REMOTE SENSING AND GIS INTEGRATION: TOWARDS INTELLIGENT IMAGERY WITHIN A SPATIAL DATA INFRASTRUCTURE

MOHAMED ABDELRAHIM

October 2001

GEODESY AND GEOMATICS ENGINEERING
UNB

TECHNICAL REPORT NO. 210

REMOTE SENSING AND GIS INTEGRATION: TOWARDS INTELLIGENT IMAGERY WITHIN A SPATIAL DATA INFRASTRUCTURE

Mohammed Abdelrahim

Department of Geodesy and Geomatics Engineering
University of New Brunswick
P.O. Box 4400
Fredericton, N.B.
Canada
E3B 5A3

October 2001

PREFACE

DEDICATION

To

My father Prof. Dr. Ing. Mahmoud Hosny Abdelrahim,
my mother, my sisters, my father and my mother in law
for everything.

Special dedication goes to my wife Samar and my son Omar.

PREFACE

This technical report is a reproduction of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Geodesy and Geomatics Engineering, June 2001. The research was jointly supervised by Dr. David J. Coleman and Dr. Wolfgang Faig. Funding was provided by the Government of Egypt.

 The citation to this work should appear as follows:

Abdelrahim, M. (2001). *Remote Sensing and GIS Integration: Towards Intelligent Imagery Within a Spatial Data Infrastructure.* Ph.D. dissertation, Department of Geodesy and Geomatics Engineering Technical Report No. 210, University of New Brunswick, Fredericton, New Brunswick, Canada, 316 pp.

ABSTRACT

The integration between remote sensing imagery and GIS vector data has been a major research concern for more than two decades. This integration facilitates monitoring and analyzing of spatial phenomena. Within the existing situation, two main integration approaches are being used. The first approach depends on processing remotely sensed data and transferring the results to a GIS. Existing GIS vector data can also be processed and used within the image processing techniques to extract better results. Second, remote sensing imagery can be used as a backdrop for the vector layers for better visualization and for vector layers updating purposes. In both approaches, remote sensing imagery is playing a passive role in the spatial query and analysis process; either as a source of information and/or a background to the vector layers.

As remote sensing imagery continues to improve in terms of spatial resolution (< 1 m), it has the potential to provide a closer, clearer, and sharper representation of real-world phenomena and promises a reliable medium if used actively and directly to query and explore these real world phenomena.

Although existing integration techniques satisfy the needs of several spatial applications, they are not optimum for either direct/on-the-fly usage of GIS vector layers as a base in interpreting the image content or active involvement of high quality imagery in spatial query and analysis. These techniques degrade the quality of the image scene, disturb the analysis performance, and may hide important spatial patterns that appear within the image.

In this research, an "Intelligent Imagery System Prototype" (IISP) was developed. IISP is an integration tool that facilitates the environment for active, direct, and on-the-fly usage of high resolution imagery, internally linked to hidden GIS vector layers, to query the real world phenomena and, consequently, to perform exploratory types of spatial analysis based on a clear/undisturbed image scene. The IISP was designed and implemented using the software components approach to verify the hypothesis that a fully rectified, partially rectified, or even unrectified digital image can be internally linked to a variety of different hidden vector databases/layers covering the end user area of interest, and consequently may be reliably used directly as a base for "on-the-fly"

querying of real-world phenomena and for performing exploratory types of spatial analysis.

Within IISP, differentially rectified, partially rectified (namely, IKONOS GEOCARTERRA™), and unrectified imagery (namely, scanned aerial photographs and captured video frames) were investigated. The system was designed to handle four types of spatial functions, namely, pointing query, polygon/line-based image query, database query, and buffering. The system was developed using ESRI MapObjects 2.0a as the core spatial component within Visual Basic 6.0.

When used to perform the pre-defined spatial queries using different combinations of image and vector data, the IISP provided the same results as those obtained by querying pre-processed vector layers even when the image used was not orthorectified and the vector layers had different parameters. In addition, the real-time pixel location orthorectification technique developed and presented within the IKONOS GEOCARTERRA ™ case provided a horizontal accuracy (RMSE) of +/- 2.75 metres. This accuracy is very close to the accuracy level obtained when purchasing the orthorectified IKONOS PRECISION products (RMSE of +/- 1.9 metre). The latter cost approximately four times as much as the IKONOS GEOCARTERRA™ products.

The developed IISP is a step closer towards the direct and active involvement of high-resolution remote sensing imagery in querying the real world and performing exploratory types of spatial analysis.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and gratitude to my supervisors Dr. David Coleman and Dr. Wolfgang Faig for their encouragement and never ending support towards my work to make it possible.

I am grateful to Dr. Y.C. Lee, Dr. Bernd Kurz, and Dr. Yun Zhang for the discussions and suggestions which were extremely helpful.

I would like to thank David Fraser for his help and technical support during the period of my research.

Thanks to Linda, Wendy, and Kim for always being there to help me out with my never-ending problems and needs.

Thanks to all my friends and colleagues in the department, especially Songnian Li, Saleh Al Hubail, Rima Ammouri, Kevin Pegler, Stephen Rawlinson, Mele Rakai, Sam Ng'ang'a, Thomas Widmer, Michael Sutherland, Kutalmis Saylam and Frank Neidhart.

I would like to take the opportunity to express my deepest gratitude to the Egyptian Government for the financial support during my research period.

I would like to thank my wonderful parents for their never ending love and moral support.

This work could never be completed without the love, support, and understanding of my wife, Samar.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 CONTEXT OF THE RESEARCH

Remote sensing (RS) and GIS integration has taxed the minds of researchers for more than two decades (for example, see Baker and Drummond, [1984] and Barker [1988]). As computer hardware and software technologies advanced, new visions and approaches for accomplishing this task in a more seamless and complete way were developed. More than a decade ago, Ehlers et al. [1989] formulated a very popular model that describes both how the integration was performed at that time and the final goal regarding this task. The so-called "total integration" was and still is the final aim of researchers and software vendors. At such a level, for example, one can rely on GIS vector data and functionality to provide an internal on-the-fly (directly within the query/analysis process) basis that may help in interpreting remotely sensed imagery and facilitating the active/direct usage of imagery in spatial query and exploratory analysis. This is accomplished with minimal pre-processing requirements and without any interruption/disturbance to the image scene or to the process.

Despite great improvements in remote sensing image quality and in GIS technology, total integration has not yet been reached [Ehlers, 2000]. Until recently the integration task was performed mainly in two broad approaches. The first approach is represented by a graphical integration. Using this approach, we can use this very high quality imagery as a background for the vector data to be verified, updated, and/or for image interpretation. In addition, this approach may rely on the user to perform the integration visually by simultaneous display of both datasets. The second approach is based upon conversion techniques. The conversion is accomplished from vector-to-raster to be able to include the vector information as an additional channel within the image in order to improve the image interpretation task, or from raster-to-vector to transfer the results from the image processing/interpretation back to the vector domain.

Although these two approaches satisfy the needs of several spatial applications, they are not optimum for either (1) direct/on-the-fly usage of GIS vector layers in interpreting the image content; or (2) active involvement of the high quality imagery in spatial query and analysis. The superimposition technique degrades the visual quality of the image by overlaying vector maps on the image, disturbs the analysis performance, and may hide important spatial patterns that appear within the image. In addition, the conversion techniques have a problem of maintaining the attribute information during the conversion process.

As remote sensing imagery continues to improve in terms of spatial resolution (< 1 m), it has the potential to provide a closer, clearer, and sharper representation of the landscape for the surface of the earth. This promises to be a reliable medium if used actively and directly to query and explore real world phenomena. This imagery now deserves to be reconsidered in terms of its usage within the integration framework for spatial query/analysis and in the spatial decision making processes. In addition, integration techniques should go beyond the graphical or conversion levels, to allow high quality remote sensing imagery to play a more active role within an integrated GIS environment and to facilitate the usage of GIS vector data in extracting/ interpreting the image content in an efficient manner.

The main goal of this research is to determine the possibility, identify the interface requirements, emphasize the effectiveness, and declare the drawbacks of building an internal on-the-fly link between remote sensing imagery, with different rectification levels, and GIS vector data, having different characteristics. This link will facilitate the direct/active use of remotely sensed data as a base for querying the real world and, consequently, as a reference in our spatial analysis and support of direct/disturbance free initial interpretation of the image content.

The "intelligent imagery" concept presented in this research represents one possible way of integration that internally processes and links the GIS vector data layers with remote sensing imagery. This allows the image to access all features' related information and GIS functionality without any disturbance to the image scene or to the process. While other approaches stack vector data layers on top of the imagery, this approach keeps the corresponding vector data hidden and utilizes them internally until results need to be displayed. This approach creates a better environment for the on-the-fly usage of GIS vector databases/layers and functionality to help interpret the imagery. In such an environment, for example, the user does not have to be concerned about the image rectification level, degrading the image scene by superimposing or simultaneously displaying vector layers to access the image feature's existing attributes, or image-vector required pre-processing to align/match both data sets. Furthermore, the environment facilitates the direct and active use of remote sensing imagery as an active reference layer to retrieve information about real world features on the image, perform spatial queries, and display the results of these queries on the image. Consequently, the environment paves the way for performing exploratory spatial analysis based on a clear/undisturbed image scene.

1.2 RESEARCH OBJECTIVES

Based upon the literature review conducted within the area of remote sensing and GIS integration and spatial software components, the following can be extracted:

1- Nowadays, widely available very high resolution imagery (<1.0 m) promises to provide almost continuous representation of the real world.

2- Huge numbers of previously interpreted vector data bases exist and are being updated on almost a regular basis.

3- Software components technology (encapsulating the required functionality's objects) pave the road to new, robust and more flexible integration techniques.

4- GIS vector data and remotely sensed imagery integration techniques currently employed have not gone far beyond simultaneous display, superimposition, and data conversion/transformation.

5- There is a need to develop integration techniques that provide a direct and active involvement of the high-resolution imagery, almost identically representing the real world, in spatial query and, consequently, spatial analysis.

The objective of this research is to design, implement, and test a general-purpose system/tool prototype of the intelligent imagery concept (Intelligent Imagery System Prototype "IISP") using available software components/toolkits to handle different integration cases, with an increasing level of complexity. These cases will include remote sensing imagery with different rectification levels (differentially rectified, partially rectified, and unrectified) and GIS vector layers with different parameters/characteristics (projection, coordinate system, scale), as shown in section 3.1.1. Although the system developed within this research work is not application-specific, it identifies and solves the technical, theoretical, and interface issues/problems of building an internal direct/on-the-fly link between the image and vector layers in different cases. In other words, the system developed within this research is serving as an essential core component that can be extended in the future to serve certain specific tasks.

In order to build such a system, the following steps were taken:

1- Conceptually design an "Intelligent Imagery" system and point out the major steps towards direct linkage between vector databases/layers and image data to be able to:

   a- perform on-the-fly image queries and display the attributes based on a specific cursor location in the image;

   b- query the database and display the results accurately and directly in the image;

   c- build a query based on a polygon of any shape drawn onto the image; and

   d- perform advanced spatial operations such as the buffering operation within the system.

   This must be achieved without the need for vector/raster overlay or raster-to-vector/vector-to-raster conversions within the different integration cases, as mentioned earlier.

2- Identify the system components and requirements and use these requirements as a base for a subsequent research study throughout the GIS and remote sensing software market to extract the technology necessary for building the system.

3- Develop the appropriate techniques to establish the linkage between the two data sets, in different integration cases with an increasing level of complexity, by integrating the appropriate software toolkit component(s) obtained from the previously conducted research.

4- Implement and develop a prototype system using the developed techniques in three of the previously mentioned integration cases and propose solution techniques to develop the fourth case, (please refer to section 3.1.1 for full details about these cases).

5- Test the developed prototype to ensure that it delivers accurate results for the processed queries.

6- Implement the developed prototype in an Internet-based environment for better access and distribution approach.

7- Identify the limitations of the developed techniques, if any, and give suggestions and recommendations for further improvement.

This research addresses an area that will become of even greater concern and interest as remote sensing image quality improves. In other words, as the image resolution gets finer and more closely represents our continuous real world phenomena, we have to find better ways to introduce, link, attach, and integrate directly to the image the information associated with real world features stored within a GIS. For maximum benefits, this should be accomplished without the need to disturb the image scene and/or the process by stacking vector layers on the image, applying certain conversions, or performing pre-processing transformations.

1.3 THESIS ORGANIZATION

This thesis is divided into five chapters. In Chapter 2 the background research will be presented. The chapter will start by laying out a general overview of the existing techniques and approaches for integrating remote sensing and GIS. Emphasis will be given to a discussion of the use of remote sensing imagery and GIS vector data as well as to the possible roles that remote sensing imagery can play within an integrated GIS (IGIS). This leads to the introduction of the intelligent imagery approach, its associated required components and techniques, and relevant research. Furthermore, the definition and history behind the software components technology is also presented and the benefits and requirements of using software components technology for an ideal integration between remote sensing and GIS was demonstrated. A research study result is presented, highlighting the available software components within the GIS and remote sensing software markets

Chapter 3 is dedicated to the developmental stages of the (IISP). The first section discusses the analysis of the system requirements, which identifies the system functions, techniques, and required components. The subsequent four sections represent the system interface design, internal design, implementation, and process workflow for each case within the IISP. The problems encountered as well as actions taken in implementing those cases are emphasized. The last part of this chapter will focus on the system's

Internet migration process. This section discusses the technology, developmental stages, problems, and solutions within the Internet migration process.

Chapter 4 discusses the testing procedure performed on the developed system. The testing methodology and its stages are presented. During this stage of the research, the development techniques were tested to ensure that the designed process was mapped correctly within the programming code and to ensure the delivery of spatially and aspacially accurate results. The chapter starts by identifying the testing components and stages. The test results are then presented, and the required modifications are identified. The author's vision regarding possible modifications for the developed system was presented.

Finally, the conclusions and recommendations arising from all the work undertaken in this project will be stated in Chapter 5.

CHAPTER 2

BACKGROUND RESEARCH

The integration of remote sensing raster imagery with GIS vector maps has proven to be a need for many spatial applications. The aim of this chapter is to review remote sensing and GIS integration techniques, to present the IISP concept, to introduce relevant research efforts, and to identify the position or the category the IISP may have within the integration framework.

2.1 IMAGE/VECTOR DATA INTEGRATION

2.1.1 Problems in Reliable Integration of Image and Vector Data

Remote sensing imagery (airborne or spaceborne) serves as an effective, fast, current, and reliable tool for visualizing and collecting information about spatial phenomena. During the last two decades, many organizations throughout the world have dedicated much of their efforts to collecting, storing, and distributing remote sensing image products in an efficient manner. For example:

1- Service New Brunswick (SNB) is in the process of constructing a large orthophotomap database covering all of the coastal areas of New Brunswick at a scale of 1:10 000. SNB aims to extend this effort to eventually cover the entire province [Abdelrahim et al., 2000 a].

2- The United States Geological Survey (USGS) is creating one node in a "National Geospatial Data Clearinghouse" that is designed to help users find information about geospatial or spatially referenced data available from

USGS. One division of the clearinghouse is for national mapping and remotely sensed data [USGS, 2000]. That division offers a variety of choices of satellite imagery at different resolutions and spectral bands, and digital orthophotos at a scale of 1:12 000.

3- The Space Imaging Company recently has launched the first one-metre resolution satellite image from the IKONOS satellite [Space Imaging, 1999].

4- The Canadian and European space agencies will continue to launch radar sensors, that can collect night-time imagery.

5- More than 30 earth observation and monitoring satellites with a variety of different resolutions and spectral bands are scheduled for launch early in the twenty first century [Stoney and Hughes, 1998].

With these few examples, it is apparent that humankind is really heading towards the so-called "Decade of Imagery" [Stefanidis, 1997] or, as I would say, "decade of high-resolution imagery."

On the other side of the spatial information highway, a large amount of digital vector map data has already been collected either photogrammetrically, or by digitizing existing hardcopy maps, or through GPS and/or via Computer Aided Design (CAD) files, and these data are being updated on a regular basis. For a long time, governments and private organizations have been collecting map data in digital form. For example:

1- SNB provides digital topographic maps at a scale of 1:10 000 covering the entire province.

2- The USGS manages a collection of 70 000 digital topographic map sheets at different scales. One can order digital maps from the nearest "*map dealer*" across the United States.
3- Topographic maps and digital spatial data for Australia at scales of 1:250 000, 1:100 000, and 1:50 000 are available from the Australian Surveying and Land Information Group [AUSLIG, 2000].
4- Geomatics Canada developed National Topographic Database (NTDB), a digital data base that covers the entire Canadian landmass at the scales of 1:50 000 and 1:250 000 [NRCAN, 2001]

These examples provide just a glimpse of data providers in the so-called Spatial Data Infrastructure.

Integrating remote sensing raster imagery with already existing vector data is an essential requirement for efficient handling and monitoring of Earth's phenomena. This integration should be accomplished within a powerful system that has the capability to handle diverse types of geographic data. A Geographic Information System (GIS) is a system by which we can input, manipulate, maintain, analyze, and output multiple forms and layers of spatial data [Aronoff, 1996]. With recent advances in technology, we now can handle remote sensing imagery and vector data within a GIS environment in a certain manner and with certain limitations.

Two main approaches exist for integrating the two data types. One approach is to first process the remote sensing imagery within an Image Processing System (IPS), and then digitize the extracted information for input into a GIS. GIS data might also be used to better extract information from remote sensing imagery through pre-defined conversion/transformation. The other approach is to use the remote sensing imagery as a backdrop for the vector map for the purposes of map updating and/or better visualization and consequently the decision making process.

These two approaches are suitable for many applications where the image is playing a passive role in performing spatial analysis, as a source of information or as a medium for vector data quality and integrity checks. When trying to directly/on-the-fly utilize the GIS vector data in image interpretation or to actively involve this imagery, especially the widely available high resolution imagery, in spatial analysis and/or the decision making process, these two approaches will not be adequate. In such cases, we need to extract on-the-fly any information related to image features and also to perform GIS functions, which can help to interpret the image content or to query the real world, without any interruption to the image scene or to the analysis process. In other words, we need to implement the integration beyond the graphical and/or data conversion levels.

In this research work, a different approach to using raster remote sensing imagery, internally supported by vector database layers, is examined for efficient spatial information query and retrieval. This approach is based on the idea of using remote sensing imagery, supported by *hidden* vector data plus the associated attributes stored in a GIS, as an interactive medium. Through this medium we can, for example, issue a cursor specific query, display Data Base Management System (DBMS) query results directly through the image pixels, issue a localized query based on a delineated polygon\line of any shape on the image, and perform advanced spatial operations, such as the buffering operation, through this medium. This method presents a better integration approach in trying to use the available information stored within a GIS vector database to facilitate the active use of remotely sensed imagery in our spatial analysis and/or decision making process. Also, it paves a better way for the use of existing vector data to aid image interpretation and, consequently, for the feature extraction process. This method eliminates the need for simultaneous display of vector and raster data or for vector/raster overlay/conversion that might lead to a loss of accuracy, confusion, slow analysis performance, or hide important spatial patterns that could affect the final decision criteria. This research project is considered to be bringing us one step closer towards these goals.

2.1.2 Limitations in Available Commercial Off-The-Shelf Software Packages

In order for the intelligent imagery concept in general, and IISP in particular, to be designed and implemented, the general design specifications of the system have to be defined. The main issue is relying on the image as an interaction medium with the real world. In order to do so, we need to provide the image pixels with the information required to serve this interaction/query process. This can be done by internally and on-the-fly linking of pixels of desired image features with the corresponding vector layers. Because the image may not be at the appropriate rectification level (an orthophoto, for example) and both data sets may come from different sources with different parameters, embedded on-the-fly rectification procedures should be designed to transform/orthorectify image feature's pixel(s) coordinates to the correct corresponding vector layers ground coordinates. To accurately retrieve the correct results, certain interfaces as well as procedures have to be designed and implemented.

As will be discussed in section 2.2.1.3, most of the existing commercial off-the-shelf (COTS) software packages deal with the integration issue mainly at graphical and data conversion levels. If the intelligent imagery concept proposed is to be implemented within any of the existing software packages, the following have to be considered:

1- Vector layers must have the same parameters as the image, i.e., the user has to transform/reproject all the layers to comply with the reference layer, i.e. the image, characteristics.
2- The image must have a high rectification level (an orthophoto) or an accurate registration procedure should be accomplished allowing the image to be geocoded properly with the corresponding vector layers;
3- The system interface has to be modified in order to assure accurate results retrieval;
4- The system functionality should be modified so that it is unnecessary to treat each data set as a whole when performing the transformation and/or orthorectification procedures.

While the first task is a time consuming one, the second task, in addition, may require trained personnel and certain tools that may not exist within the software itself. Another problem is that recent commercial high-resolution imagery, such as IKONOS GEOCARTERRA™ (Space Imaging, 1999), may need further studies and new algorithms in order to be orthorectified. These new algorithms then have to be implemented within the software, and then the image has to be orthorectifed. Concerning the third issue, existing software packages allow for interface modifications but with limited capabilities. For example, the ArcView GIS package allows for the modification of its interface and functionality through the Avenue scripting language but the developer will not be permitted to create new objects [Razavi, 1999]. This is a crucial issue if the interface and/or functionality requires the creation of new objects that may not exist within the existing software. The fourth issue is very important if the user has limited computer resources. Existing software has to transform/reproject the whole data layer and the entire image has to be orthorectified. This is not necessarily true within the Intelligent Imagery concept. Within which, only the desired image pixel(s) and/or vector layer feature(s) need to be treated at the time of process execution.

One more issue in dealing with the integration between image and vector data is the integration level. While some existing software packages integrate some of the image processing and GIS functionality, the system has to perform each function separately. One function has to be performed first and the results have to be retrieved and used as input for the next function. In other words, the integration is the sum of all functionalities. This may cause processing delays and may require unnecessary user interaction within the process. Within a real total integration (as will be discussed in sections 2.2.1.3 and 2.4.2), many tasks have to be automated and consequently the process/analysis procedures may be speeded up. For such integration to be achieved, the objects that handle that functionality have to be internally integrated based upon pre-designed process flow and minimal user-system interaction. As a result of such internal object integration, these automated procedures may be accomplished in a more reliable, well-defined, and flexible manner and consequently the total integration aim will be much closer, as will be discussed in sections 2.4.2 and 2.4.3.

2.2 INTEGRATION OF REMOTE SENSING AND GIS

Integration of remote sensing imagery and GIS is a paramount requirement to efficiently handle the ever-increasing amounts of spatial data. Over the past two decades, the integration has been performed at different levels, with different operational and functional characteristics. In this section, background research on the techniques and levels of performing the integration, the general categories of applications representing the use of remote sensing imagery and GIS within such integration levels, as well as the role of remote sensing imagery in each of these categories is presented.

2.2.1 Integration Levels

In order to set a framework for the integration of remote sensing and GIS, three main levels of integration have been identified by Ehlers et al. [1989]. The characteristics of these levels are described in the following three sub-sections.

2.2.1.1 Level I: Separate But Equal

The first level of Ehlers' hierarchy is called "Separate But Equal," which indicates the separation of image processing and the GIS system. In this approach, the user would be able to simultaneously display vector data and remote sensing imagery and to move either the image analysis results into the GIS and digitize a classified image or use GIS data to georeference the imagery. Level I is based upon mainly exchanging data between systems. This level is considered to be a very low (or early) level of integration. With the advances in computer technology, this level is now disappearing and being replaced with a more advanced level of integration, Level II. Figure 2.1 shows the system architecture for Level I integration.

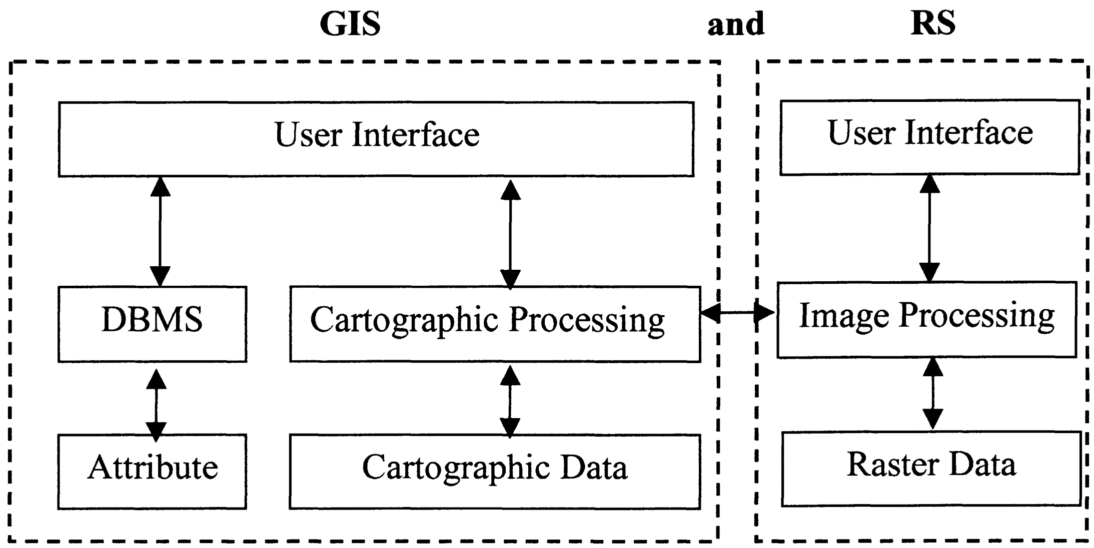


Figure 2.1
GIS and RS Level I integration (after Ehlers et al. [1989]).

2.2.1.2 Level II: Seamless Integration

This level involves "Seamless Integration" in which raster-vector processing is allowed. GIS and image analysis systems are stored in the same computer. Simultaneous access to the functions of both systems is allowed through a common interface, but the separate systems operate independently, and data must be exchanged regularly between the two systems. Although problems of format conversion and raster/vector overlays have been reported for the usage of that level (for example, see Laurer et al. [1991] and Hinton [1996]), it provides a temporary solution to the integration task. Figure 2.2 shows the system architecture for Level II integration.

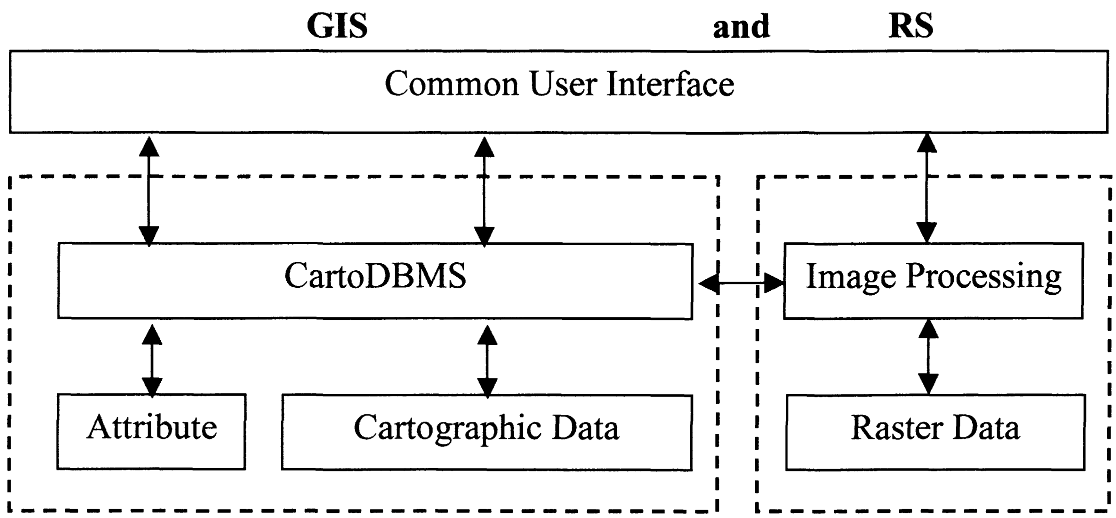


Figure 2.2
GIS and RS Level II integration (after Ehlers et al. [1989]).

2.2.1.3 Level III: Total Integration

This system is called "Total Integration" in which we have one system that allows the user to process remote sensing data and vector data simultaneously. This makes use of full GIS and image analysis functionality simultaneously with no need for data conversion between systems. Figure 2.3 shows the system architecture for Level III integration.

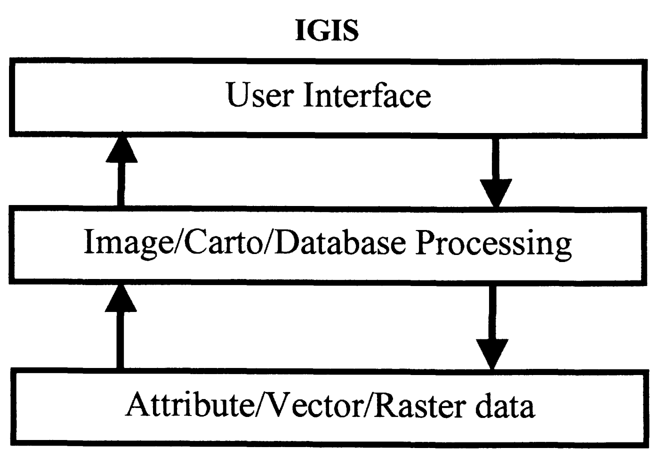


Figure 2.3
GIS and RS Level III integration (after Ehlers et al. [1989]).

Most of the software packages that exist either in the GIS or in the image processing fields support only the first two integration levels or at most a very early stage of the third integration level. For example, ESRI ARCVIEW embeds limited remote sensing image processing functionality by using software extensions from ERDAS IMAGING [ERDAS, n.d. b; ESRI, 2000 a]. Another example would be the capabilities of remote sensing packages (such as PCI, ERMapper, ERDAS, etc.) to import vector map layers and attribute tables and display them in a superimposition mode with the image. The CARIS GIS software package from Universal System Limited implements its own image handling capabilities such as, orthorectification, registration, mosaicking, image enhancements, etc., within the system [USL, 2001]. In other words, any of these existing systems mainly contain more or less the sum of their own functionality plus some functionality from the other side. The functionality integration is being done at a higher level, i.e., one function starts where the other one ends. Furthermore, the integration of the actual data and processing results is still performed within the graphical level and/or using procedures which are out of the main application scope process (superimposition, simultaneous display, conversion, user-based inputs/operations). Total integration as described by previous and recent research efforts (for example Ehlers et al., [1989]; Turker [1997]; Abdelrahim et al., [2000 b]; Ehlers [2000]), should be done at a lower level, i.e., datasets and functionality should be integrated internally. Users should have the freedom of selecting the appropriate medium (either remote sensing imagery or GIS vector data or both) to perform the required analysis while having information and supported functionality available from both sides to serve their needs. As shown in Figure 2.3, total integration allows for the use of vector database layers, remote sensing imagery, or both for spatial analysis. These data sources should be internally linked together and based on the situation, either vector maps or raster remote sensing imagery can be used to query the real world, retrieve spatial information, and perform spatial analysis. The long-term goal of Level III integration, i.e., total integration, will not be reached by just one big step but rather by several small steps.

As will be discussed later in section 2.2.3.3, most of the previous and current work on integration has ended up transferring and digitizing pre-processed remote sensing data into a GIS and using vector data as a base for querying the real world and/or for using the imagery as a passive backdrop for the purpose of either updating vector layers or enhancing the visualization for better spatial analysis. Relying on vector data layers alone or having an image as a backdrop for querying the real world has several disadvantages as discussed in section 2.3.2.

With the advancements in remote sensing technology, we can now easily obtain very high-resolution imagery (1~ 4 m). High-resolution imagery contains a large amount of detail, and the identification of features becomes easier and more reliable. Those images serve as a great source of information and primary views of the real world phenomena. It is uneconomic and confusing in some cases to still use the imagery, especially high-resolution ones, as a backdrop for vector maps for visualization purposes and for relying mainly on vector maps for making spatial decisions. It should be rather used as a reference layer through which we can directly retrieve spatial information and execute GIS functions such as buffering or neighborhood operations serving spatial tasks. This can be reached by internally linking the huge well-organized vector layers/databases to the image and by letting the image serve as the rich visual medium, and the vector data as the information/intelligence source of the image.

2.2.2 Use of GIS and Remote Sensing Data

In general, techniques and levels of integration between remote sensing and GIS are divided into the following categories, as identified by Wilkinson [1996]:

1- Category A: Remote sensing used as an information source for GIS.

2- Category B: GIS data used to aid image processing, image interpretation, and feature extraction from remote sensing imagery.

3- Category C: Remote sensing and GIS data used together in environmental modeling and analysis.

The approach of using remote sensing imagery supported by existing GIS vector data as a tool or reference layer in querying the real world belongs to the second and third categories. The following sub-sections represent the major roles that remote sensing imagery can play within the integration framework.

2.2.3 Roles of Remote Sensing Imagery within an Integrated GIS (IGIS)

Remote sensing imagery can play three main roles in a GIS with respect to the collection, retrieval, visualization, and query of spatial information, namely passive, stand alone, and active [Derenyi and Fraser, 1996].

2.2.3.1 Passive Role

In the passive role the image is registered and geo-referenced to the ground coordinate system with the aid of several features or points, well defined in both the image and on the ground. Then, classification and feature extraction algorithms are applied to the remotely sensed imagery, producing thematic maps. The results can be vectorized and entered into the GIS as a vector layer. For another typical application, the original or processed image is attached to the multi-layered GIS system and serves as a backdrop to the existing vector layers in the GIS for the purpose of either updating these GIS layers or else querying/analyzing spatial phenomena using the superimposed collections for better visual analysis (this may be considered as an advanced passive role). The superimposition of vector layers over an image covering the area under consideration aids the viewers and the analysts to better understand the recorded geographic entities.

2.2.3.2 Stand-Alone Role

Nowadays, many software tools exist that allow the user to create a digital orthoimage, i.e. a rectified image corrected for relief displacements, for example, PCI [PCI, n.d.] and ERDAS IMAGING [ERDAS, n.d. a]. This digital orthophoto can serve as a base map by itself. The popularity of creating and using orthoimages is rapidly increasing among many users and organizations. For example, SNB is now establishing a large database of orthophoto maps covering all of the coastline of the province, and is planning to eventually cover the entire province. Other examples include, the USGS's huge archive of Digital Orthophoto Quadrangles [USGS, 2000], Massachusetts Institute of Technology's (MIT) Orthophoto Server [MIT, 1998] , Department of Information Systems and Telecommunications (Montgomery County's Digital Orthophoto Image Server) [DIST, 2000], etc.

2.2.3.3 Active Role

At this stage, remotely sensed images are directly involved in spatial information retrieval, visualization and analysis, and as a result provide a more complete, current, and accurate view of the real world for better decision making. This research deals mainly with the active use of remote sensing imagery. The schematic diagram in Figure 2.4 gives an overview of the framework of remote sensing and GIS integration and illustrates where this research fits into the total picture.

As we can see from Figure 2.4, there are three major ways of utilizing remote sensing and GIS vector data. In the first category (A), as mentioned earlier in section 2.2.2, remote sensing imagery is used as a data source for GIS, and then the GIS vector data is used alone as a tool for spatial analysis. At that stage, the image plays a passive role. For example, Baumgartner and Apfl [1994] classify satellite imagery and transfer the results by digitizing them into a GIS to create snow maps. Hodgson et al. (1988) use Landsat TM imagery to create "wet" and "dry" year maps of foraging habitats. The maps are then used within a raster-based GIS ERDAS, by that time, to perform the monitoring analysis. Welch and Madden [1988] produced 1:10 000 and 1:24 000 scale aquatic vegetation maps from color infrared photographs at different dates. The maps were then converted forming a raster-based database which was used in a raster-based GIS to assess the factors affecting the plant growth within the study area. Chuvieco and Congalton [1989]

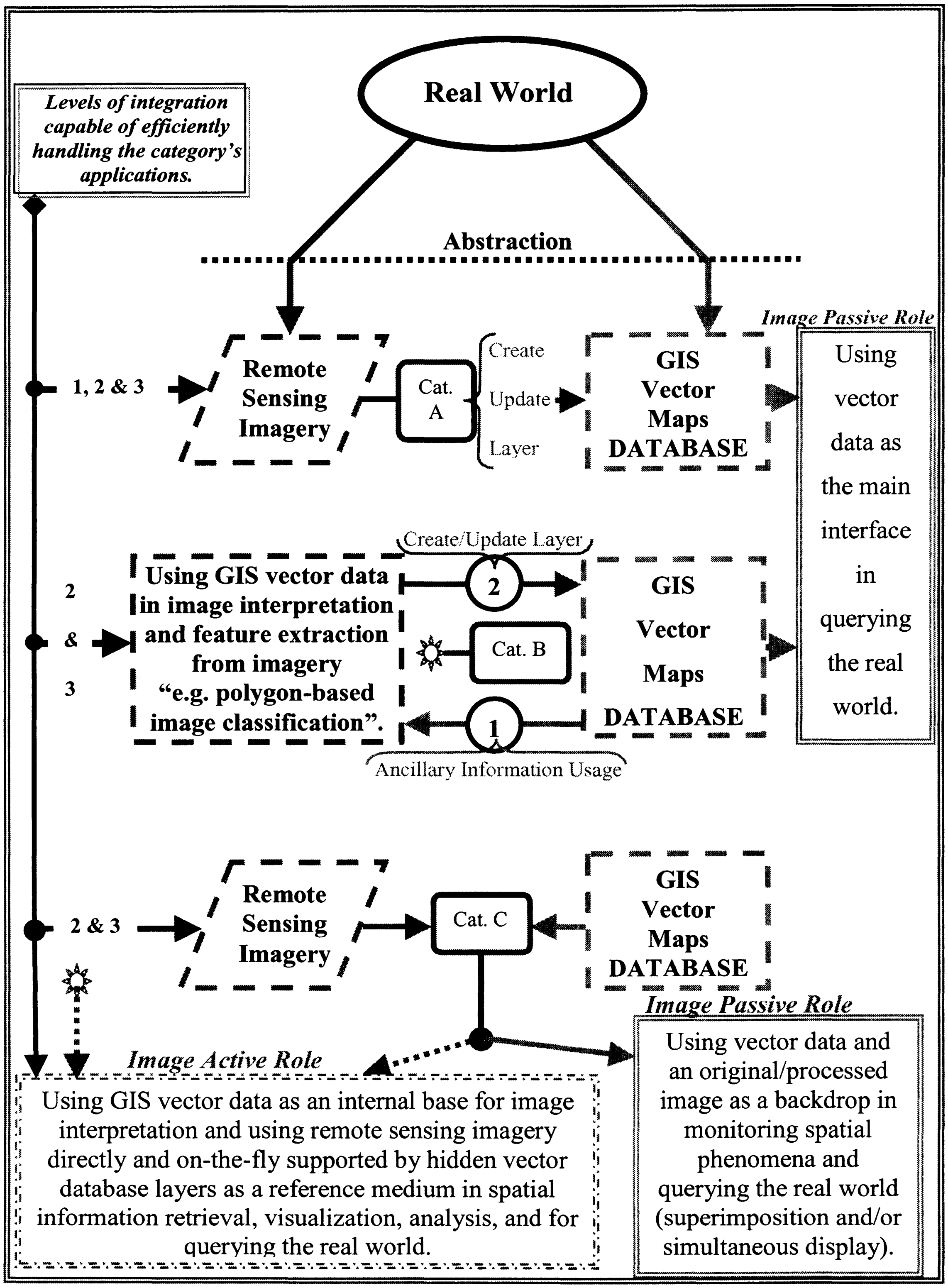


Figure 2.4
GIS and remote sensing: Integration framework.

utilized the digitally processed Thematic Mapper data as a layer in their GIS to map the forest fire hazard along the Mediterranean coast of Spain. Jakubauskas et al. [1990] used Landsat MSS images taken at two different dates to create vegetation maps, and then a GIS system was used to examine the effects of fire in the vegetation changes in a Michigan pine forest. Jazouli et al. [1994] use edge detection filters to extract roads and class boundaries from SPOT and Landsat images and then update the existing GIS layers. Mickelson et al. [1998] utilize the Gopal-Woodcock fuzzy set process in extracting forest classes from Landsat TM images. Considerable effort has gone into extracting 2-D and 3-D buildings from remote sensing imagery and then updating the GIS databases (see for example Kim and Muller [1998]). Sharma and Sarkar [1998] used a modified contextual classification technique to extract classes from Landsat TM in which the spatial context of a pixel is involved in the classification process. Contextual classification techniques proved satisfactory in extracting information/features over the single/independent pixel classification. A good review of the use of remotely sensed data as an information source for GIS can be found in Trotter [1991].

In the second category (B) (see Figure 2.4), GIS data are used as aids for extracting accurate information from the image, e.g. extracting forest polygons through image classification using GIS ancillary data. This information is then vectorized and entered into the vector GIS for further analysis. Also in this case, the image plays a passive role. A number of examples exist that use GIS data as the knowledge base for image classification. For example, Molenaar and Janssen [1992] used GIS data in a knowledge-based system to improve the image segmentation and classification for the Polder areas in the Netherlands. Cleynenbreugel et al. [1990] utilized GIS data to automatically extract road structures from remote sensing imagery. Kontoes et al. [1993] developed a method and a "reasoning program" to post-process the classified image data using GIS data. Janssen et al. [1990] incorporated GIS topographic data to identify polygons in the image and to identify the pixels that belong to each polygon in a per-polygon image classification. An increase in the classification accuracy by 12~20% over the conventional per-pixel classification was reported.

In the third category (C) (see Figure 2.4), GIS and remote sensing imagery can be used together in spatial analysis. Such a situation would use the original/processed image as a backdrop to the vector data for better visualization, analysis or map updating. For example, Derenyi and Turker [1996] developed an integration technique by which they could integrate GIS vector data and satellite imagery to perform a polygon-based image analysis and classification. This was then used to update existing land-use polygon layer classes by using image processing and GIS software packages, and transferring the data between them. In order to examine the main cause of oak mortality in rural and urban environments throughout central Texas, Ware and Maggio [1990] used interpreted aerial photos overlaid with urban barriers, such as streets, water lines, and houses, to monitor the effect of these barriers on the oak disease. Chuvieco and Congalton [1989] used classified TM data and integrate it with GIS data layers to create a forest fire hazard map. Ambrosia et al. [1998] used GIS ancillary information overlaid with raster airborne imagery and assisted by communication means in allocating resources against fire in a near-real-time process to detect fires at early stages and to minimize the damage. Gamba and Casciati [1998] utilized GIS and image understanding techniques in the RADATT (RApid Damage Assessment Telematic Tool) project, funded by the European

Commission, for early and near-real-time assessment of earthquake damage in the Umbria region (central Italy). Koch and El-baz [1998] used Landsat TM imagery to create a surface change map and a classification map. Then, these maps, were integrated/overlayed with SPOT panchromatic images, topographic maps, geomorphic maps, and surface sediment maps of Kuwait in a GIS to asses the effect of the Gulf war on the geomorphic features of Kuwait. The image here is playing a passive/advanced-passive role or at most a "semi-active" role.

As shown in the previous examples, in all three categories, the main roles of images are either as a source of data or as a backdrop to the vector data, i.e., the image is in a "passive state" or at most in a "semi-active state". In the author's opinion, the intelligent imagery approach may fall into categories (B) and (C).

As can be seen from the previous literature review, an excessive amount of remotely sensed data and GIS vector databases/layers are available but with limited integration tools. Both data sets have been used for a variety of applications but still the integration techniques are relying on superimposition and/or conversion. These integration techniques have been reported to have different functional as well as operational problems. Furthermore, if the new coming era of very high resolution satellite imagery is to be used in a more active manner, and the analysis of spatial phenomena is to be directly based upon unobstructed, primary data sources, and complete and current views of this imagery, then new integration tools should be developed.

2.3 INTELLIGENT IMAGERY APPROACH: CONCEPT, RELEVANT RESEARCH AND ANTICIPATED SIGNIFICANCE

The "intelligent imagery" concept, in general, is a new paradigm for integrating remote sensing and GIS for spatial information retrieval and visualization and, consequently, for spatial analysis. The author's approach relies on using remote sensing imagery as an active reference layer to query and retrieve attribute information about real-world features on the image and display the results of specific queries regarding these features within the image. In this system, orthorectified, partially rectified/geocoded, or unrectified multi-resolution imagery covering a specific area is displayed on the screen and linked internally to the vector database file(s) covering the same or a larger area. The vector files may have different projections, coordinate systems, and scale parameters. Without superimposition or simultaneous display, the system on-the-fly compensates for the image rectification level of distortion as well as the variations in image and vector data parameters in order to perform the query tasks. Figure 2.5 illustrates a general overview workflow of the IISP, as concerned in this research.

First, the user has to *identify* the required vector layers to be involved in the task. Those layers, as concerned in this research, may reside on the user local machine or can be downloaded from a specific FTP server. Another option, which was not of concern in this research, is that the layer may reside on a remote server and the system sends query requests to the server and the server responds back. As those layers are identified, the system should be able to recognize some parameters of the layers. Second, the desired image as well as the associated parameters, such as the projection, coordinate system, rectification level, should be directly recognized and/or entered into the system. Then the system loads the image.

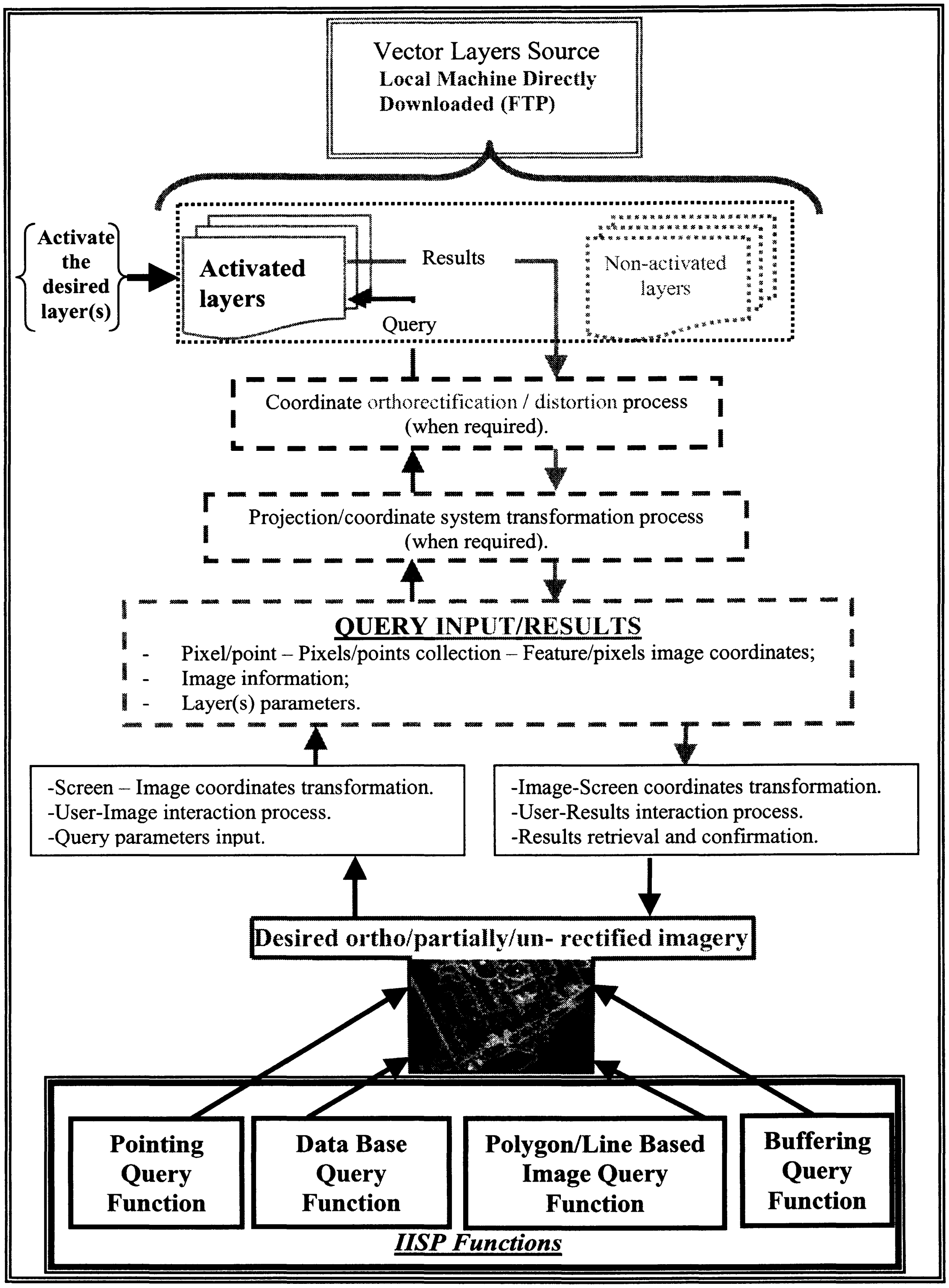


Figure 2.5
IISP general processing workflow.

Based on the query function selected, four functions will exist within the proposed system concerned in this research, the system executes a certain process. The process in the query flow direction takes the image coordinates, performs certain transformations (as will be shown later in designing the system) and queries the desired/activated layers. The results are then back-transformed/distorted (if required) to match the image and the layers previously entered/recognized parameters for display purposes.

2.3.1 Relevant Research

Little research has been found to date in the area of intelligent imagery or active use of remote sensing imagery. McKeown [1984; 1987] explains the steps and components of a system that was designed to include aerial photos in the spatial query analysis. This system was called MAPS. MAPS can be defined as [McKeown, 1987, p.334].,

> *...is a large scale image/map database system for the Washington, DC. area that contains approximately 200 high resolution aerial images, a digital terrain database, and a variety of map databases from the Defense Mapping Agency (DMA).*

The MAPS spatial database was developed between 1980 and 1984, and supported by the Defense Advanced Research Projects Agency (DARPA) Image Understanding Program. MAPS allowed the user to perform spatial query using aerial imagery as spatial indexing into the spatial database. MAPS made use of an artificial intelligence system and some decision rules that are based on the Washington, D.C. area, to perform or retrieve the query results. While representing leading-edge work at the time, the design of MAPS did not address the following:

1- MAPS was an area specific project, i.e. it was not a general purpose system.
2- MAPS did not make use of satellite data, only aerial photos.
3- MAPS did not consider the problem of using vector files from different organizations with different characteristics and/or different image rectification levels.
4- MAPS did not address the problem of querying features within a single image, but it did consider the problem of retrieving images from an image database based on a specific query statement.

Hepner [1988] created the so-called query language process (IDQUERY) for decision making regarding the Cross-Country Movement (CCM) in an image-based GIS. In that system, several data layers that contain soil, surface, and vegetation data were queried. The user could apply Boolean and arithmetic queries, and display the results in any raster plane. The raster plane could be a rasterized map or raster classified satellite image. Hepner reported that one of the limitations was the significant amount of image and attribute data file preparation, which means that it is not an on-the-fly process. Also the system was mainly based upon a raster data structure and did not address the idea of integrating and querying the data files in their native format, i.e. raster and vector.

Arminakes et al. [1999]introduced the concept of producing intelligent raster maps. This concept uses the raster maps as interactive medium in receiving inputs and displaying the results of performing the shortest path analysis within the vector data domain.

Little work has been done on the use of GIS information to build intelligence into remote sensing imagery and to start to use the images, with their identical representation of the real world, in querying and analyzing spatial phenomena. Much work has been accomplished, however, in the inverse problem, namely, extracting features from the image and feeding them into the GIS and using GIS vector layers to perform spatial analysis, as shown previously in category A and B. For a review of the utilization of information extracted from remote sensing imagery for different analysis purposes and applications see, for example, Buiten and Clevers [1993]. Not only satellite imagery but also softcopy aerial photographs have been used in extracting ground features, in 2-D and 3-D, and feeding them to the GIS vector layers. For an extensive review in that area, see for example Henricsson et al. [1996]; special issue in Computer Vision and Image Understanding CVIU [1998]; Niederost [2000]; and Zhang and Baltsavias [2000].

2.3.2 Anticipated Significance

The intelligent imagery system proposed by this research in Chapter 1 would provide a means of integrating vector and image data in such a way to be able to:

1- involve the image (the primary real world representation) actively and directly in spatial query, thereby reducing some of the confusion or obstructions introduced by superimposition techniques;

2- provide an on-the-fly basis for accessing the intelligence and information within the vector layers for better image interpretation

3- reduce the pre-processing requirements by embedding them internally within the process flow, i.e., on-the-fly, when appropriate and required.

The *intelligent imagery* concept will have future significant effects on many disciplines and applications. The importance of this concept can be explained through several examples:

1- Spatial Query and Analysis

a- Maps serve as powerful tools for handling and analyzing spatial data. With the advanced technology presented in a GIS, the geographical data within any particular area of interest can be stored in separate layers. Each layer represents a theme of topologically related features (e.g., road network layer, hydrographic features layer, forest polygons layer, contour layer, etc.). Complex spatial analysis for this particular area requires the combination or superimposition of all of these layers to completely perform the spatial analysis task. Relying only on these map layers alone or superimposed with an image as a backdrop to perform spatial analysis tasks suffers from several weakness, namely [Derenyi and Fraser 1996; Shears, 1998]:

i- Superimposition of all data layers at once complicates the visual analysis performed by the analysts to pre-examine the situation at hand and might hide important spatial patterns which might lead to wrong or imperfect decision criteria. A temporary solution is to handle a few layers at a time.

ii- Using remote sensing imagery as a backdrop underneath map layers will increase the visual interpretation of the situation at hand but only, in the case of handling a few data layers at one time.

iii- Maps represent the landscape boundaries as sharp lines and, based upon cartographers perspectives, might provide misleading information for spatial analysis.

Furthermore, superimposing map layers onto an image serving as a backdrop is not the best way to get the maximum benefit from the heterogeneous amount of raster and vector data available today for performing spatial analysis tasks. The superimposition solution might be sufficient if both the area of interest and the spatial analysis task are not complex (i.e., few data layers are involved in the analysis). As the complexity increases, more data layers should be involved and more confusion can arise by superimposing vector and raster data. With this in mind, the intelligent imagery method permits *"on-the-fly"* query of whatever data layers are available and necessary, all with less confusion.

The concept of querying an image covering a particular area with the support of hidden vector database layers will provide a better environment to perform such queries. Using the imagery as an interactive medium with the real world will provide a synoptic undisturbed view of the real world and with the support of the well-organized vector database this will help to effectively and efficiently query that image.

b- Using remote sensing imagery as an interactive medium has several advantages as stated by Clark [1974], and McKeown [1987]. Remote sensing imagery contains more information than vector maps. Users can better orient themselves with images rather than a map overlay, especially if they are not familiar with the area under consideration. People may even have more faith in images than maps. As stated by Shears [1998],

> *a symbolized vector map product is a representation of the landscape as defined by a cartographic specification or as filtered by a cartographer's pen. An image, on the other hand, is a real life picture. The content in an image far exceeds that of a conventional map product, because it does not conform to a set of feature extraction rules. As a result, image maps inherently include every detail that is afforded by the image resolution, which on 1 m imagery would allow hedges, sheep tracks, gates and underlying ground cover to be identified with virtual impunity.*

c- Within a spatial data infrastructure, image and vector data files might come from different organizations using different parameters, such as projections, coordinate systems, models, scales, etc. Furthermore, the image used may be at low rectification levels, i.e. not orthorectified. Integrating remote sensing imagery with GIS vector data only at the graphical level, as in the case of superimposing vector data on an image, requires that the vector data and the image have the same parameters. While differences in projection and coordinate systems may be solved by the time consuming task of reprojection, relief distortion in non-orthorectified imagery may have a large effect on the analysis task being accomplished. It is highly recommended that, the image has to be orthorectified first before it can be involved in any subsequent spatial analysis.

The problem of having files from different sources with different parameters might be solved to a large extent by the proposed intelligent system. Multi-resolution remote sensing imagery is used as a *consistent* reference layer and on-the-fly procedures to accommodate for variations in projection, coordinate system, and image rectification level (as will be explained in section 3.1.1) can be embedded within the query process. Without any superimposition, simultaneous display, and/or vector layer preprocessing or image transformation/rectification we can retrieve objects information from those vector layers, based on their corrected image coordinates and utilize them in the query process.

d- Baltsavias [1996] mentions that orthoimages can be used as geographic reference layers in which spatial operations like polygon retrieval, buffer, COGO function, etc., can be executed, and the results can be displayed. He mentions that the orthoimage provides a complete view of the world and permits visual interpretation by the operator. The proposed system would deal with not only orthophotos but also with partially rectified and unrectified imagery.

2- *Image Classification and Training Area Selection:*

One of the most important applications of combined vector data and remote sensing imagery is image classification. Digital image classification is "the process of assigning pixels to classes" [Campbell, 1995, p. 313]. The first step in most supervised image classification techniques is the selection of training areas that represent the terrain classes of interest. Training areas are usually obtained by using maps and aerial photos. As stated by Campbell [1995], the analyst is assumed to be familiar with the area of interest. Selecting the training areas is usually performed by superimposing map layers over image data or by the operator doing a visual match through simultaneous display of both data sets. In this regard, Hinton [1996, p. 882] states that:

> *to be of maximum benefit, the system used must be able to perform the so called raster/vector intersection query [Ehlers et al., 1991], (i.e., given an image and a polygon file, which pixels fall within which polygon?), without carrying out any conversion or raster overlays/combinations.*

With the intelligent imagery approach we can link the pixel coordinates to vector map coordinates, query the database, and display/save the results directly through the image. As a result, the analyst will be able to select the desired training areas without either data conversion or raster/vector overlay. Although this research is not concerned with implementing image classification functionality within the developed system, it would solve the problems that exist in trying to internally link the image and the vector information for that purpose. When extended to handle image classification tasks, the intelligent imagery concept/system may provide image classification with a *fast* and *semi-automatic* process of selecting the training areas and a better way towards an initial visual assessment of the classification results.

3- Image Interpretation

As reported by Jensen [1990], the aerial photo interpretation task is too time consuming, especially for a large area. Also, there is the risk that interpretation will vary among interpreters due to their different background knowledge. The method of intelligent imagery will provide an on-the-fly base to help interpreters reduce the risk of arriving at completely different interpretations in some cases.

4- Data Transfer and Conversion

As discussed by Laurer et al. [1991] and emphasized by Hinton [1996], to minimize any loss of accuracy and/or data quality, the data must be left in their native format without conversion. In the intelligent imagery system, raster remote sensing imagery and GIS vector data layers would be left in their native format, and an accurate linkage between the two established to integrate both data types. In other words, the approach would provide an integration strategy based on information exchange rather than actual data transfer. It would allow the user to exchange information between both data types, through an appropriate link, without the need for data transfer and/or conversion. In addition to that, it would allow for the analysis and the final decision to be made based on imagery that is a closer representation of the real world than vector maps.

The above-mentioned points are far from a complete list, but they give an idea on how the intelligent imagery concept might improve the ability of users to handle raster and vector data sets in a diversity of applications.

As it can be seen from this section, the intelligent imagery concept requires a certain interface design and functional process flow in order to handle on-the-fly imagery and vector data with different characteristics and rectification levels. The current integration tools and techniques rely on transferring data between systems, conversions, or superimposition/graphical integration levels. While existing remote sensing image processing and/or GIS software packages have the interface and functionality that facilitate the handling of existing integration techniques, they are not flexible enough to handle the intelligent imagery concept requirements. Fortunately, several vendors release software component-based toolkits as part of their product lines. These components encapsulate the objects required to handle the functionalities of the vendor's line of products. Software components can be utilized within a pre-designed system flow and interface to build new applications serving pre-defined tasks.

2.4 SOFTWARE COMPONENTS

In order to build the IISP, three options have been studied. The first option was to build the system from scratch. The second option was to use an existing off-the-shelf software package and modify its interface and functional process to suit the IISP requirements. The third option was to use software components technology. Within this research project, several reasons motivated the use of the software component technology, over the other two options, to build the desired system, namely:

1- It was not intended to build the system from scratch because this requires a very high level of programming experience and the existing possibilities of using any of the other two options.
2- As mentioned in section 2.1.2, existing software packages provide constrained tools to modify the system interface and functional processing which may constrain the desired application design and may be difficult to expand.
3- The interface of existing GIS and image processing software packages does not necessarily support direct image query.
4- Existing GIS and image processing software packages do not necessarily support the real-time transformation required.
5- Object integration furnished by the software component technology may provide the opportunity to automate the query process.
6- The desire to extend the product to a Web-based query system cannot be easily accomplished with stand-alone packaged software.
7- The need to utilize existing/same functionality but packaged in a different manner.

The promising flexibility of having ready-to-use software components was concluded/assumed by the author to give a reasonable opportunity to design such a system. In other words, the Intelligent Imagery System Prototype (IISP) developed within this research project was built by integrating spatial and aspatial software components within a visual development environment.

Several definitions for the term "Software Components" were found in the literature. For example:

> *Software components are objects or collections of objects that stand alone independent of hardware, software, and other elements of the computing environment [Hartman, 1997, p.16].*

> *A software component is a separately identifiable piece of software that delivers a set of meaningful services that are only used via a well-defined interface [Sprott and Wilkes, 2000].*

> *Software components are software pieces built and continuously improved by an expert or organization and encapsulating business logic or technical functionality [Pharoah and Brooke, 2000].*

In the author's opinion, a software component may be defined as a tool box that contains encapsulated existing software objects that provide specific functionality and which can be utilized/communicate/integrated with other components to build a new application. Considering Figure 2.6, for example, a new desired application was required to have map display, database query, charting, statistical analysis, and remotely sensed image classification capabilities/functionality. Software components, each of which represents a candidate of the previously mentioned functionality, should be obtained, integrated within a desired development/programming environment, and handled together following a specific processing flow in order to perform a new task. The new application is being assembled from different parts (components) obtained from different vendors following a well-defined standard.

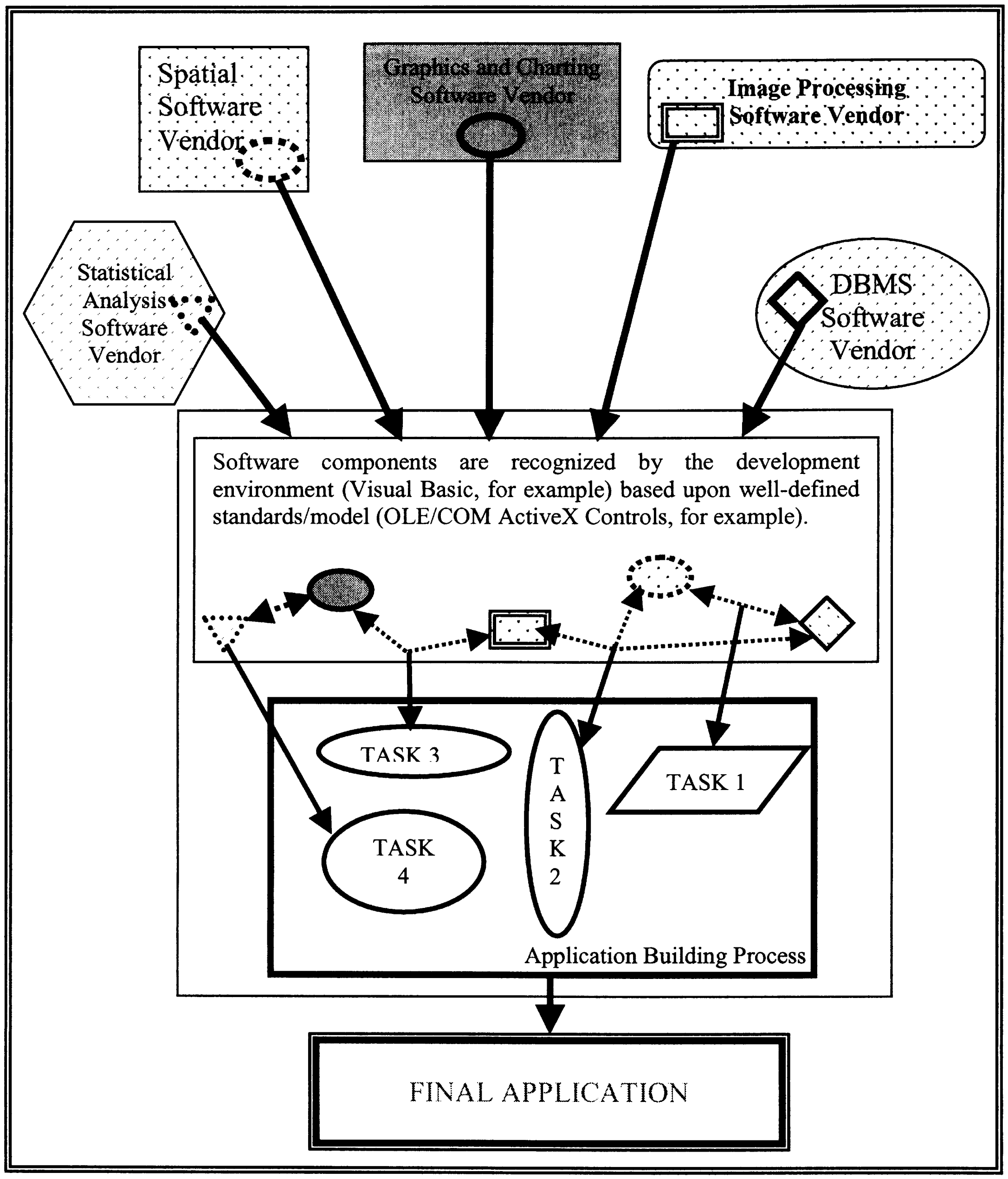


Figure 2.6
Application building using software components.

A full discussion of the history, the models for component architecture, and recent developments in component-based approaches are beyond the scope of this background research. For a well-documented description of this subject, see, for example Marangozov and Bellissard [1996]; Hartman [1997]; Meyer [1999]; Castek [2000]; SEI [2000]; Veryard [2000]; Pharoah and Brooke [2000]; and Sprott and Wilkes [2000].

A component-based development environment is an integrated approach by nature. In other words, compatible software components from different vendors serving different tasks are integrated within a pre-designed interface to create a certain application. Based upon the research conducted within the area of remote sensing and GIS integration, this approach may be an appropriate one for achieving the total integration between remote sensing and GIS in general and in implementing the intelligent imagery concept in particular. This is because the software components allow the integration to be performed at lower levels, i.e., the object level, and provide the flexibility in designing the applications' functional process.

In the author's opinion, the total integration should be application specific and can better be accomplished through a component-based approach. This approach is better than: (1) enhancing a specific off-the-shelf software package, (2) producing plug-ins and/or extensions, or (3) building scripts to link the existing software functionality, because it provides flexibility for expansions/improvements and is not restricted to a certain software interface or functional process.

Toolkits, also called component suites, are complex and comprehensive tools that enable users to pick up the required tools for their applications and to customize these tools to fit their needs. As identified by Hartman [1997], the motivation behind the development of toolkits is that the users might not need all the functionality offered by a GIS system, for example, but rather may need to simply strengthen specific functions, develop new functions, and ignore others. These functions could be broken down to sets of components/groups of functions and integrated by users into their own applications using a pre-defined developing/programming environment. As technology advances, new components can be added and integrated with existing ones to perform more advanced operations or even enhance the existing ones.

The main disadvantages of this approach are that it strongly depends on the existing technology, requires more effort to configure and to understand how the components work for proper integration, and is difficult to predict the integration errors before the integration begins [Vigder et al. 1996].

The process of releasing well-maintained software components has already been realized at least by the GIS community. For example, ESRI Inc. has released its MapObjects product which encapsulates the collection of objects required to perform many GIS functions. This component consists of "an ActiveX control and collections of dozens of automation objects that provides GIS and mapping functionality" [Hartman, 1997, p. 120]. Since MapObjects component was implemented as an ActiveX control, it can be embedded within any programming environment that supports that technology, such as Visual Basic, Visual C++, and Delphi. For reviews in the ActiveX technology, see, for example Hartman [1997] and Eidahl [1997]. Through well-maintained technical documentation and training programs, MapObjects is widely used now by developers and trained personnel within different national and international organizations to satisfy their spatial applications. As will be seen in section 2.4.1, the scenario is more or less the same within many other GIS vendors such as INTERGRAPH, CARIS, and MAP INFO, among others.

While many developed applications have been based upon a variety of software object components, they were based upon utilizing vector layers only, or superimposed with image data as the core target, consult section 2.4.1 for the usage history of various software components. No previous research efforts have been found that specifically treat the issue of utilizing the high resolution imagery as a generic reference layer in querying the real world in particular.

2.4.1 IISP: Development Environment and Components

The IISP developed within this research project is a first but essential step in implementing the Intelligent Imagery concept, which may be extended and improved to perform more spatial tasks and to integrate image processing algorithms based on a specific application requirement and as the technology advances in the future. As mentioned in section 2.4, the IISP was developed using software component technology.

The IISP functionality, as identified in section 3.1, is composed of two primary categories, namely, spatial and aspatial/user interaction functionalities. The aspatial/user interaction functionality could be handled using different components that exist or can be created within the programming environment itself. The spatial functionality requires spatial software components to be implemented.

Nowadays, many spatial software vendors, namely remote sensing and GIS vendors, are making their software technology available through a variety of software function libraries/toolkits/components, as shown in Table 2.1. A research study throughout the mature remote sensing and GIS market was conducted in order to narrow the choices and identify the appropriate tools to perform the previously defined *spatial* tasks/functions for this research project.

Table 2.1 shows the software vendors contacted for that task as well as their available toolkits/software components that may be usable for this research project. The selection of the appropriate software toolkits/components was based upon the following criteria:

Table 2.1: Software vendors contacted for the software components selection stage.

	Web Contact	**Software Development Components**	**Community**
ESRI	www.esri.com	MapObjects 2.0 a	GIS
Intergraph	www.intergraph.com	GeoMedia Customizable ActiveX Components	GIS
MapInfo	www.mapinfo.com	MapInfo MapX	GIS
Universal Systems (USL)	www.caris.com/	CARIS Spatial Framework	GIS
PCI Geomatics	www.pci.on.ca	No tools found beneficial for this research	Remote sensing
ERMapper	www.ermapper.com	C-based function libraries	Remote sensing
ENVI	www.envi-sw.com	No tools found beneficial for this research	Remote sensing
ERDAS	www.erdas.com	No tools found beneficial for this research	Remote sensing

1- The potential to satisfy the previously defined spatial functional requirements.
2- The compatibility with any of the standard visual programming environments, for a better and easier interface development process.
3- The previous usage history to avoid immaturity issues.
4- The existence of well-documented and regularly maintained components.
5- The support of various image as well as vector data file format support.

6- The support for image processing functionality and algorithms (either within GIS components or separate remote sensing/image processing components), For possible future improvements.

7- The available resources represented by the current license agreements available within the Department of Geodesy and Geomatics Engineering at UNB.

Tables 2.2 (a) through (e) summarize the main findings of this study.

Table 2.2 (a): ESRI MapObjects 2.0a (major characteristics).

Factor \ Vendor	ESRI (MapObjects 2.0a)
General Description	- MapObjects 2.0a is a collection of embeddable mapping and GIS components from ESRI that consists of an ActiveX control (OCX) and a collection of more than 45 programmable ActiveX Automation objects. [ESRI, 1999 a and b].
Commonly Used Programming Language	-Visual Basic -Visual C++ -Delphi -Power Builder
Usage History	- Used by several organizations for different mapping applications. For complete details, see (http://www.esri.com/software/mapobjects/mapobjectspartners/index.html).
Literature and Technical Support	-On-line help -Code examples -Published tutorials and books -Technical support -Training courses
Vector Data Supported Format	-ESRI Shape file -ARC/INFO Coverage -SDE Layer -CAD files -VPF files
Image Data Supported Format	ADRG - ASRP-BIL – BIP – BMP – BSQ – CADRG – CIB – CRP – ERDAS – GeoTIFF – GIF – IMAGINE – IMPELL – JFIF – MrSID – NITF – SUN – SVF – TIFF - USRP
Image Processing Functionality	None
GIS Functionality	A variety of GIS functionality is supported. For more details, see (http://www.esri.com/software/mapobjectslt/mo_vs_lt.html).
System Requirements	- Any development environment that supports OLE controls; - Requires Windows 95 or higher and Windows NT 4.0 or higher; - Applications created with MapObjects will have the same system requirements.
License	The Department of Geodesy and Geomatics Engineering at UNB has a site license for all ESRI products.

Table 2.2 (b): INTERGRAPH GEOMEDIA (major characteristics).

Vendor / Factor	INTERGRAPH (GEOMEDIA)
General Description	- GeoMedia contains 17 ActiveX controls (OCX), nine data servers, and over 90 programmable ActiveX Automation objects [Intergraph, n.d.].
Commonly Used Programming Language	-Visual Basic -Visual C++ - Power Builder
Usage History	Used by several organizations for different mapping applications. For complete details, see (http://www.intergraph.com/gis/quotes/).
Literature and Technical Support	-On-line help -User manual -Technical support -Training courses
Vector Data Supported Format	-ArcInfo -MGE Data Manager (MGDM) -Arc/ShapeFiles -Oracle® (Spatial Data Option) -FRAMME -MicroStation MGE
Image Data Supported Format	BMP – BSQ – GeoTIFF – GIF – TIFF
Image Processing Functionality	None
GIS Functionality	Several GIS and spatial query functionalities are supported.For more details, see (http://www.intergraph.com/gis/geomedia/spatial_ana.asp).
System Requirements	Windows 95, 98, NT v4.0
License	NA (evaluation copy may be requested).

Table 2.2 (c): MAPINFO MAPX (major characteristics).

Factor / Vendor	MAPINFO (MAPX 4.5)
General Description	MapInfo MapX is an ActiveX component that can be used by standard visual programming tools [MapInfo, 2000].
Commonly Used Programming Language	-Visual Basic -Visual C++ - Delphi - Power Builder - Oracle Express Objects
Usage History	Used by several organizations for different mapping applications. For complete details, see (http://www.mapinfo.com/company/customer_successes/index.cfm#telecom).
Literature and Technical Support	- Reference guide -User manual -Sample application - On-line help - Technical support
Vector Data Supported Format	-ESRI Shape files -MIF -Oracle8i Spatial
Image Data Supported Format	Raster images such as aerial photographs can be used to add a bitmap layer to maps.
Image Processing Functionality	None
GIS Functionality	Several GIS and spatial query functionalities are supported. For more details, see (http://dynamo.mapinfo.com/products/web/Features.cfm?ProductID=41).
System Requirements	Windows 95, 98, NT v4.0. At least 15 Mb of available disk space.
License	NA (evaluation copy may be requested).

Table 2.2 (d): UNIVERSAL SYSTEM Ltd. (USL)

CARIS SPATIAL FRAMEWORK 3.4 (major characteristics).

Vendor / Factor	Universal Systems Limited (USL) CARIS SPATIAL FRAMEWORK 3.4
General Description	"The CARIS Spatial Framework provides over 200 object classes contained within a number of C++ libraries. There are two main types of objects within the framework: Spatial and Non-Spatial Objects. Objects of both types provide a complete interface for customizing and extending applications within the Geographic Information System domain" [USL, 2000].
Commonly Used Programming Language	The most commonly used object-oriented development language can be used to call the functional libraries required.
Usage History	CARIS Spatial Framework is used by companies in many fields including telecommunications, marine, and defense. For more details, see (http://www.spatialcomponents.com/faqs/framework.html).
Literature and Technical Support	-On-line help -User Manual -Technical Support -Training Courses
Vector Data Supported Format	-ESRI Shape files -MIF -CARIS Files -Oracle8i Spatial
Image Data Supported Format	-GeoTiff -Tiff -BSB -HCRF raster
Image Processing Functionality	None
GIS Functionality	A variety of GIS functional libraries are supported. For more details, see (http://www.spatialcomponents.com/solutions/development/architecture.html).
System Requirements	- Framework V3.2 is available on the following platforms: SOLARIS 2.6 CC 4.2 HPUX 10.20 CC Windows NT 4.0 MSVC 5.0/MSVC6.0.
License	Site license for all USL products at the Department of Geodesy and Geomatics Engineering at UNB.

Table 2.2 (e): ERMAPPER Functional Library (major characteristics).

Vendor / Factor	ERMAPPER (C-Based functional Library)
General Description	"The ER Mapper C library can be used to create stand-alone programs that can read, write and manipulate ER Mapper images and algorithms" [ERMAPPER, 1999].
Commonly Used Programming Language	Not Specified.
Usage History	NA
Literature and Technical Support	-Code examples -Technical support
Vector Data Supported Format	-ESRI Shape files - ARC/INFO coverage
Image Data Supported Format	ERMAPPER
Image Processing Functionality	All functionality and algorithms exist within the ERMAPPER products.
GIS functionality	Restricted by the available functionality within ERMAPPER. For more details, (http://www.ermapper.com/cgi-bin/product/feature.cfm).
System Requirements	Not specified.
License	NA (evaluation copy may be requested).

Based on this study, the following conclusions can be extracted:

1- The GIS market is far more productive than remote sensing in terms of software components.

2- Most of remote sensing/image processing vendors provide software-specific developing languages to extend the algorithms/process that exist within the package. Based on this research result, no software components were available to be directly integrated and involved in building new spatial applications during the course of this research.

3- Most GIS software components *commonly* support the following vector format: ESRI shape files – ArcInfo coverages – MapInfo MIF/MID files.

4- Most GIS software components support a variety of raster formats.

5- Concerning GIS functionality, most of the GIS software components support simple as well as advanced spatial functionality and attribute queries.

6- From a remote sensing imagery perspective, at least, most of the components provide a means to display and manipulate the display of a variety of image formats;

7- None of the GIS toolkits supports image processing functionality.

8- GIS software components are used by a world-wide variety of users/organizations, which is an indication of a reasonably mature market.

A decision was made to use the ESRI MapObjects 2.0a software component as the core components to implement the spatial functionality within the IISP and to use Visual Basic 6.0 as the development environment because of the following three reasons:

1- In general, the pre-defined selection criteria factors are satisfied, except for the support of image processing functionality.

2- Full-featured component was available

2- Documentation and technical support for the components were available.

2.4.2 Components/Object-Based Remote Sensing and GIS Total Integration

Despite the fact that much has been accomplished towards the main goal of closer integration between remote sensing and GIS, namely: total integration, this integration is still being done in a data transfer mode. As mentioned by [Ehlers, 2000], *The current status can still be described primarily as data exchange between GIS and an image analysis system or an add-on of some image processing functionality to a separate GIS.*

As a practical example of such a statement, as mentioned previously in section 2.2.1.3, ESRI ARCVIEW attempted to embed some image processing functionality within the system and cooperate with ERDAS Inc. through plug-in/extensions called "IMAGINE Image Support" and the "Image Analysis Extension". The extensions allow ESRI ARCVIEW users to perform, for example, image enhancement and visualization-related functions, image-to-map registration, image categorization, raster conversion utilities, and change detection procedure [ERDAS, n.d. b]. Another example is that remote sensing image processing software vendors, such as PCI, ENVI, ERDAS, etc., import and export GIS vector layers and possibly the associated attributes to the system and display those layers on top of the image primarily for updating purposes. Ehlers [2000] also stated that,

Images are seen as another layer, integration consists more or less of a georeferencing and overlay process…which resulted in the creation of another 'dump' GIS layer.

The main characteristics of a system of total integration, as may be extracted from previous research efforts, for example Ehlers et. al. [1989]; Glenn et al. [1993]; Derenyi and Fraser [1996], Turker [1997]; Abdelrahim et al. [2000 b]; Ehlers [2000], is as follows:

1- A sophisticated design that can handle raster, vector, and attribute data in a transparent manner.

2- Integrated remote sensing and GIS functionality rather than the sum of both systems' functionalities. The user interaction requirements, as well as processing steps can be reduced leading to the automation of many tasks.

3- An integration at the lowest level of processing.

4- Minimal or no conversion procedures, such as raster-to-vector or vice versa.

5- A one-user interface overlying a shell of hidden processes that handles raster and vector data simultaneously.

6- Based on the user requirements and the application needs, the imagery must have access to all the capabilities, information and intelligence that exist within GIS vector databases. Consequently, the imagery can be used directly for performing the spatial task without any degradation of the image quality or to the processing flow.

To pave the way for such totally integrated systems, we should consider three issues:

1- In the author's opinion and as mentioned by other researchers, for example Ehlers et al. [1991]; Johnston et al. [1997]; and Ehlers [2000], the requirements for a totally integrated system are project-driven. In other words, it will be difficult to have a general-purpose integrated system. This is because a pre-defined process representing the integration procedure should be embedded in the system design itself. Within such systems, particular image processing/analysis functionality and GIS data analysis modules and procedures should be integrated/melt together in a well/pre-defined process and within a certain interface to serve a specific task/project/application.

2- Total integration not only integrates the functionality of the systems and performs tasks in a subsequent mode, but also requires the cooperation of the objects and methods associated with those functionalities. Consequently, objects can internally communicate back and forth and the required tasks can be accomplished in a cooperative and simultaneous mode leading to the automation for many of those tasks.

3- We have to be realistic in terms of vendors' cooperation and we should consider the commercial, administrative, and other non-technical/scientific issues/restrictions facing the total integration goal.

Intelligent imagery concept is considered to be a one feature of the total integration. In that feature the user can rely on remote sensing data as a reference layer in querying and analyzing spatial phenomena while having an internal/on-the-fly access to the GIS attribute and functionality. In addition it relies on performing the integration via an information exchange mode by integrating the required functionality objects.

2.4.3 Possible Total Integration Solutions

One possible solution for the GIS and remote sensing total integration issue lies in internal functional integration. Software component technology offers a great opportunity to the field of integration between remote sensing and GIS technologies. It enables the integration to be done at the function level through an integration between GIS and remote sensing functions' objects within the components and consequently to meet some of the previously mentioned total integration requirements.

For example, if we need to perform a polygon based image analysis for detecting and monitoring dynamic changes in land cover classes/conditions within an existing land use boundaries layer residing in a GIS database, Turker [1997] identified the process to be summarized as follows:

1- In the case of having high terrain within the image or if the image is taken with high elevation angles, i.e. an off-nadir viewing imagery, the image should be orthorectified in order to be displayed correctly with the vector layer. This process is beyond the application scope, requiring trained personnel and special software modules.
2- Identify the classification classes of interest (training areas).
3- Perform image classifications.
4- Identify the existing polygons within the GIS database along with their attributes.
5- Identify the polygon locations within the image by superimposition.
6- Identify the image pixels inside these polygons.
7- Perform a certain statistical procedure within each polygon to check that the pixels located within a selected polygon belong to the same class and represent the polygon's existing class/attributes. This is done using a specific threshold with a pre-specified confidence level.
8- Identify the significantly changed polygons so that they can be updated while the unchanged polygon can be ignored.
9- Perform a manual/user-driven inspection within each changed polygon in order to identify the significance of change and assign the polygon to its

appropriate new class. If necessary, divide existing polygons into new sub-polygons representing different classes.

10- Add the newly created polygon(s) to the appropriate layer and delete the old one in order to update this layer.

Turker [1997] developed such a procedure and reported that one of the major disadvantages of the environment existing at the time of his research was the need to switch back and forth between GIS and image processing systems, namely, between CARIS and PCI. Furthermore, the statistical assessment procedure and database updating had to be done manually or outside of the application's main process, requiring several additional processing steps. Another problem was that the assessment had to be done for one class at a time and that a new procedure/program to count pixels located within each polygon had to be developed.

Considering the previously outlined task/procedure and others, in order for a system to perform such a task in a seamless way both GIS and image processing functionality objects should be able to communicate based upon a pre-defined/developed interface to accommodate user-system interaction requirements. Two possible solutions may exist to do that. First, which may face administrative and commercial problems, both GIS and remote sensing vendors should perform technical cooperation and design such system for that specific purpose. Second, the software components, which hold the main required functionality objects for both GIS and image processing sides, should be released allowing developers within organizations to handle the communication issues based on their specific application requirements. If the second solution is attempted, the integration process will be achievable and the total integration goal will be approached faster.

In the author's opinion, if a components-based solution is attempted to address such an application, several tasks can be automated and the process accelerated. For example:

1- As concerned in this research (within the IISP), there will be no need to exit the application scope and orthorectify the image or to find a projection module to re-project the vector data. These two processes can be carried out on-the-fly by impeding the projection facilities and required orthorectifictaion techniques within the query process and can be accomplished with minimal interruption and only when needed.
2- Classification, assessment, changed polygon identification, and database updating processes may be performed in a single step as shown in Figure 2.7.

Other than the components shown in Figure 2.7, others may be used for interface and other application-related issues. This is just an outline/rough vision/design of the process to demonstrate how the software components may help in performing a better integration for one possible application. The whole procedure may be automated, even in the process of selecting the training areas. If known reference/unchanged land use polygons layer exist, the system can access the database/layer internally, perform the required transformation to accommodate the issues of different projections and image rectification level, and use them as the training areas representatives.

If efficiently programmed and if the necessary components have been integrated following a pre-defined procedure, this process can be accomplished by clicking a button and with little or no user interruption and/or process delay.

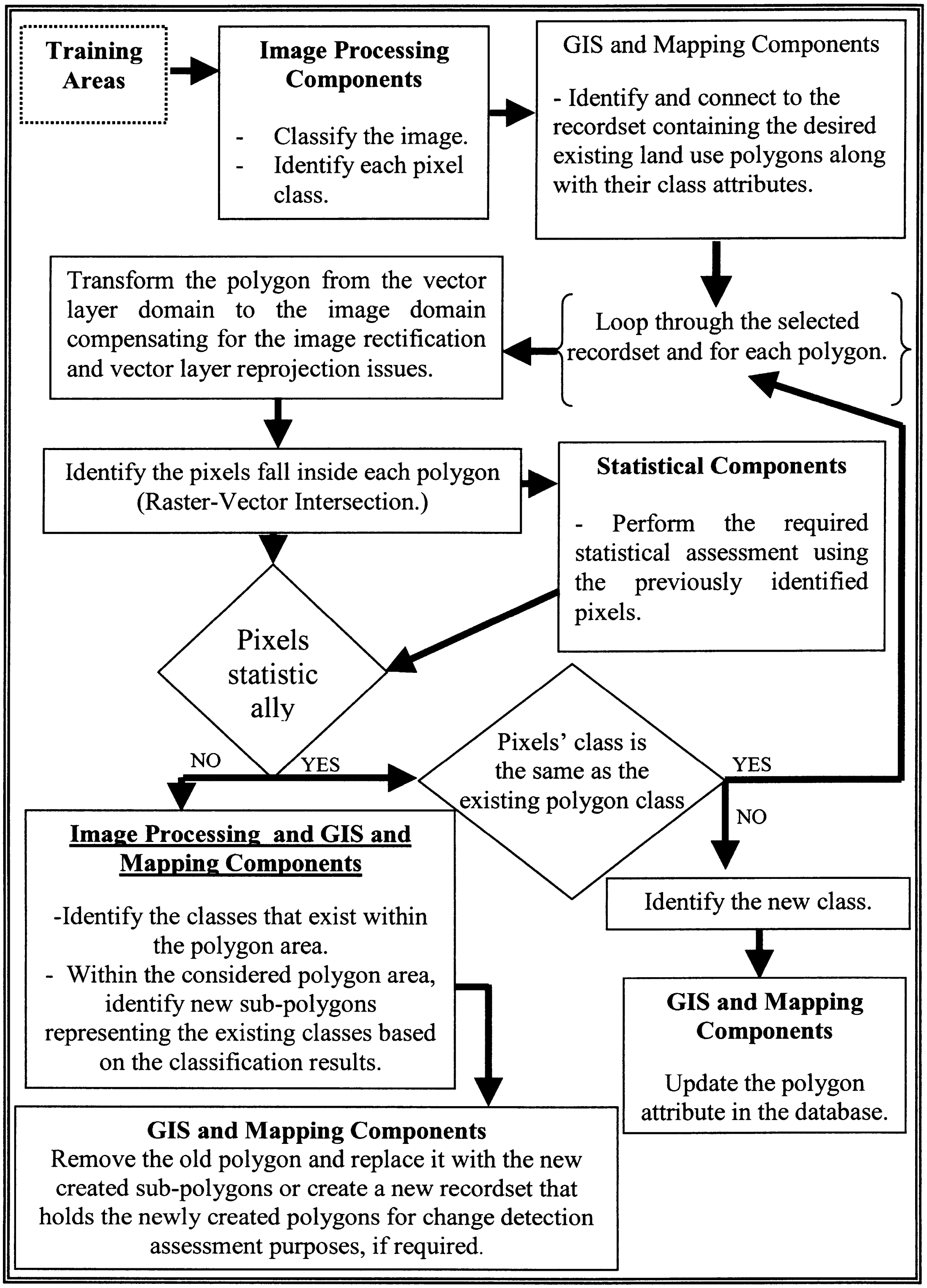


Figure 2.7
Polygon-based image analysis using software components approach.

It is not meant here to state that the software components approach requires no effort. The integration issues between different components as well as the study of the components' objects and associated methods have to be considered carefully. Once an application developer becomes familiar with the available tools, such an approach will be of great benefit for developing applications related to the concern of an organization and for quickly satisfying the dynamic needs of such applications.

The main message that the author is trying to communicate here is that the software components will:

1- Provide flexibility in creating and modifying desired application systems.

2- Allow for the creation of an application-specific workflow/process and map out the integration procedures between the selected components based upon the previously defined workflow.

3- Be suitable for expansion to handle new tasks by tracking down the system processing flow for modification, plugging any new desired components related to the new tasks (if required), and communicate the new objects with the existing ones, when appropriate, based on the newly developed processing flow.

4- Pave the road for faster integration methodology and open new opportunities towards the main goal of "Total Integration" or more accurately "Application-Oriented Total Integration" because it allows for the integration to occur at low processing levels.

As mentioned in section 2.4.1, GIS vendors are much more advanced in terms of releasing their software components than remote sensing image processing vendors. They are not only releasing the components but are also maintaining, updating, and furnishing their products with the technical support required. They also provide appropriate training programs to help the users in their development tasks. For total integration to be reached through the software components approach, both communities have to have well-established, maintained, and documented components that include most of their products' functionalities. The more functionality we may have within the component objects, the more applications we can develop, the wider the spectrum of integration-related applications can get, and the faster the total integration to be achieved.

Although the developments of such an approach still have to get more advanced in terms of the components and their functionalities offered by GIS and remote sensing vendors, it is the author's opinion that this direction will lead shorter and more directly towards the total integration goal.

Through the period 1999-2000 during which this research was undertaken, no compatible software component were being offered by any commercial remote sensing image processing vendors. In the author's opinion, that without the release of such remote sensing image processing software components, along with the appropriate technical information and training programs, expanding the IISP to build intelligence within the image processing by internally utilizing GIS vector data particularly and in remote sensing and GIS total integration in general will not be achieved soon.

Chapter 3

INTELLIGENT IMAGERY SYSTEM PROTOTYPE (IISP): FUNCTIONAL REQUIREMENTS, DESIGN, AND DEVELOPMENT

IISP, developed within the course of this research project, was built using software toolkits/components available commercially. It contains four major functional components and accommodates four integration cases. These functional components and cases present an increasing level of complexity and cover a wide range of interfaces, theoretical, and technical issues that might be of concern to different users and applications. The system was intended to be general-purpose, extendable, and component-based.

In order to build that system, the functional requirements of the system were first defined, and then a research study was conducted to allocate the available software components to satisfy these pre-defined system functional requirements. For each of the cases considered, the system interface requirements/design, internal design, implementation, and process flow had to be developed, and the problems and solutions within each element had to be identified. Finally, the process and tools involved in migrating the developed system to the Internet were designed and implemented.

The following sections present the steps taken towards the design, implementation, and Internet migration of the IISP.

3.1 IISP FUNCTIONAL REQUIREMENTS

The IISP is a system prototype demonstrating the concept of relying on remote sensing images, internally linked to vector data layers, to query the real world. The main development concerns within the IISP are to identify the user-image interaction issues, to establish the appropriate required link between both data sets, and to solve technical and theoretical issues associated with performing direct on-the-fly spatial queries based on the image scene. These will be performed for different cases, with increasing levels of complexity, and for four categories of spatial functionality. The following sub-sections present the IISP selected cases and functionality.

3.1.1 Rationale Behind Selected Cases and Functionality for IISP

One of the most important aspects of this research was to establish an efficient and appropriate link between the image and the vector layers. This link should be an on-the-fly, rather than a pre-defined one, i.e. the user should be able to load an image covering a particular area of interest, identify the vector layer(s) that share more or less the same extent as the image, insert certain required parameters, press a button, and start querying the image. Inappropriate links due to geometric shift or inefficient programming may produce slower performance, confusion, and inaccurate results.

Due to the fact that the image and the vector layers will not have the same characteristics within a spatial data infrastructure and the image may not be an orthorectified one, the link between the raster image and the vector databases/layers should be developed as to accommodate those differences. In other words, there are increasing levels of complexity in such an application. For example:

1- the image is differentially georectified (as with a digital orthophoto) and shares a common coordinate system and map projection with the corresponding vector data;

2- the image is differentially georectified but is related to a different coordinate system than the vector data;

3- the image is only partially rectified (perhaps with no corrections for relief displacement) and related to the same or different coordinate systems;

4- the image is not rectified, but a limited number of photo-identifiable control points are available.

To track the possible cases for integrating remote sensing imagery with vector databases/layers in general and to develop the IISP in particular, the following factors should be considered:

1- Projection: We may have different projection systems for both data sets.

2- Coordinate system: Coordinate systems might be different for both data sets.

3- Scale: The image and the vector data covering the same area may have different scales.

4- Level of image rectification: The image may be differentially rectified, partially rectified, or unrectified.

For the purpose of identifying all possible levels/cases, a framework was developed as shown in Figure 3.1. Figure 3.1a represents a case with the following characteristics:

1- differentially rectified imagery (orthoimage);

2- image and vector files with different scales;

3- image and vector files with the same projection and coordinate systems.

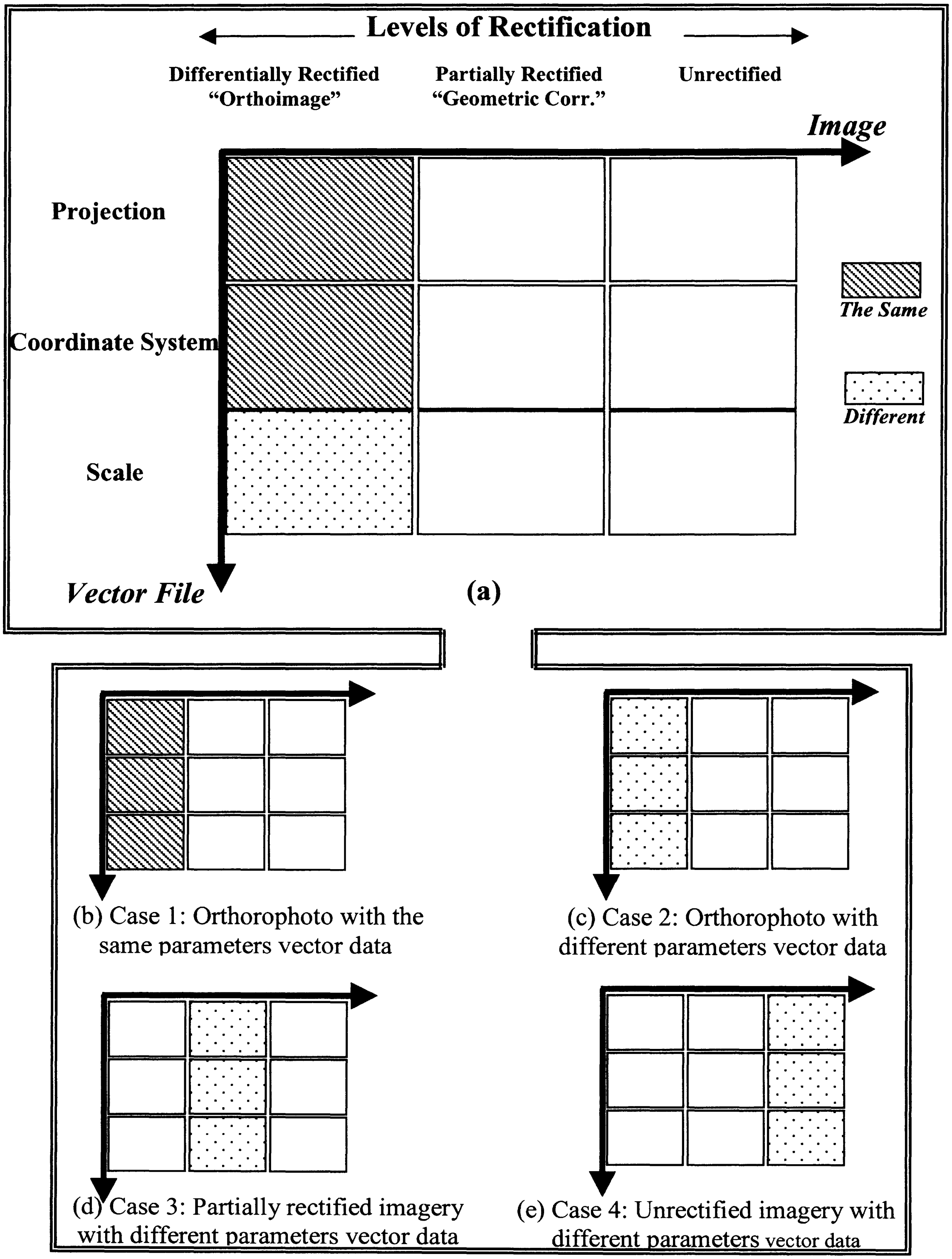


Figure 3.1
Framework of handling remote sensing imagery and vector files and IISP research cases.

Using the framework shown in Figure 3.1, we can identify all possible cases for combining/integrating remote sensing imagery and vector files. Figures 3.1 b through e summarize the cases that were investigated within this research project [Abdelrahim et al., 2000 b]. Four research cases are included within this research project, namely:

1- ***Case 1***: Differentially rectified imagery (orthoimage) and vector data covering the same area with the same scale, same projection, and same coordinate system for both data sets.

2- ***Case 2:*** Differentially rectified imagery (orthoimage) and vector data files covering different portions of the image with different projections, coordinate systems, and scales.

3- ***Case 3:*** Partially rectified imagery, geocoded, and vector data files covering different portions of the image having different projections, coordinate systems, and scales. (The image may or may not have a georeferencing information file). In this research IKONOS GEOCARTERRA™ imagery was considered.

4- ***Case 4:*** Unrectified image (for example, scanned aerial photo and captured video frame) and vector data files covering different portions of the image having different projections, coordinate systems, and scales (usually the image is not georeferenced).

In deciding on the functions that should be implemented within the IISP for the purpose of this research, we may consider four main broad categories, namely:

1- Location specific query.
2- Local query.
3- Global query.
4- Relative query.

The first category is concerned with the need to query a specific feature located on the image, i.e. the need to point to the feature and retrieve its information. This is very important for the purpose of fast clarification. The second category deals with a localized query concerning the features that exist within a certain area or across a certain strip/line. This category is very important in terms of patch/pattern analysis. The third category deals with inquiring about features of certain types within the whole image which may provide an indication about the significance of the image to a certain application within a certain area. The fourth category deals with querying features relatively. In other words, locating the features that exist within a certain distance of another feature or checking if a certain feature is located within this distance. This last category is important for planning purposes.

In order to satisfy these four categories, the IISP was designed to perform four main spatial functions, namely:

1- Pointing Query

2- Polygon/Line Based Image Query

3- Data Base Query

4- Buffering Operation

3.1.2 Requirements Definition

The first step in designing any new system or subsystem is the so-called requirement analysis/study. The main goal of this analysis is to identify and document the desired functions and/or operations that the new system or subsystem is expected to do [Razavi, 1999]. The requirement analysis stage may be defined as the stage in which the system requirements and functions are identified, outlined, and declared. The requirement study stage starts when recognizing that a solution is required for a specific problem [Jacobson, et. al. 1999]. As mentioned by Jacobson et al [1999], in that stage the developer should not care about how the system will do the function(s) but rather that it does them. The following sections illustrate the functional requirements of the intelligent imagery system prototype of concern to this research project.

3.1.2.1 IISP Functional Requirements Specifications

In identifying the system's functional requirements, we have to keep in mind that this system was designed to fulfil four major tasks in the four pre-defined cases developed by the author in section 3.1.1. A general description of the main functions that can be performed within the IISP cases may be presented as follows:

1- Pointing Query

With the pointing query, the user should be able to point to a certain feature on the image and, through a database search, retrieve the information related to that feature. This search can be accomplished within a search circle as well to accommodate any geometrical shift introduced within the image and the vector layer.

2- Polygon Based Image Query

The system allows the user to localize the image query by drawing a polygon/line of any shape on the image, performing a search through the selected layers based on a user-defined search criterion, and retrieving the results. Furthermore, the user may apply a search expression within the results to check whether a certain feature exists within the resulting features.

3- Data Base Query

With the database query the user is allowed to query a certain layer/database, the system identifies the features that meet the specified query, and the results can be displayed directly on the image. It is important to note that this query is concerned with the features that exist within the image extent. Therefore, the query process should filter the query results so as to produce the resulting features located within the image extent in case of having the vector layer(s) extent beyond the image extent.

4- Buffering Operation

In the buffering operation the user can select a feature on the image, a group of features, or a user-defined/drawn feature as the buffering core feature. By specifying the buffering distance, the system creates a buffering zone and retrieves all the information related to the selected layers within the buffering zone. In addition, the user may filter the results by using a search expression within the resulting features. The results then can be displayed.

All these functions require input data for processing. The input data are loaded and identified using the following two functions:

1- Image Loading and Display

In the image loading and display, the system should allow users to locate the image file within their machine, or download the file first from an FTP site, and display the image. In addition, the system should provide the user with an overview window showing a thumbnail of the image and the extent under consideration. The cursor location on the image, as well as the image file name, should be displayed to the user for more clarification.

2- Vector Layers Identification

The system should be able to allow the user to identify desired vector layers, store them in an archive, and allow the display of the name of each. Furthermore, the users need a tool to select/deselect certain layers from the archive they may need to be involved with within the query process and/or to remove any layer(s) from the archive. In addition, as the layers are selected for the query, the user may need to obtain an indication about the extent of a vector layer's coverage to avoid any confusion.

In general, the IISP was designed to include and implement four cases, each of which has four main functions. Within each function, the processing steps were divided into three main stages, namely "Data loading and identification", "Main function process", and "Result displays." Figure 3.2 shows an overview of the IISP organization. It is important to note that only the first three cases of the suggested four were implemented. The fourth case involving unrectified imagery was discussed theoretically, and a design outline was presented.

All of the previously mentioned functions are composed of a set of sub-functions. Appendix A shows the required sub-functions for all the cases within the IISP. In addition to the above mentioned IISP functional requirements, the system was required to be an Internet enabled application in order to allow for simultaneous access and for a better distribution approach. For that purpose, the system was designed as a stand-alone executable application, and then an Internet migration process was applied, as explained in section 3.6. The design requirements/considerations and implementation issues for the IISP cases are discussed in details in sections 3.2 through 3.5.

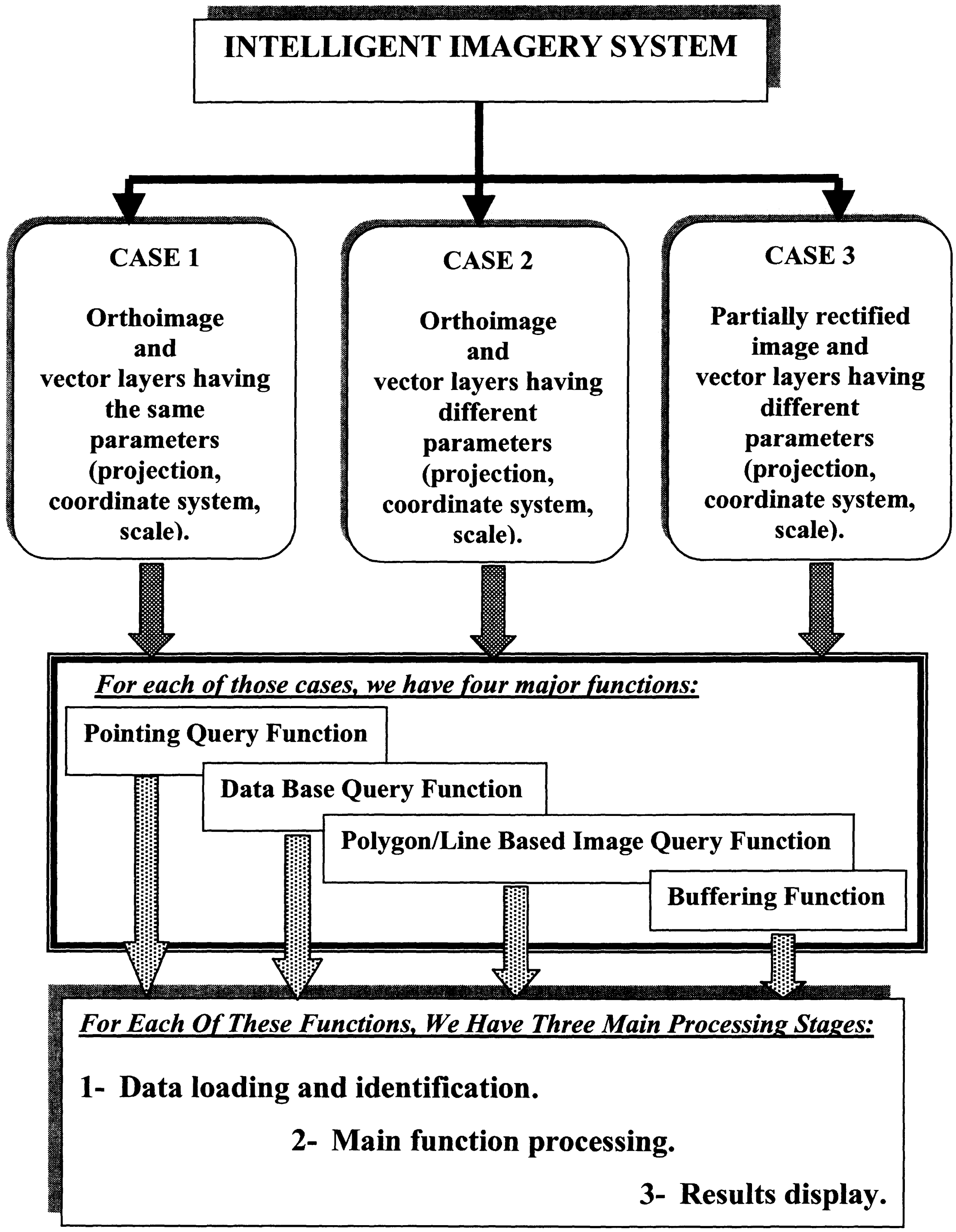


Figure 3.2
IISP: system organization.

3.2 IISP CASE 1: ORTHOPHOTO WITH SAME PARAMETERS VECTOR DATA

For Case 1 the assumption was that the image is differentially rectified (an orthophoto) and the vector data have the same projection, coordinate system, and scale as the image. In order to design, implement, and test this case the following data sets were used:

- An orthophoto obtained from SNB with a scale of 1:10 000 covering the Bouctouche area of New Brunswick, Canada. A sub-scene was extracted and georeferenced based upon the Stereographic and the North American Datum (NAD83) coordinate system.
- 1:10 000 building layer, river layer, property layer, and road network layer having the same projected coordinate system (Stereographic/NAD83).

All layers were extracted from SNB topographic map sheets in NTX format and converted to ESRI Shape files using ArcView3.2. Please, refer to Appendix B for metadata information.

3.2.1 Stage 1: "Data Loading and Identification"

The data loading and identification stage is considered to be an independent stage, because the processing steps as well as the requirements for that stage will remain the same no matter which function is to be performed. This stage is an essential step before performing any of the subsequent operations/functions. In this stage the user specifies an image to be *loaded and displayed* within the application and also *identifies* the vector data layers that will be accessed and involved in any of the subsequent image queries. Four main issues arise during the design of this stage, namely:

1- What are the main tasks that need to be accomplished in that stage, and what are the controls required for each?
2- How will the user interact with the data layers of interest (select/discard)?
3- The required data may exist within the local machine directories or may reside on an FTP server ready to be downloaded.
4- What is the appropriate interface design to fulfil the requirements of the previous issues?

In order to consider these issues, we first have to decide on the implementation technique of that stage. The loading and identification of the raster image and the vector files may be implemented either statically or dynamically. In the "static" approach, users have to identify/attach the desired files to the image. Every time they load the image, those files will be loaded in accordance. This method may be appropriate for applications that use the same files regularly, such as "change detection" applications. If a static link is selected, users do not have to look for the files covering the same area every time, rather they just load the image and the files covering the appropriate area as the image would be identified automatically.

The static link approach has the problem of excessive storage space, especially if we have the images stored in different databases. A vector file may share the same area or part of it with several images. Consequently, based on the storage technique, the vector layer itself or a pointer to it may be stored as many times as it may be associated with those images.

Dynamic implementation will allow users to identify the files they may need for the problem at hand only at the time of execution. In other words, users have to look for the desired files every time they use the image. In order to avoid storage duplication, since the IISP is intended to be a general-purpose system, and for more flexibility, the dynamic approach was decided to be a reasonable approach for that stage within the IISP.

The first step in designing the system within this stage was to emphasize the tasks as well as the required controls. Table 3.1 shows a summary of the main tasks for this stage and the controls required for each.

Table 3.1: Stage 1 -tasks and required controls.

TASK	Required Controls/Functions
1- Image file selection	- (File) menu item and (Load Image) sub-item
2- Image file loading and display	- (CommonDialog) control and MapObjects 2.0a Map control
3- Image file name appearance	- Picture Box
4- Vector layers selection	- (File) menu item and (Identify Layer) sub-items
5- Vector layers identification	- (CommonDialog) control
6- Vector layers name appearance	- Picture Box for each identified layer
7- File access through FTP	- FTP function and Internet Transfer control
8- Cursor location report	- Two Label controls for X and Y coordinates
9- Overall image/vector layers extent view window	- MapObjects 2.0a Map control
10- Image file and vector layer removal	- (Clear) menu item and (Clear Image/Layers) sub-items
11- Warnings – notifications	- Message Boxes

3.2.1.1 Interface Design and Processing Flow

To illustrate how the above mentioned tasks were performed, a detailed flow chart for the "Data loading and identification" stage was created as shown in Figure 1 in the FLOWCHARTS.PDF file included within the accompanying CD.

The starting point of interaction between the IISP and the user is the main form in the application (Form1). This form has been set to appear automatically on the screen when the application starts.

The first step in the "Data loading and identification" stage is to allow the user to select the desired image for loading and display and to *identify* the vector data layers of interest. To perform these two tasks in a convenient way, a menu item (File) was created. see Figure 3.3. Using this menu item the users can access an image file within their machine's local directories or to download the image from an FTP site. Figure 3.3 shows the main Form with (File) menu item and the (Load Image) sub-items.

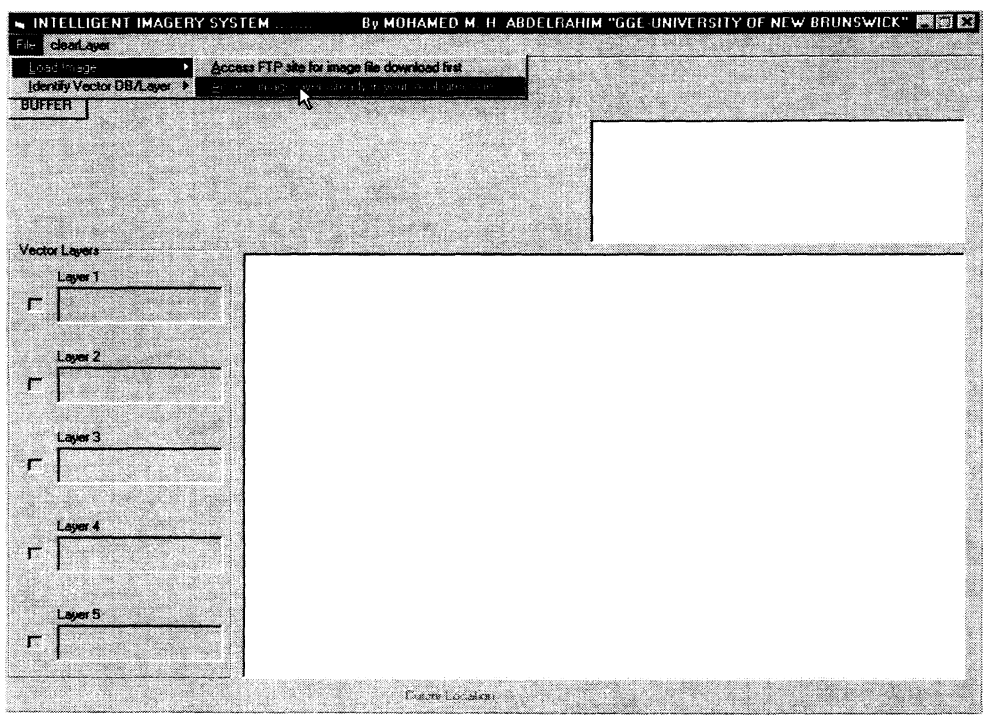


Figure 3.3
Image file loading.

To access an image file in a local directory, a CommonDialog control was embedded within Form 1. The control "ShowOpen" method was used to display a standard

Windows "Open File" dialog, as shown in the IISP_CODE.PDF file included within the accompanying CD. The user can then navigate through the local directories and allocate the desired image file for usage. To display the selected image within the application, MapObjects 2.0a Map control was used. Utilizing the Map control "Add Layer" method, the image is then displayed to the user.

Three more requirements need to be satisfied. First, users may need to know the name and the location of the image file they are using. Second, users also may require a cursor position reporting during the work. Third, users may require an overview of the image with an emphasis on the area displayed within the Map control and the vector layers extent. To perform the first task, a picture box was added to the form to display the image file name and location. For the cursor reporting facilities, two label controls were provided to report the cursor position at any place on the Map control. The problem here is that the cursor position is read by the system in the screen coordinate system. A conversion process was accomplished programmatically, to convert from screen to the ground coordinate system, using the Map control "ToMapPoint" method and the image world file that holds the georeferencing parameters. For the third requirement, an additional Map control was added to the form. When an image file is loaded and displayed within the original Map control, it will be also displayed within the second Map control. If during the process, the original Map control had to display only part of the image, as in the case of zooming and panning (as will be shown later in section 3.2.2), the second Map control will display a rectangle to dynamically represent the extent of the area displayed within the original Map control. In addition to that, showing the user the area shown by each vector layer is important in portraying to the user whether a certain feature on the image that does not exist within the vector layer is due to an updating or coverage shortage problem. When the user checks the layer to be used for the query, the label of this layer takes a certain color and a rectangle having the same color and representing the layer extent will show up within the second Map control. This was accomplished by declaring an object, assigning the layer extent to it, and then using the "DrawShape" method to draw the rectangle/object.

After performing the image file loading and display tasks, the next step is to *identify* the vector layers that may be needed for the subsequent image query operations. Similar to the image loading tasks, the user can use the (File) menu option, select the layer number, and identify that layer in the local directory. Figures 3.4 shows the steps to identifying a vector layer. The user can access an FTP site to download any desired vector file before identifying it.

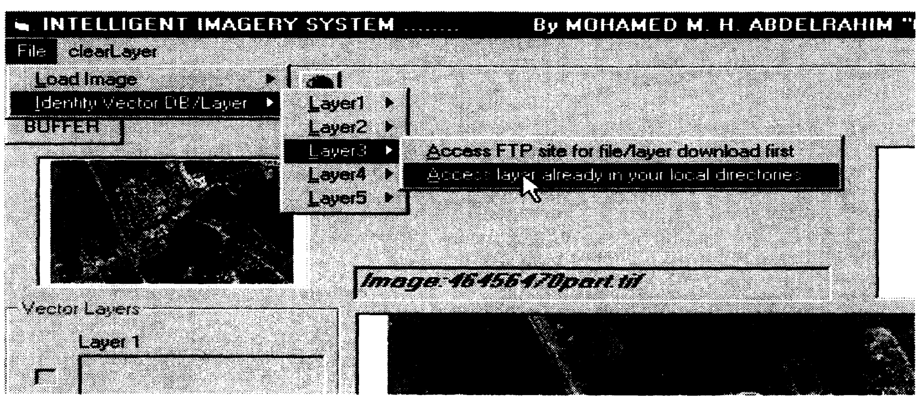


Figure 3.4
Vector layer identification.

In identifying any desired vector layer, the layer will not be displayed or even added to the Map control, but rather a new variable will be declared to hold the name and the physical location of the selected layer and then will be assigned those values. In order to allow the user to get an indication about which layers were selected, the selected layer name will appear in the picture box related to this layer. Figure 3.5 shows the main form (Form 1), a displayed image, and four identified vector layers. As can be realized from Figure 3.5, the form contains several other interface components that will be discussed later in sections 3.2.2 through 3.2.5.

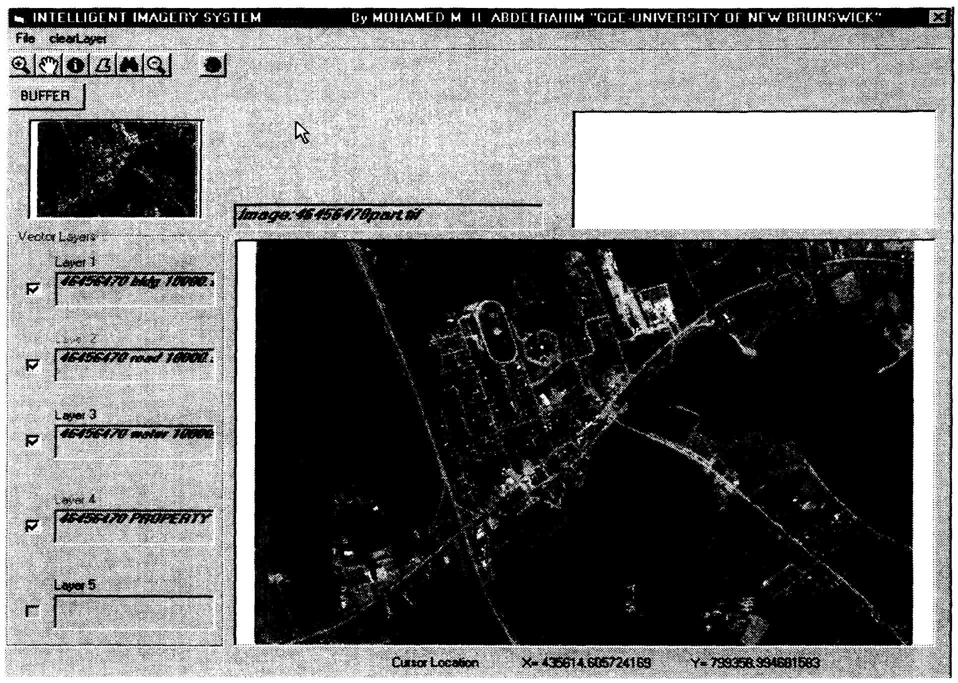


Figure 3.5
Data loading and identification stage completed.

3.2.1.2 FTP Function

In some cases neither the image file nor the vector data layers will be available within the user's local directories, but rather they may be residing on an FTP site ready for downloading. Instead of having the user start another application to access those FTP sites for downloading purposes, the user is provided with an FTP access and downloading functionality within the IISP.

In order to embed the FTP functionality within the developed system, the Internet Transfer Control (ITC) was used. ITC provides any Windows-based application with the tools to implement the HyperText Transfer Protocol and/or the File Transfer Protocol within the applications.

For the purpose of starting an FTP session and loading the desired files, several steps have to be accomplished in sequence. These steps are:

1- connect to the server;
2- pass the authentication;
3- get the folders/files list from the remote server;
4- navigate through this list;
5- identify the required file(s) for download;
6- download the selected file(s) into the local machine;
7- close the connection.

Figure 2 the FLOWCHARTS.PDF file shows a flow chart of these steps.

If the user selects to download an image file or vector layer(s) from an FTP site, the "FTP Session" form appears, as shown in Figure 3.6.

In order to connect to any desired server, the user has to provide the server address and the appropriate authentication information, i.e., a user name and a password, if required. Three text boxes were added to the "FTP Session" form allowing the user to type in this information. Following that is the connection and log-on procedure. A command button, labeled "Connect", was added to the form to execute the connection.

To connect to the remote server, having the ITC "Protocol" property set to (icFTP), the system makes use of the "Execute" method within the ITC. For the purpose of monitoring the connection process as well as any other process within the FTP session, a

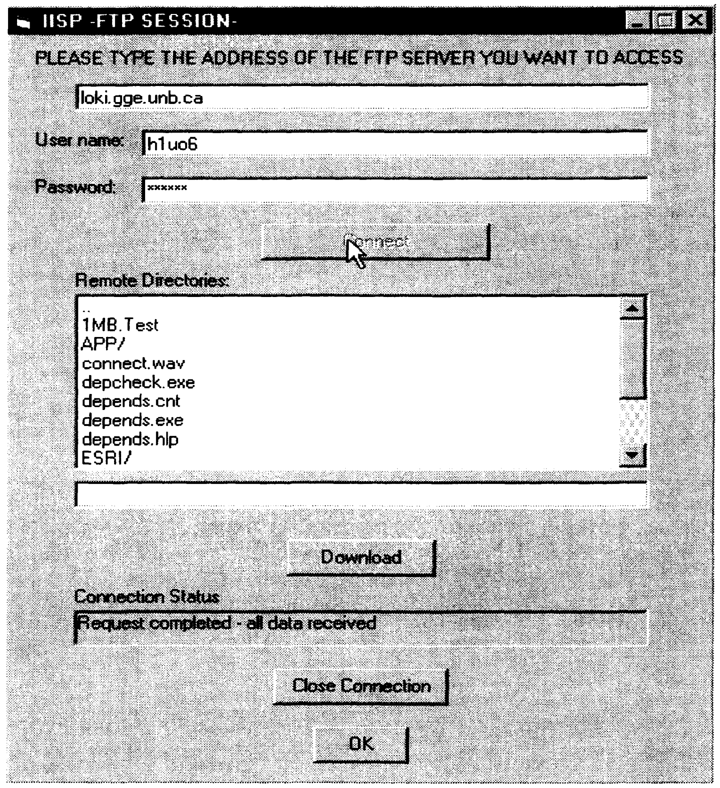


Figure 3.6
FTP session form.

Picture Box was added to the form to show the existing process status, which was accomplished through the ITC "StateChanged" event. When the control status changes from sending a request to receiving a response for example, the "StateChanged" event detects that change and sends to the user a certain pre-defined message reflecting the current operation/processing status of the control.

After establishing the connection, the user must retrieve all the files and folders existing within the remote server, navigate through them, and select one or multiple files to be downloaded. For that purpose, a ListView control was added to the form. When the connection is established, the "ServerList" function was designed to automatically retrieve the files and folders that exist within the highest level of the remote server file tree and display the results within the ListView. This initial files/folders retrieval was accomplished through the ITC “Execute” method using the key word “Dir” as an argument to retrieve the available data on the server, and the ITC “GetChunk” method to retrieve the buffer content within the ITC.

By double clicking on an item within the ListView control, when trying to navigate through the directories, the ListView “DblClick” event is triggered based on the item clicked. If the item clicked was a file, it will be highlighted and made ready to be downloaded. If the selected item is a folder, the system will navigate through the folder content using the ITC “Execute” method with the “CD” and “CDUP” keywords. “CD” gets the content of the directory and “CDUP” gets the higher level of directories on the tree. In order to download the selected file(s), the system calls the “DoFileDownload” function within the "shdocvw.dll" library file. This function allows for a common “File Download” window to appear and the user can specify the desired location to save the downloaded file, monitor the downloading process, and get notified of the download process completion in a convenient way. After downloading the required files, loading the desired image, and identifying the necessary vector layers, the system is ready to carry out any of the subsequent query operations designed within the IISP.

3.2.2 Pointing Query Function:

The main purpose of the pointing query function is to allow the user to inquire about a certain feature within the image, by pointing to that feature, and to quickly retrieve the attribute information associated with that feature from the database. Since GIS as well as remote sensing users tend to perform that function while the vector layer(s) are visible, and the IISP is totally hiding the vector data and connecting the image with the vector layers internally, several issues had to be carefully examined. These issues are:

1- How can we ensure that the information retrieved is related to the correct feature?
2- How will the user interact with the system, and what processing steps are involved?
3- How can we deal with any geometric shift, e.g. a georeferencing error, that may affect the final result or cause confusion?

As mentioned earlier in section 3.1, each function within the IISP functions has three basic processing stages, namely data loading and identification, main function processing, and results display. The data loading and identification stage (stage 1) is to be performed as explained in the previous section. The following sub sections summarize the tasks, required components, interface design, and processing steps involved within the pointing query function.

3.2.2.1 Stage 2 - "Main Function Process"-

In this stage, several required tasks had been identified. Table 3.2 summarizes the main tasks as well as the required components for this stage.

Table 3.2: Pointing query function (Stage 2 -tasks and required controls).

TASK	Required Controls/Functions
1-Zooming, panning, full extent capabilities	-A ToolBar with zooming and panning buttons - Zooming, panning and full extent functions
2- Vector layers activation/deactivation	- Check Box beside each layer Picture Box
3- Pointing query function triggering	-Pointing query button -Pointing query function(s)
4- Indicate the found feature(s) to the user	- FlashShape method
5- Search radius input (in case of using a search circle)	-Input box
6- Warnings – notifications	- Message Boxes

3.2.2.1.1 Interface design and processing flow

As shown in the flow chart for this stage, Figure 3 in the FLOWCHARTS.PDF file, the first step is to allow the user to indicate which layer to query. Since the user will point at one particular feature on the image, only one layer will be queried at a time. In order to indicate to the system which layer is to be queried, five Check Boxes were added to the form beside the previously designated layers' Picture Boxes.

During the pointing query function, as well as the other functions, the user may need to zoom, pan, or display the image to its full extent. A ToolBar was added to the form which includes several buttons, four of which were designed for these three functions. If the zoom-in button is pressed, the system then triggers the Map control "TrackRectangle" method within MapObjects 2.0a. This method allows the user to draw a rectangle around the area of interest, and then uses this rectangle extent as the new Map control extent, i.e. zooming. In order to zoom out, the user has to click on the Map control and the system takes these point coordinates as the center of the new map extent and reduces the original displayed map extent by a certain factor producing the new extent. In the Pan and the Full Extent case, the system makes use of the direct "Pan" and "FullExtent" methods that exist within the MapObjects 2.0a Map control.

In addition, one of the buttons within the ToolBar was dedicated to trigger the process related to the pointing query function. Assuming that we have a specific layer indicated for the query and the pointing query button pressed, the system is ready to start the process. The process starts when the user clicks on a feature/location on the image, point (P), and the Map control "MouseDown" event is triggered.

The process starts by calling the "Call Layer(i)" function, based on the layer selected. The "Call Layer(i)" function was built to connect the image and the desired vector layer and to use this connection for the rest of the process when required. This function creates a Data Connection object as well as a new Recordset object. Both objects work as shown in Figure 3.7.

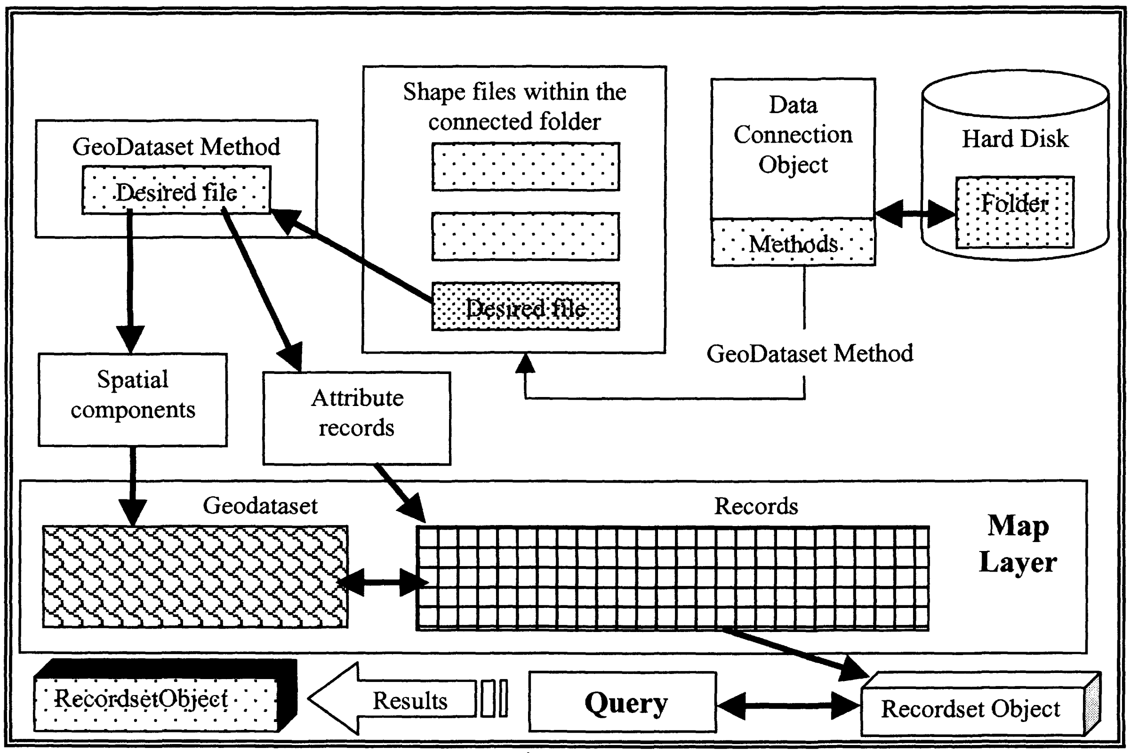


Figure 3.7
Vector layer connection and query process.

First, the DataConnection object identifies the location of the folder that contain the desired shape file layer. Second, using the file name and the Data Connection "FindGeoDataset" method, the system identifies the desired geographical data set/layer spatial and attributes components. Third, the Map Layer spatial data assigned the spatial components of the identified file by setting the Map Layer "Geodataset" to be the same as the Data Connection selected file/layer. The records attribute information of the map Layer are also set as attribute components of the Data Connection "GeoDataset". Finally, in order to navigate the records associated with the map layer, which definitely will be an essential operation for any subsequent query, we declare a new Recordset object and set it to hold the records of the map layer. The Recordset is a key object for navigating the records within the map layer. When performing any query, the system contacts the Recordset object of the map layer, performs the query, and returns a Recordset object containing the query results. After identifying the map layer's spatial and attribute components, the system identifies the screen coordinates of the point location (P). The system then reads the image georeferencing information and using the Map control "ToMapPoint" method, it converts the point location from screen coordinates to the ground coordinate system. If the map layer selected to be queried contains polygon features, the system then searches the layer's data set using the point (P) as the search point and the "IsPointIn" method. This method performs a point-in-polygon search within the selected layer. On the other hand, if the layer contains linear or point features, the Map layer "SearchByDistance" method will be used to initially search the layer with a search circle having a pre-defined radius of 30 m and the point (P) as its center. This is done with point and linear features layers because, due to geometric shift errors, it is rare that the user may click on a position that exactly corresponds to a linear or point feature position. The process would continue as follows:

1- To ensure that the resulting feature is the correct and desired one, it should be displayed to the user. For the purpose of fast recognition and less disturbance, among the methods that exist to display a feature, it was decided to use the Map control "FlashShape" method to identify the locations of the feature within the image pixels and flash the feature.

2- In the case of a linear or point features layer, the results may contain more than one feature. In that case the system will display the resulting features one by one allowing the user to select the desired feature, as mentioned in the next step.

3- It is important to realize that the resulting feature may not be the correct/desired one due to a geometric error, eg. georeferencing shift, map accuracy. To accommodate for this situation, a message box would be presented asking the user whether this is the required feature.

4- If the answer is positive, the system then loops through the layer associated Recordset object to retrieve the attribute information related to the feature. Otherwise, the system will display a warning message allowing the user to input a user-defined search circle radius and the process will continue until the desired feature is found, as shown in Figures 3.8 and 3.9.

5- If no features are found, and the user does not want to expand the search circle further, the system will display a message indicating the non-existence of the query feature and, consequently, this feature may need to be updated.

The resulting feature attributes, if any, will need to be displayed as well. This leads to the next stage, namely the "Results Display" stage.

3.2.2.2 Stage 3 - "Results Display"-

In this stage the resulting attribute of the query feature will be displayed. Table 3.3 lists the main tasks as well as the required controls/functions to perform those tasks.

Table 3.3: Pointing query function (Stage 3 -tasks and required controls).

TASK	**Required Controls/Functions**
1- Results display	- ListView control
2- Number of resulted features	- Text Box

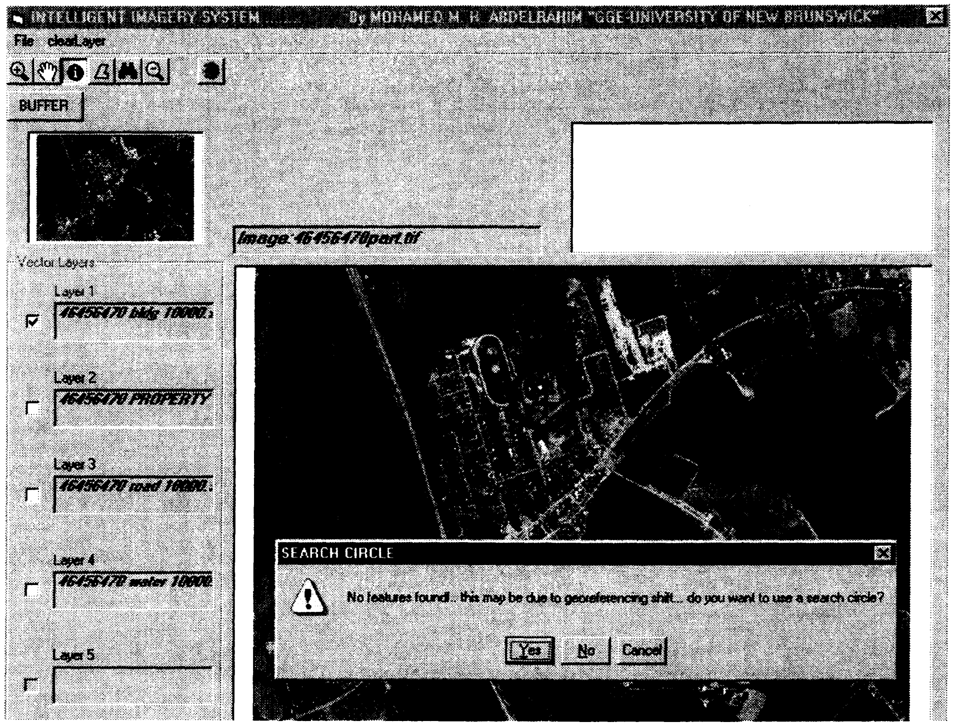


Figure 3.8
Pointing query function (georeferencing error message).

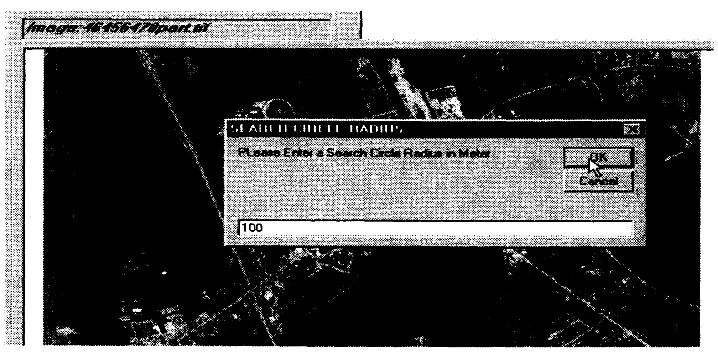


Figure 3.9
Pointing query function (search circle radius input box).

Since the pointing query function was designed for fast information retrieval of a specific feature on the image, there is no need to add another form to display the results. In other words, it is recommended to have the function and the results within one container, the main form (Form 1). In order to display the attributes of the resulting feature, a ListView control was added to the main form. In addition to that, a Text Box was added to the form to display the number of resulting features in case where more than one feature results, as in the case when using the "SearchByDistance" method.

Two main columns were set within the ListView control, namely Field and Value, to display the attribute categories and values within the layer under consideration for the selected feature. A loop through the created Recordset is performed to identify the fields within the Recordset and their values. These results are then displayed in the control as shown in Figure 3.10. Figure 4 in the FLOWCHARTS.PDF file shows the processing flow of that stage.

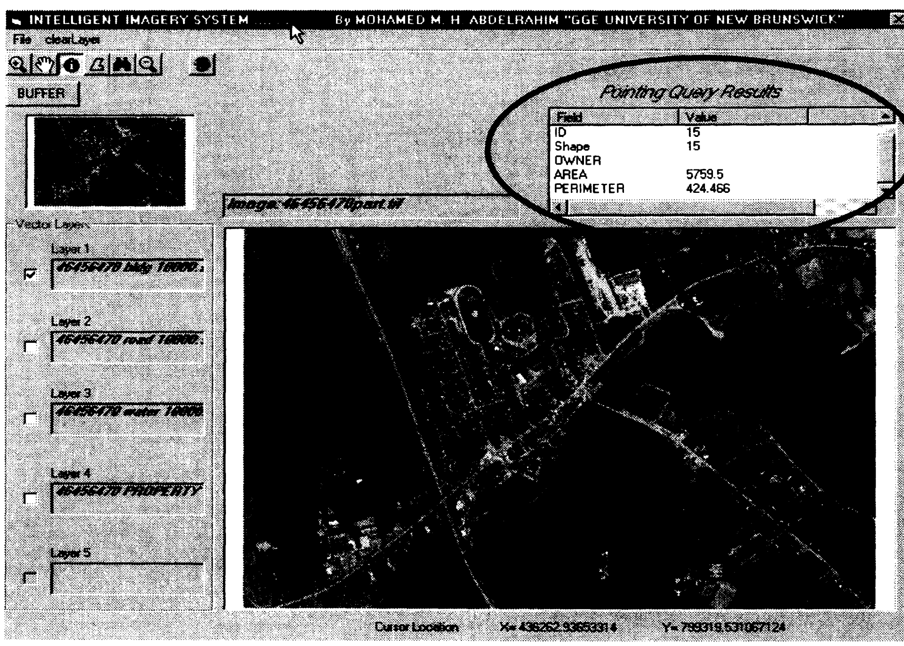


Figure 3.10
Pointing query function (selected feature attributes)

3.2.3 Polygon/Line-Based Image Query

In the polygon/line-based image query function, the user localizes the query by drawing a polygon or line around or by intersecting with the area of interest, selects whatever layers are to be queried at once, performs a certain search criterion, and retrieves the results that match the pre-defined criterion. As mentioned earlier, any function within IISP starts with the data loading and identification process stage as explained in section 3.2.1.

During the design of this function, the following issues had to be considered:

1- Users may need to apply a certain search criterion, such as retrieving only the features that intersect the user-defined/drawn polygon/line.

2- Users may need to retrieve all the features within/crossing the drawn polygon/line or they may need to check the existence of specific feature(s).

3- Users should be informed in case no results are obtained for a certain layer.

4- We should consider performing the query on several layers simultaneously.

The following sub-sections illustrate the next two processing stages within this function.

3.2.3.1 Stage 2 -"Main Function Process"-

The first step was to identify the main tasks and controls/functions required for this stage. Table 3.4 summarizes the main tasks and controls/functions.

Table 3.4: Polygon/line-based image query function (Stage 2 -tasks and required controls).

TASK	Required Controls/Functions
1- Polygon/Line-based image query function triggering	- Polygon/line-based image query button - Polygon/line-based image query function
2 - Polygon/Line based image query (Options) form to select the search criterion	- Form - Option buttons
3- Search expression input	- Input boxes
4- Warnings - notifications	-Message Boxes

3.2.3.1.1 Interface design and processing flow

As shown in Figure 5 in the FLOWCHART.PDF file, After loading the image and identifying the vector layers required for the query, the user has to select the required layers for the query by checking beside them. In order to execute the function, a Toolbar button was added to perform that task. When pressed, the system starts the function with a message box asking the user to select the query method, query by a user-defined polygon or line.

When the user performs the query, two situations may exist. Users may need to retrieve all the resulting features based on their search criterion, or perform a quick check to see whether a certain feature exists within the selected area, for example, retrieve only the hospital buildings that are affected by a flood within a flood zone area. In order to facilitate the quick check purpose, another message box will appear. It inquires whether to perform/apply a search expression within the results, as shown in Figure 3.11 a. If the system gets a positive response, the system was designed so that a number of input boxes will appear, based on the number of selected layers, asking the user to input a search statement, as shown in Figure 3.11 b. On the other hand, if the system gets a negative response or if any of the input boxes are left empty, the system will retrieve all the results within the user-defined polygon/line.

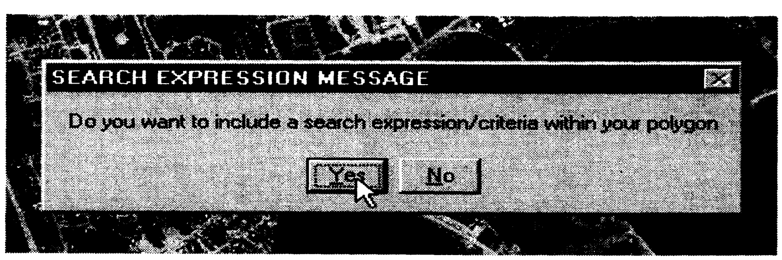


Figure 3.11 a
Polygon/line-based image query function (search expression message).

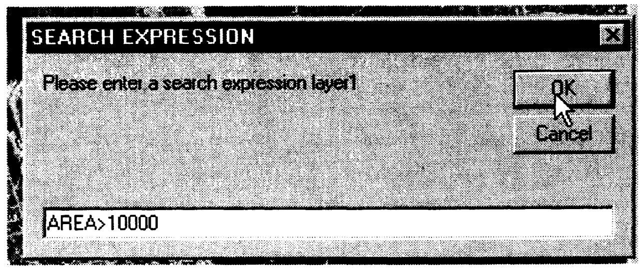


Figure 3.11 b
Polygon/line-based image query function (search expression input box for layer 1).

The next step allows the user to perform the intersection operation between the user-defined polygon/line and the selected layers and to retrieve the results based upon a certain criterion. MapObjects 2.0a provides several search criteria, which in the author's opinion cover most of the requirements for this function. In order to allow the user to select the desired search criterion, a "Polygon/Line-Based Image Query (Options)" form was created. This form contains the search criterion description, as shown in Figure 3.12.

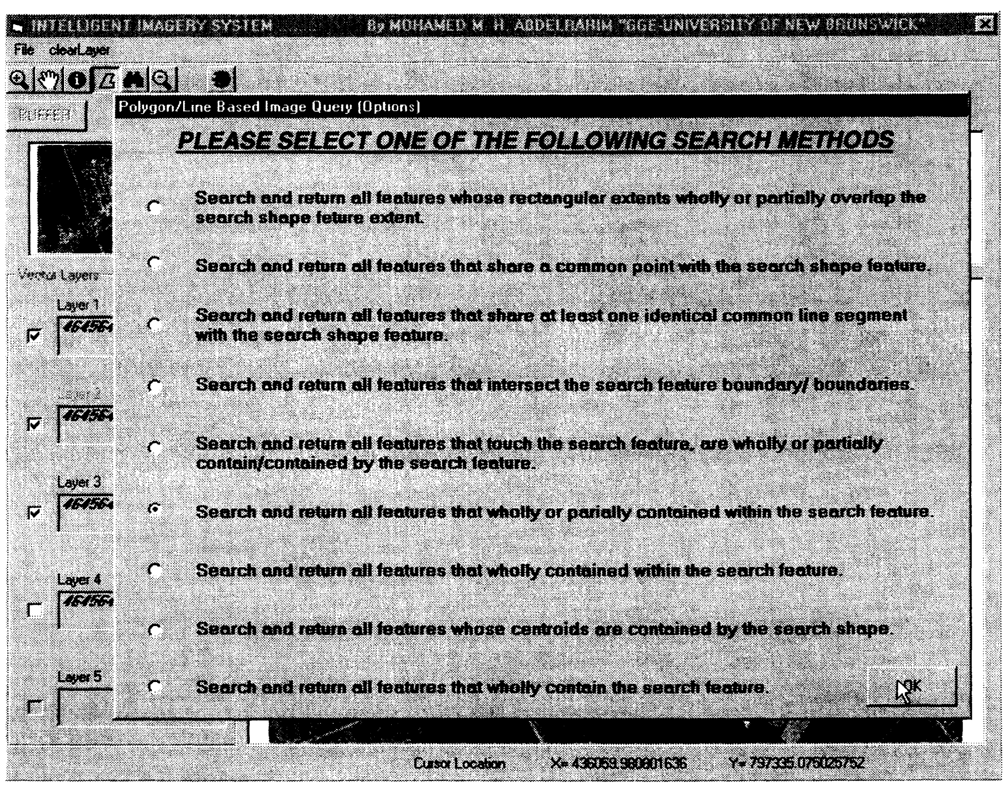


Figure 3.12
Polygon/line-based image query function (searching options form).

After selecting the desired search criterion, the query process starts when the user draws the polygon/line on the image around/crossing the area under consideration, as shown in Figure 3.13. This triggers the Map control "MouseDown" event and functions that are related to the pressed button on the Toolbar.

Using the Map control "TrackPolygon/line" methods, the system keeps track of the user- defined polygon/line. A new polygon/line object will be created and assigned to the user-defined polygon/line. Using the image georeferencing parameters, stored in a TIFF WORLD file (.tfw), the system relates the drawn polygon/line to the ground coordinates (and to the vector layers domain, accordingly).

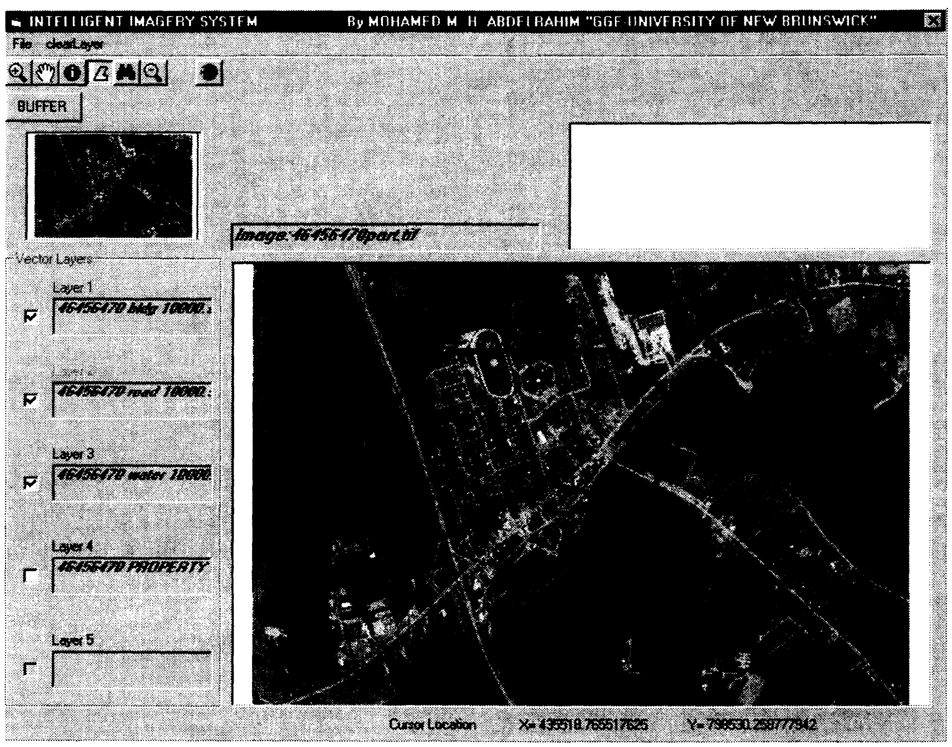


Figure 3.13
Polygon/line-based image query function (user defined polygon).

Based on the selected layers, the system calls the function "Layer(i)" in order to connect the previously defined files to Map layer and Recordset objects, as explained in section 3.2.2.1.1. A Map layer "SearchShape" method is then performed in each of the selected layers using the user-defined polygon/line, the selected criterion, and the search expression as its arguments. A new Recordset object will be created and assigned the resulting query features. Finally, the results are ready for display.

3.2.3.2 Stage 3 -"Results Display"-

The aim of the results display stage is to display the results obtained from the main function processing stage and to provide flexibility to the user to manipulate the results within the display tools. Two main questions were raised during that stage, namely:

1- How would these results be displayed in such a way as to allow the user to:

    a- display the results totally or partially;

    b- emphasize certain features;

    c- carry the results into subsequent operations, without any interruption to the image scene?

2- What are the required displaying tools according to the previously mentioned requirements?

Table 3.5 summarizes the tasks and the required controls/functions for this stage.

Table 3.5: Polygon/line-based image query function (Stage 3 -tasks and required controls).

TASK	Required Controls/Functions
1- Display the results	-Display forms
2- Attributes display	- ListView control
3- Number of record display	- Text box
4- Flash all the results	- Command button
5- Flash any particular resulted feature	- Flashing Function
6- Zoom or pan to any particular resulted feature	- Zoom/Pan buttons - Zoom/Pan functions
7- Save the results	- Command button

3.2.3.2.1 Interface considerations and processing flow

As shown in the function workflow, Figure 6 in the FLOWCHARTS.PDF file, the input to the results display stage consists of the resulting Recordset objects obtained from the performed query. The user should be informed when there are no resulting features for any of the previously selected query layers. In that case, an appropriate message would be sent to the user and the results obtained from the other layers will be displayed.

To allow the user to handle the results obtained from different layers with more flexibility, the system was designed to separately display the results obtained from each layer. In other words, several display containers, having the same tools, were developed, one for each layer. The display containers were designed as shown in Figure 3.14.

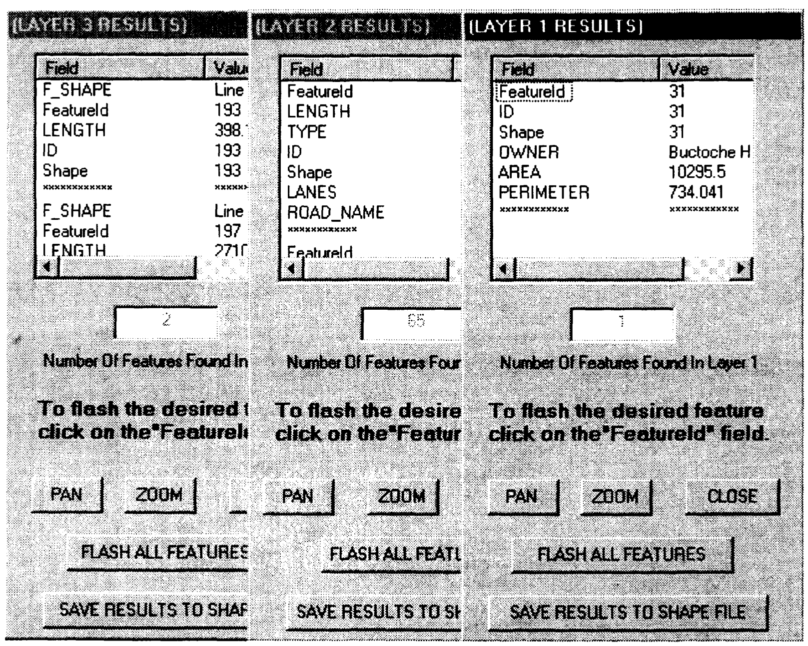


Figure 3.14
Polygon/line-based image query function (results display forms).

Each of these containers is a form that contains different controls. In order to display the results, a ListView control with two columns labeled "Field" and "Value" was added to the display the records. The system first loops through the resulting Recordset to obtain all possible attribute fields names and their values for each record/feature. Since the user may need to know instantly the total number of the resulting features, this number is obtained through the Recordset "Count" method and displayed to the user in a TextBox.

The first action that the user may perform is to display all the results in order to see and judge the distribution of the resulting features and their locations against the previously drawn user-defined polygon/line. A command button was added to the form allowing the execution of that task. When this button is pressed, the system creates a new object. This object works as a container for the whole resulting features. The system loops through the Recordset, adding each feature into the created object/container. Every time a new feature enters the container, the Shape "Union" method is used to perform a union operation between the feature in the container and the new feature. By the end of this operation, a new feature is created representing the union of all resulting features and this is then flashed to the user.

After examining the results globally, the user may need to closely examine a specific feature. A function was added to the display results form that allows the user to click on the "FeatureId" attribute and the system then loops through the Recordset, identifying the feature using the feature "ID" value, and flashes the feature. If the user wants to zoom/pan to that particular feature, the same process can be accomplished. When the user presses the Zoom/Pan command button provided within the form, the system performs the same process as when flashing a feature, but by applying the Zoom/Pan function instead of the "Flash" method.

The Zoom function creates a new Rectangle object and sets that rectangle to the selected feature extent, scales that rectangle up three times by using the “ScaleRectangle” method (A method to reduce or enlarge a rectangle object) within MapObjects, and then sets the Map control extent to that rectangle extent.

The Pan function sets the center of the Map control extent rectangle to the selected shape extent rectangle center. The selected shape will then be panned to approximately the center of the Map Control.

Furthermore, the user may need to carry the results into subsequent operations. Another command button was added to the form allowing the user to save the results as a Shape file. The same process applied in identifying a vector layer is applied in saving the results with mainly two differences. First, the system calls the Microsoft CommonDialog control “Save” window to show up instead of the “Open” window. Second, the system makes use of the MapObjets’ Recordset “Export” method (a method that exports a recordset to an ESRI Shape file) to create the new layer. The user then gets notified of the success or failure of creating the layer.

Finally, a command button was added to the form allowing the user to close the form and perform another function.

3.2.4 Database Query

In the database query function, the user is querying the database/layer table first by using a certain “where” clause of an ANSI SQL statement, and the results can then be manipulated. The following issues should be kept in mind when designing this function, namely:

1- In some cases, the vector layer/database extent may exceed the image extent and we may get some of the resulting features outside the image extent. Consequently, we should obtain the results that are within the image extent.

2- In order to create the search expression, how would the user specify the search expression and how will it be built?

The first stage of this function, the data loading and identification process, is performed in the same manner as explained in section 3.2.1. The following sub-sections illustrate the next two processing stages within this function, as shown in Figure 7 in the FLOWCHARTS.PDF file.

3.2.4.1 Stage 2 -"Main Function Process"-

The first step was to identify the main tasks and controls/functions required for this stage. Table 3.6 summarizes those tasks and controls/functions.

Table 3.6: Database query function (Stage 2 -tasks and required controls).

TASK	Required Controls/Functions
1- Database query function triggering	- Toolbar button - Database query process function
2- Database query builder	- Form
3 – Search expression building tools and execution	- Two ListView control - Twelve Command buttons - One Text box
4- Results filtering to bring the features located within the image extent only	- Filtering function
5- Warnings - notifications	- Message Boxes

3.2.4.1.1 Interface design and processing flow

The first step in this process is to select which one of the identified layers is to be queried. The user performs this step by checking the check box beside the layer to be queried. A button was added to the main form Toolbar to execute the process. When pressed, the “ DataBase Spatial Query” form appears, as shown in Figure 3.15.

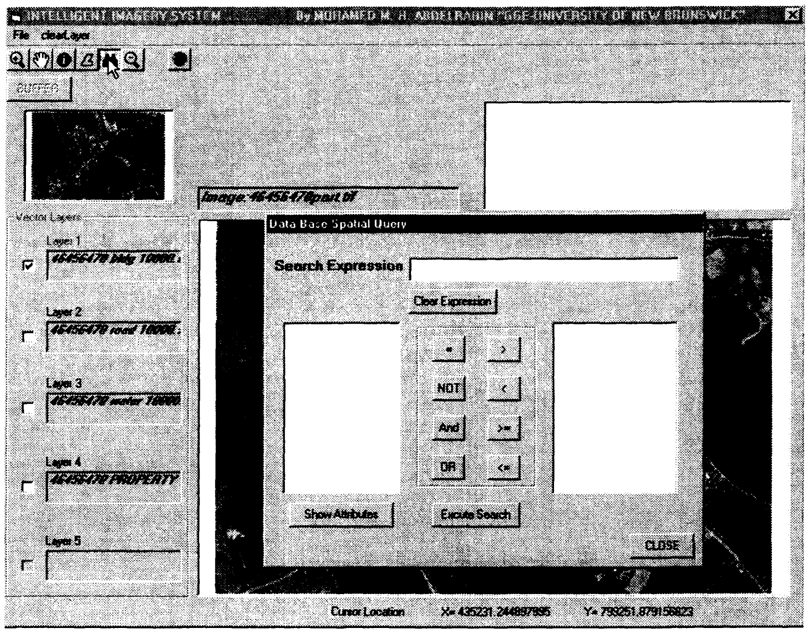


Figure 3.15
Database query function (main form).

The form contains a Text box to allow the user to input the required query. The required query expression can be entered by typing the expression manually or by creating it using query builder tools. A command button labeled “Clear Expression” is provided within the form to remove the expression if required. Before entering the query expression, the attributes as well as the records within that layer should be displayed. For the purpose of showing the layer attributes, a ListView as well as a command button, labeled “Show Attributes”, is provided. When the user presses the Command button, a search through the Recordset assigned to the selected layer is being conducted and all the attribute fields related to that layer are returned, as shown in Figure 3.16.

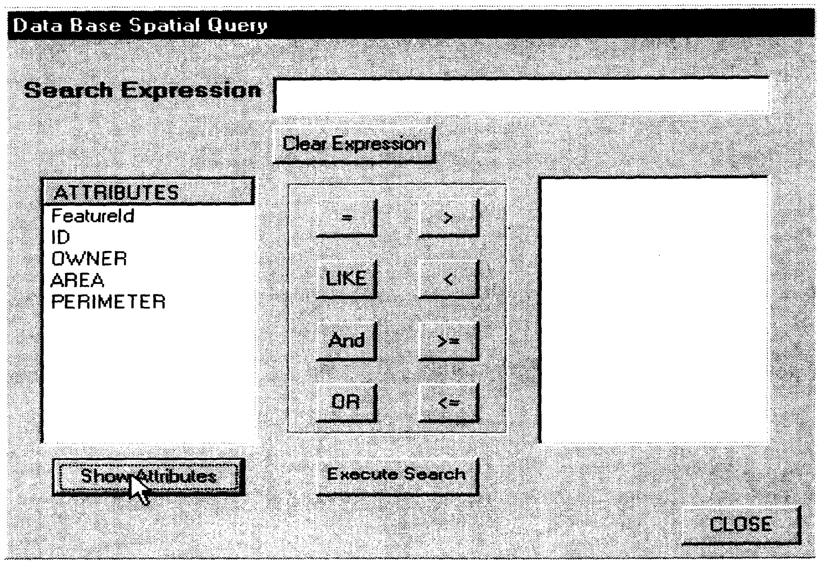


Figure 3.16
Database query function (selected layer attributes).

In order to display the range of records associated with one attribute, a second ListView was added to the form. When the user clicks on any of the attribute fields within the first ListView the attribute records/values associated with that attribute field will be displayed. This is done by performing a loop through the Recordset and the first string within the query expression statement will be displayed in the Text box. In order to build the rest of the expression, eight command buttons were provided within the form which contain some possible operations needed to build the expression directly. If the required operation is not included, the user has to enter the search expression manually. When any of the buttons is pressed and followed by a selection of the attribute value from the second ListView, the rest of the expression can be built, as shown in Figure 3.17.

For the purpose of executing the search expression, a command button, labeled "Execute Search", was added to the form to execute the search. When pressed, the Map layer "SearchExpression" method is performed within the selected layer using the built-in expression as its argument. A new Recordset object is created holding the query result, which is ready for display.

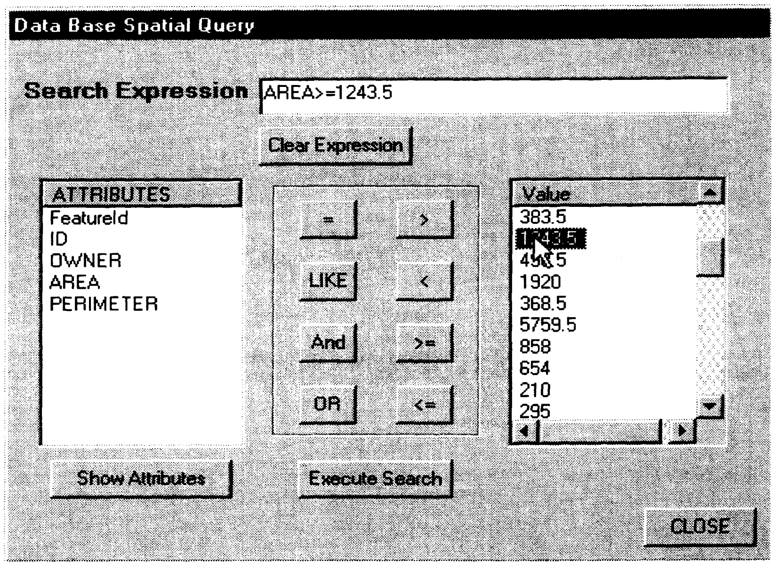


Figure 3.17
Database query function (building search expression).

It is not necessarily true that the image extent will be the same as the vector layer extent. In other words, some of the features resulting from the Database query may be located outside the image extent. Since we are interested in the features within the image extent only, the system was designed so that the results obtained can be filtered. The filter works by comparing the features extent with the image extent. For any feature within the results, if the intersection between the feature extent and the image extent is void (null), then the system skips the feature and removes it from the results.

3.2.4.2 Stage 3 -"Results Display"-

In this stage the results obtained from the previous stage are displayed. The processing steps, as well as the functionality, within this stage is the same as the polygon/line based image query function (see section 3.2.3.2). One exception is that the results obtained from the query have to be filtered in the case of displaying or saving functions to represent the features that match the query expression and which are located within the image extent.

3.2.5 Buffering Function

In the buffering function the user identifies a certain feature or group of features as a core of the buffering operation and specifies a buffering distance. The system then performs the buffering operation and retrieves the results. The Data loading and identification stage takes place before carrying on with this function. Several issues had to be considered when designing this stage, namely:

1- How may the user define the buffering features and what are the possible situations?
2- In the case of having a linear features layer, are there any problems due to the fact that those features are stored as segments, i.e. each feature composed of several segments, and how to solve this problem?
3- The user may need to retrieve all the features within the buffering zone or just check for the existence of a certain feature (within one or multiple layers).

3.2.5.1 Stage 2 -"Main Function Process"-

The main tasks and the associated required controls/functions for this stage had been identified as shown in Table 3.7

Table 3.7: Buffering query function (Stage 2 -tasks and required controls).

TASK	Required Controls/Functions
1- Buffering query function triggering	- Buffering query button - Buffering query function
2- Select how to identify the buffering feature (one feature, group of features, user drawn feature)	- Message Boxes
3 – Identification of the type of shape the user may draw to use as a core for the buffering operation	- Form - Option buttons
4- In case of linear features, connect the feature segments	- Connection function
5- Selection of the layer(s) to perform the buffering operation on	- Form - Check boxes
6- Buffering distance input	- Input Box
7- Buffering results filtering by search expression	- Input boxes
8- Warnings - notifications	- Message Boxes

3.2.5.1.1 Interface design and processing flow

In order to start the buffering operation, a command button was added to the main form. Within the buffering operation, a core feature for the buffering operation has to be identified first. The feature selection procedure was designed to satisfy most of the user's needs within this function. First the user may need to select a certain feature on the image to start the buffering operation, which is an obvious case. Second, the user may need to draw a certain shape on the image to start the operation. This will be very useful for planning purposes, if, for example, the user wants to retrieve all the properties that could be affected or required for purchase in case of having a new road construction plan. Third, the user may need to carry on a certain process and use a group of features resulting from previous operations as the core for the buffering operation. Based on these demands, the system was designed to accommodate three possible selections, namely:
an existing feature on the image, a user-defined shape (drawn on the image), or a group of features already stored in a file.

The process starts, as shown in Figure 8 in the FLOWCHARTS.PDF file, by a message on the screen asking the user to either identify the buffering feature or to use an existing group of features. If the user selects to identify a specific feature, the system then allows the user either to draw a feature on the image and use it as a buffering core feature or to select an existing feature on the image. After the selection is made, the user then has to identify which layer(s) to be queried against the buffering zone that will be created. For that purpose, a new form was created as shown in Figure 3.18.

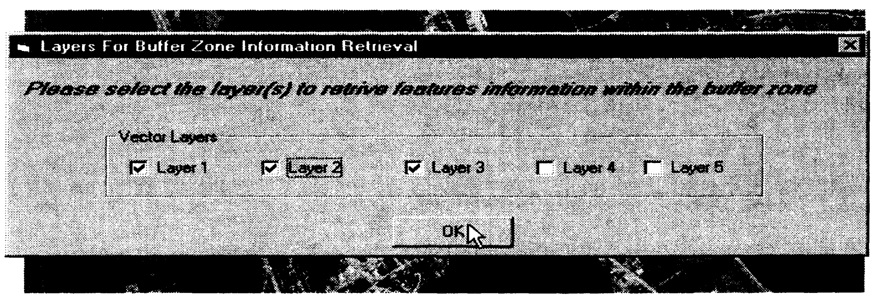


Figure 3.18
Buffering query function (layers selection).

This form allows the user to specify the layers involved in the query process in a manner similar to the layer selection within the main form. The following processing steps depend on the previous user selection of the core buffering shape. The following three sub-sections explain the processing steps involved in each case.

3.2.5.1.2 Buffering operation using an existing feature on the image

This buffering process starts by identifying the feature on the image. This identification process is the same as the identification process used in the pointing query function, as explained in section 3.2.2, without allowing the user to use a search circle method in order not to confuse the user or the function process. If the feature identified is a point or a polygon, the system then allows the user to specify the buffering distance using an Input box. On the other hand if the user selects a linear feature, a certain pre-process should take place.

Due to the fact that linear features (e.g. roads) within non-topologically structured files (e.g. ESRI Shape files) are stored as segments, each segment represents a feature by itself. This situation may be the case for data files obtained from a variety of organizations, for example, Statistics Canada. We may find that the same road has several records representing the road segments. In the buffering operation, the user may need to perform the operation on an existing road within the image. It is impractical due to the data structure limitations to let the user perform such operations several times, on each segment, and then combine the results. A solution had to be found to solve this problem.

If we look at a linear feature layer, we will find that in most cases the linear segments will have different records except for one record, namely the feature name. In other words, the road may be divided into segments having the same name. A decision was made to internally connect all segments representing the same linear feature, thus creating a new feature, holding the complete linear feature as one segment using the segment name as the connection key. In order to achieve that, the "Union" method was used. However, it is possible that we have a situation, as shown in Figure 3.19, of one road with multiple names. The user is interested in the whole road not in the names. In other words, we need to create a buffer zone around the whole road. Therefore, an input box was provided to the user to enter an additional road name to be connected. Using the "Union" method, the original name, and the additional names, if any, the system will join the segments and create a feature representing the whole road. In order to avoid errors in the buffering result, the system flashes the created feature and the user then has to confirm the validity of the created feature. The system now is ready to carry on the rest of the process.

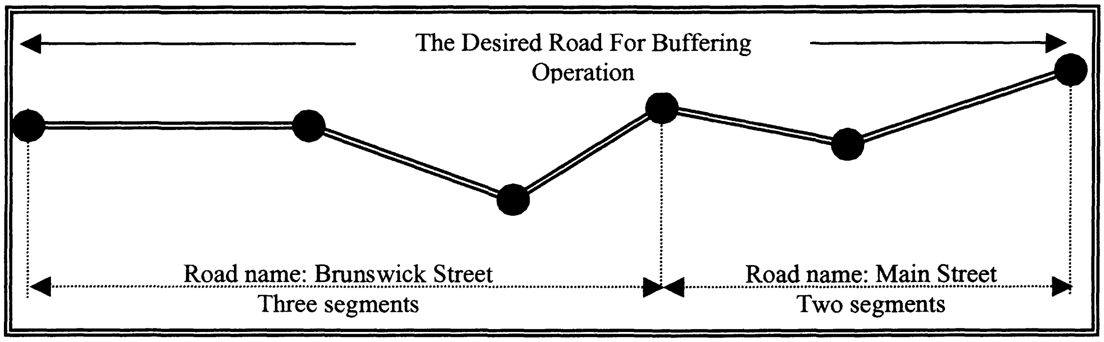


Figure 3.19
Buffering query function (linear feature structure within an ESRI Shape file).

An input box will appear allowing the user to specify the desired buffering distance. By selecting the buffering core shape (the feature(s) to create the buffer zone around) and using the input buffering distance, the system creates the buffering zone. The shape's "Buffer" method within MapObjects 2.0a is utilized in this step. When applied, this method creates a buffer zone and the system automatically creates a polygon object to hold the created zone as a polygon shape to be carried into the rest of the process.

After creating the buffering zone shape, the system checks for the selected layers to be queried. In some cases, users may need to retrieve all the features located within the buffering zone or they may need a quick check whether a certain feature is located within the buffering zone. For the second purpose input boxes were provided, as many as the selected layers, allowing the user to filter the results using the search expression.

A "SearchShape" method is then applied within each selected layer using the created buffering zone and the search expression, if any, as its arguments. New Recordset objects are created, each of which holds the result for each selected layer to be displayed.

3.2.5.1.3 Buffering operation using a user-defined drawn shape on the image

As mentioned earlier, the user may need to draw a certain shape on the image and use that shape as a core for the buffering operation. If so, a form will appear at the beginning of the process asking the user to identify the type of shape to be drawn. This form contains four option buttons allowing the user to select one of four possible shape types to be drawn; namely, point, line, polygon, and circle. The user then selects the desired layers to be involved in the query as well as the buffering distance. When the user starts drawing the shape on the image, the "Track(shape)" method is called, based on the selected shape, i.e., "TrackLine", "TrackPolygon", or "TrackCircle". If the selected shape is a point, the "ToMapPoint" method is called to identify the point and convert it from screen to ground coordinates.

The system creates an object to hold the drawn shape as the buffering operation core shape, and uses that shape along with the entered buffering distance to create the buffering zone polygon. The process continues as explained for the previous case.

3.2.5.1.4 Buffering operation using a group of features

If the selection was to use a group of features as the buffering core shape, the "Buffering Shape" common dialogue appears allowing the user to select the file that contains the desired group of features. As this file is identified, the system creates a new Recordset object to hold the records of that file. The user then identifies the layers to be involved in the query and also the buffering distance.

Two main directions can take place. First, using the buffering distance and each individual feature within the selected group, the system can create several buffering zone polygons for each feature and then connect them using the "Union" method to create one buffering zone polygon. Second, we may first connect the features within the selected group to create one feature and then, using the buffering distance, create the buffering zone polygon around that feature. The second solution was more appropriate because of the performance issue. The first solution performs the "Buffer" method as many times as the number of features within the group creating several buffering zone polygons and then applies the "Union" method to connect them. The second solution performs the "Union" method within the features of the group to create one buffering core feature and then performs the "Buffer" method once on that feature. After creating the buffering zone polygon, the process for creating the resultant Recordset remains the same as in the previous two cases.

The results obtained from the previous stage are now ready for display. In order to display the results, the same Display forms, their functions, and the process, as used for the Polygon/Line based image query, are utilized.

3.3 IISP CASE 2: ORTHOPHOTO WITH DIFFERENT PARAMETERS VECTOR DATA

In Case 2 we have a differentially rectified image, i.e. an orthophoto, covering a particular area of interest and the vector data sets covering the same or larger extent, but having different projections and coordinate systems. The main concern within Case 2 was to design and implement the appropriate approach for the on-the-fly projection capabilities within the IISP. A search through the appropriate software components available commercially to perform that task was conducted. Then, the user interaction with the system and the required modification to the process in case 1, to accommodate the requirements of case 2, were identified.

In order to design, implement, and test this case the following datasets were used:

- An orthophoto obtained from SNB with a scale of 1:10 000, Stereographic projection, NAD83 datum, and covering the Moncton area in New Brunswick, Canada;
- 1:10 000 building layer with a Stereographic projection and ATS 77 datum;
- 1:10 000 river layer with a UTM zone 20N projection and ATS 77 datum;
- 1:10 000 land cover layer with a UTM zone 20N projection and NAD 83 datum;
- 1:1000 000 road network layer with a NAD 27 datum.

The building, river, and land cover layers were extracted from SNB topographic map sheets in NTX format and converted to ESRI Shape files using ArcView3.2. The road network layer was obtained from Statistics Canada Digital Map Library in ArcInfo Coverage format and then imported to ArcView 3.2 and converted to an ESRI Shape file. Appendix B contains metadata information.

3.3.1 Available Projection/Transformation Software Components

Prior to designing the second case, an investigation into the available projection/transformation software components in the market took place. This investigation included mapping, GIS, remote sensing, and projection/transformation-oriented software through the Internet and the literature. This research focused on the answers to the following three questions:

1- What software components are available for performing the projection/transformation tasks?

2- If any, are they compatible with Visual Basic 6.0, the development environment?

3- Does the component support worldwide varieties of projections and coordinate systems?

Three software components were found satisfying the above mentioned conditions, namely:

- "GeoView" from Blue Marble Geographic Company.
(http://www.bluemarblegeo.com)
- "MapObjects 2.0a projection components" from ESRI.
(http://www.esri.com)
- "MapInfo MapX 4.5" from MapInfo company.
(http://www.mapinfo.com)

In order to select the optimum product, decision criteria had to be developed. These criteria were based upon several factors, namely:

1- number of projections supported;

2- number of coordinate systems supported;

3- number of pre-defined geographic/projected coordinate systems;

4- the ability of creating new user-defined/customized coordinate systems;

5- previous usage experiences;

6- previously identified integration problems with the MapObjects 2.0a components (the spatial components selected for IISP);

7- Shape file support;

8- availability and price.

Table 3.8 summarizes the main characteristics of the three products in terms of the developed criteria:

Table3.8: Projection/transformation software components characteristics.

Components / Factor	GeoView	MapObjects 2.0a	MapInfo MapX 4.5
1	32	32	~ 26
2	Large number	246	~ 118
3	Large number	~ 1000	Not specified
4	Yes	Yes	Yes
5	Several companies and organizations	Several companies and organizations	Several companies and organizations
6	Not known	None	Not known
7	Yes	Yes	Yes
8	US$ 999	Site license	Not specified

From the above listed factors, it can be concluded that:

1- All three components have a wide variety of projection and coordinate system definitions;

2- It is possible to customize or build a new desired coordinate system;

3- All of the components have a reliable history (as indicated in their Web sites).

Three advantages exist for the MapObjects 2.0a projection component over the other two products. First, the projection component is already within MapObjects 2.0a and there will be no possibility of having integration or compatibility problems. Second, the component is already available along with the appropriate documentation and technical support. Third, the use of the MapObjects 2.0a projection component is already included within the IISP application package and will not increase the application file size. In addition to that, there will be no need to purchase any additional components.

A decision was made to use the MapObjects 2.0a projection component to implement the on-the-fly projection/transformation capabilities within the IISP.

3.3.2 Design Considerations

In order to implement the on-the-fly projection/transformation capabilities within the IISP, the following questions had to be answered:

1- In general, what are the projection/transformation limitations and capabilities within the selected software component?
2- What are the possible cases we may have? How can we design the system in such a way as to handle these cases? How will the user interact with the system?
3- Should the image be transformed to the vector layers' domain or vice versa?
4- If the vector layers will be the slaves (the ones being transformed to the image domain), should the system transform the whole layer or should another approach be followed?

The above-mentioned questions will be analyzed and answered in the following sections.

3.3.3 MapObjects 2.0a Projection/Transformation Capabilities and Limitations

Further research was conducted to emphasize the capabilities as well as the limitations of the MapObjects 2.0a projection component. The major capabilities of the MapObjects 2.0a projection component may be summarized as follows [ESRI, 1999 b].

1- includes a large number of pre-defined projected/geographic coordinate systems;
2- allows for building customized coordinate systems;

3- allows an existing layer to be exported to a new layer with a different coordinate system;

4- has a one-step transformation of a feature of any shape from one coordinate system to another;

5- allows for automatic reading of the appropriate format vector layers' coordinate system, and also provides the capabilities to distinguish between the geographic and the projected coordinate systems.

On the other hand, during this research, one major disadvantage was discovered, namely, Map Objects 2.0a currently does not allow for direct or indirect reading of any image layer coordinate system information.

Despite this disadvantage, the MapObjects 2.0a projection component was used successfully to implement the projection/coordinate system on-the-fly transformation within the IISP.

3.3.4 Possible Cases and Design

The concept behind the IISP relies on remote sensing imagery as an interaction medium with the real world, and, consequently, this medium should be stable at all times. In other words, we should consider the image coordinate system domain as the master domain and the vector layers should be transformed to bring them to the master domain.

Image layers as well as vector layers may have either geographic coordinate systems or projected coordinate systems. The system should be designed to accommodate both types and to internally transform between the same types or from one type to another.

One of the disadvantages of the projection components within Map Objects2.0a is that they do not support direct or indirect reading of the image layer projection, as mentioned earlier in this section. This means that the user will have to input the image coordinate system properties and parameters during the image loading process. Three forms were designed to allow the user to identify to the system the appropriate parameters that match the existing image geographic/projected coordinate system, as will be explained later in this section.

No matter how the internal on-the-fly projection/transformation function will be implemented within the IISP, the image coordinate system identification process should take place during the image loading stage.

The "Image File Projection Information" form appears during the image loading process allowing the user to identify the projection/datum/coordinate of the image file. This form, as shown in Figure 3.20, was designed so that it contains two selection boxes. The first box contains a list of predefined projected coordinate systems and the other contains a list of pre-defined geographic coordinate systems. The geographic/projected coordinate system "PopulateWith" method was used to attach all the available pre-defined coordinate systems within the MapObjects 2.0a component to these two boxes. Users can then browse and select the projected/geographic coordinate system that matches their image case.

If users cannot find the required projected/geographic coordinate system within the list, they will need to customize the parameters to create the desired one. A command button labeled "Customize The Parameters" was added to the form for the user to build/customize the required geographic/projected coordinate system step-by-step.

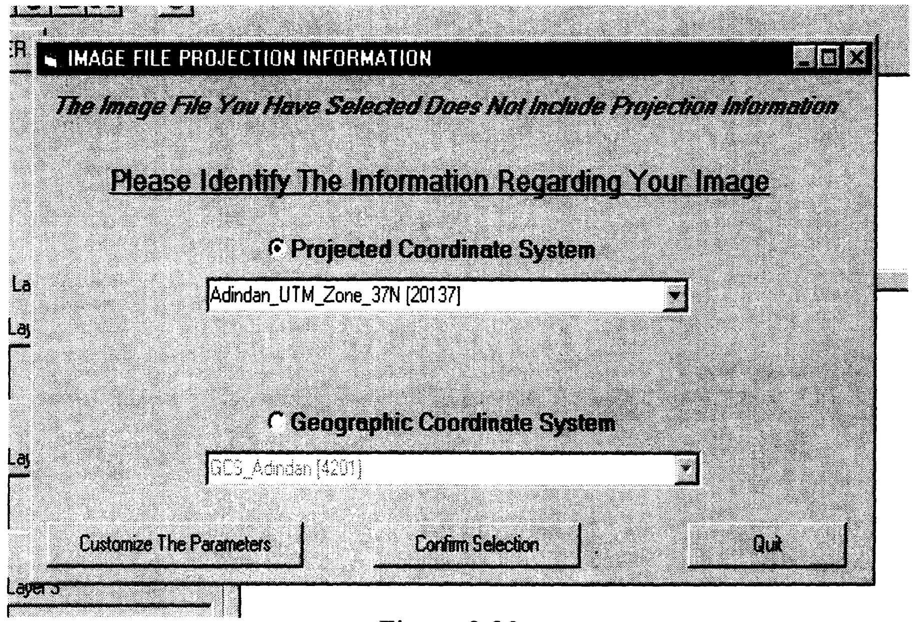


Figure 3.20
Image file projection information.

By pressing that button the "Image Coordinate System" form will appear allowing the user to define the projection, the unit of measure and to define or build the image coordinate system. When users confirm their selection, the "Image Coordinate System Parameters" form appears allowing the user to set different advanced parameters related to the previous projection/coordinate system selection such as false East, false North, scale factor, etc., Figure 3.21 and Figure 3.22 show these two forms for Stereographic projection with Metre unit and North American Datum (NAD 83).

After defining the image geographic/projected coordinate system, the system will be ready to automatically identify the coordinate system of the vector layers, apply the required on-the-fly projection/transformation, and query the image. Figure 3.23 shows a schematic diagram representing the image coordinate system identification process.

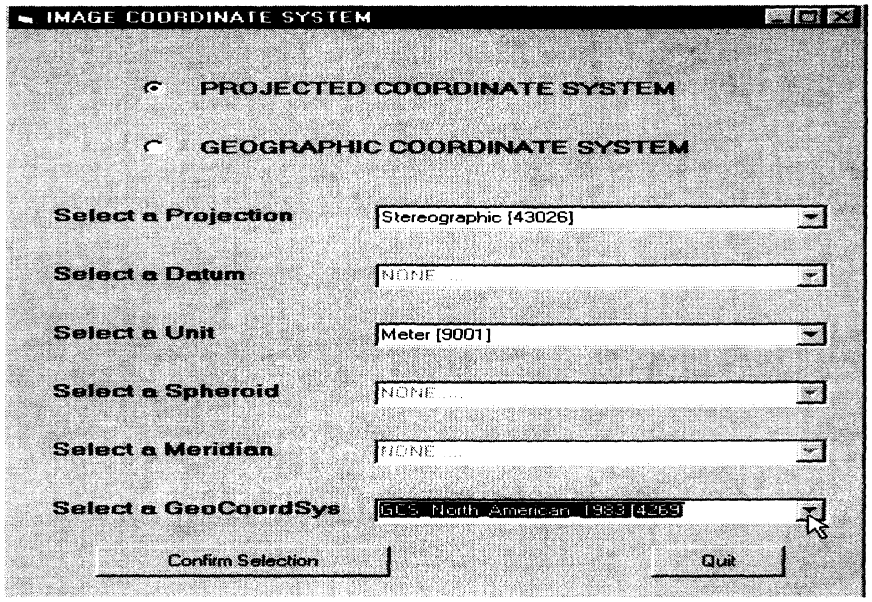


Figure 3.21
Image file projected/geographic coordinate system customization.

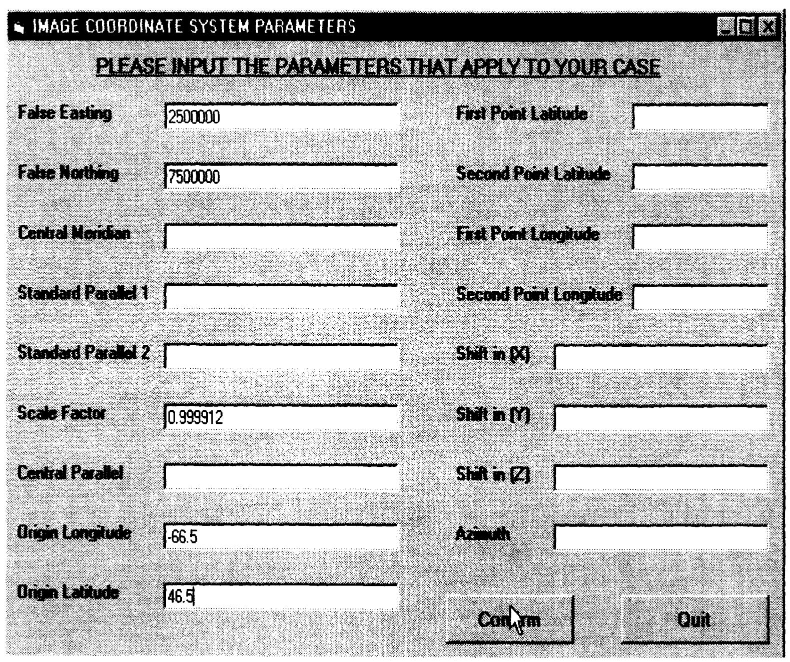


Figure 3.22
Image file projected/geographic coordinate system parameter input.

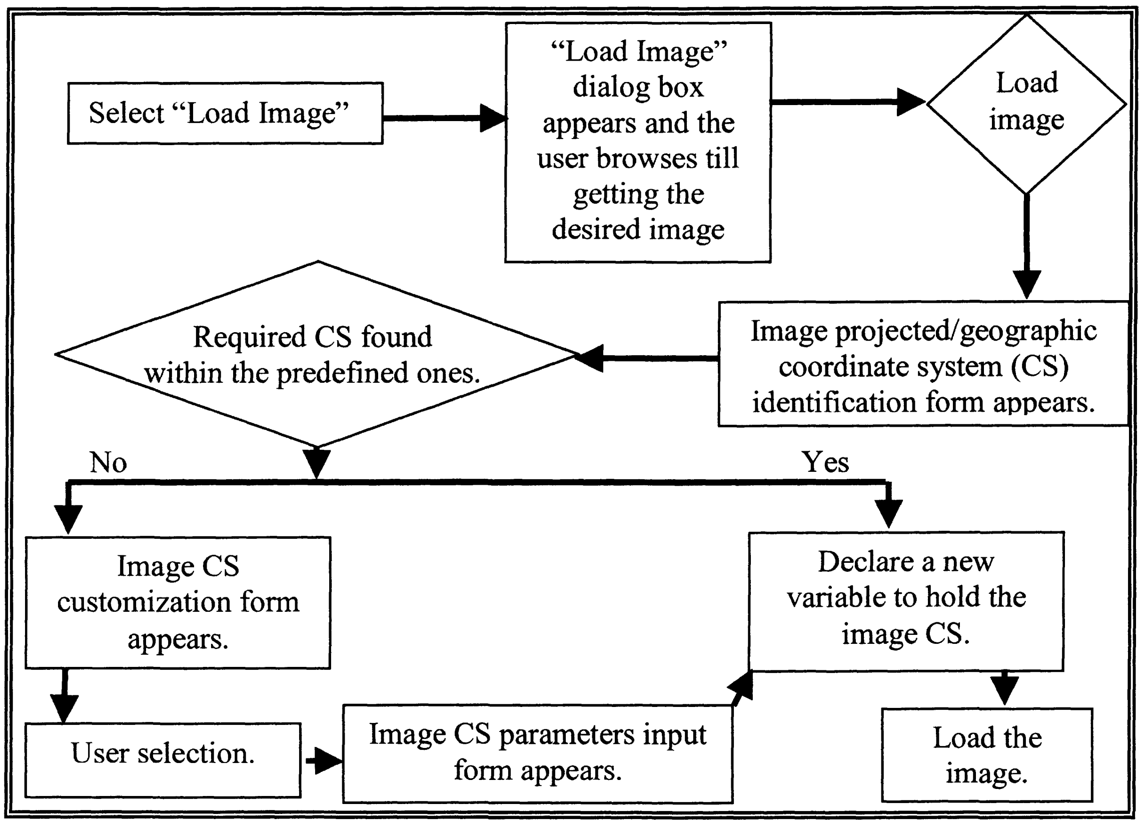


Figure 3.23
Image coordinate system identification process.

3.3.5 Possible Projection/Transformation Approaches

In order to decide on the appropriate approach to implement the projection/transformation utilities within the IISP, two possible methods were identified, designed, and implemented. The following sub-sections demonstrate these two methods along with their advantages and disadvantages.

3.3.5.1 Method I: "Direct on-the-fly Whole Layer Projection"

In this method a direct on-the-fly projection approach was applied. As the user identifies each layer that will be needed for the query, the system reads the projection information of that layer and then transforms the whole layer to the image domain. The steps involved in this procedure are as follows:

1- The user inputs the projection, datum, and coordinate system information of the image during the image loading process;
2- The user identifies the layer(s) required in the query process;
3- The system reads the projection information of the layer;
4- The layer is then transformed to the image domain using the Map Layer "Export" method within MapObjects 2.0a;
5- A new Recordset object is created and the new transformed layer is attached to it;
6- The new Recordset object created is used to receive and invoke any query regarding the attached layer, i.e., the transformed one.

In order to accommodate the projection/datum/coordinate system on-the-fly transformation using the first approach, some modifications were applied to the vector layer(s) identification process within the IISP Case 1.

When the user identifies any vector layer, the system first reads the projection information of that layer to decide whether the layer has a projected or geographic coordinate system. The system then identifies the required transformation for the layer to the image domain and performs the transformation. It is assumed here that an appropriate projection information file containing the layer coordinate system information exists and that it is stored with the layer files in the same directory.

In order to perform the transformation, the system makes use of the "Export" method associated with the Map Layer object within MapObjects 2.0a to write a new transformed file to a temporary directory. Following that, the transformed layer replaces the original one in the Data Connection object. A new Recordset object is created holding the new transformed layer and redirecting the link between the image and the identified layer to the transformed layer rather than to the original one. With the whole layer(s) transformed and the image linked to the new transformed layer(s), the rest of the system process and functionality remains the same as designed and explained in Case 1. Figure 3.24 shows a simplified schematic diagram of that process. This approach was implemented within 4 layer options within IISP, namely layers 2 through 5.

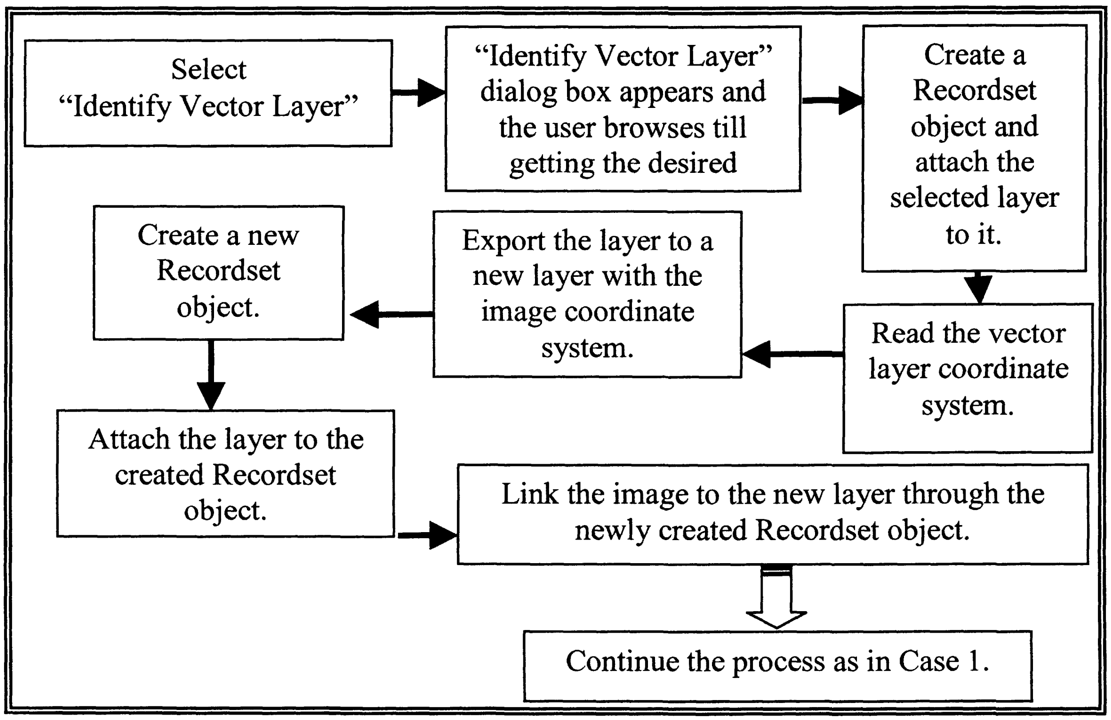


Figure 3.24
Vector layer identification and export process (Projection Method I).

3.3.5.2 Method II: "Minimal on-the-fly Results Projection"

The concept behind the IISP facilitates this method. While Method I is an essential requirement for displaying image and map layers in a superimposition mode, it is not a necessity within the IISP. Since the IISP internally links the vector data to the image and there is no need to display the vector layer(s), it is also possible that there is no need for the whole data set to be transformed beforehand. In other words, a different approach can be applied. This approach is based on the idea of having the vector layer(s) identified and linked to the image without transformation. Then, the query input as well as its results can be transformed from the image domain to the map layer domain or vice versa as required by the process. This approach makes use of the Shape's "Transform" method within MapObjects 2.0a.

The "Transform" method of the Shape object allows the transformation of any type of shapes such as polygon, line, point, or points collection to be transformed from one coordinate system to another [ESRI, 2000 b]. The disadvantage of the "Transform" method is that it does not support the transformation of a group of features simultaneously. In other words, if we have a group of features, a query result for example, we will not be able to transform them at once from their layer domain to the image domain. This is required when displaying/flashing the results from a certain query at once or when we use a group of features as a core for a buffering operation. Two solutions were attempted to solve this problem. First, using the Map Layer "Export" method, we can export the group of features, usually created from a certain query, to form a new layer, transform this layer, and then reconnect the image with the new transformed layer. Second, using the Shape's "Transform" method, we can transform the resulting features one by one and then use the Shape's "Union" method to create a new shape that represents the transformed resulting features. It was decided that the second solution was more appropriate because it involved fewer processing steps than the first approach and it was more flexible from the implementation perspective.

Figure 3.25 and Figure 3.26 represent schematic diagrams of the processing steps involved in each of the solutions.

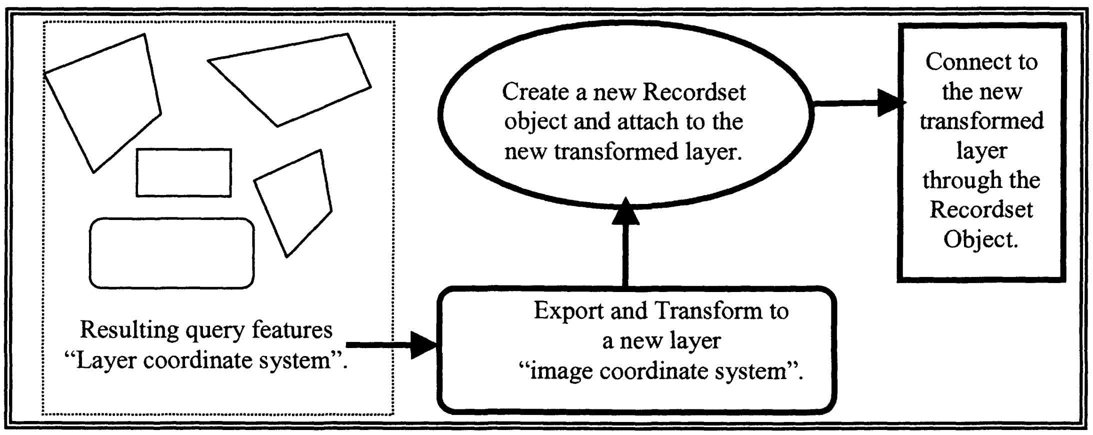


Figure 3.25
Query results transformation through the Export method.

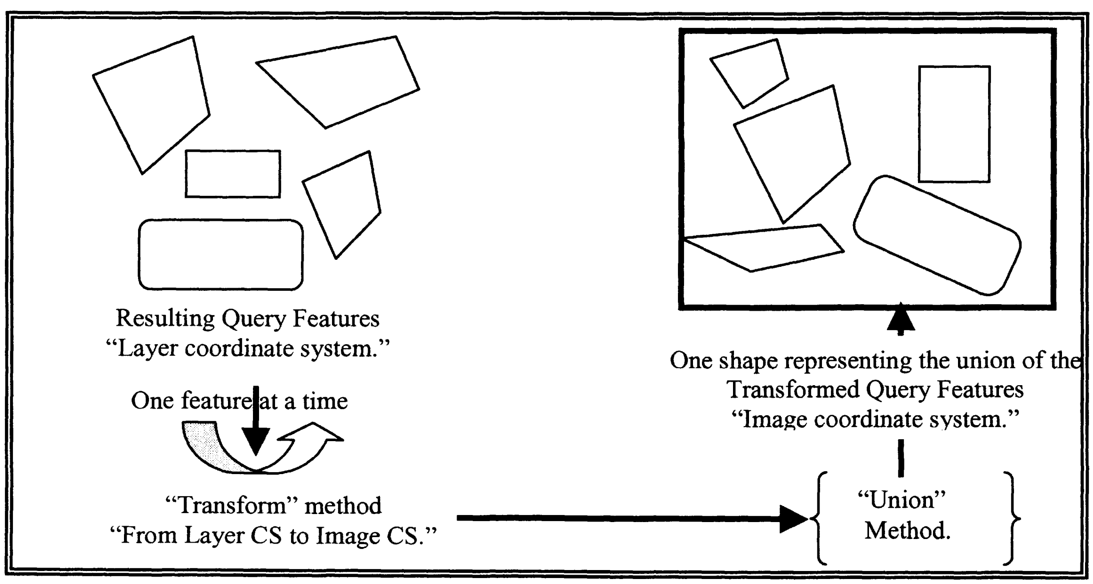


Figure 3.26
Query results transformation through the Transform and the Union methods.

Using this method, the first stage (Data loading and identification) process will remain the same as explained in Case1 (see section 3.2.1) with the exception of defining the image coordinate system. The following sub-sections represent the processing steps involved in the IISP Case 2 using the second approach, "Minimal on-the-fly results projection", in implementing the projection capabilities within each function of the system.

3.3.5.2.1 Pointing query function

The pointing query function, as shown in Figure 3.27, involves the following steps:

1- The user clicks on a position (P) on the image, and the system converts the position from screen coordinates to ground coordinates based upon the image coordinate system.

2- The system reads the projection information of the desired layer and transforms (P) from the image coordinate system domain to the vector layer coordinate system domain using the Shape's point feature (P) "Transform" method.

3- A search through the layer records is conducted and the same procedure as for Case 1 is followed.

4- When the required feature is found, another backward transformation takes place, from the layer coordinate system domain to the image coordinate system domain, using the detected feature as the argument to the "Transform" method.

Any intermediate process, such as the Search Circle procedure, makes use of the same transformation procedure for the image coordinate system domain to the vector layer coordinate system domain and vice versa.

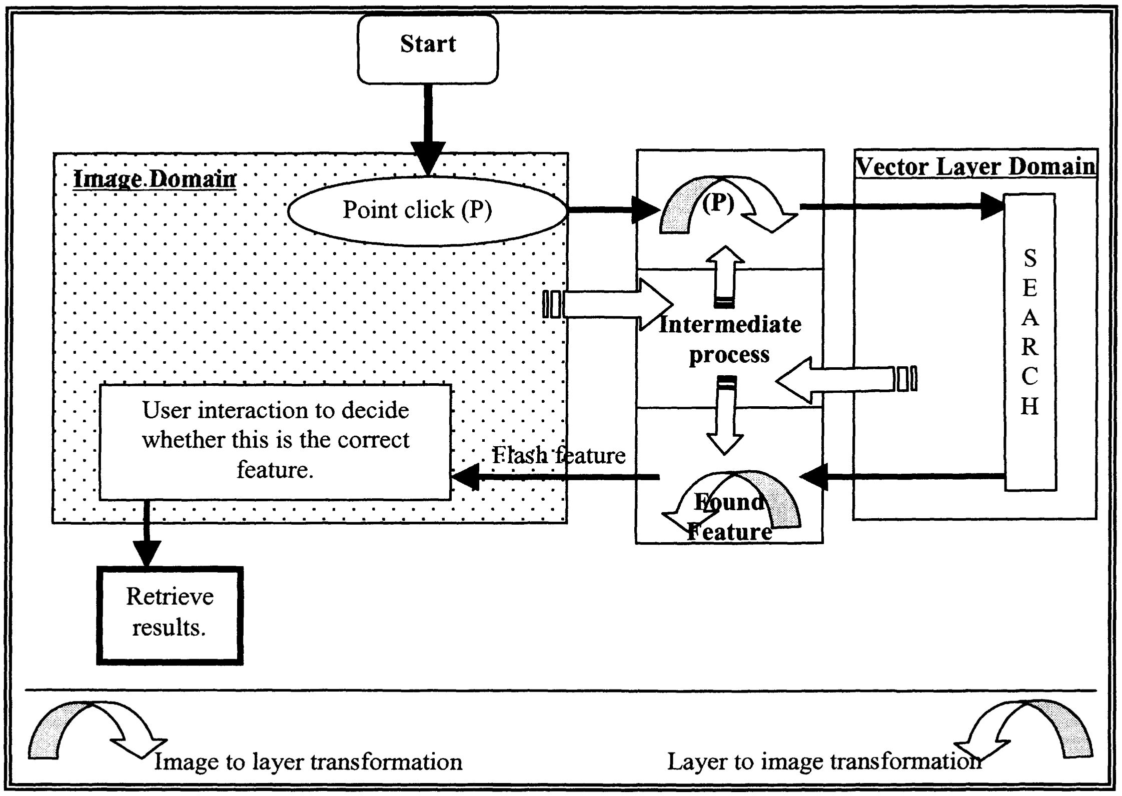


Figure 3.27
Processing steps of the pointing query function (IISP Case 2- Projection Method II).

3.3.5.2.2 Polygon/line-based image query function:

In order to perform the polygon/line-based image query function within the second method of projection, the following procedure has to take place as shown in Figure 3.28:

1- The user draws a polygon or line around/crossing the areas of interest on the image;

2- The system identifies that user-defined shape within the image coordinate system domain and then checks for the first layer selected to be a candidate in the query process;

3- The system should read the projection information of that layer and then convert the user-defined shape from the image coordinate system domain into the layer coordinate system domain, making use of the "Transform" method.

4- A search through the selected layer is conducted, using the search criterion identified by the user, and the resulting features are obtained;

5- The display form assigned to that layer appears holding the resulting query features along with their attributes;

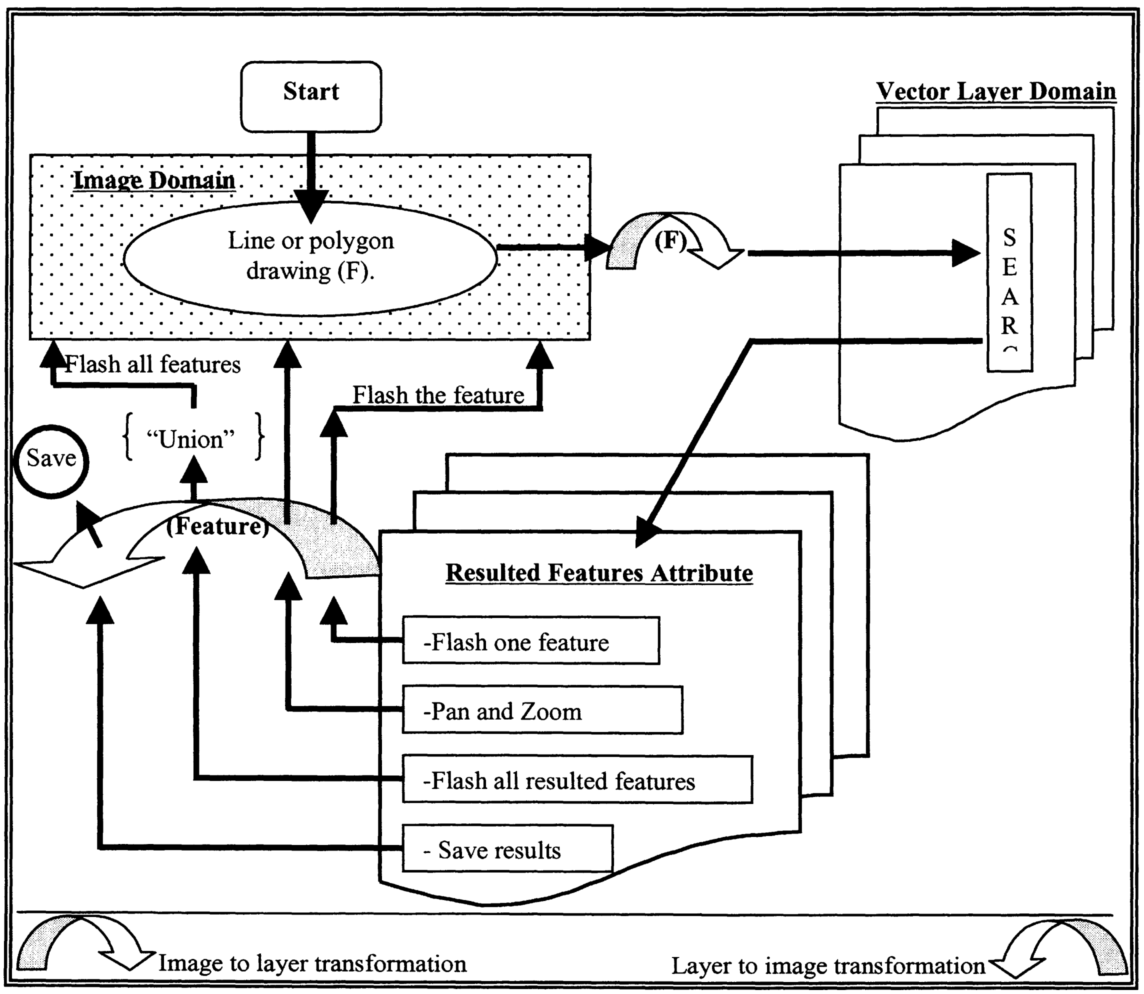


Figure 3.28
Processing steps of the polygon/line-based query function (IISP Case 2- Projection Method II).

6- After the display form appears, the user can select which function, within the display form, should be executed. Based on this selection, a certain process has to be executed as follows:

   a- If a certain feature is selected to be displayed/flashed, panned to, or zoomed to, the feature has to be transformed from the layer coordinate system domain to the image coordinate system domain, and then the appropriate procedure takes place, as explained in Case 1, using the transformed feature as its target.

   b- If the user decides to flash all the features at once, the system transforms the resulting features one by one, performs the "Union" method to form a new transformed shape representing the union of all the resulting features in the image coordinate system domain. It then flashes all features using the same flashing process as explained in Case1.

   c- If the resulting features must be saved as a new layer, the system uses the "Export " method to create a new layer that contains the resulting features in the image domain coordinate system.

If more than one layer is selected for the query, steps 2 through 6 must be repeated for the other layers.

3.3.5.2.3 Database query function

In the database query function, the image is loaded and the vector layer(s) identified, as explained previously in sections 3.2.4 and 3.3.5.2. The process of querying the vector layer is exactly the same as it was in Case 1with no need for modifications. In other words, the layer is identified and queried without any transformation. After the display form appears, the same process as explained in steps 5 and 6 of the previous function takes place. Figure 3.29 shows a schematic diagram representing the major steps involved in performing this function.

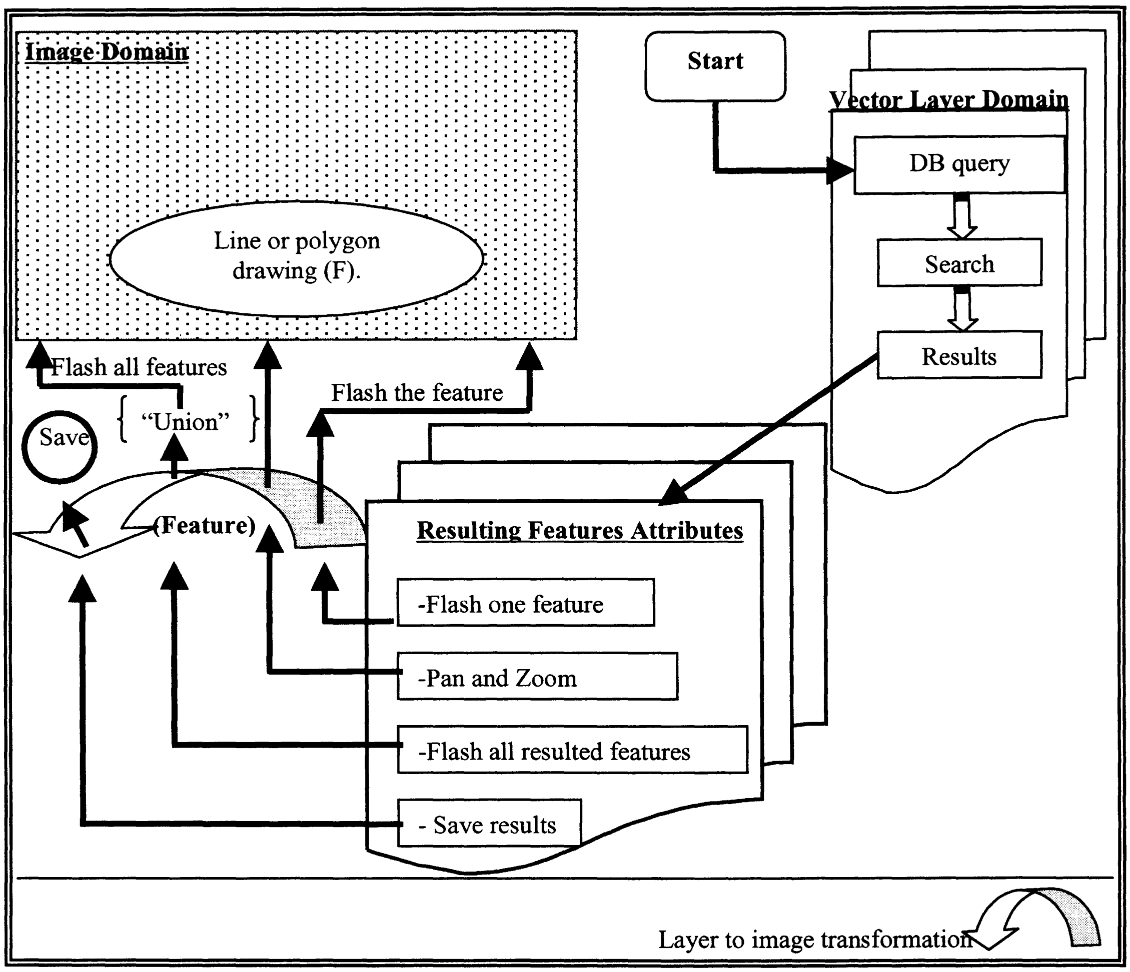


Figure 3.29
Processing steps of the database query function (IISP Case 2- Projection Method II)

3.3.5.2.4 Buffering operation

In performing the buffering operation, the system was designed to allow the user to select a certain feature within the image, i.e. a user-defined/drawn feature, or a group of features already stored in a file as core feature(s) for this operation. As shown in Figure 3.30, the major steps involved in performing that function may be summarized as follows:

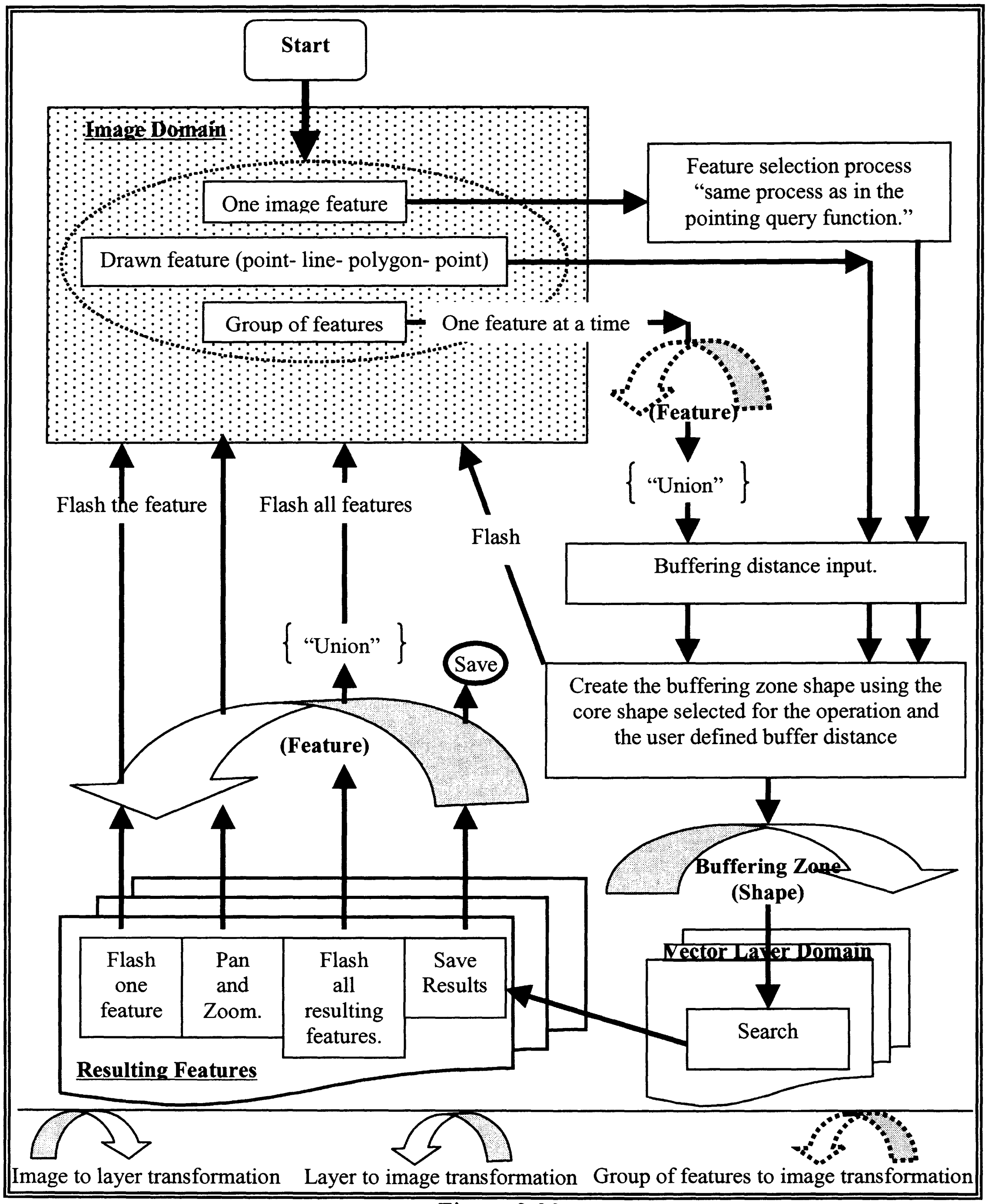


Figure 3.30
Processing steps of the buffering function (IISP Case 2- Projection Method II).

1- If a certain feature is selected to be the core for the buffering operation, the same process as identified within the pointing query function is executed until the desired feature is selected. This feature will be in the image coordinate system domain as a result of the pointing query process.

2- When having a user-defined/drawn feature, the system identifies the feature within the image coordinate system domain.

3- If the user selects a group of features stored in a file as the core buffering features, the system has to read the projection information associated with that file, export the file to the image coordinate system domain, connect to the exported file, and then apply the "Union" method to create a new feature. This new feature represents the transformed group of features to be the core-buffering feature. It is important to note that this step will be taken if we use any of the projection methods (I or II).

4- After identifying the core-buffering feature, the user inputs the buffering distance as well as the desired layers to be queried against the created buffering zone;

5- Using the core buffering feature and the user-defined buffering distance, the buffering zone is created and then transformed from the image coordinate system domain to the first selected query layer coordinate system domain.

6- After obtaining the results, the same process as identified in the polygon/line based query function, steps 5 and 6, is carried out.

7- The process presented in steps 5 and 6 should be repeated for each selected query layer.

It is important to note that if both projection methods are used simultaneously within the query process, in querying different layers, the system has to consider the appropriate transformation of the query shape (the buffering zone or the user-defined polygon/line) from the image to the desired query layer domain or vice versa, when required. This is important since the projection Method II does not project the entire layer at the beginning of the query process. This projection Method II has been implemented within the first layer in the IISP.

When using either of the projection methods, the process works properly when there is a physical shape/layer transformation. However the methods need special modifications when dealing with numbers. For example, when trying to perform a search using a Search Circle or in using a user-defined buffering distance, there may be a problem if the point position on the image has a geographic coordinate system and the user enters a search radius in metres. This way, the system will add the point position in degrees to the radius in metres. To avoid such a problem, one possible solution was accomplished by tracking down the process transaction that deals with this issue and converting the entered values from metres to degrees. This is done based on an earth radius of 6372 km.

3.3.6 IISP Projection/Transformation Utility: Method I vs. Method II

As explained earlier, two methods of an on-the-fly projection/transformation utility were implemented within the IISP Case2. Each method has its advantages and disadvantages. Table 3.9 represents a comparative study of the two methods.

Instead of having one fixed method implemented within the system, an optimum solution would be to use both methods within the same system. The user can select the appropriate method for each layer, mainly based on the size of the dataset and the user's experience with system resources. This technique will benefit from the advantages of each method and will avoid some of the disadvantages.

Table 3.9: Projection/transformation methods: A comparison

	Method I	Method II
Implementation	Easier to implement	Harder to implement
Transformation approach	Make use of the "Export" method.	Make use of the "Transform" method which can be applied to any single shape of any type.
Initial vector layer identification	Takes more processing time to identify and transform the layer in the beginning of the process. (May cause a problem in case of a large data set and limited computer resources.)	No additional processing time required if compared with Case 1 because no transformation or modification applied in the vector layer(s) identification process.
Internal processing problems	None	Problem when having more than one feature to transform (as explained earlier).
Query performance	Faster (same performance as in Case 1).	Slower (slower than Case 1 because of the processing time required to perform the transformation process during the query.)
Further modification	Easy to modify because the modification will be applied to the vector layer(s) identification process.	Difficult to modify because the modification should take place along the process.
Relative accuracy	Both methods provide the same results when used to project/transform point, linear, and polygon layers, in terms of the number and the display of resulting features.	
Recommendation	This method is recommended if the original dataset is not too large.	This method is recommended if the original dataset is very large.

3.4 IISP CASE 3: PARTIALLY RECTIFIED IMAGERY "IKONOS GEOCARTERRA™" WITH DIFFERENT PARAMETERS VECTOR DATA

The previous two cases covered the interface design, user interaction, and processing requirements involved in connecting vector layers to orthophotos having the same or different parameters (projection, coordinates system, and scale). Using these orthophotos as a reference layer in querying the real world using different spatial query procedures have also been dealt with. In Case 3, the system is designed to handle partially rectified/low level geo-corrected imagery. It means that geometric distortion within the image has been partially corrected but not completely removed.

The main concept of the Intelligent Imagery System relies on *high-resolution* remotely sensed data in querying and analyzing spatial phenomena. Consequently, a decision has been made to dedicate this case to solving the issues related to one of the most recently launched commercial high resolution imagery on the market, namely, IKONOS imagery. For Case 3 the assumptions were that the image had the lowest rectification/geo-correction level in the IKONOS line of products (IKONOS GEOCARTERRA™) and that the vector data parameters may differ from those of the image in terms of projection, coordinate system and scale.

The following data sets were used to design, implement, and test this case:

- IKONOS GEOCARTERRA™ image obtained from the City of Fredericton representing the Fredericton area, New Brunswick Canada. The image used was panchromatic and resampled (1.0 m Ground Sampled Distance GSD) using UTM (zone 19N/WGS 84) as its projected coordinate system.

- Two 1:10 000 building and road layers with NB. Stereographic projection and ATS 77 datum, obtained from SNB;
- Two 1:10 000 building and road layers with UTM zone 19N projection and WGS84 datum, obtained from SNB;
- 1:250 000 land cover layer with a UTM NAD 27 coordinate system, obtained from Statistics Canada from the on-line digital map library.

Please refer to Appendix B for metadata information.

3.4.1 IKONOS GEOCARTERRA™ Imagery

Space Imaging's IKONOS 2 Satellite, launched on September 24, 1999, provides 1-metre resolution panchromatic imagery and 4-metre resolution multi-spectral imagery. As of March 28, 2000, four main products may be ordered based upon IKONOS imagery, namely [Gerlach, 2000]:

1- CARTERRA™ Geo Pan. and Multispectral 11 bit;

2- CARTERRA™ Reference Pan. 11 bit;

3- CARTERRA™ Precision Pan. 11 bit;

4- CARTERRA™ Precision Plus Pan. 11 bit.

Table 3.10 describes the rectification and accuracy considerations for the four products. For complete details on the satellite sensor, products, and pricing see for example Gerlach [2000]; ERSI [n.d.]; Toutin and Cheng [2000]; Space Imaging [n.d], and Space Imaging Europe [n.d.].

As shown in Table 3.10, IKONOS GEOCARTERRA™ is the product with the lowest rectification level. The image is collected, corrected due to the sensor and the collection geometry errors, and then it is projected and resampled to a predefined projected coordinate system. Neither Ground Control Points (GCPs) nor a Digital Terrain Model (DTM) has been used within this rectification process. As released by Space Imaging, the Circular Error (CE 90%) of this product is 50 m excluding the relief displacement.

Table 3.10 : IKONOS satellite products.

Product	GCPs	DTM	CE90%	RMSE
Geo	No GCPs used in the rectification process.	No terrain model is used in the rectification process.	50.0 m + terrain displacement	23.3 m
Reference	No GCPs used in the rectification process.	Terrain model is used in the rectification process.	25.4 m	11.8 m
Precision	GCPs used in the rectification process.	Terrain model is used in the rectification process.	4.1 m	1.9 m
PrecisionPlus (same as precision with higher collection elevation angle to reduce the relief effect).	GCPs used in the rectification process.	Terrain model is used in the rectification process.	2.0 m	0.9 m

In Case 3, a new method has been developed in order to obtain on-the-fly a precise geographic position of an image feature and to approximately display the ground feature correctly on the non orthorectified imagery whenever it is needed during the image query process. This provides the user with the convenience and accuracy of querying an orthophoto. The following sections provide a detailed explanation of the developed technique.

3.4.2 IKONOS GEOCARTERRA™ Collection Geometry

The IKONOS satellite sensor is a *push-broom* type sensor. This means that the satellite uses an array of Charge Coupled Devices (CCD) that scan one line at a time in the across-track direction and the satellite movement allows for scanning in the along-track direction. The IKONOS satellite has the capabilities of off-nadir pointing. Along with the usage of a rotating mirror, the satellite itself can pivot in orbit allowing for up to 45 degrees off-nadir pointing in-track and cross-track, as shown in Figure 3.31 [UCSB, n.d.]; [Space Imaging Europe, n.d.]; and [Gerlach, 2000].

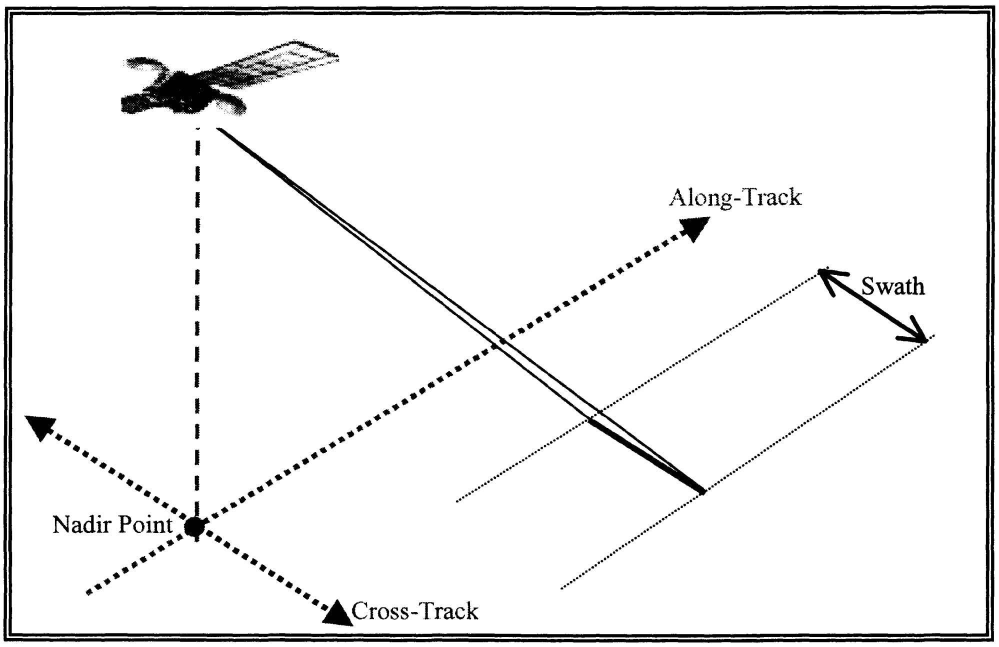


Figure 3.31
IKONOS satellite in and cross-track pointing capabilities.

3.4.3 IKONOS GEOCARTERRA™ Rectification Procedure and Error Sources

The rectification process of the IKONOS GEOCARTERRA™ product basically eliminates the image distortion introduced by the collection geometry factors, namely, earth curvature and rotation, panoramic distortions, and variations in the altitude and the attitude of the sensor [Space Imaging, n.d.]. The image is then resampled to a uniform ground sample distance and a specific map projection. During this process, no GCPs or terrain model is used but rather the orbital ephemeris, as well as satellite attitude and altitude information are used to determine the pixel positions on the ground [Gerlach, 2000].

After performing the previously mentioned georectification process, two main sources of errors remain within the image pixels. The first one is the relief displacement. This error is produced from the actual terrain elevation within the area. Despite the fact that the satellite altitude is 681 km, the collection elevation angle for this type of imagery could be as low as 50 degrees, which introduces a significant relief displacement on the image [Gerlach, 2000].

The second source of error is due to a feature's height above ground. In some cases, features such as buildings/towers may have a large height above the ground. In such cases, the feature's top and bottom points will have a displacement based upon this height. This source of error should be considered within the developed system, as will be explained later in section 3.4.6.2.

3.4.4 Theoretical Solution

A great amount of research has been done in order to orthorectify the push-broom type of imagery. For example, SPOT (System Pour l'Observation de la Terre) Satellite is one of the satellite systems that has characteristics similar to IKONOS in terms of orbital type, sensor type, and pointing capabilities. Modeling the orthorectification process in SPOT has been widely examined, and a great number of articles exist in the literature that deal with this issue. Most of these methods make use of a modified version of the photogrammetric collinearity condition to model the rectification process; see, for example, Lee and Chen [1988] and El-Manadili [1994]. Other modeling techniques consider modeling the entire scene using polynomials (see, for example, Westin [1990] and Radhadevi et al. [1994]). Another technique is to modify analytical photogrammetric plotters to handle SPOT imagery [Kruck, 1988]. The Canada Centre for Remote Sensing (CCRS) developed a method to process raw SPOT imagery [Cheng and Toutin, 1994].

The main issue here is that, in all of the previously mentioned cases, the image orientation and detailed orbital information were being provided as metadata along with the image file and the sensor model is well established. Having this information and using the pixel coordinates within the image coordinate system, photogrammetric techniques can be utilized in order to model the *cause* of image distortion, and an orthophoto can be produced.

In our case, Space Imaging is providing insufficient orbital information and few parameters related to the image viewing angle. Another complication arises from the fact that the raw data is not released and if the width of the desired area is larger than 11 km the image is provided as georectified tiles. It means that the satellite is imaging a continuous strip that contains the client area of interest and then the required area is processed and delivered to the client. In other words, the client will get a georectified image tile/scene with no indication to the location of the image center pixel/line. In addition, the image may have different elevation angles and different pointing directions which means that the first tile is taken and then the satellite is programmed to reorient its viewing and pointing angles to image the next tile. Without the appropriate information, employing photogrammetric techniques to orthorectify IKONOS GEOCARTERRA™ imagery will be difficult and problematic and other solutions should be attempted. Toutin and Cheng [2000] have developed a method in order to orthorectify the IKONOS GEOCARTERRA™ imagery.

In this research, a new approach and procedure, making use of metadata parameters and photogrammetric methodology, has been developed in order to obtain on-the-fly the correct pixel position as if the image were orthorectified. This approach performs an on-the-fly multi-stage transformation eliminating the error sources in a subsequent mode without disturbing the system processing/flow. The sensor error (the error introduced within the image pixel coordinates during the georectification process due to the error in calculating the sensor orientation parameters and the resampling process) in identifying the image pixels' coordinates is eliminated first, and the relief displacement component(s) are then identified and treated. In other words, the approach developed within this research represents an attempt to model the *effect* of the sensor orientation and position as well as terrain characteristics and an approximate prediction to image georectification/processing steps for the existing distortion within the georectified imagery.

3.4.4.1 Elimination of Sensor and Georectification Error

The georectification process is accomplished by obtaining the sensor position and orientation on-board and using these parameters to triangulate in ground features on the image (Gerlach, 2000). The georectification error can be mainly introduced due to:

1- the error in determining the satellite position.

2- the error in determining the satellite orientation angles over time.

3- the resampling procedure employed.

This error produces the geometric relationship shown in Figure 3.32.

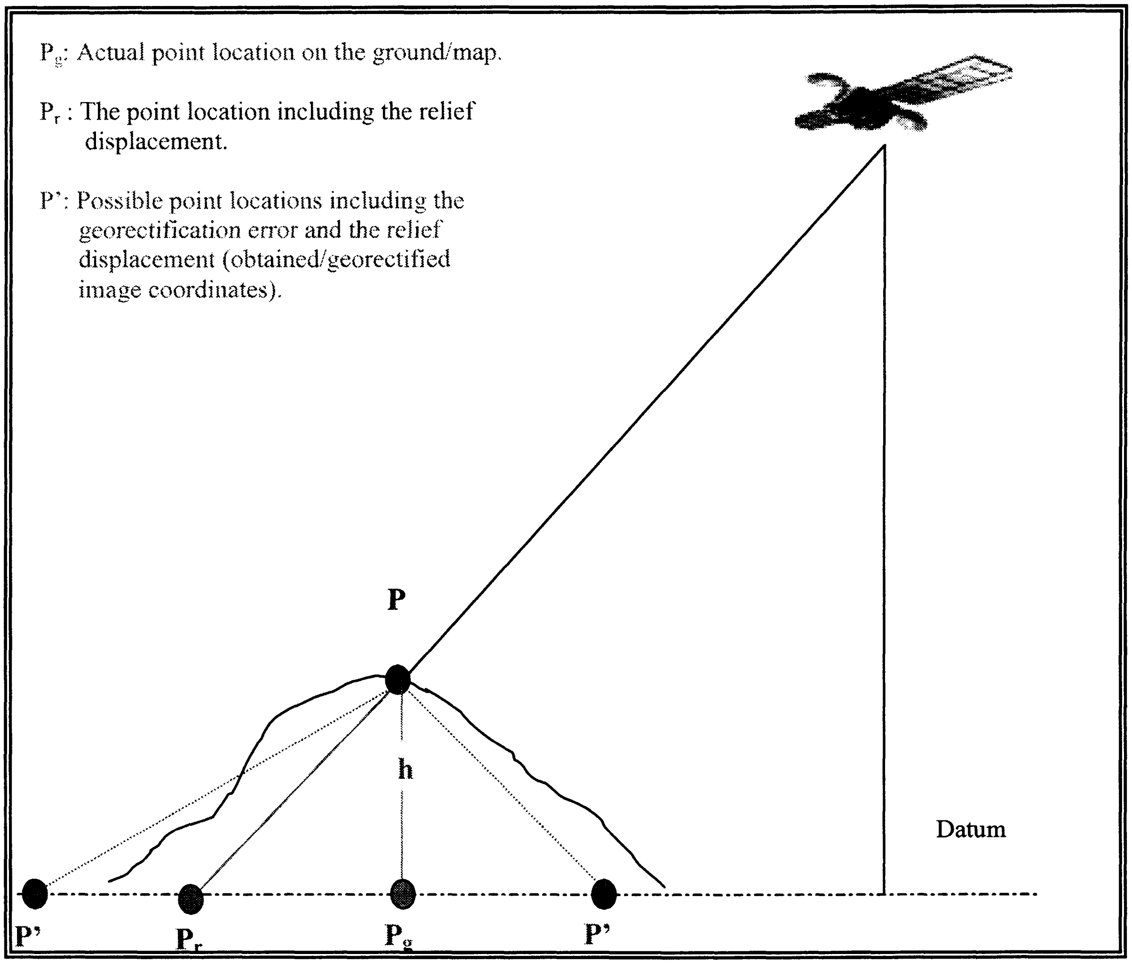


Figure 3.32
IKONOS sensor and relief displacement errors.

In order to eliminate the georectification error, we have to perform a coordinate transformation that brings any image pixel to its position where the relief displacement is the only remaining error, i.e. the (P_r) position.

The georectification process is accomplished by determining the position of the image center pixel(s) based on the sensor information. Based on this position and the predefined Ground Sampled Distance (GSD), the other pixels' coordinates may then be obtained. In other words, the error introduced in determining the sensor location and orientations (and, consequently, the center pixel position) is propagated through the whole image. Assuming that this error is almost constant across the image, this error can be resolved into shifts in X and Y directions, rotation angle, and scale in x and y directions. A six-parameter affine transformation should be enough in order to accommodate this error.

To perform this transformation, at least 3 control points must be identified within the image and on the ground. Since these GCPs may have errors as well, a decision was made to select 4 control points across the image and perform a Least Squares solution. In fact more GCPs can be selected to obtain a more accurate result. Only four GCPs will be used in this solution to allow the user to perform this task quickly, accurately, and without delaying the process.

The obtained transformation parameters are calculated based upon utilizing the four GCPs locations in the image (P') and their ground locations (P_g). By performing such transformation we are not only considering the georectification process error but also the average relief displacement that exists in the GCPs. As a result, using these obtained transformation parameters with any other point on the image will approximately produce the (P_r) location for this image point based upon a new datum (rectification plane). This new datum represents approximately the average height of the selected GCPs. Since we are using only four GCPs to estimate the georectification process error across the entire image, and for accurate results, we need for those GCPs to ideally possess almost the same elevation. Furthermore, the relief displacement within any of the GCPs selected is a function of both the elevation at this position and the sensor elevation angle at this location. Since we do not know the actual elevation angle at the GCPs location and since we need to eliminate the georectification process error only, we need to select the lowest GCPs within the image area. In other words, for this step to be completed accurately the user has to select four GCPs that are very low. After performing such a transformation and obtaining the required transformation parameters, any image point/pixel may be transformed to obtain (P_{rx}, P_{ry}).

3.4.4.2 Relief Displacement Correction

By applying the previously mentioned transformation parameters to any of the image points, the produced/transformed coordinates should be approximately corrected for the georectification processing errors (P_{rx}, P_{ry}) and based upon average GCPs elevations as the new datum. The next step in the process is to obtain the correct ground/orthorectified coordinates of the desired point (P_{gx}, P_{gy}). For this we need to consider the geometry shown in Figure 3.33. Based upon this conventional geometric situation, the relief displacement component can be calculated by knowing two main elements, namely, the angle (I) or (E) and the point elevation based upon the newly established datum as shown in Figure 3.33.

In order to obtain the point (P) elevation, a DTM must be available for this area. If a DTM is not available, an average elevation for the entire area should be considered. Even with a DTM we still have a problem. If the transformed point is used within the interpolation procedure the problem is simply that we are not interpolating at the desired point/feature location. This problem can be either ignored or solved to a certain extent. It can be ignored if the relief displacement is not too large or if the image is a nadir image. In this case the transformed point (P_r) and the ground point (P_g) will probably be located

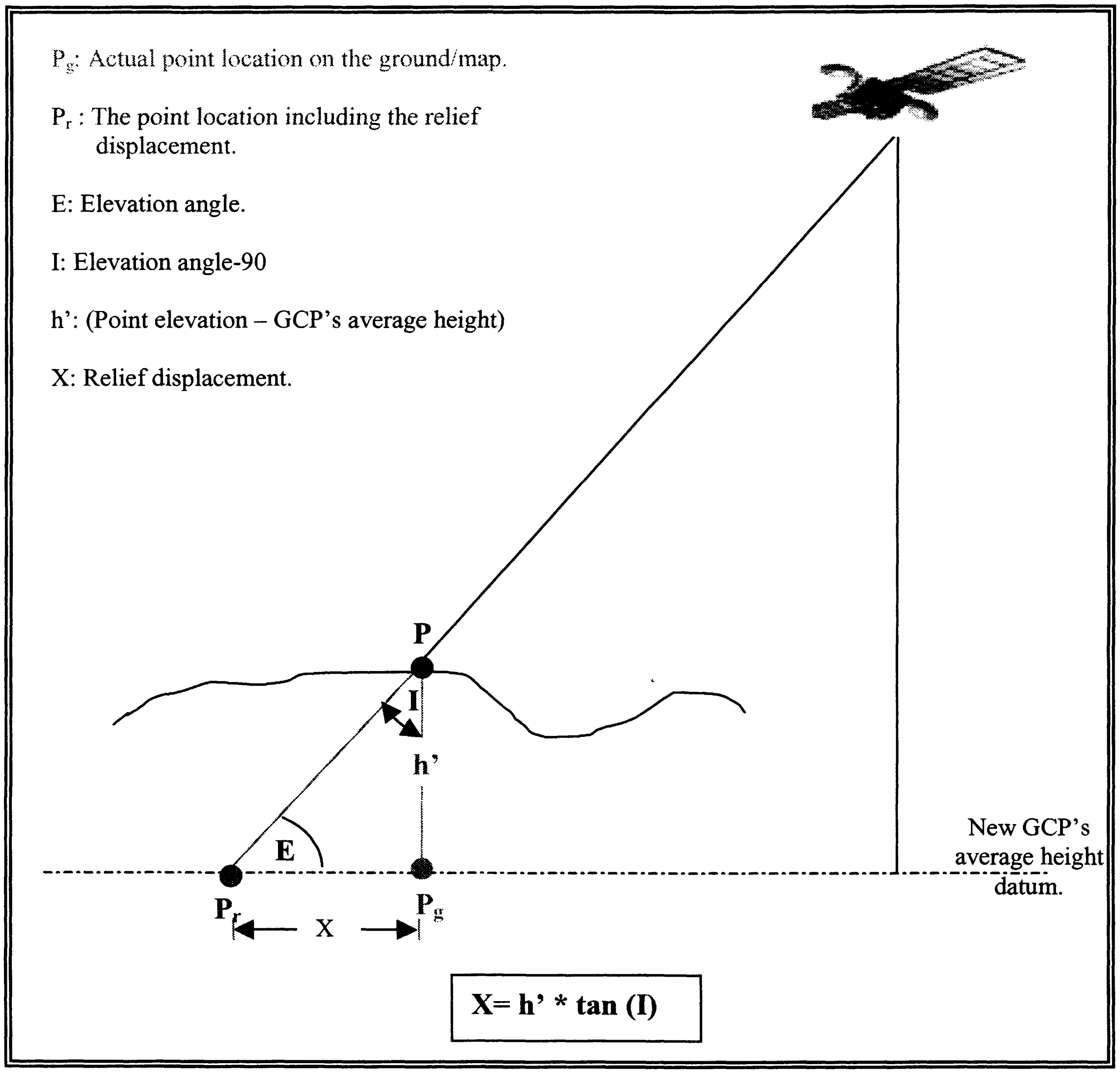


Figure 3.33
Relief displacement component of the ground points.

within one cell or neighboring cells in the DTM. In other words, if we interpolate at either of the two locations we will get very close interpolation results. If this problem has to be considered the following procedure can be followed:

1- Obtain the minimum and maximum elevations within the whole image area.

2- Calculate the minimum and maximum relief displacements expected within the considered area, corresponding to the obtained minimum and maximum elevations (the value of the calculated/approximated angle (I) in order to calculate the relief displacement will be considered later this section.)

3- The point (P_g) will be located away from the point (P_r) and towards the satellite nadir line within a distance range that falls between the minimum and the maximum displacement values obtained. Then, we can calculate the distance between the point (P_r) and the nearest DTM points and select all the points at distances within the specified range, as shown in Figure 3.34. (Locating the satellite nadir line position will be discussed later this section.)

4- Identify the DTM points located within the previously specified range and obtain an average elevation for the terrain within this range.

5- This average value is the elevation of the desired point (P_g).

After interpolating the desired elevation value, we can obtain the magnitude of the relief displacement of this point. Three main questions remain to be answered, namely:

1- What is the elevation angle that should be used to calculate the relief displacement of any desired point within the image?

2- Where is the satellite nadir line located relative to the image ?

3- The obtained relief displacement component is the resultant of two relief components in X and Y directions. How to obtain the amount of these shifts? Should each shift be added to or subtracted from the transformed coordinates (P_{rx}, P_{ry}) in order to obtain the actual ground coordinates (P_{gx}, P_{gy}) of the desired point?

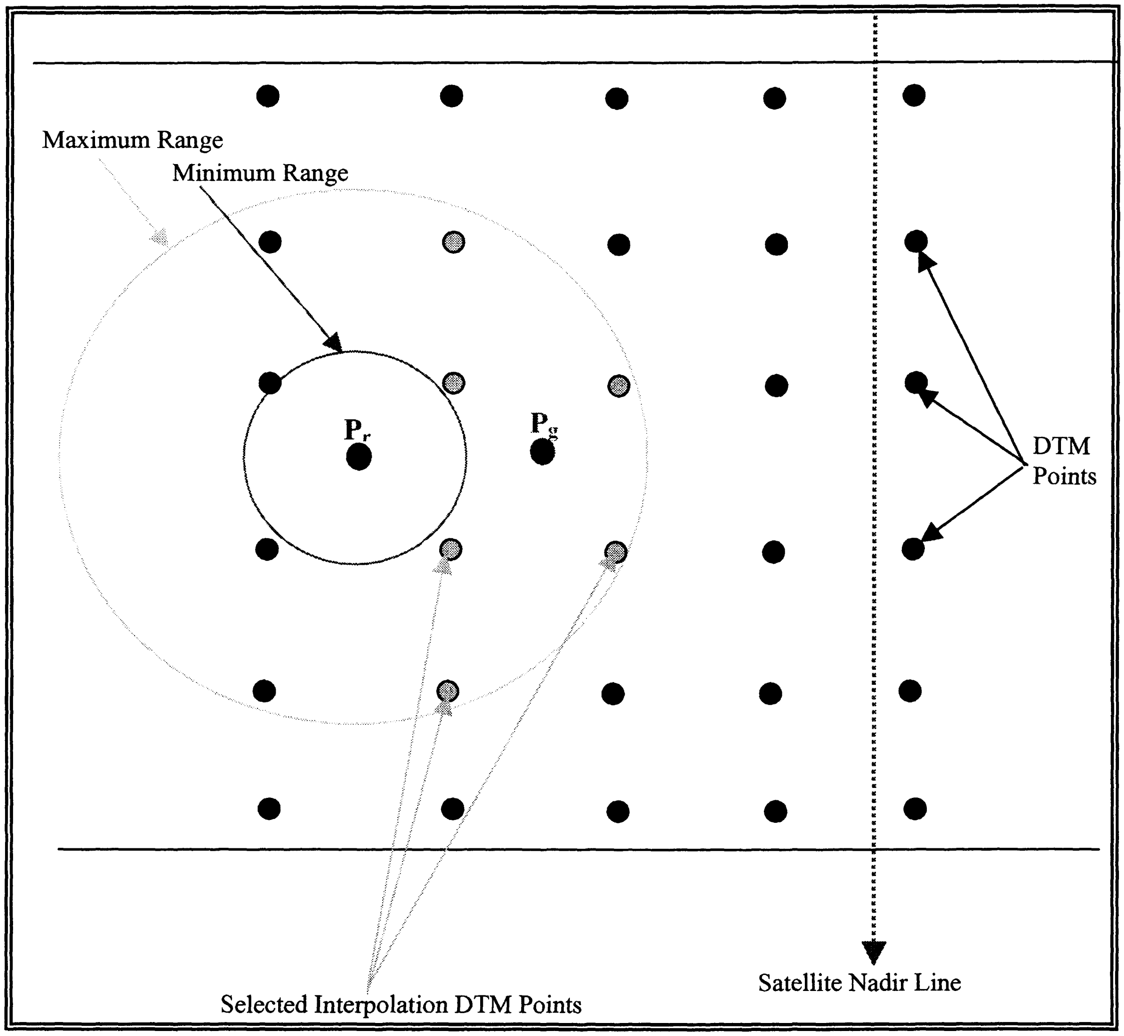


Figure 3.34
DTM interpolation.

The metadata file accompanying the image file provided by Space Imaging has two parameters that are used to answer the above questions. These parameters are the "Nominal Collection Elevation" and the "Nominal Collection Azimuth". Extensive research was conducted to obtain a reliable and an accurate definition of these two parameters. Among the few sources that deal with the IKONOS imagery specifications, the following definitions have been found [Space Imaging Europe, n.d.]:

Nominal Collection Elevation: is the angle between the target point's horizon (Tangent Plane) and, the line connecting the target point with the satellite.

Nominal Collection Azimuth: is the angle between the line connecting the target point and the satellite's projection on the target point's horizon (Tangent plane) and, the N-S direction. The direction of this angle is always clockwise.

Considering Figure 3.35, the Nominal Collection Elevation and Azimuth angles are represented by the angles (E) and (Φ) respectively.

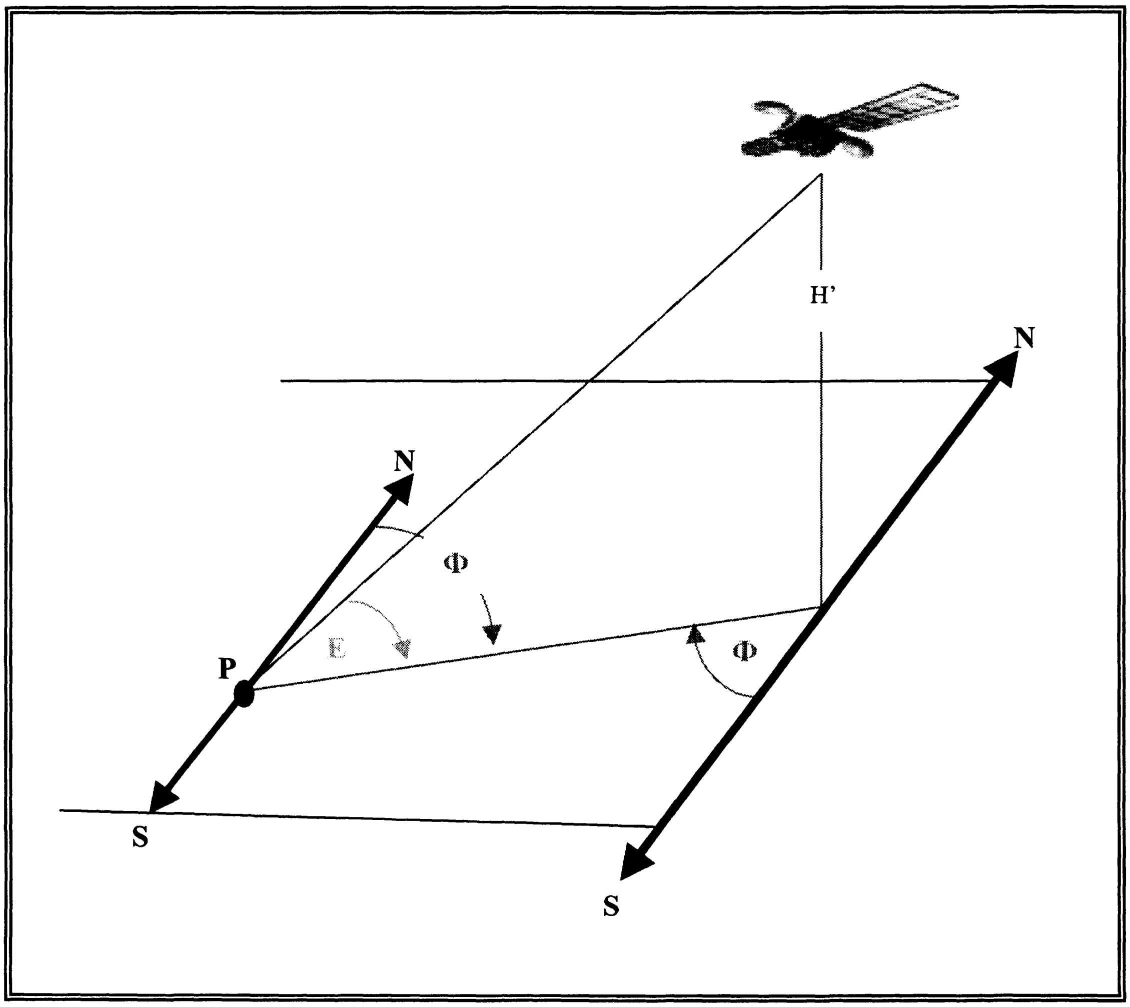


Figure 3.35
IKONOS Nominal collection Elevation and Azimuth.

In order to check the correct understanding of these two definitions and to verify the previous assumption, certain calculations were carried out. Using the Elevation angle (E), we can approximately calculate the acquired nominal GSD in the cross-track and along-track directions. This can be done utilizing the following equations [Lillesand and Kiefer, 1994];

$$GSD1 = H' * [SEC(90 - E)]^2 * \beta$$

$$GSD2 = H' * [SEC(90 - E)] * \beta$$

where: GSD1=Acquired Nominal GSD in the cross-scan direction.

GSD2=Acquired Nominal GSD in the along-scan direction.

H'= Sensor height. β = Instantaneous Field Of View (IFOV).

E= Nominal collection elevation angle.

Using the IKONOS characteristics:

β = 1.2 micro radians

H'=681 km

90-E= 90-68.18=21.82 degrees (Fredericton's selected IKONOS image)

we obtain:

GSD1=0.94 m

GSD2= 0.88 m

These two values matched the values given by Space Imaging's delivered image metadata (0.95, 0.88 m). The same procedure was performed to check other metadata files posted on different Web sites, see for example, MODIS [2000], and similar results obtained. This provides a reasonable confidence that the elevation angle provided within the IKONOS image metadata is the angle (E) as shown in Figure 3.35.

The nominal acquired GSD could be located anywhere based on the specified elevation angle. The nominal collection azimuth determines the location at which this GSD was acquired based upon the angle (Φ) as shown in Figure 3.35. Based on the angle (Φ) associated with the Fredericton image and by examining the building features appearing on the image, the definition of the angle (Φ) matched the one in Figure 3.35. The angle (E) may have any value between 90 degrees (nadir image) and 50 degrees (off-nadir) viewing [Gerlach, 2000]. Obviously, the angle (Φ) may range between 0 degrees and 360 degrees.

Within the developed method, the nominal collection elevation was considered to be the same across the entire scan line and consequently the entire image. This is a valid assumption for off-nadir viewing imagery. The validity of this assumption is based on the fact that the sensor viewing angle (90-E) in an off-nadir viewing image is large (up to 40~45 degrees) compared to the sensor's FOV (0.931). For example, as a percentage of the relief displacement itself, it will not make a significant difference if we considered the sensor viewing angle to the point of interest to be 30 degrees or (30 +/- 0.931/2 degrees) in the extreme case. In the nadir-viewing situation the relief displacement will not have a significant effect. For example, if we have a point with a 1000 m elevation above ground, the maximum relief displacement would occur if the point with this height is located at a viewing angle of (0.931/2) at the two ends of the scan line/swath with a magnitude of (1000* tan (0.931/2)=8.1 metre). Consequently, in a nadir-viewing imagery, the angle can be approximated as (0.931/4 degrees), i.e., we consider the point of interest to be in the middle of each side of the scan line and we can take an average height for the area under consideration. Based on this assumption, the magnitude of the relief displacement can be calculated and the first of the above mentioned three questions is answered.

In order to answer the second question, we have to consider the angle (Φ). As illustrated in Figure 3.36, four situations may exist. At point (1) the nominal collection azimuth angle (Φ) ranges between (0 – 90) degrees, at point (2) the range is (90-180) degrees, at point (3) the range is (180-270), and at point (4) the range is (270-360). Based upon the angle (Φ) the satellite nadir line can be located either to the right or to the left of the georectified and north-up aligned IKONOS image.

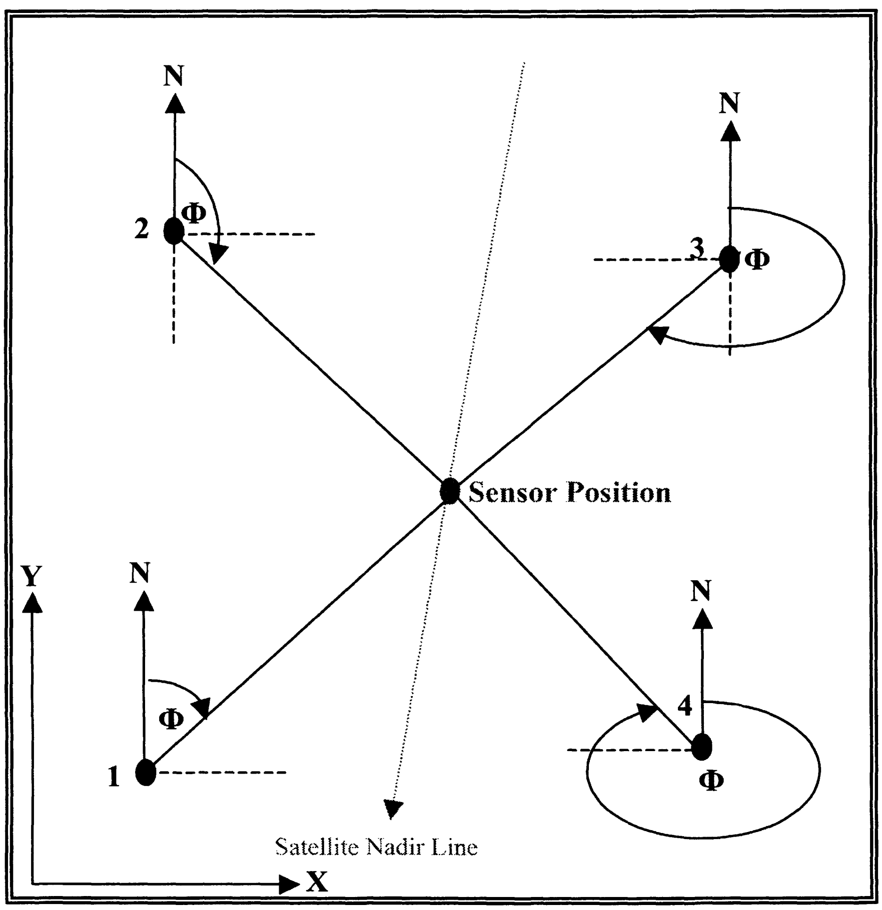


Figure 3.36
Nominal Collection Azimuth (possible cases)

Determination of the location of the image relative to the satellite nadir line provides two more benefits in the off-nadir viewing case, which will lead to the answer of the third question. First, based on this angle we will be able to identify whether the relief displacement components (X,Y) should be added to or subtracted from the transformed image position (P_r), considering the X,Y coordinate system as shown in Figure 3.36. Since, for most cases, the relief displacement direction is always away from the nadir line, the correction for the relief displacement components in both directions in Case 1 in Figure 3.36 should be added. Similarly, in Case 2, the relief displacement in X direction must be added and in Y direction must be subtracted. In Case 3 the relief displacement components in X and Y directions must be subtracted. In the last case, Case 4, the relief displacement X component must be subtracted and Y component added.

The next step is to determine the X and Y components of the relief displacement. In this step we have to make two assumptions. First, we assume that the collection azimuth for any point on the image is constant and equal to the nominal collection azimuth. Considering Figure 3.37, this assumption is valid. The distance PP_e is very small compared to the distance O'P or O'P_e. Using a simple calculation, at a 60 degree elevation angle the length PP_e is approximately 6.5 km and the distance O'P approximately equals the distance O'P_e at 393 km. Secondly, the angle (D) then can be calculated and is approximately < +/- 1 degree. Consequently, this will produce the same angular difference in the collection azimuth angle. A maximum of one-degree difference in the collection azimuth angle will not produce a significant effect, especially when having moderate terrain elevation within the area under considerations, in calculating the relief displacement X and Y components.

In order to determine the X and Y components of the relief displacement, we have to consider the direction in which this relief displacement took place. In a normal situation in nadir-viewing or when we have a cross-track pointing capability, the relief displacement takes place in the scan line direction and away from the satellite nadir line. Due to the fact that IKONOS imagery is taken with both in-track and cross-track off-nadir pointing, the displacement now is not only in the scan line direction but also has components in the direction perpendicular to the scan line.

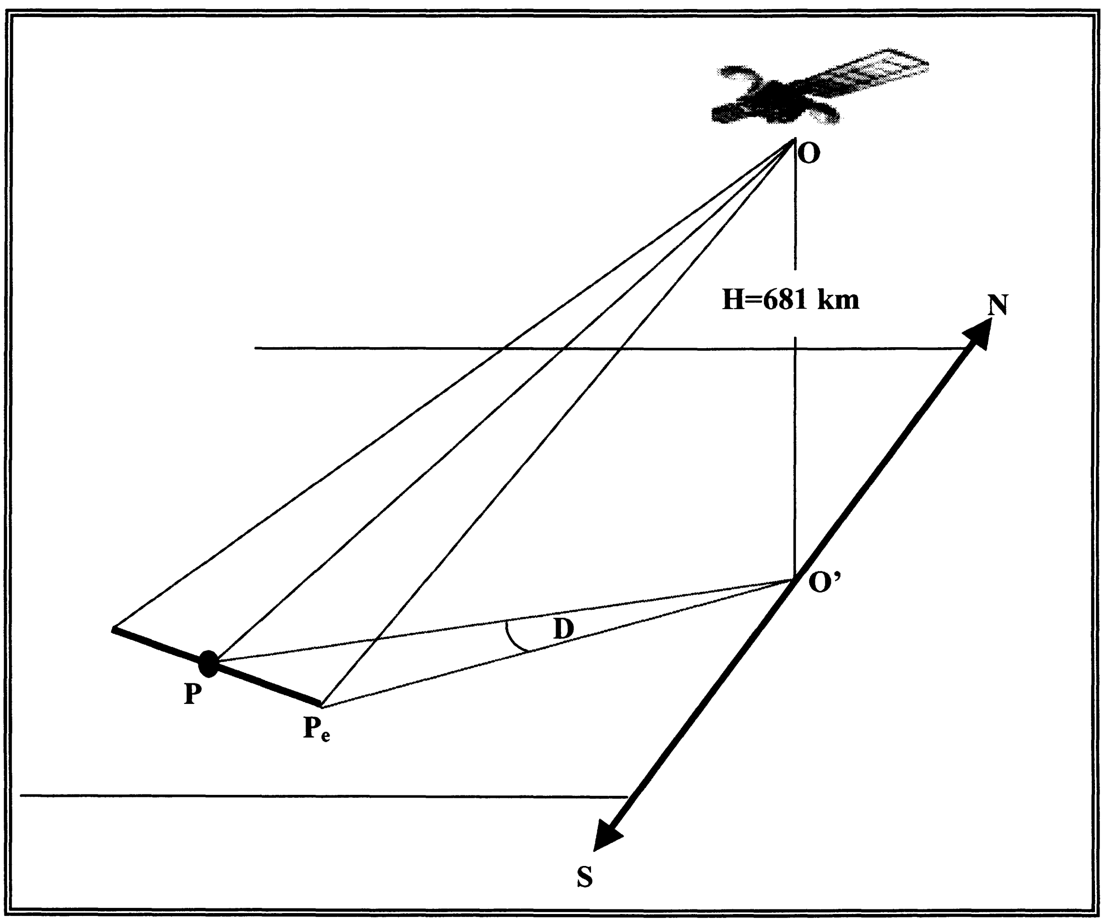


Figure 3.37
Collection Azimuth across the image.

Considering Figure 3.38, the X and Y components of the relief displacement for any point (P) on the image are a function of the actual elevation angle and the actual collection azimuth of this point. Since we assumed that the elevation angle and the collection azimuth in any point in the image is constant and equal to the image Nominal Elevation Angle and Nominal Collection Azimuth respectively, an assumption has been made that the relief displacement at any point will occur in the collection azimuth direction. Although this assumption may not represent the exact geometrical situation, it provided a reasonable overall accuracy within the final output when tested using different image points with different elevations and at different positions on the Fredericton image.

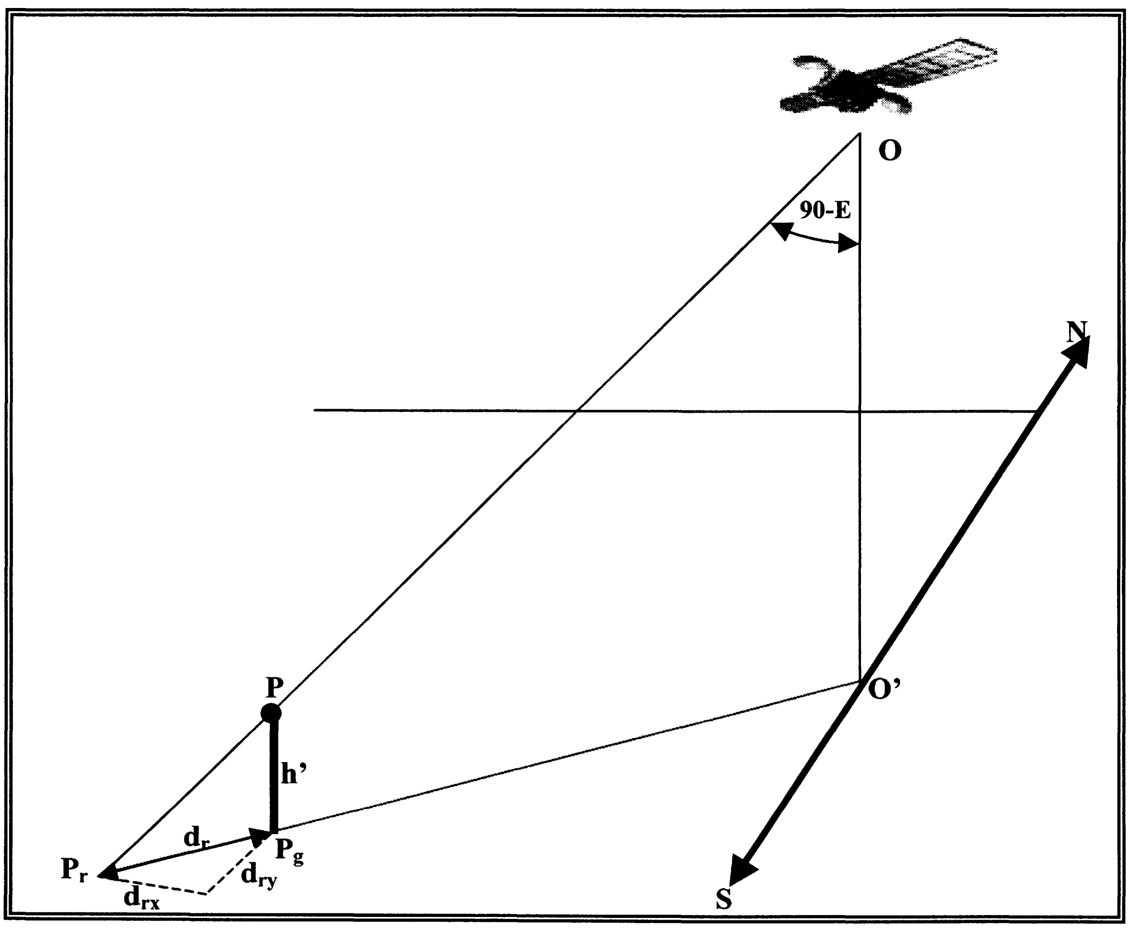


Figure 3.38
Relief displacement X and Y components magnitude and direction.

The correct and approximately orthorectified ground position (P_{gx}, P_{gy}) may be obtained using the following two equations:

<u>If (Φ) < 90 Degrees</u>

$$P_{gx} = a1 + a2 * (P'_x) + a3 (P'_y) + (h' + Z_f) * Tan\ (90 - E) * Sin\ (\Phi)$$

$$P_{gy} = a4 + a5 * (P'_x) + a6 (P'_y) + (h' + Z_f) * Tan\ (90 - E) * Cos\ (\Phi)$$

<u>If 90 Degrees <(Φ)< 180 Degrees</u>

$$P_{gx} = a1 + a2 * (P'_x) + a3 (P'_y) + (h' + Z_f) * Tan\ (90 - E) * Cos\ (\Phi - 90)$$

$$P_{gy} = a4 + a5 * (P'_x) + a6 (P'_y) - (h' + Z_f) * Tan\ (90 - E) * Sin\ (\Phi - 90)$$

<u>If 180 Degrees <(Φ)< 270 Degrees</u>

$$P_{gx} = a1 + a2 * (P'_x) + a3 (P'_y) - (h' + Z_f) * Tan\ (90 - E) * Cos\ (270 - \Phi)$$

$$P_{gy} = a4 + a5 * (P'_x) + a6 (P'_y) - (h' + Z_f) * Tan\ (90 - E) * Sin\ (270 - \Phi)$$

<u>If 270 Degrees <(Φ)< 360 Degrees</u>

$$P_{gx} = a1 + a2 * (P'_x) + a3 (P'_y) - (h' + Z_f) * Tan\ (90 - E) * Sin\ (360 - \Phi)$$

$$P_{gy} = a4 + a5 * (P'_x) + a6 (P'_y) + (h' + Z_f) * Tan\ (90 - E) * Cos\ (360 - \Phi)$$

where:

P_{gx}, P_{gy}: Corrected ground coordinates of the desired image feature

P'_x, P'_y: Georectified image point position (feature's position from image)

a1,…..,a6: Transformation parameters

h': Feature interpolated elevation measured from the GCPs average height datum

Z_f: Feature's height above ground (if applicable), for example, buildings and towers

E: Image Nominal Elevation Angle ***Φ***: Image Nominal Collection Azimuth.

3.4.5 Sensor Error Elimination (Method 2)

As explained earlier in section 3.4.4.1, the sensor location error can be approximately treated by using a 4 GCPs affine transformation. This section presents another technique for eliminating the sensor error without performing the previously specified transformation.

The sensor error associated with IKONOS GEOCARTERRA™ products depends on the accuracy of the satellite location, altitude and attitude determination over time, and, consequently, in identifying the image center pixel coordinates. This error will be propagated through the entire resampled and projected image and will be presented within the entire coordinates of the image pixels. This error will approximately cause the center pixel and, consequently, all image pixels to shift away from the correct position mainly by X and Y components/offsets. This method assumes that this error can be resolved into two main components, namely, shifts in X and Y directions, and then these two components will propagate constantly across the entire image. If these two components can be approximately determined, the image pixel coordinates can be corrected from the sensor error producing new coordinates that contain almost pure relief displacement.

The determination of average and approximate values of the two shift components can be accomplished using GCPs. As considered within this research project, 3 GCPs have been used to perform such a task.

For any control point, the difference between the image's point coordinates and the actual point's ground coordinates presents the final shift introduced mainly by the sensor error and the relief displacement. As shown previously in Figure 3.32, the relief displacement brings the theoretical/ground/correct location of the point (P_g) to the location (P_r) and then the sensor error/shift brings the location (Pr) to the actual georectified image location (P'). By considering the height, the elevation angle, and the collection azimuth of the GCPs, to be equal to the nominal ones, we can calculate the shift introduced due to the sensor error. It is important to note that different formulas representing different cases/values for the angle (Φ) should be considered to be able to match the exact scenario for each case. For example, considering Case 1 in Figure 3.39, the location (P'_x) is produced by shifting the ground location (P_{gx}) by the relief displacement's X component and then by the sensor error shift in X direction.

Following the formulas developed as shown in Figure 3.39, the sensor error shift components in X and Y directions can be calculated for each of the 3 GCPs used and then averaged. The corrections to the two shift components will have the same values as the shift components but with different directions. The two average shift corrections (C_x, C_y) can be added or subtracted, based on their sign, to any image point coordinates, thus producing the location of the point containing only relief displacement (P_{rx}, P_{ry}). By treating the relief displacement as explained previously, good approximate of the actual/correct/orthorectified coordinates of the desired point can be obtained and used to query the data layers. This method has been tested and the results show an accuracy (RMSE) of +/- 2.75 metres that is better than using the 4 GCPs affine transformation and are approximately within the same range for the IKONOS orthorectified imagery, based upon the Fredericton image used (0-122 m elevation variations), as shown in Appendix C.

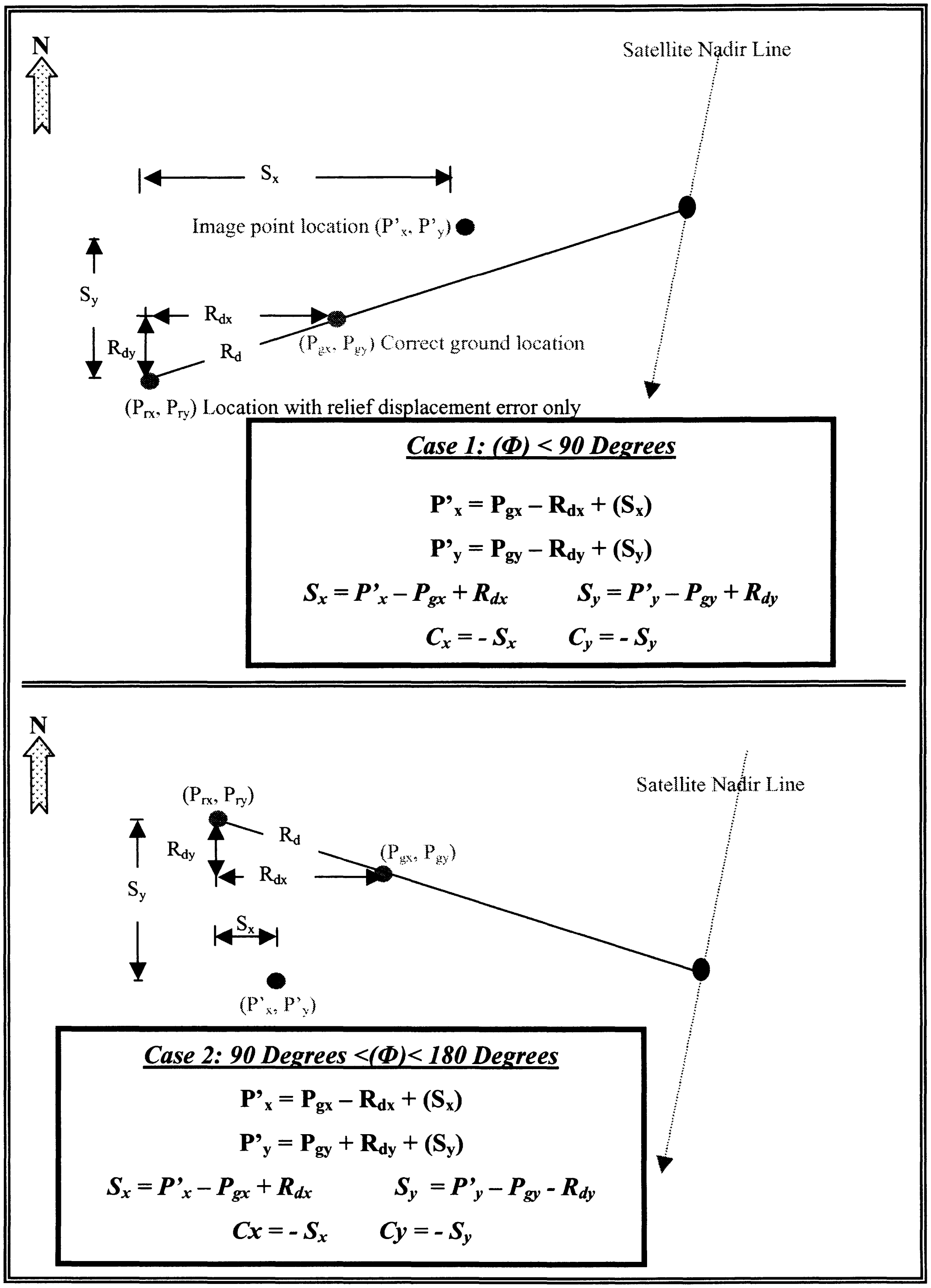
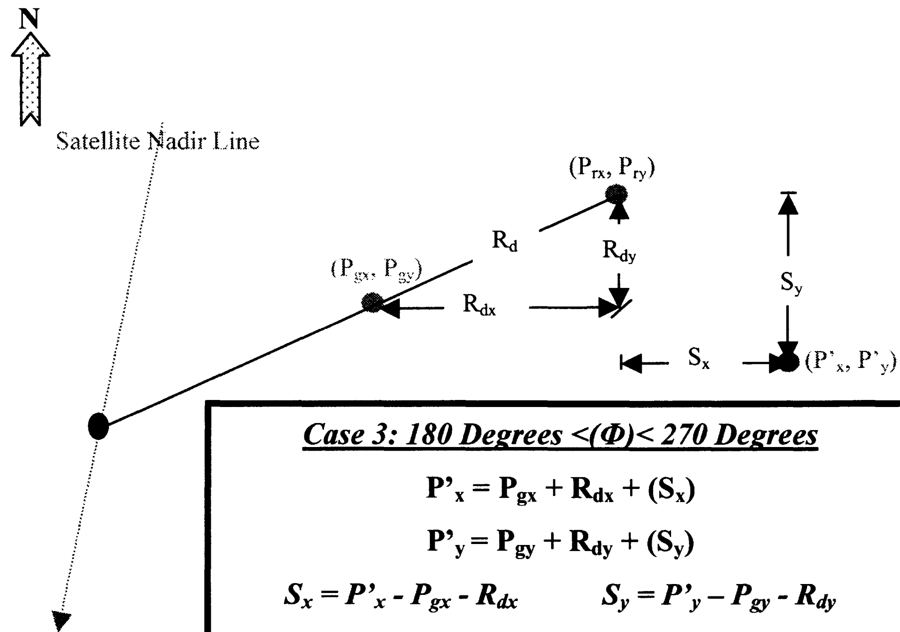


Figure 3.39
Sensor error elimination (Method II).



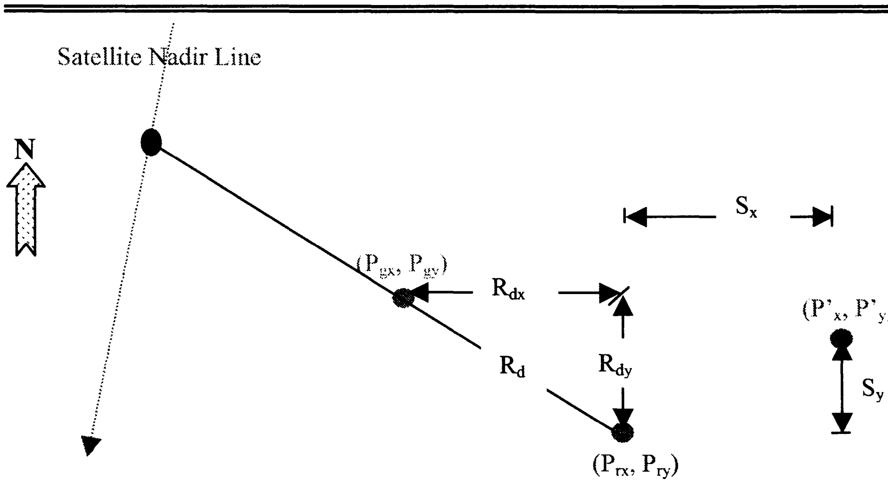
Case 3: 180 Degrees <(Φ)< 270 Degrees

$$\mathbf{P'_x = P_{gx} + R_{dx} + (S_x)}$$

$$\mathbf{P'_y = P_{gy} + R_{dy} + (S_y)}$$

$$S_x = P'_x - P_{gx} - R_{dx} \qquad S_y = P'_y - P_{gy} - R_{dy}$$

$$C_x = - S_x \qquad C_y = - S_y$$



Case 4: 270 Degrees <(Φ)< 360 Degrees

$$\mathbf{P'_x = P_{gx} + R_{dx} + (S_x)}$$

$$\mathbf{P'_y = P_{gy} - R_{dy} + (S_y)}$$

$$S_x = P'_x - P_{gx} - R_{dx} \qquad S_y = P'_y - P_{gy} + R_{dy}$$

$$C_x = - S_x \qquad C_y = - S_y$$

Figure3.39
Sensor error elimination (Method II) (cont.)

3.4.6 IISP: Case 3 Design and Implementation

The main issues here is to map/implement the theoretical solution developed as explained in sections 3.4.4 and 3.4.5 within the system process. This includes identifying the requirements to handle such a process, and designing and modifying the interface components and also the system workflow in order to meet the requirements for this case.

Clearly, a DTM for the desired area may not always be available. In order to identify and handle the problems and the requirements of the worst case, it was decided to implement a rectification process that handles the 4 LOW GCPs affine transformation and then using an average terrain elevation within the area to calculate an approximate amount of the relief displacements. In other words, in this implementation stage, no DTM interpolation was implemented. In addition to that, this case was implemented with the projection Method I as explained in section 3.3.5.1.

3.4.6.1 Design Considerations

In order to implement this case, the design considerations as well as the interface/process requirements have to be identified first. Consequently, the following questions had to be answered:

1- In specifying the GCPs and required metadata parameters, how will the user interact with the system, and what are the main requirements within this process?

2- How and when should the system perform the required affine transformation process, and what are the components required for this process?

3- How and when should one consider the input to the terrain height?

4- How can one consider layers that have man-made features with their own height above the ground?

5- How should the system be designed and modified to handle an interpolation component as well as to calculate the average sensor shift errors using 3 GCPs, as explained earlier?

6- How and when should one perform the back transformation in order to display the vector layer (query results) features in their correct position in the distorted/partially rectified imagery?

The above questions are analyzed and answered in the following sections.

3.4.6.2 Stage 1: "Data Loading and Identification"

In stage 1, the image layer with its required projection, coordinate system, and metadata parameters is loaded into the system. In addition, the physical location vector layers are identified to the system to facilitate the image-map linkage process for later stages and functions. The next step is to allow the user to input the GCPs required. Two methods exist within the IISP (Case 3) allowing the user to input the required GCPs. The user can input these GCPs in an interactive process if it is the first time the image is used or through a direct reading from a text file if the image has been used before and the GCPs have been identified previously. The selection of either one of these two options is done using a Message Box as shown in Figure 3.40.

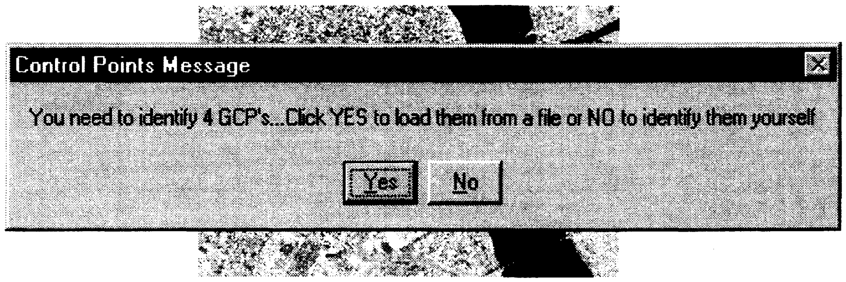


Figure 3.40
GCPs identification options Message Box.

If the user selects the interactive process, a Message Box pops up asking the user to identify the desired GCPs by right-clicking the mouse on the desired location. The user then selects the first GCP, the selected location is read by the system and then flashed to the user for confirmation, and the "GCPs Identification" form appears. The user then can input the values of the ground coordinates (X, Y, Z) of the selected GCP. Within this form the user can flash the GCPs for visualization purposes, enter the required metadata parameters, and save the GCPs entered into a Comma Delimited text file for further usage. The "Load From File" command button was added to the form in order to allow the system to read and input the entire group of GCPs from a file without the need for interactive selection, in case of having the GCPs file created previously, Figure 3.41 shows the "image and metadata input" form.

In order to perform a least-squares solution using the selected GCPs and to calculate the transformation parameters (a1 through a6), the system uses the image and ground (X, Y) coordinates to create the design matrices (A, L) and the normal equations with dimensions (8x6) and (8x1), respectively. The system then calculates the required parameter matrix (X) with the dimension (6x1) using the following well-known formula, assuming equal weights:

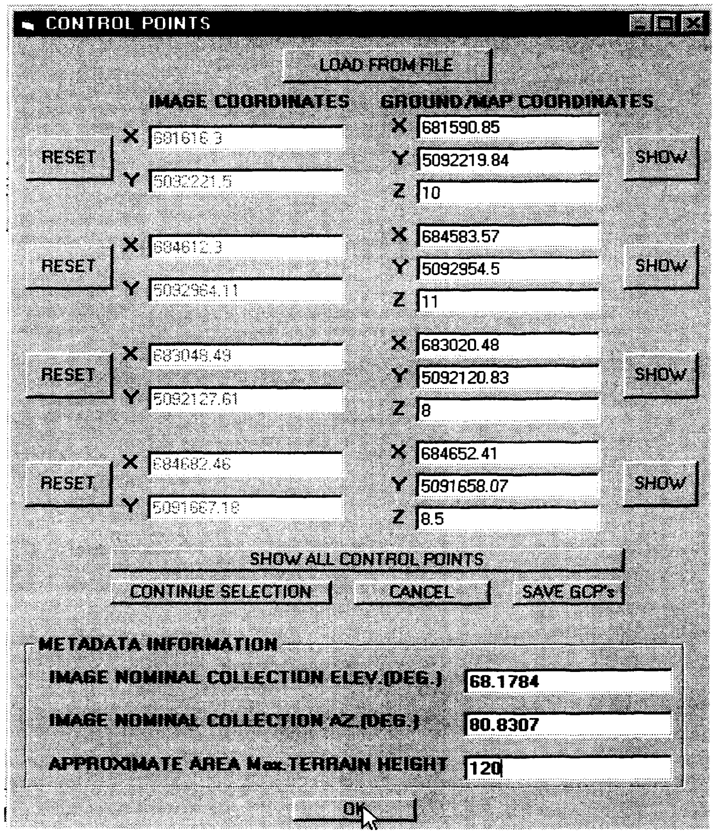


Figure 3.41
GCPs and image metadata input Form.

$$ {}_{6}\hat{X}_{1} = ({}_{6}A^{T}{}_{8} * {}_{8}A_{6})^{-1} * ({}_{6}A^{T}{}_{8} * {}_{8}L_{1}). $$

In order to create and manipulate these matrices, the system includes a reference to the Microsoft Excel Object Library, and then calls on this library to perform the desired task, using MMULT, MINVERSE, and TRANSPOSE methods, as shown in the code attached to the GCP form in the IISP_CODE.PDF file. By the end of this process, the system calculates and stores the affine transformation parameters to be used later within the query process. The average elevation within the image area is then calculated based on the maximum elevation that exists within the image and the average GCPs elevation, as shown in Appendix C. The image is now ready for the query. Figure 3.42 shows a schematic diagram for the process of this stage.

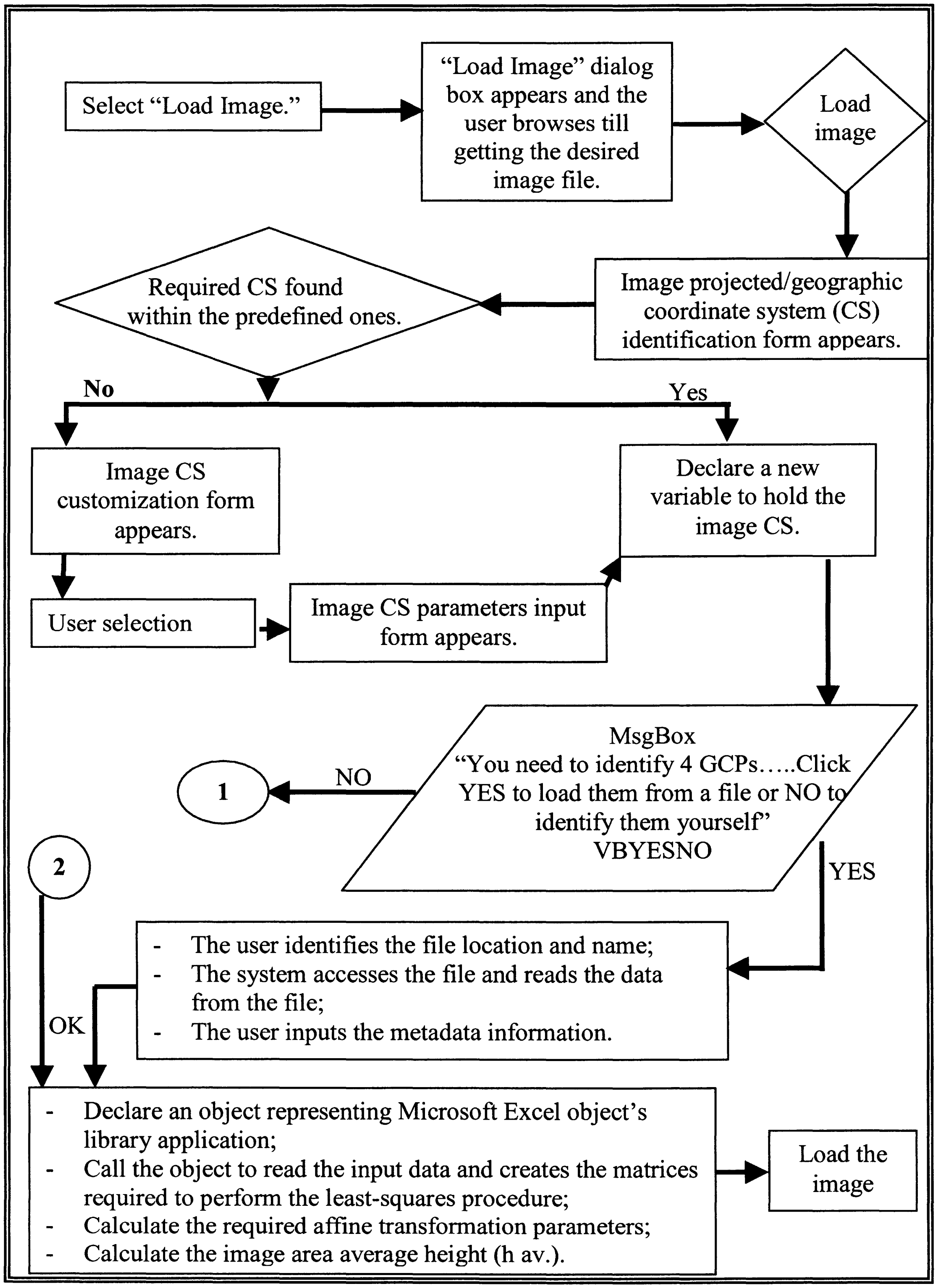


Figure 3.42
Case 3 "image loading" process.

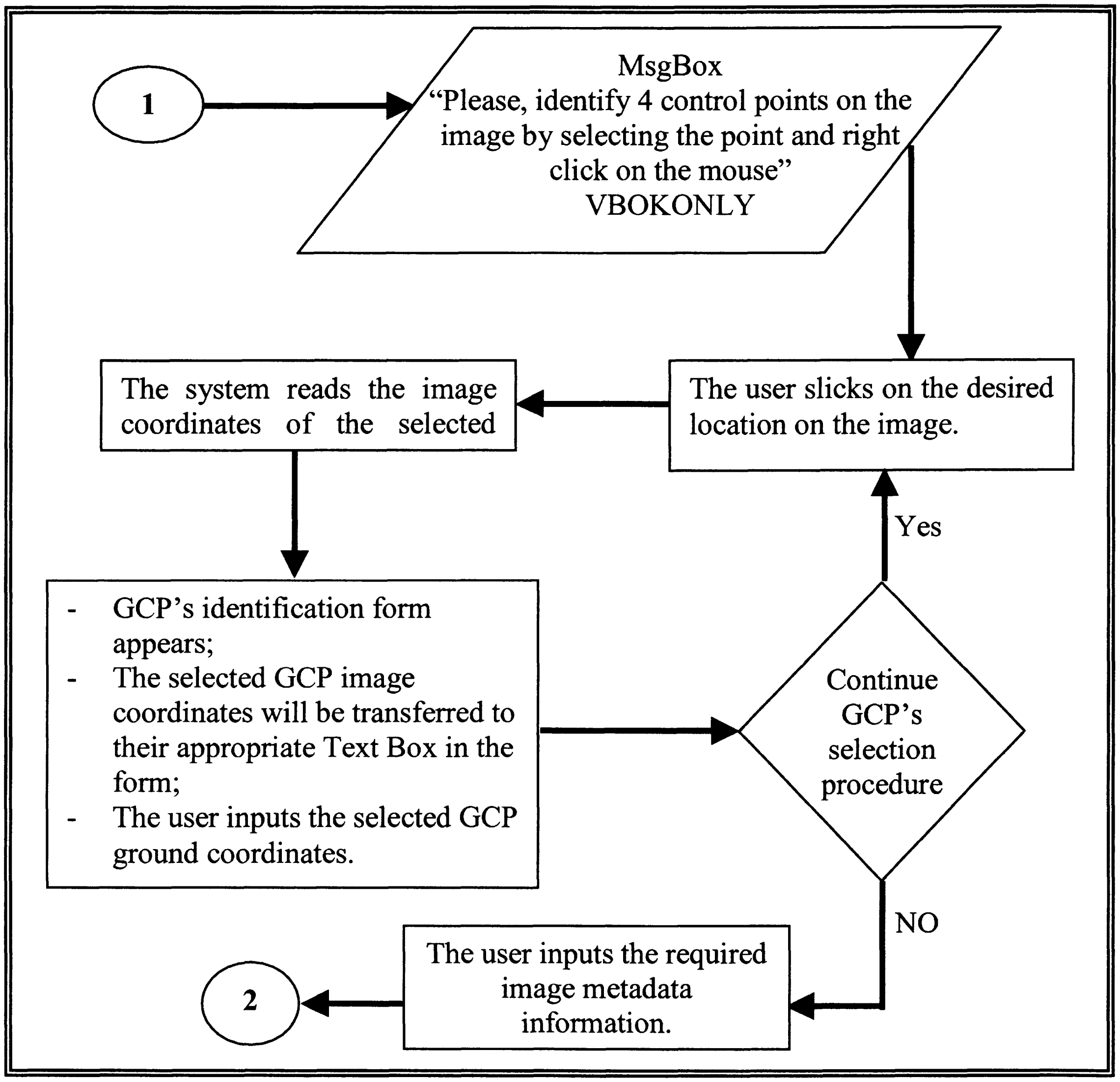


Figure 3.42
Case 3 "image loading" process (cont.)

In identifying the vector layers, the system follows the same process as explained previously in section 3.3.5.1 with one modification. As the layer is being identified, the system identifies the layer type (point, line, polygon). If the layer is a polygon layer, it may contain building features (features that have their own height above ground). The system, therefore, presents another form with two options to confirm whether or not the layer contains features with heights above the ground, as shown in Figure 3.43. The system then assigns a code to this layer based on the user's confirmation (0 or 1) and an Input Box appears to the user allowing for an average features' heights, as a one possible solution, for this layer to be entered (Z_f). The system is now ready to perform the image query process.

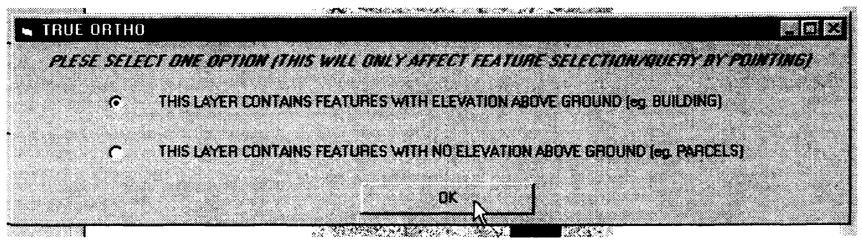


Figure 3.43
Identification if the polygon layer's feature type

3.4.6.3 Pointing Query Function

The pointing query function, as shown in Figure 3.44, involves the following steps:

1- The layer to be queried is selected.

2- The user clicks on a position (P) on the image, and the system reads the georectified image coordinates of this position.

3- The system uses the transformation parameters (a1 through a6) to transform the image coordinates to a new position in which the coordinates contain only the effect of the relief displacements (P_{rx}, P_{ry}).

4- Using the previously calculated average terrain elevation (h av.), the feature height (Z_f), if any, and based upon the two angles (E) and (Φ), the system calculates the two relief displacement components at the desired location (drx, dry).

5- Based on the value of the angle (Φ), the system recognizes whether to add or subtract these relief components to or from the transformed coordinates (Prx, Pry), based upon the four cases presented in section 3.4.4.2. This produces the desired ground coordinates (Pgx, Pgy).

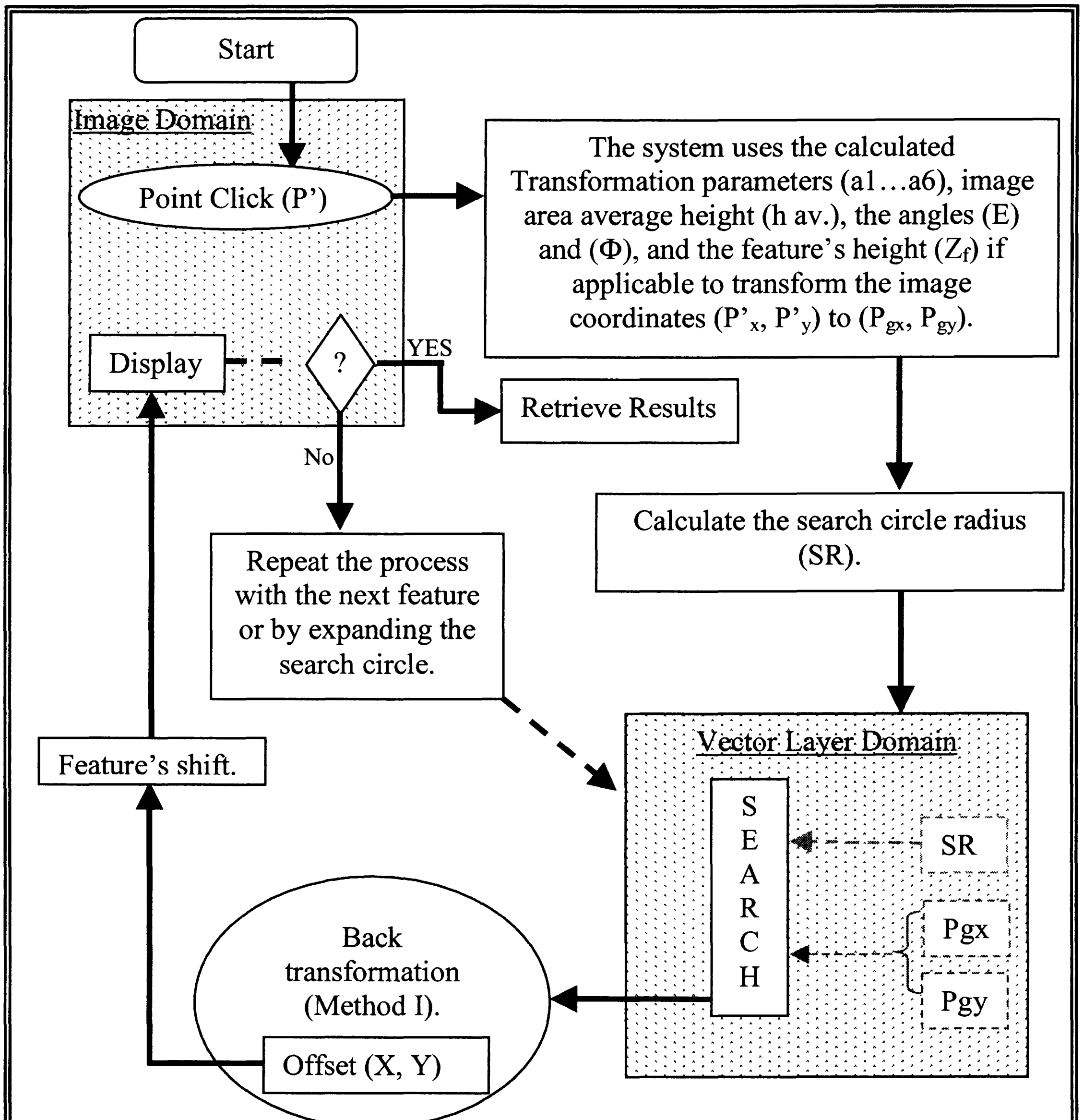


Figure 3.44
Case 3 "Pointing Query Function" processing steps.

6- The system then uses the final coordinates to search the vector layer using a search circle with a certain radius. This radius is based upon the expected error existing within the selected point. By recognizing the maximum elevation of the area and the average terrain height used, the system calculates the maximum expected CE within the image and then assigns it to the search radius using the following formula:

Search Radius = (h max. – h av.) * TAN(90-E).

7- If no features are found, the user gets confirmation and is allowed to increase the search radius if required in a process similar to the one explained in section 3.2.2.

8- If a feature or group of features is found, the system displays it to the user one by one allowing the user to select the desired feature.

In displaying the feature to the user, unless being treated, we should note that there will be a discrepancy between corresponding features because the image is not orthorectified and the map feature is at its true ground position. In order to address this problem, we can perform the first method of the back transformation from ground to image coordinate system. If we assume that the produced coordinates of the transformed and relief corrected image point fall exactly on the desired ground point, then the difference between the coordinates of the original user-identified/clicked image point and the transformed coordinates represents the actual X and Y shift components between the point on the image and the actual ground point.

For display purposes, the resulting feature is shifted with the same calculated shift values, using the feature's "Offset" method that can move the feature in both X and Y directions, and then is flashed to the user. Although this method may not produce the exact resulting feature position on the image, in the case of using the average elevation (h av.), it brings the ground feature to a closer position of its distorted image position and, consequently, is displayed with better approximation on the image. It is important to note that, when implemented and tested with an accurate DTM elevation interpolation, this method affected the displayed results in a significant manner and brought the resulted feature to its almost exact location on the image.

Another back transformation method was implemented with the other functions, as will be explained later in section 3.4.6.4.2.

3.4.6.4 Polygon/Line-Based Image Query Function:

In this Case 3, the Polygon/Line Image Query Function involves the same processing steps as identified in the previous two cases, as explained in section 3.2.3 and section 3.3.5.1, with two modifications.

3.4.6.4.1 Modification I: polygon/line transformation

Within this function, it is of great importance to recognize the intent of the user. When the user draws a polygon or a line around/across the area of interest, he/she is interested in the features that are located *on the ground* within this specified area. In other words, we do not have to consider any heights introduced within the feature itself (above the ground). For instance, if we are querying a building layer, we are interested in the building features located within this ground area on the image. As shown in Figure 3.45, the processing steps may be summarized as follows:

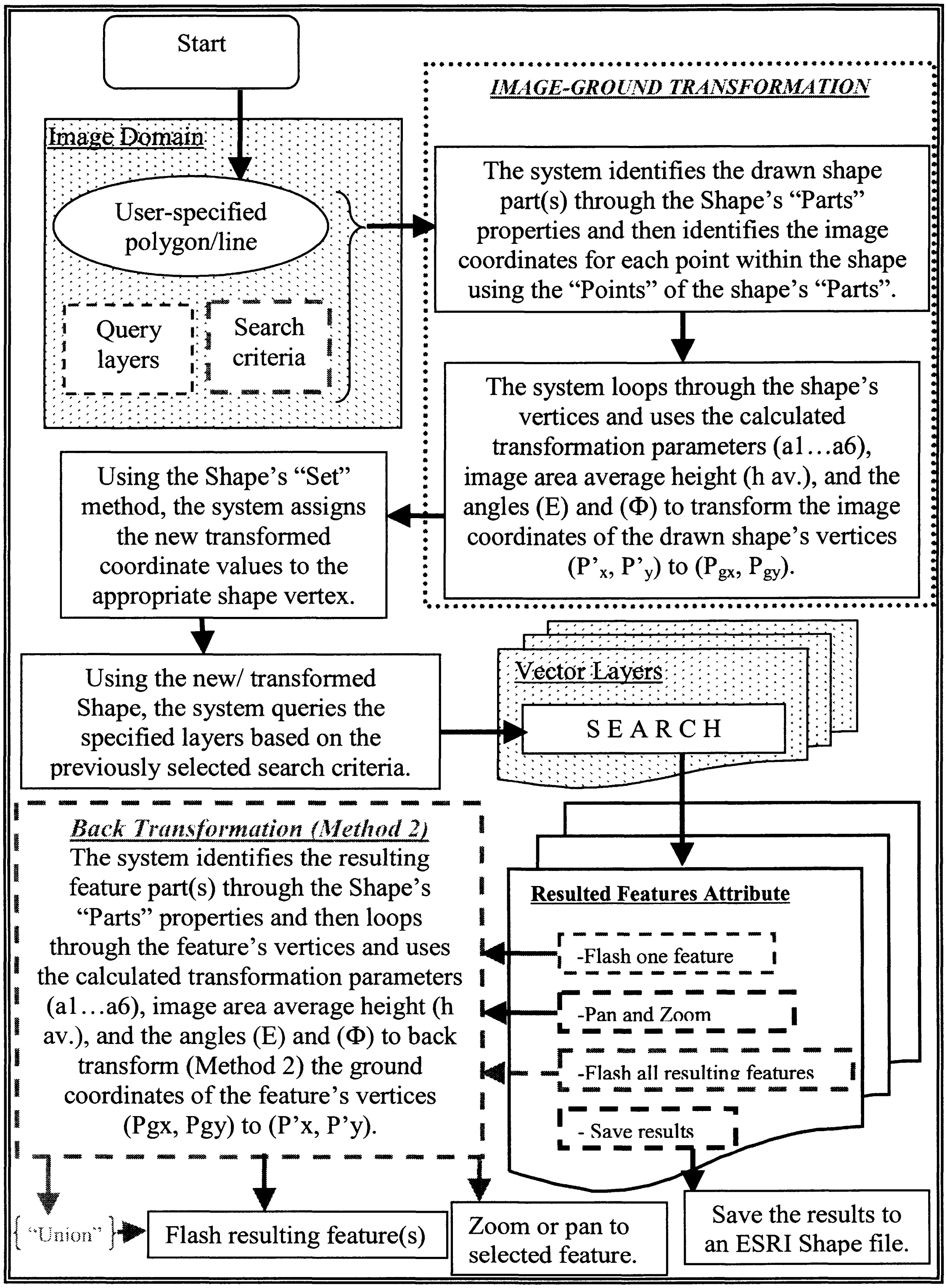


Figure 3.45
Case 3 "Polygon/Line Based Image Query Function" processing steps.

1- The user identifies the layers to be queried as well as the search criterion and expression and draws the polygon or line to query within/along.
2- The system identifies the user-defined drawn shape and converts it from screen to image coordinates.
3- Since the image coordinates of the vertices of this shape are distorted due to the relief and the sensor errors, the system has to transform these vertices from the distorted image coordinates to the ground coordinates. A loop through the vertices of the user-defined drawn shape is performed applying the transformation process for each vertex/point, as explained in the Pointing Query Function. This process brings the user-defined drawn shape to approximately its actual ground position.
4- Using the transformed shape, the process steps involved in querying the previously identified layers remain the same as explained in section 3.2.3.

3.4.6.4.2 Modification II: Resulted features back transformation

Modification II mainly takes place at the "Results Display" stage within the polygon/line-based image query function. The process of identifying the results and displaying them is the same as explained in section 3.2.3.2. When displaying, zooming to, and panning to a certain resulting feature, a back transformation had to be applied. Since the coordinates of this resulting feature are on the domain of the vector layer, these coordinates represent the "correct" coordinates.

In order to perform such functions we have to transform the feature's coordinates back to the distorted image domain. The back transformation within this function will not be performed as explained previously when displaying resulting features in the Pointing Query Function because the image coordinates of the feature are not known and, consequently, the X and Y offsets can not be calculated. Considering the normal transformation procedure used previously, the ground location (Pgx, Pgy) was calculated based upon the affine transformation parameters, the elevation angle (E), the azimuth angle (Φ), and the calculated elevation of the point. In the backward transformation, (Method 2), the image coordinates of each of the resulting ground feature's vertices (P'x, P'y) is being calculated based upon its ground coordinates (Pgx, Pgy), the transformation parameters, the elevation angle (E), the azimuth angle (Φ), and the point's calculated elevation (average elevation in our case). For example, when having the angle (Φ)>90 degrees, the following two formulas may be used to perform such tasks:

$$P'_y = \frac{[a5 * P_{gx} - a5 * Rd * \text{Sin}(\Phi) - a5 * a1 - a2 * P_{gy} + Rd * \text{Cos}(\Phi) * a2 + a2 * a4]}{a5 * a3 - a6 * a2}$$

$$P'_x = \frac{P_{gx} - Rd * \text{Sin}(\Phi) - a1 - a3 * P'_y}{a2}$$

Appendix D contains the developed formulas that are used to directly perform the backward transformation. It is important to note that to allow the system to loop and recognize the shape vertices, the shape "Parts" property has to be declared as a Map Objects 2.0a "Parts object" that holds the set of points forming the shape [ESRI, 2000 b]. Then, each point of the parts point collection is being recognized and transformed. It is important to note that when applying the average shift method, i.e. the second method of sensor error elimination, different sets of formulas should be developed, in order to perform the back transformation, based on the scenario used in this method.

After transforming all the vertices of the feature, a new object is created that contains the transformed feature and this is then flashed to the user. When tested using the *DTM interpolated elevations*, this back transformation displayed the features approximately in their correct positions on the image.

3.4.6.5 Data Base Query Function

In Case 3, this function was performed in the same manner as explained in sections 3.3.5.1 and 3.2.4 until the results are displayed. When the "Display Results" form appears with the resulting query features, the process of manipulating the results remains exactly the same as explained in the previous function (see section 3.4.6.4.2). Because the "Results Display" forms will perform the same functions, the same processing steps should be followed and, consequently, no modification was needed for displaying the results in the Data Base Query Function.

3.4.6.6 Buffering Query Function

This buffering function starts with identifying the core buffering feature(s). The process of selecting this feature was modified in order to accommodate the requirements of this case (Case 3). The rest of the query process remains the same as explained previously in sections 3.3.5.1 and 3.2.5. The functionality and the processing steps to perform those functionality in manipulating the query results within the "Results

Display" stage are the same as explained for the previous two functions, see section 3.4.6.4.2 and section 3.4.6.5.

When the user selects an existing image feature as the core buffering feature, the system follows the same process as in the Pointing Query function until the desired feature is selected as explained in section 3.4.6.3. The user can select to draw a certain feature (point, line, circle, polygon). He/she draws the desired shape, and the system converts this shape the same way as in the user-defined drawn shape explained in the Polygon/Line Based Image Query except when the drawn shape is a point or a circle. If the selected shape is a point, the point's image coordinates are transformed directly as explained previously in the Pointing Query function , section 3.4.6.3 steps 2 through 5.

If a circle is selected to be drawn and used, the system declares a point object, assigns the circle center coordinates to the previously declared point object and then transforms the created point object into the actual ground coordinates (P_{gx}, P_{gy}), as explained in section 3.4.6.3 steps 2 through 5. Then the whole circle itself is shifted, using the "Offset" method, so that the transformed circle is centered at the new ground-transformed object point.

When a group of features is to be selected as the core buffering feature, the system applies the "Union" method to create a new feature representing the projected and coordinate system transformed features.

The output from any of the previous three selections is an object representing the ground location of the desired core buffering feature(s). Then this object will be used, along with the user-specified buffering distance, to create the buffering zone/polygon, as explained in section 3.2.5. This buffering zone or polygon first has to be displayed to the user. In order to perform this task, the system has to transform all of the vertices of that polygon from the correct/ground coordinates to the distorted image coordinates. This can be done by creating an empty object, assigning the buffering zone/polygon to that object, looping through the created object vertices, performing a back transformation on each vertex, and creating the new back transformed polygon, as explained in section 3.4.6.4.2. Then the new transformed object will be displayed using the flash method.

Using the originally/non-transformed/ground buffering zone/polygon, the system then performs the query process against the selected layers as explained in section 3.2.5. Following that, the "Results Display" forms appear carrying the resulting features and the process associated with this stage starts. No process modification has been applied to that stage. In other words, the same processing steps as explained in the previous two functions, Polygon/Line Based Image Query and Data Base Query, will be followed in order to manipulate the resulting features. Except for the mentioned modifications, the rest of the processing steps involved in this function remain the same as explained previously in section 3.2.5. Figure 3.46 shows a schematic diagram representing the main processing steps associated with this function.

3.5 IISP CASE 4: UNRECTIFIED IMAGERY WITH DIFFERENT PARAMETERS VECTOR DATA

For Case 4, it is assumed that we have unrectified imagery (tilted/near-vertical scanned aerial photo or captured video frame) covering a particular area of interest, and vector data layers covering more or less the same area, but having different projections, coordinate systems, and scales. In addition, image-identifiable full ground control points are available. In Case 4, the methodology, major problems, solutions, and main design considerations are presented.

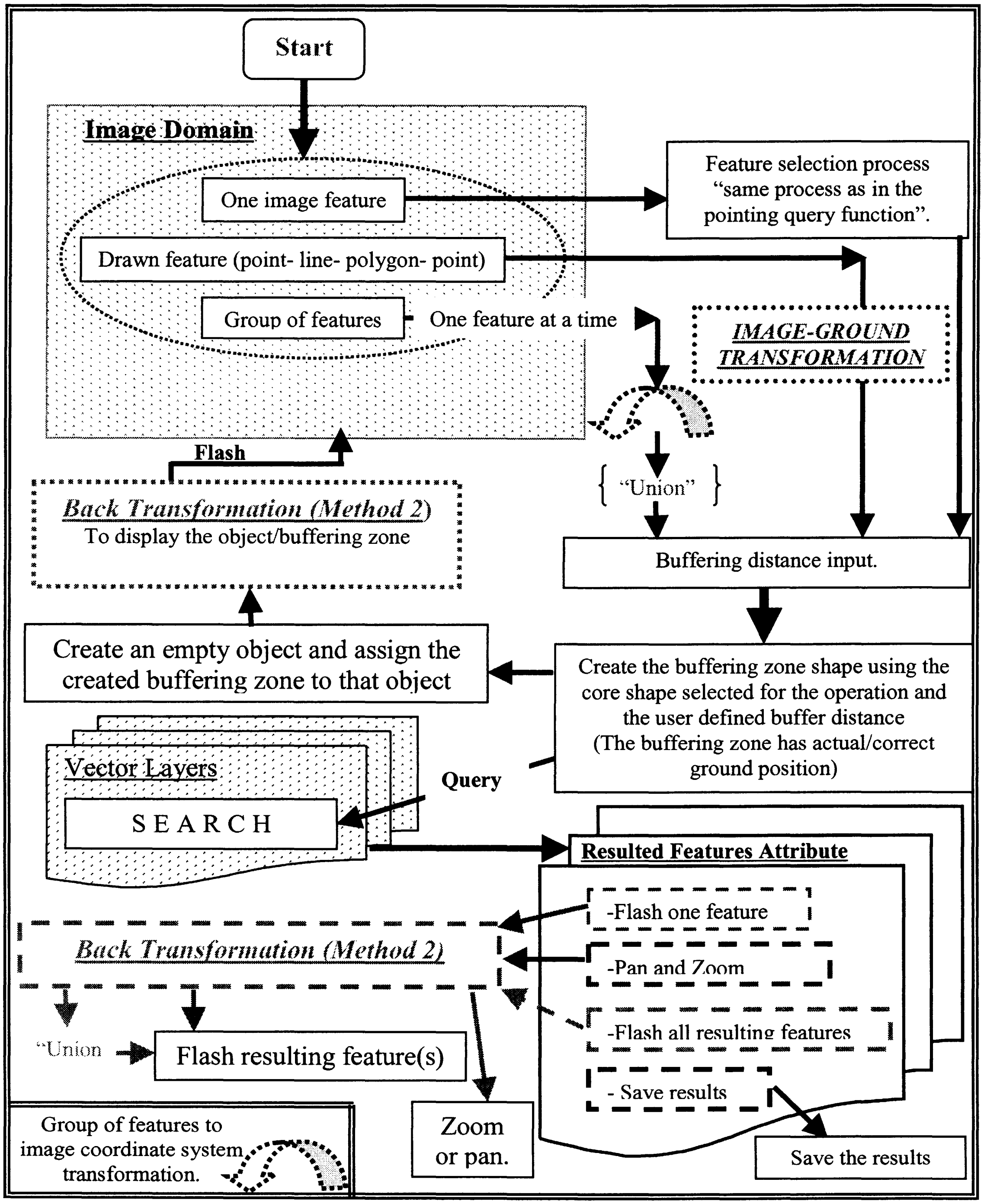


Figure 3.46
Case 3 "Buffering Query Function" processing steps.

There are several design issues that should be taken into account, including:

1- The image is not formally pre-registered to a ground coordinate system, but rather it is assumed that several ground control points are available and visible within the image and they can be used to establish the image-ground relationship within the process.

2- The system should establish the image coordinate system using user-specified parameters.

3- Since the system reads the coordinates based upon the screen coordinate system and the image does not have a georeferencing file, the system should provide a means to convert from the screen coordinate system to the image coordinate system.

4- The image contains systematic errors that affect the image point coordinates;

5- In addition to the errors due to lens distortion, film distortion, and atmospheric effects, the image has tilt distortion and relief displacement that affect the position of features on the image.

6- Rather than proceeding with an image correction procedure, image points and shapes should be corrected and related to the ground, and the ground features should be rectified and presented within the distorted image, when required.

7- A feature, such as a building, has its own elevation above the ground that introduces an additional displacement to the position of the feature on the image.

8- Are there any differences in the solution to this case when using the two projection methods introduced in section 3.3?

Assuming that the first projection method (direct on-the-fly whole layer projection) is used, the following sub-sections present the methodology and processing steps involved to solve the problems that may be introduced within the IISP functions.

3.5.1 Image Loading and Vector Layers' Identification

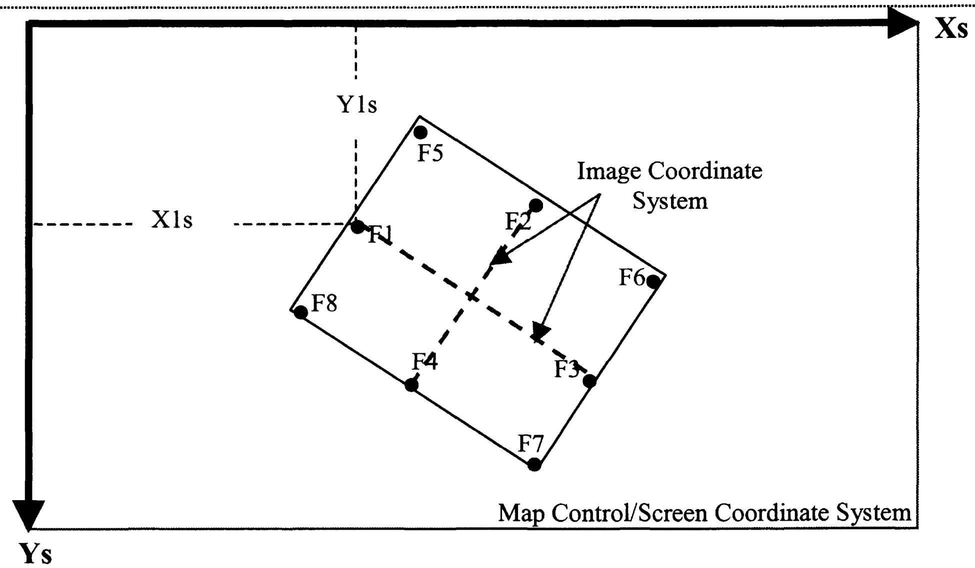
The image loading stage in this case is concerned with selecting an image file, loading it, relating it to a ground coordinate system, and preparing the image for subsequent query processes.

After loading the image, the system has to identify the camera calibration parameters by directly reading the camera calibration report file associated with the image. The camera calibration parameters, as concerned in this case, are fiducial mark coordinates, focal length, and principle point coordinates (if needed). Because the system recognizes the screen coordinate system as its main coordinate system, the system has to establish the image coordinate system and to obtain the transformation parameters to convert from screen to the image coordinate system. This can be accomplished as follows:

1- *Establishment of Image coordinate system*

   a- The system has to recognize the fiducial marks by asking the user to input the fiducial marks, sides or sides and corners, and the screen coordinate system by clicking on the fiducial marks on the image. The user should enter the four side fiducial marks first and then the other four corner fiducial marks (if available).

   b- The system then arranges the fiducial marks by using the screen coordinates and calibration report coordinates using the same procedure as shown in Figure 3.47. This is an important step to avoid any possible mismatch to the fiducial screen and calibration report/image coordinates.



<u>Arrange F1 – F4 (side fiducials)</u>

1- *Among the four points, take the mark with min. screen x coordinate and the mark with max. screen x coordinate and assign them to F1 and F3 respectively.*
2- *For the other two, take the mark with min. screen y coordinate and the mark with max. screen y coordinate and assign them F2 and F4 respectively.*

<u>Arrange F5 – F8 (corner fiducials)</u>

1- *Among the four points, take the two marks with min. screen x coordinates and consider them F5 and F8. Then among these two marks, take the mark with min. screen y coordinate, assign it to F5, and the other will be F8.*
2- *Take the other two marks with max. screen x coordinates and consider them F6 and F7. Among these two marks, take the one with min. screen y coordinate, assign it to F6, and the other one will be F7.*

AFFINE TRANSFORMATION EQUATIONS

$$Xi = a1 + a2*Xs + a3*Ys$$

$$Yi = b1 + b2*Xs + b3*Ys$$

Where: *Xi, Yi = Image coordinates - Xs, Ys = Screen coordinates*
a1, a2, a3, b1, b2, b3 = Transformation parameters

Figure 3.47
Fiducial marks arrangement and "Affine Transformation".

c- The screen coordinate system, Map Control coordinate system, is working as a monocomparator coordinate system. In other words, using the fiducial marks image and screen coordinates, the system can apply an affine transformation to produce the required transformation parameters from screen to image coordinates.

d- By performing the previously mentioned transformation, the system internally establishes the image coordinate system and can convert any point from the screen to image coordinates or vice versa.

2- *Systematic error corrections*

a- Because of the conditions introduced within the camera and the atmosphere during exposure, we may have several systematic errors within the image that represent errors in the measured/calculated image coordinates. Those systematic errors are well studied and modeled, for example, see Abdelrahim [1994].

b- A systematic error component can be added/subtracted from the image coordinates prior to subsequent usage.

3- *Image-ground relationship*

a- As an initial step towards relating the image to the ground, the user needs to identify the available control points within the image as well as their 3-D ground coordinates. At least 3 full control points must be provided for this stage.

b- In order to identify the geographic/projected coordinate system of the ground control points, a similar process, forms, and controls as used in identifying the image coordinate system in Case 2, section 3.3.4, can be used.

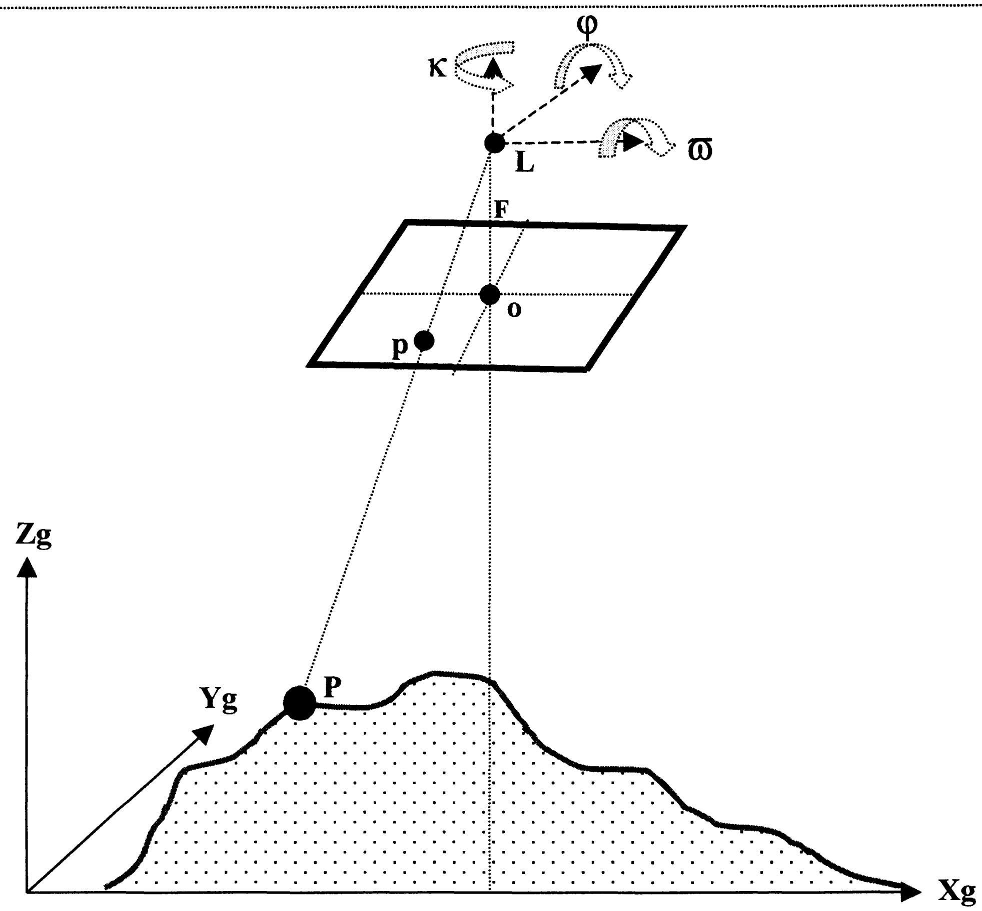
c- The direct relationship between the image points and the corresponding points on the ground can be obtained by applying the well known photogrammetric collinearity condition, as shown in Figure 3.48 [Barnes and Vonderohe, 1985].

d- Using the previously identified control points we can uniquely (for 3 points) or by least-squares adjustment obtain the exterior orientation parameters of the image, through the "Space Resection", which results in the image's three rotation angles and the exposure station's coordinates.

e- As soon as the system obtains these parameters along with the screen to image transformation parameters, the corrected ground coordinates, with respect to tilt and relief displacement, of any screen/image domain point can be obtained *if its elevation is known.*

With respect to the vector layers' identification stage, the same process as explained previously in section 3.3.5.1 should be followed. In that process, we should note that the image coordinate system is now represented by the coordinate system of the ground control points. In addition to that, the system should inquire whether the layer contains features that have their own elevations above the ground (building features, for example). The information provided within this additional step will be used within the pointing query function, as explained later in section 3.5.2.

After performing the previously mentioned steps, the system is now ready to carry on any of the subsequent query functions. Each of the subsequent query functions has its own processing requirements. The following subsections represent the requirements as well as the process associated with the subsequent functions.



$$Xpg = XL + \frac{(Zpg - ZL)[m11(xp - xo) + m21(yp - yo) - m31F]}{m13(xp - xo) + m23(yp - yo) - m33F}$$

$$Ypg = YL + \frac{(Zpg - ZL)[m12(xp - xo) + m22(yp - yo) - m32F]}{m13(xp - xo) + m23(yp - yo) - m33F}$$

Where:

Xpg, Ypg, Zpg = Ground Coordinates of Point (P)
xp,yp = Image Coordinates of Point (P)
xo, yo = Principal Point Coordinates
XL, YL, ZL = Camera Exposure Station Coordinates in the Ground Coordinate System
F = Calibrated Focal Length
m's = Elements of a three-dimensional Rotation Matrix as a Function of Camera's Rotation Angles (ϖ, φ, κ) Around X, Y, Z Axes

Figure 3.48
Collinearity condition equations.

3.5.2 Pointing Query Function

The processing steps involved within this function are considered to be similar to those in Case 2 with particular modifications to accommodate Case 4 requirements and assumptions. The processing steps of this function may be summarized as follows:

1- The user identifies the layer to be queried and clicks on an image position (P) that represents a feature on the image.

2- The system then reads the screen coordinates of that position, converts them to image coordinates using the previously defined affine transformation parameters, and then applies a correction for the systematic errors to these coordinates.

3- If the query layer contains features with no elevation above the terrain (parcel layer, soil sheets,…,etc.), we can directly apply the collinearity condition in an *iterative process* to obtain the correct ground location of point (P). The process will be as follows [Barnes and Vonderohe,1985]:

   a- In order to use the collinearity condition for that purpose, the system has to know the elevation of the point (P) above the datum.

   b- We can get an initial estimation of that elevation by getting the average terrain heights within this area from the previously identified ground control points, control points file associated with this area, or from a DTM, if available.

   c- By applying this elevation value within the collinearity equations, we get an initial estimation of the ground position of point (P) knowing its image coordinates.

d- Using this position, the system can extract a better estimate for the elevation by using an inversely weighted distance interpolation from each control point, or using the DTM, based on the latest estimate of the point (P) ground position.

e- Using this new elevation value, we can extract a better ground position for point (P) using the collinearity condition equations.

f- The process is repeated until the change in the elevation value reaches a pre-defined threshold.

4- In case of the query layer containing features that have elevations above the ground (building layer, for example), we may use a modified version of the c*ollinearity iterative process*. When users click on such a feature, it is more likely that they will pick a point at the top or near the top of the feature. This will not be a problem when there are short features with a wide bottom area (townhouse, for example). The problem arises if the feature is tall and has a small bottom area (tower building, for example). Due to the relief displacement introduced within the feature because of the additional elevation of the feature above the ground, any image point except at the bottom of the feature will not produce the correct ground position of the feature when using the collinearity condition. In other words, we need to obtain the image position that corresponds to the bottom of the user-selected feature on the ground. The following process for vertical photography could be followed to approximately obtain an image point that represents a point on the bottom of the desired feature:

a- In order to perform such a correction, we need to know an approximate elevation of the feature. This can be done by :

  i- allowing the user to enter an approximate value of that elevation;

  ii- enter an average value to the building heights within this area (can be obtained from external sources);

  iii- allow the user to select from several categories, as follows:

	Approximate Height Value
> 10 stories building	50 m
5-10 stories building	30 m
<5 stories building	20 m
Townhouse building	10 m

b- Using this height and assuming that the user clicks near or on top of the building, we can get closer to the actual ground position of the feature. This can be accomplished by obtaining the position of the feature's bottom on the image as a function of the feature height, the elevation of the feature bottom above the datum (terrain elevation), and the image coordinates of the point (P), where the user originally clicked on the image, as shown in Figure 3.49. The collinearity condition can be utilized in an *iterative* process substituting the originally selected image point position with the new corrected one.

c- It is important to note that:

  i- The obtained image point may not exactly fall within the feature bottom but this error will be compensated for when the user selects a search circle to search the layer.

ii- The image is tilted (not vertical), as assumed for this case, which practically means that we have tilts of between one and three degrees. In such cases, relief displacements on a tilted photograph may be calculated based upon vertical photo calculations with sufficient accuracy, as shown in Figure 3.49, [Wolf and Dewitt, 2000].

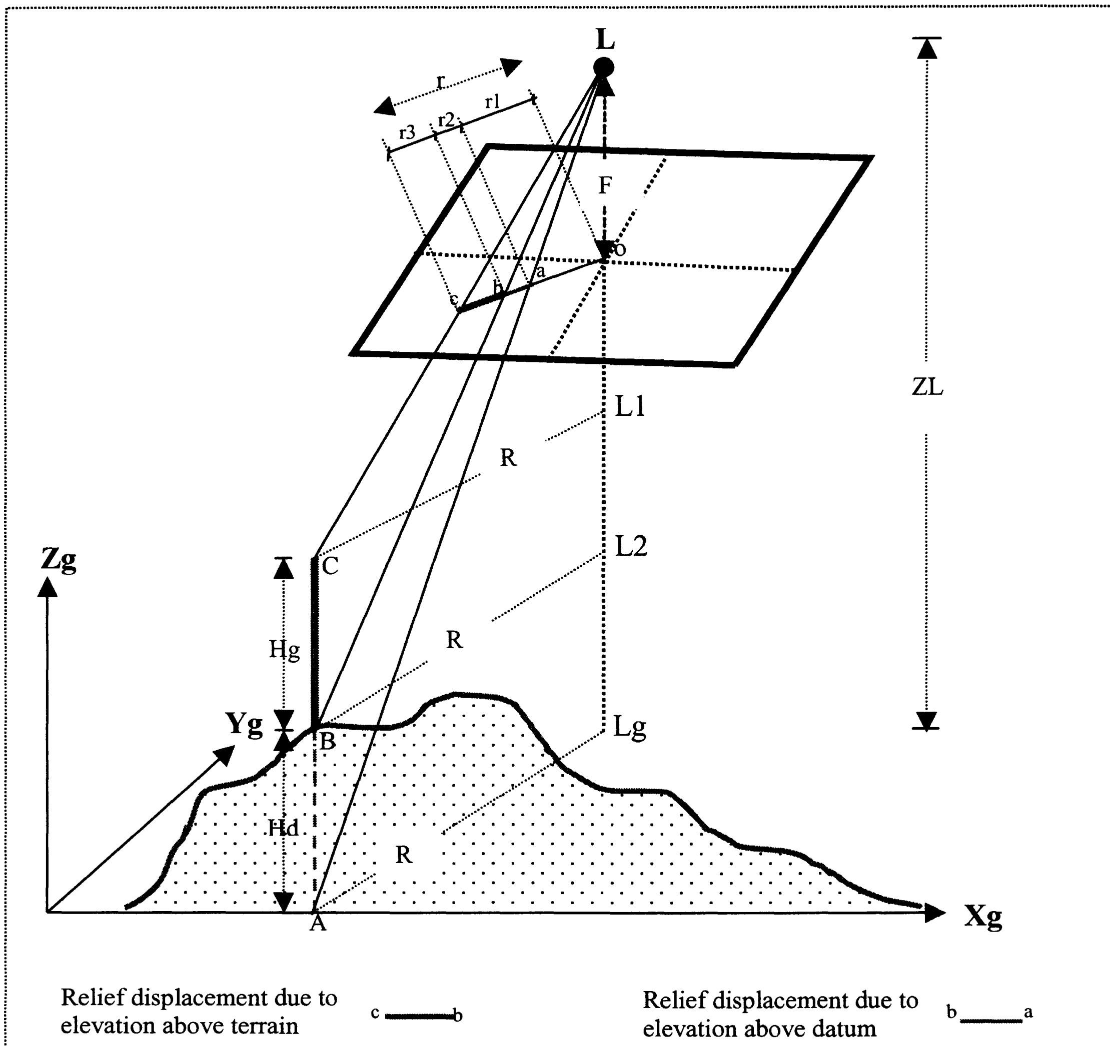


Figure 3.49
Relief displacement due to feature's height above ground

Δ (L-C-L1) and Δ (L-c-o)

$$\frac{R}{r} = \frac{(ZL - Hd - Hg)}{F} \longrightarrow \boxed{R * F = r * (ZL - Hd - Hg)}$$

Δ (L-B-L2) and Δ (L-b-o)

$$\frac{R}{r - r3} = \frac{(ZL - Hd)}{F} \longrightarrow \boxed{R * F = (r - r3) * (ZL - Hd)}$$

since Xc/Xb = Yc/Yb = r/(r-r3),

where: Xc, Yc = Image coordinates of the actual point clicked by the user (hypothetically feature's top or near the top point);

Xb, Yb = Image coordinates of the desired feature bottom point;

$$Xb = Xc * [\ \frac{ZL - Hd - Hg}{ZL - Hd}\ ] \qquad Yb = Yc * [\ \frac{ZL - Hd - Hg}{ZL - Hd}\ ]$$

In collinearity equations, we can multiply the image coordinates of the actual point (Xc,Yc) by the following factor

$$\frac{ZL - Hd - Hg}{ZL - Hd}$$

Figure 3.49
Relief displacement due to feature's height above ground (cont.)

This page was left unintentionally blank by the author.

d- The same processing steps 2 through 6 within the original *collinearity iterative process*, as explained previously, may then take place.

5- After obtaining the ground position of the desired feature, a link to the layer and a search through the layer is conducted, as undertaken within the same function in the previous two cases.

To flash a certain feature, the feature's vertices on the ground are related to the image coordinates one by one. This is accomplished using the collinearity conditions. The system has to interpolate the elevation value of each feature point as explained earlier and then use the collinearity equations to obtain the image coordinates of that point knowing its ground coordinates. If a systematic error model is considered, its components should be added to the obtained image coordinates with opposite value in order to obtain the distorted image coordinates of the desired feature/shape and consequently to represent the correct shape location based upon the existing image coordinate system (fiducial system).

3.5.3 Polygon/Line Based Image Query

Within this function, as mentioned in Case 3 (section 3.4.6.4.1), it is of great importance to recognize the intent of the user . The user-specified drawn shape is meant to query features that exist on the ground within this drawn shape. In other words, any heights issue of the features (if any) should not be of concern within this function. The processing steps may be summarized as follows:

1- The user identifies the layers to be queried as well as the search criterion and expression and draws the polygon or line to query within.

2- The system identifies the user-defined drawn shape and converts it from screen to image coordinates.

3- Using the previously mentioned *collinearity iterative process*, we can relate the user-defined drawn feature to the ground by relating its vertices one by one.

4- A search through the selected query layers can then be conducted, and the results can be displayed within the "Display Results" form(s).

5- Flashing a feature will take the same processing steps as followed in flashing a feature within the pointing query function.

6- Zooming or panning of a certain feature will be accomplished by identifying the desired feature extent, relating that extent to the image and screen coordinates, and then the zooming and panning functions can be performed using the identified feature's screen extent as arguments.

7- Saving the results will be accomplished directly with no need to perform any kind of transformation, and the results will be a shape file having the ground control points coordinate system.

8- In order to flash a group of features at once, each feature should be converted from the ground coordinate system to the image coordinate system using collinearity conditions as explained earlier in section 3.5.2, and to the screen coordinate system using the affine transformation. By applying the union method, we can create a resulting feature that represents the feature group within the screen coordinate system and then we can apply the "Flash" method to flash it.

3.5.4 Data Base Query Function

In this function the user selects the layer to be queried and presses the database query toolbar button. The process then proceeds with the same steps as explained earlier in Case 1, section 3.2.4, until the Results Display form appears. Then, the methodology, techniques, and processing steps presented in section 3.5.3 in terms of handling the resulting features within each Results Display form will be followed.

3.5.5 Buffering Query Function

The process in this buffering query function starts with the identification of a buffering core feature. This is done by identifying a feature on the image, a user drawn shape on the image, or a group of features that exist in a certain shape file. In the first case, the same procedure as presented in the Pointing Query function, section 3.5.2, will be followed until a feature is selected. If the user selects a linear feature as the buffering core feature, the same process of connecting the linear segments as explained in section 3.2.5.1.2 will be followed. In the second case, the user draws a shape on the image and the system converts the shape from screen coordinates to image coordinates. The system then relates that feature to the ground in exactly the same procedure undertaken in relating the user-defined query polygon or line to the ground within the Polygon/Line Based Image Query function. If the user selects a circle to be the core buffering feature, the system then identifies the center of the circle and any random point around on the circle, relates these two points to the ground, and creates the desired circle on the ground. In case of having a group of features to be used as the buffering core feature, no transformation will be needed within this stage, but rather the system can perform the

"Union" method on this group of features creating one single feature representing that group. The outcome from this process is a single feature in the coordinate system of the ground control points representing the buffering core feature.

The next step is the user input to the desired buffering distance. The system then uses the created buffering core feature and the buffering distance to create the buffering zone feature (shape). Using the collinearity condition, the system can transform the points of the buffering zone feature from the ground to the image and then to the screen coordinate system in a similar manner as followed in flashing the selected feature within the Pointing Query Function. The buffering zone can then be flashed within the image pixels. The system then identifies the layers selected to be queried as well as the search expression, if any, connects to these layers, performs an intersection operation, and retrieves the results for each layer. The methodology, techniques, and processing steps presented in section 3.5.3 in terms of handling the resulting features within each Results Display form are followed.

In the case of using the second method of projection (minimal on-the-fly results projection) developed within this research, as explained in Case 2, the same approach and techniques presented within sections 3.5.1 through 3.5.5 are applied with minor modifications. These modifications are due to the fact that the vector layers will not be projected to the ground coordinate system of the control points within the identification process. Rather, the identified vector layers will remain within their own coordinate system domains and only particular features will be projected on-the-fly when required. In other words, we have to project on-the-fly from the ground coordinate system domain to the desired layer domain and vice versa, when required, as demonstrated previously in section 3.3.5.2.

3.5.6 Accuracy Considerations

The proposed process and techniques designed within this case depend on several issues in order to ensure accurate results, namely:

1- Fiducial marks should be identified accurately in order to maintain an accurate establishment to the image coordinate system. A library of standard fiducial mark shapes can be built within the system. The user then identifies the correct shape from the library. When the user clicks on the location of a fiducial mark, image matching components, if available, can be integrated into the process by searching around the clicked point for the same selected pattern within the library. It is then possible that the system can recognize an accurate position of the fiducial marks and automatically detect the others.

2- The accuracy of calculating the camera's exterior orientation parameters will depend on the number of control points identified to the system. The more control points we have the more accurately we may identify those parameters, and the more accurately the system functions can proceed.

3- In many cases, we have a digital control points file that contains available control points' XYZ ground coordinates. We can allow the user to use few control points to relate the image to the ground and then use the whole file in interpolation when required. Furthermore, if a DTM is available of the area under consideration, this will be of tremendous help to increase the accuracy of the elevation interpolation process of ground locations and, consequently, it will increase the accuracy in identifying the corresponding locations on the image.

4- In treating the relief displacement existing due to feature's height (building feature, for example), although the input of a feature's height will not be exact either through a user input or an average height value within the area, it will definitely be very helpful for approximately identifying the feature position within the vector layer. The remaining error can be compensated for through a search by distance (search circle) method and by using a user confirmation to the results obtained.

The techniques presented within this case will be very helpful for quickly interpreting the image especially in applications where, for example, a fast assessing of the damage within a disaster area is required considering an emergency situation or limited technical/practical resources in obtaining an orthophoto. By embedding the previously mentioned rectification procedure, we can utilize the aerial photos on-the-fly, as they are, to explore and assess spatial phenomena in a fast and accurate manner. In addition to that, the same developed procedure, with some modifications, can be used in querying video imagery.

3.5.7 Airborne Video Imagery

Video imagery plays an important role in time-sensitive applications such as disaster management. The fast availability of the imagery, real-time collection procedure and cost-effectiveness considerations give this technique an upper hand over aerial photos for such applications [Meisner, 1986; Marsh, et al. 1991; Mausel, et al. 1992]. Video imagery is collected and stored on videotape. Either a real-time in-flight or normal post-flight procedure can be followed to convert any selected analogue video signal/information into digital frame (photo). This procedure can be accomplished using the so-called "Frame Grabber" [King, 1991; Maggio and Baker, 1988]. A near real-time transmission of the desired frames to a ground location can be accomplished by selecting the desired image(s), compressing them, and transmitting them through a mobile phone and a modem as applied in the Multi CCD Camera System developed by Savopol, et al. [2000]. Furthermore, a GPS exposure position is recorded with any desired selected frame. This leads to a near real-time final product that is competing with traditional aerial photos and that is ready to be used in analyzing spatial phenomena. The IISP unrectified aerial photo case can then be configured and used to directly and on-the-fly link any available GIS vector data to help interpreting the image and consequently better understand and analyze the situation at hand. This means there is no need for extensive pre-processing conversion/transformation or superimposition that may delay the process, confuse the analysis, and/or hide important pattern/information that may be crucial for such applications.

In the case of using the captured video frame, two main considerations should be taken into account in order to reconstruct a situation similar to that of an aerial photo, namely:

1- *Image coordinate system:*

A captured video frame does not have fiducial marks in order to establish the image coordinate system. Fortunately its exact size (pre-defined captured frame size/ digitization sampling frequency) is known based upon the characteristics of the digitizing process, the video camera, and the frame grabber used, for example, see Wright and Um, [n.d.]. Consequently, the system can internally divide the frame into four equal quadrants. By asking, the user clicks on the corner of the captured frame, considered now as the four corner fiducial marks within a typical aerial photo. The system then can distinguish the frame and screen coordinates of the corner points and can establish the required transformation parameters between the screen coordinate system and the frame coordinate system.

2- *Camera interior and exterior orientation parameters :*

The camera parameters such as focal length, lens distortion, etc. are usually known from a pre-defined calibration procedure. Those parameters can be used in a similar way as in the aerial photo case in photogrammetric calculations or in modeling systematic error effects. The camera (exposure station) position as well as the exact roll, pitch, and heading of the aircraft can be measured to a high accuracy using a differential Global Positioning System and Inertial Reference Unit (IRU) [Terra Surveys, 2000]. By matching the exact time stamp of the frame with this information, the exact exterior orientation of the desired frame can be obtained. This on-the-fly exterior orientation development procedure can eliminate the step of identifying ground control points within the image. In other words, the exterior orientation parameters obtained along with the image/frame coordinates for any desired point can be used directly within the collinearity condition. This condition uses the same procedures and techniques explained earlier, within the aerial photo case, to obtain the correct/desired ground position and consequently retrieve features' information from vector databases. The video mapping system usually includes GPS components to obtain the recording position at any time, for example, see Cooper, et al. [1995]; King, [1995]; Slaymaker, et al. [n.d.]; Airborne Videography Ltd., [n.d.]; and Terra Surveys, [2000], but the Inertial Reference Unit is not necessarily included. In this case, the same procedure in identifying at least 3 full ground control points on the image followed by the collinearity condition can be used to obtain the captured frame exterior orientation.

Assuming the center pixel of the frame as the origin (principal point) and knowing the exterior orientation, a similar situation to the aerial photo case can be simulated and the same query process can be followed. In order to practically verify the accuracy and the performance of the developed techniques and procedures either within a scanned aerial photo and/or a captured airborne video frame, a future testing and implementation for this case and associated techniques needs to be considered.

3.6 IISP: MOVING THE FUNCTIONALITY TO THE INTERNET

One of the requirements of the developed system was to allow different end users to access it and use it simultaneously. Instead of creating a stand-alone application and having the users install the application individually within their workstations, a better distribution approach would be to migrate the developed application to the Internet. If the application is an Internet-enabled application, any user can access it at any time and use the most updated version of the application without the need to distribute/install software.

Nowadays, the Internet is the appropriate environment for exchanging information and distributing application software. A variety of tools exist to create an Internet-enabled application. For the project at hand the application had already been created and tested using the Visual Basic Standard EXE environment. Thus, the main concern, at that stage of the research, was to find the most suitable and appropriate techniques/tools to convert the developed Visual Basic stand-alone application into an Internet-enabled application with minimum modifications.

The migration process was accomplished using the Visual Basic ActiveX document technology. This technology allowed the author to create the desired Internet-enabled application while keeping the functionality and the components of the original system unmodified. The main objectives of this section are to:

1- Present the technology used to migrate/convert the intelligent imagery system prototype from a stand-alone application to an Internet-enabled application.
2- Demonstrate the implementation procedure taken towards the migration process.
3- Discuss the problems faced and actions taken towards the migration process.
4- Lay out the advantages and disadvantages in using the ActiveX document technology in migrating the IISP.

3.6.1 ActiveX Document Technology

Visual Basic 6.0 provides three basic tools in order to create a Web-based application, namely, IIS (Internet Information Server) applications, DHTML (Dynamic HypreText

Markup Language) applications, and downloadable ActiveX documents [Eidahl, 1997]. In this research, the third type of tool was used to migrate the developed system to the Internet. The main two reasons for using ActiveX document technology are that, compared with the other tools, only minor programming efforts were needed to convert the already developed application to an Internet-enabled one, and no modification to the already created application was required.

An ActiveX Document can be defined as "a document from a stand-alone application that can be viewed, edited, and controlled through an ActiveX compliant browser" [DPR Consulting Ltd., n.d.]. Ravindran, [1998] defined the ActiveX Document as "...a Visual Basic application for the Internet. It can be used to implement Internet Downloadable user interface screens which can possess all the characteristics of a standard windows GUI". Eidahl, [1997] suggested that an ActiveX document is "..an application that runs inside a container, such as Internet Explorer, instead of running a stand alone program". From this research project's perspective, ActiveX Document technology can be defined as a carrier of a stand-alone application to the Internet. In other words, it may be defined as a "Visual Basic Internet-Migration Engine".

ActiveX Document projects consist of two main parts. The first part is the document itself which can be described as a special Visual Basic standard form, with some event handling exceptions. This document is called a "User Document Object". An ActiveX Document project can contain an unlimited number of such documents. In any application, the document file that contains the actual data, a word document for example, needs an application server to open, view, and edit this document file, Winword.exe, for example. The same applies for the Visual Basic ActiveX documents.

The document itself needs a server that can handle the events and the data within this document. The second element created in a ***compiled*** ActiveX Document project is the application server, EXE or DLL server. Visual Basic creates any of these two servers and, consequently, the server provides the objects/components necessary for the created documents to be viewed and activated within the ActiveX container application, Internet Explorer in our case [Appleman, n.d.; MSDN, n.d.]

The type of server to be created is based on the type of ActiveX Document project created. For example, if the project selected to be created is an ActiveX Document EXE project, then an EXE server will be created when the project is compiled. In this research project an ActiveX Document EXE project was selected because of the following two reasons:

1- The application can be run as a stand-alone application if required.
2- The ActiveX Document project created is not the actual application, but rather it works just as a holder to the previously created standard EXE application (IISP).

An ActiveX Document EXE project was created and the previously developed IISP application's forms and modules were attached to it using certain steps, as will be explained in the following sections.

3.6.2 IISP Migration Process

In order to migrate the developed standard EXE application to an Internet-enabled one, using the ActiveX Document technology, several steps had to be taken, namely:

1- Standard EXE application development and debugging.
2- Internet launching methodology.

3- ActiveX Document project creation and design.

4- Testing and compiling.

5- Packaging.

6- Launching.

The following sub-sections summarize the previously mentioned steps.

3.6.2.1 Standard EXE Application Development and Debugging

This is an obvious step. In order to migrate a standard application to an Internet-enabled one, you have to have the standard application already developed and debugged. The standard IISP application had been developed and tested using the tools and processes explained previously.

3.6.2.2 Internet Launching Methodology

One popular approach to creating an Internet-enabled application using Visual Basic ActiveX Document technology is to convert all the forms contained within the standard application to ActiveX documents or to initially select all the application forms to be of that type. Then, these documents can call each other based on a pre-designed system workflow. This approach has two major disadvantages if used for this research case. First, several code statements have to be written carefully to achieve the correct interaction between the created documents to exactly map the previously designed standard system workflow. Second, several errors may occur due to code statements, methods, and events written within the standard forms that are not supported/recognized by the created ActiveX Documents, such as load, unload,…,etc. Due to this problem, a change should take place to recover the unsupported methods and events, and consequently that may affect the designed system workflow.

To overcome both problems, a different approach was implemented to reach the Internet-enabled application goal.

This approach is based on the idea of creating only one user document object within the ActiveX document EXE project and attaching the previously developed IISP standard application forms and modules to the project as they are. The user document works as an initial interface (Menu) through which the user can select which case to use and which module to execute [Ravindran, 1998]. This approach was suitable for this research project because no code modification was necessary for the original standard application, and very few lines of code were written and attached to the user document to control and direct the user to the appropriate case based on his/her selection. At the end we reached the desired Internet-enabled application.

3.6.2.3 ActiveX Document EXE Project Creation and Design

The first actual step in the migration process was to create an empty ActiveX document project. This project contains, by default, one document/form called a "User Document object". An ActiveX document project may have one or more user document objects depending on the application requirements. For the IISP application and based on the migration approach discussed before, only one user document was required. This document is intended to serve as the main interface to the system that allows the user to interact and execute the IISP application. The user document works the same way as a Visual Basic standard application form, i.e. you have to add the desired controls and components and manipulate them using a certain code to perform the required task.

Since the developed system has three implemented cases, each of which represents one of the previously mentioned research cases, the main purpose of the user document was to hold the application files attached to it during the launching process, to direct the user to any of the three application cases, and to execute this application. As shown in Figure 3.50, the user document contains three command buttons, labeled "CASE 1", "CASE 2", and "CASE 3". A certain code was written, as shown in IISP_CODE.PDF file, and attached to these command buttons to allow the user to click on them and start the appropriate module of the IISP. For example, if the users clicks on the "CASE 1" button, the show event will be triggered and another form, Login Form, appears. If the user has the appropriate authentication, the main form of CASE 1 appears and the user starts to use CASE 1 of the IISP.

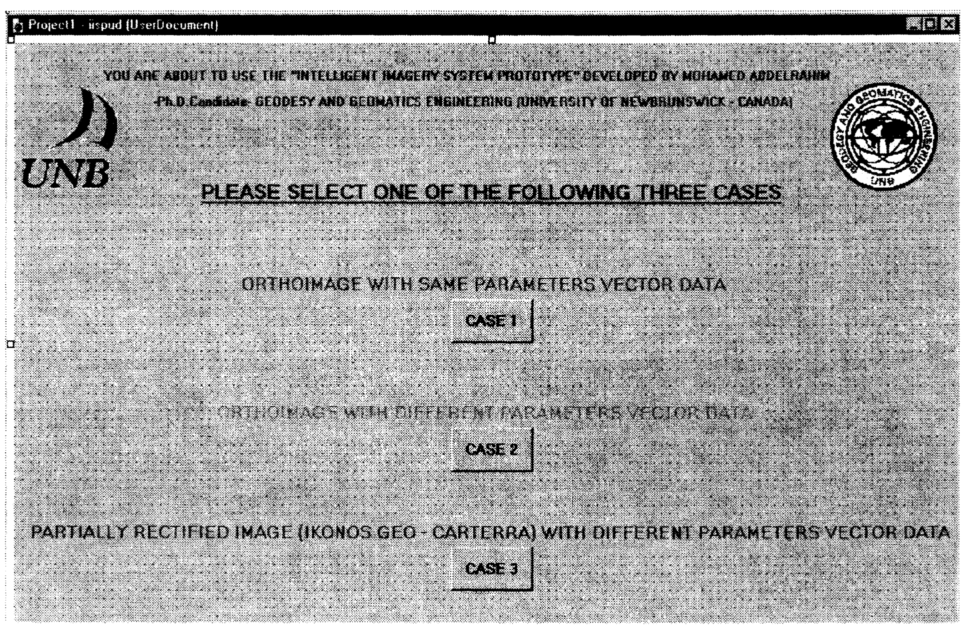


Figure 3.50
IISP user document (design time).

3.6.2.4 Testing and Compiling

After creating the ActiveX document project, attaching all the desired forms to it, and writing the appropriate code to handle the user click event on any of the command buttons within the user document, the project was tested using the Visual Basic environment. At this stage, the system was not tested against delivering the correct result, but rather against code/syntax error as well as correctness of the system workflow.

As mentioned earlier, the ActiveX document EXE project should be tested within a container (another application) that can handle ActiveX components. In the testing process, Visual Basic simulates the real situation and launches the Internet Explorer, the container, allowing the developer to test his/her application in an exact environment as it will be when used on-line. After the code testing process, the system was ready to be compiled.

The main purpose of compilation is to create the files necessary for application distribution. The compilation process involves the creation of the Visual Basic Document file (VBD), an Internet downloadable ActiveX document that allows the user to open the ActiveX document within the browser, and the application executable file (EXE), the required server, as shown in Figure 3.51. After the compilation process, the system was ready for packaging.

3.6.2.5 Application Packaging

At this stage the Visual Basic Packaging and Deployment Wizard was used to package the ActiveX document project for Internet component download. When packaging the ActiveX document project for Internet component download, as is the case in this project, the Wizard creates three *main* types of files that need to be loaded on the Web server to allow the IISP ActiveX document to be accessed through the Web. Those three files are the "cabinet" CAB file, HTM file, and Support files.

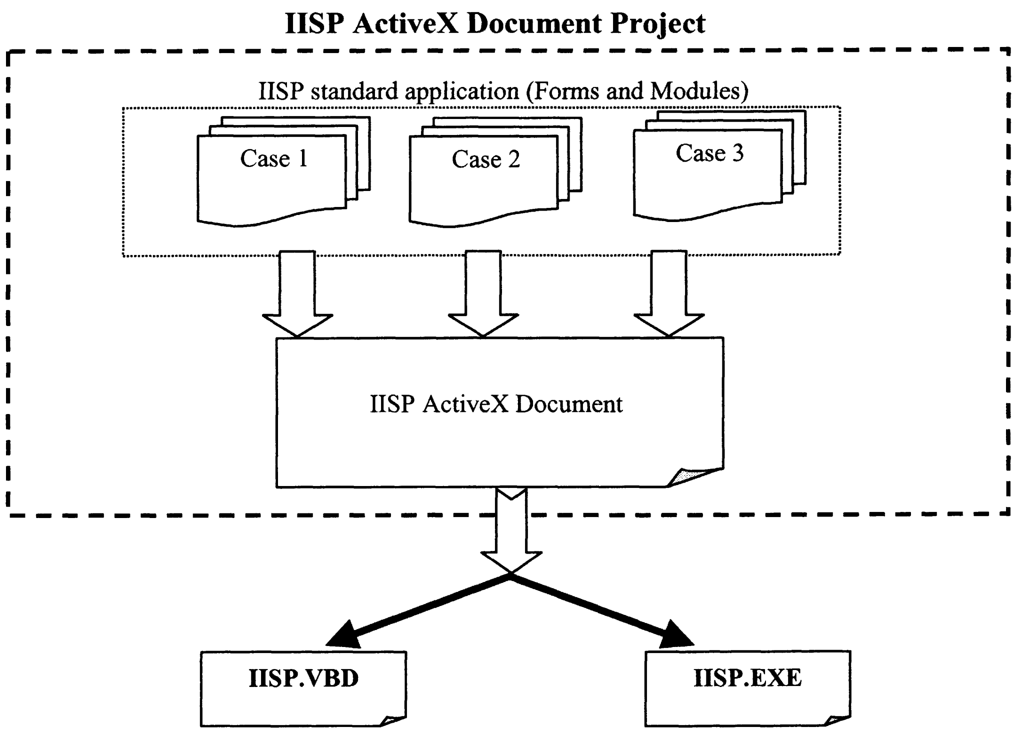


Figure 3.51
IISP compilation process

The CAB file is a Windows setup compressed file that contains all the source information, components, and controls required to run the application [MSDN, n.d.]. The CAB file contains information on how the control should be installed. The HTM file is an HTML file that contains the script necessary to download and display the created user document within the container, Internet Explorer. This file contains a link to the CAB file and is used to launch the download process. The Support files directory created contains information (INF) file and other files that contain the information regarding the CAB files created.

3.6.2.6 Launching

In this step, the IISP Web page has been created and published on the Internet. Along with the files necessary for the IISP page, the application created a VBD file, an HTM file, and CAB file that were uploaded to the server. The web site address is "http://loki.gge.unb.ca/iisp/iisp.html". This page includes a link to the User document and consequently to the IISP application.

3.6.3 Advantages and Disadvantages of ActiveX Document Technology: "IISP Perspective"

The main purpose of incorporating the ActiveX Document technology, offered in Visual Basic, within this research was to allow the developed system to have the flexibility of an Internet-enabled application. An Internet-enabled application has two major advantages. First, the user does not have to install the software in his/her machine but rather the application can be accessed at any time through a Web browser. Second, in the case of application updating, it makes the code maintainable and keeps everyone running the same version. "Creating ActiveX Documents provides a number of advantages over creating Internet applications by other means" [Eidahl, 1997]. From the IISP point of view, three main advantages had been encountered in using ActiveX Documents technology to create the IISP Internet-enabled applications, namely:

1- ActiveX documents allow the author to use the power and flexibility that exists in Visual Basic as a development environment to develop the IISP and then to migrate the developed system components and modules into Internet-enabled ones without the need to use or spend extra time to learn other programming tools or language.

2- The full capabilities of using the Visual Basic application debugging tools could be utilized.

3- The client's browser checks the downloaded CAB file before downloading. If no change exists within the CAB file (same operating system registry), the browser automatically would escape the download process and use the previously existing/downloaded CAB file. In other words, this provides an automatic way of detecting newer versions of applications and downloading the CAB file associated with the upgraded versions, when required.

On the other hand, three major disadvantages in using the ActiveX documents technology were as follows:

1- Every time you modify the application you have to rebuild the .CAB files.

2- If the user is using either a Macintosh or UNIX-based system, the ActiveX Technology will not work at all.

3- Microsoft Internet Explorer 3.2 or higher is a mandatory requirement for the ActiveX technology in general, as reported by the Microsoft Company.

In addition to the previously mentioned theoretical disadvantages, four major technical problems had been encountered during the migration process, and certain actions/solutions had to be taken. These problems and the solutions can be summarized as follows:

1- In order to create the project Internet package, the Packaging and Deployment Wizard provided in Visual Basic 6.0 professional was used. One of the resulting files, as mentioned earlier, is the HTM file. A sample of this file is shown in Figure 3.52. One of the components of this file is the so-called "ClassID" which represents the application ID within the Windows operating system. This ID is used to register the application on the client machine. It was realized that the Packaging and Deployment Wizard is creating an HTM file with an *empty* ClassID, which causes a problem in registering the application on the client machine and accordingly an application failure. In order to solve this problem, this ClassID had to be obtained first from the system registry and then entered manually to the created HTM file. This problem had been realized and reported by several users/developers and, consequently, a fix to this issue was found by Microsoft and the Visual Basic Service Pack 4 (SP4) was released. SP4 had fixes to some known issues, among which were the ClassID problem of the Visual Basic ActiveX Document technology.

2- A second component of the created HTM file is a Visual Basic Script block of code. The main purpose of this script is to direct the user to the first User Document of the application. It was realized that the HTM file contained a header which noted that the script automatically generated within the HTM file will work only with Internet Explorer 3.x. It means that for a higher version of Internet Explorer, the generated script will not work properly and consequently has to be changed. The created script had to be modified in order for the ActiveX Document to be launched automatically as soon as the browser access and download the application's CAB file. Figures 3.52 a and b show the original and modified HTM file of the IISP application.

```
<HTML>
<HEAD>
<TITLE>Project1.CAB</TITLE>
</HEAD>
<BODY>

<a href=UserDocument1.VBD>UserDocument1.VBD</a>
<!--*********** Comment Begin **********
    Internet Explorer Version 3.x HTML
    ==================================
    The following HTML code has been commented
    out and provided for ActiveX User Documents
    download support in IE 3.x only.  This
    HTML script may not work properly in later
    versions of Internet Explorer.

    Additional information about downloading
    ActiveX User Documents in IE 3.x can be
    found in Microsoft's online support on the
    internet at http://support.microsoft.com.
    *********** Comment End   ********** -->

<!--*********** Comment Begin **********
<HTML>
<OBJECT ID="UserDocument1"
CLASSID="CLSID: "
CODEBASE="Project1.CAB#version=1,0,0,0">
</OBJECT>

<SCRIPT LANGUAGE="VBScript">
Sub Window_OnLoad
    Document.Open
    Document.Write "<FRAMESET>"
    Document.Write "<FRAME SRC=""UserDocument1.VBD"">"
    Document.Write "</FRAMESET>"
    Document.Close
End Sub
</SCRIPT>
</HTML>
    *********** Comment End   ********** -->
```

Figure 3.52 a
Original HTM file.

```
<HTML>
<HEAD>
<TITLE>IISP.CAB</TITLE>
</HEAD>
<BODY>

<a href=UserDocument1.VBD>UserDocument1.VBD</a>
<!--*********** Comment Begin **********
    Internet Explorer Version 3.x HTML
    ==================================
    The following HTML code has been commented
    out and provided for ActiveX User Documents
    download support in IE 3.x only.  This
    HTML script may not work properly in later
    versions of Internet Explorer.

    Additional information about downloading
    ActiveX User Documents in IE 3.x can be
    found in Microsoft's online support on the
    internet at http://support.microsoft.com.
    *********** Comment End   ********** -->

<!--*********** Comment Begin **********
<HTML>
<OBJECT ID="UserDocument1"
CLASSID="CLSID: 9EF264F6-9703-11D4-B3E6-00A0C99B0860"
CODEBASE="Project1.CAB#version=1,0,0,0">
</OBJECT>

<SCRIPT LANGUAGE="VBScript">
Sub Window_OnLoad
   location.replace("UserDocument1.vbd")
End Sub
</SCRIPT>
</HTML>
    *********** Comment End   ********** -->

</BODY>
</HTML>
```

Figure 3.52 b
Modified HTM file.

3- When packaging the application, Visual Basic checks all the necessary components/files for the application and includes them within the package, the CAB file in particular. When the client tries to access the application, those files should be automatically downloaded and registered within the client machine. A problem has been encountered during this process. The problem was that some of these associated files failed to be automatically registered and consequently an application download and installation failure occurred. After testing all of the associated application files and components, the problem was isolated and identified to be a Map Objects 2.0a related problem. If the MapObjects components were removed from the application, the application downloading and installation process would succed. After extensive research contacts through the literature of ESRI MapObjects 2.0a documentation, forums and technical support, a possible solution has been recommended. MapObjects 2.0a control and its associated dependency files need different Microsoft system DLL files to be installed on the client machine before it can be correctly installed. Those files had been identified using a utility program, from ESRI, called "Dependency Checker". The required DLL files were added manually to the package and subsequently the application downloading and installation process was successful. In other words, at this stage the ActiveX document application would be downloaded and installed successfully including the MapObjects 2.0a control.

4- Although the application and its associated controls and components was downloaded and installed, another problem arises. The problem is that the MapObjects 2.0a cannot load any image or vector layers. When trying to add an image layer or identify a vector layer to the application, a message appears indicating that the desired layer is not a valid one. The reason for that was identified to be related to two of the MapObjects component files, namely Shape20.dll and Aiimage20.dll These two files were successfully downloaded but not correctly registered within the client machine. These two files are crucial to the MapObjects components where they are responsible for handling the display and identification of images and vector data within the components. Although these two files are self-registering, it was realized that they could not be automatically registered within the test machine (client machine). Further contacts and research were made and the conclusion was that this problem may be due to the fact that "MapObjects components were not intended/built for Internet application deployment". Several tests were accomplished by the author involving different client side computers and a temporary solution was reached. This solution is based upon the assumption that the user is an ESRI product user and, consequently, he/she would have these two files installed previously within the machine. By removing these two files from the application package, the application would be downloaded, installed, and work properly utilizing the existing Shape20.dll and Aiimage20.dll within the client machine.

Figure 3.53 shows a schematic diagram representing the detailed flow of the steps taken towards the migration process.

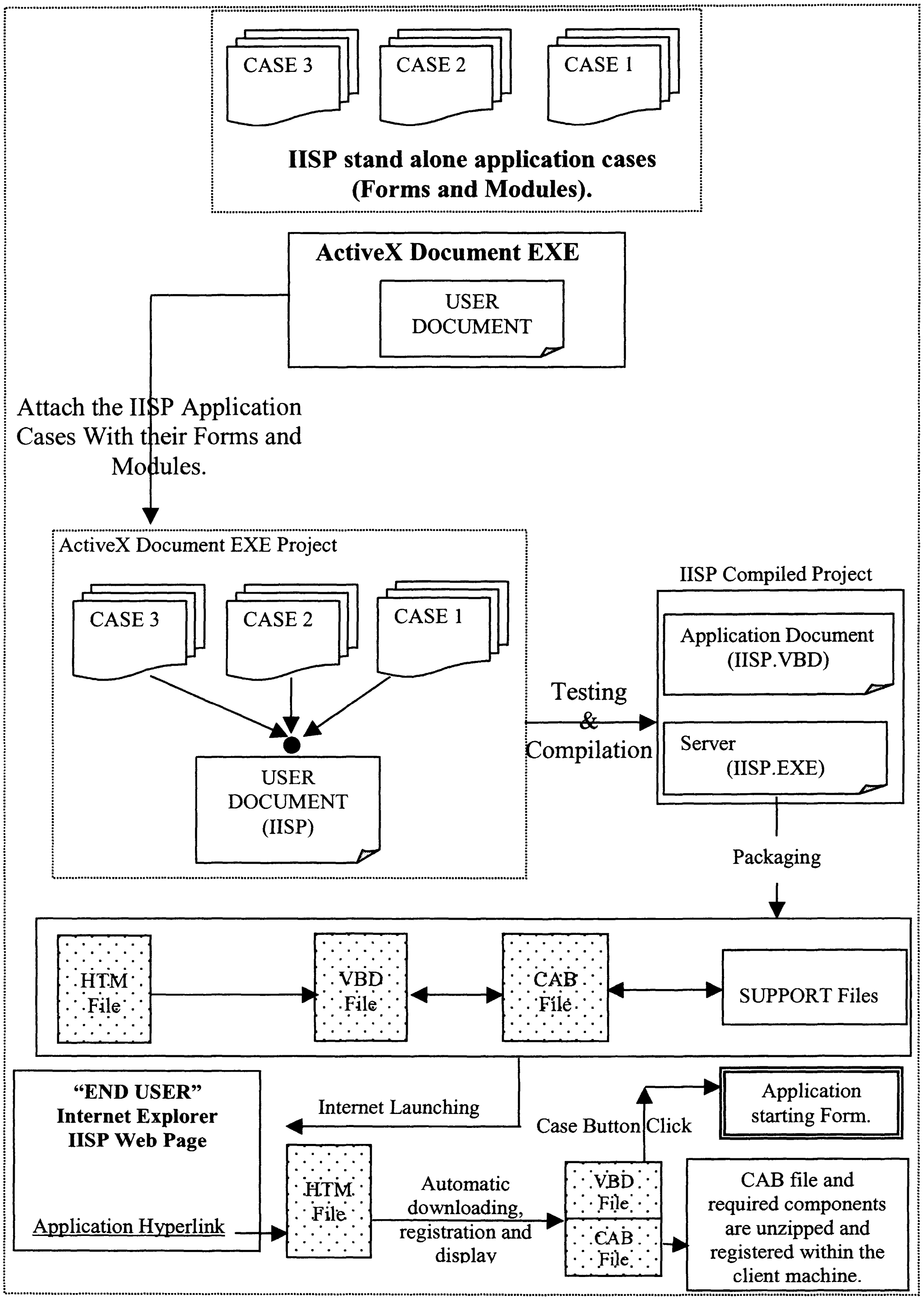


Figure 3.53
Migration process and steps.

CHAPTER 4

IISP TESTING STAGES AND POSSIBLE MODIFICATIONS

The developed system and associated techniques had to be tested to ensure accurate processing flow and correctly obtained results. In testing the system, two main stages had to be accomplished. The first stage dealt with the correct mapping of the designed process and its associated components in the developed programming code. The second one dealt with the correctness of the obtained results from the different query functions within IISP.

This chapter is dedicated to demonstrate the steps and procedures accomplished towards that goal, the results obtained, and the modifications implemented according to the result produced.

4.1 IISP: TESTING STAGE 1

As mentioned earlier, this testing stage is concerned with the System Workflow/Process Testing. This testing stage was a flow chart driven process. In other words, the previously designed flow chart for Case 1 and the modified processing flow for the other cases were used. Each function within the system's implemented three cases was tested using image and vector data sets as explained in sections 3.2 through 3.4 as well as in Appendix B.

Within this stage, the accuracy of the spatially produced results was not of concern. Rather the major concern was whether or not the function processing steps were correct and that no errors existed during the query process. Although this step has been accomplished automatically during the design stages, in order to make sure that the designed/programmed prototype functions were working as required, this stage was essential to detect any error that still existed and had not been detected during the design stage. The main concerns for all tested functions in all cases may be summarized as follows:

1- For All Functions

a- The process is mapped and working correctly for different layer types and for all five designed layers in terms of the processing steps sequence and the input/output utilities, as identified in the design stages.

b- In the case of clearing an existing layer and identifying another one, the system can recognize and connect to the newly identified one.

c- When finishing this query process and starting another function, none of the associated results or other parameters obtained within the pointing query function should exist.

2- Polygon/Line Based Image Query Function

a- The prototype is working properly when all the layers are working when selected to query by polygon or line.

b- All the search criteria within the user-defined polygon/line are working and producing different results, if applicable.

c- Manipulating results function (flash one shape, flash all shapes, zoom, pan, save results) are working for all five layers options and for all layer types.

3- Data Base Query Function

a- Building the query expression manually or using the system tools are both working and produce the same results.

b- Result filtering to obtain the features matching the search expression within the image extent only is working.

4- Buffering Query Function

a- The prototype is working properly for all the layers when selecting the buffering core feature using the pointing selection option.

b- Manipulating results function (flash one shape, flash all shapes, zoom, pan, save results) is working for all five layer options and for all layer types.

During this process, three main issues/problems/errors were encountered during this testing stage. Tables 4.1 (a) through (d) show the issues/problems/errors encountered and their respective solutions.

To avoid any run time errors after the system deployment, an error handler procedure was added to each of the main procedures/modules within the developed prototype. The error handler procedure starts at the beginning of the procedure/module by a statement that directs the program to the error handler procedure in case of any error occurrence, namely "On Error Go To Errorhandler". We need the system to deal with the error, by either ignoring it and resuming with the next step or by displaying a message with the error description, rather than halting/crashing the program or any other problems.

Consequently, the “Errorhandler” procedure located at the end of the module allows the system to generate an error description and to display a message to the user containing this description or to resume with the next step, based upon the procedure function. Following that, the user has to press the “OK” button within the Message Box to continue within the same or another query process without affecting the program itself.

Table 4.1 a: IISP testing stage 1.

Pointing Query Function					
CASE 1		**CASE 2**		**CASE 3**	
Problem	**Solution**	**Problem**	**Solution**	**Problem**	**Solution**
1- The system should clear the results displayed within the ListView in the main form in order to avoid confusion in subsequent queries.	1- The system uses the ListView “Clear” method in order to clear the results after the query results are confirmed, displayed, and another function is to be performed.	Same as in Case 1	Same as in Case 1	Same as in Case 1	Same as in Case 1

Table 4.1 b: IISP testing stage 1.

Polygon/Line Based Image Query					
CASE 1		**CASE 2**		**CASE 3**	
Problem	**Solution**	**Problem**	**Solution**	**Problem**	**Solution**
NA	NA	NA	NA	NA	NA

Table 4.1 c: IISP testing stage 1.

Data Base Query Function					
CASE 1		**CASE 2**		**CASE 3**	
Problem	**Solution**	**Problem**	**Solution**	**Problem**	**Solution**
1- When executing a search query expression and no results are obtained, either one of two messages should be displayed to the user: The first should inform the user if the query expression built had a wrong syntax. The second one confirms the non-existence of query results within the image extent. The system should identify which of the two cases exists and then sends the appropriate message.	1- When executing the query search, a condition statement was added in order to first identify whether the search expression has the correct syntax. Then, if no results are obtained, this will be due to the fact that no features exist at all or within the image extent matching the query expression.	Same as in Case 1	Same as in Case 1	Same as in Case 1	Same as in Case 1

Table 4.1 d: IISP testing stage 1.

Buffering Query Function					
CASE 1		CASE 2		CASE 3	
Problem	Solution	Problem	Solution	Problem	Solution
1- In the case of selecting the core buffering feature using the pointing query option, the same issue of removing the selected features' attributes from the ListView, as in Case 1, should be considered.	1- The same solution as taken in Case 1.	Same as in Case 1	Same as in Case 1	Same as in Case 1	Same as in Case 1

4.2 IISP: TESTING STAGE 2

Stage 2 was concerned with the accuracy of the produced/resulting features from the query functions within the IISP. This was accomplished by comparing the results obtained from querying the image within the IISP context and by performing the same query process using processed/transformed vector data within the ESRI ArcView software packages. The results obtained from the steps in this testing stage are shown in Appendix E. Figures 4.1 a through c show the major concern taken in accomplishing such tasks.

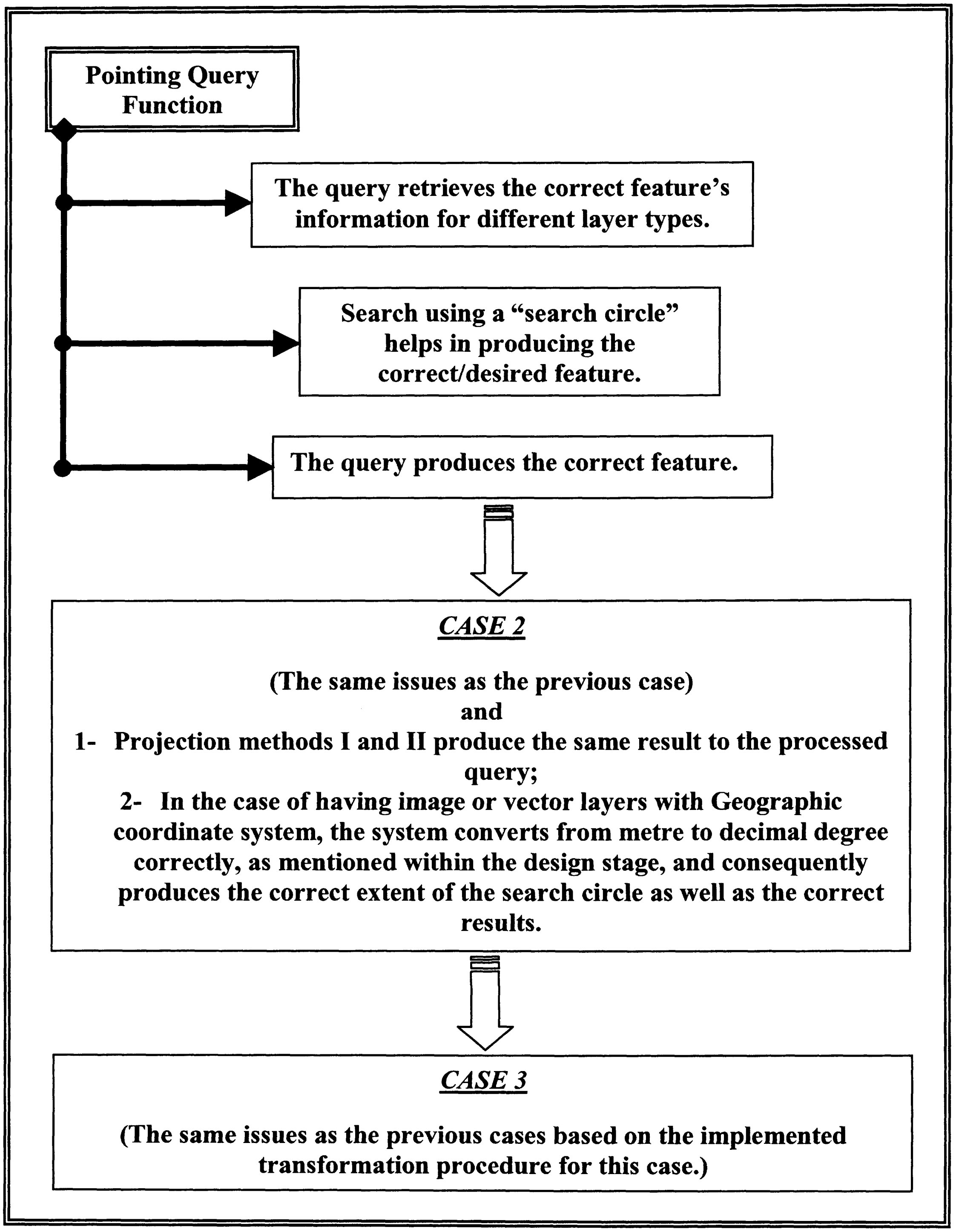


Figure 4.1 a.
IISP testing stage 2.

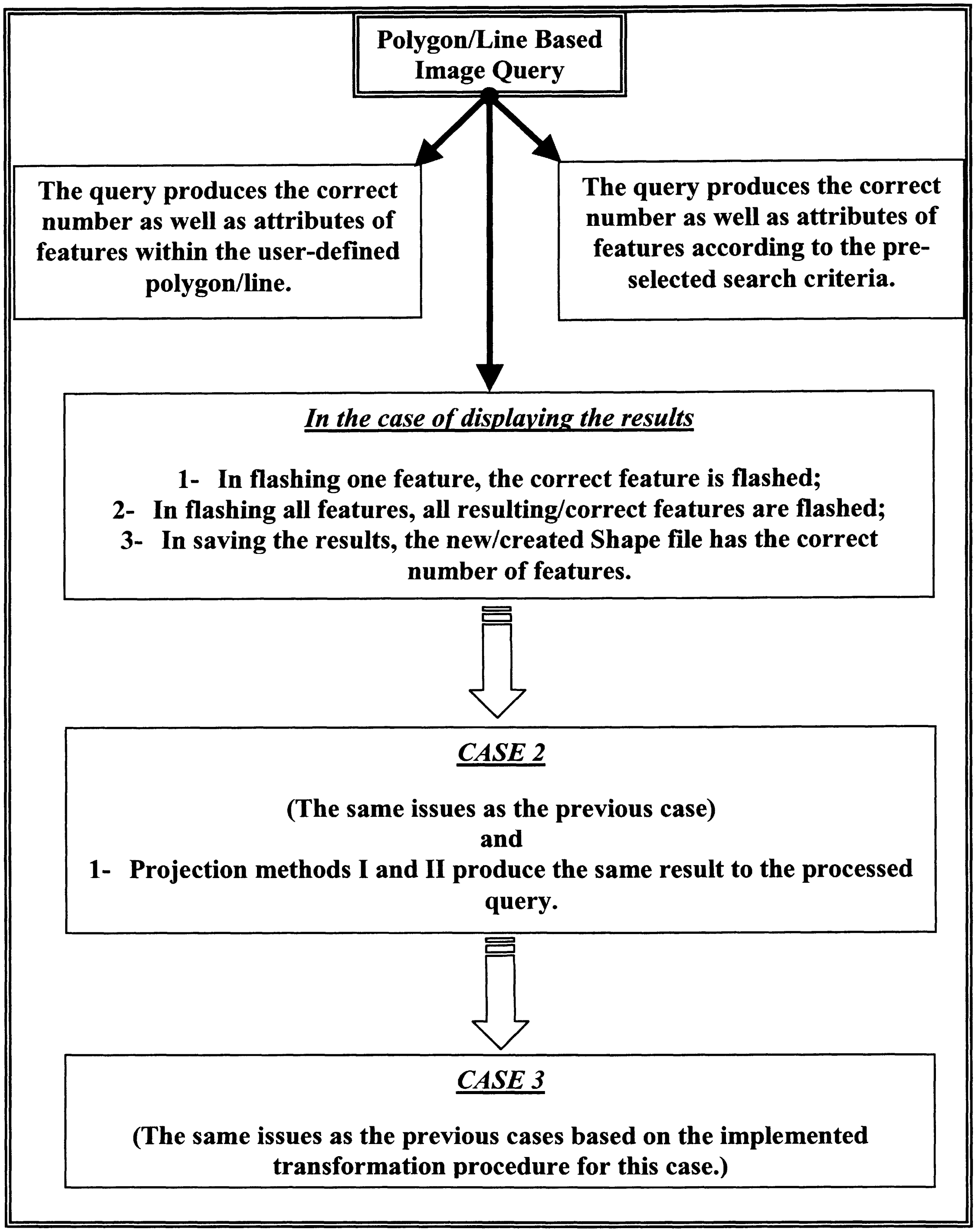


Figure 4.1 b.
IISP testing stage 2.

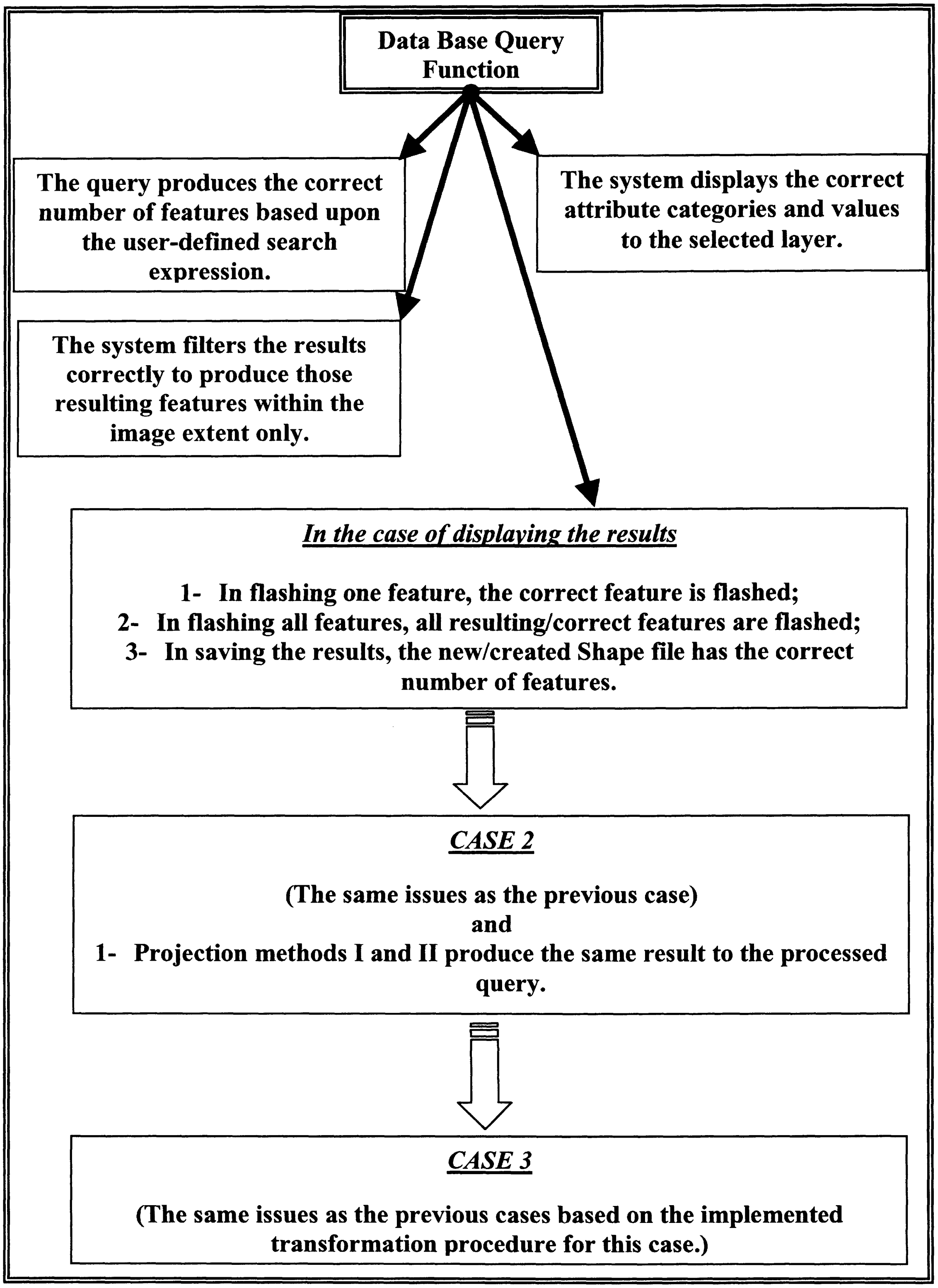


Figure 4.1 c.
IISP testing stage 2.

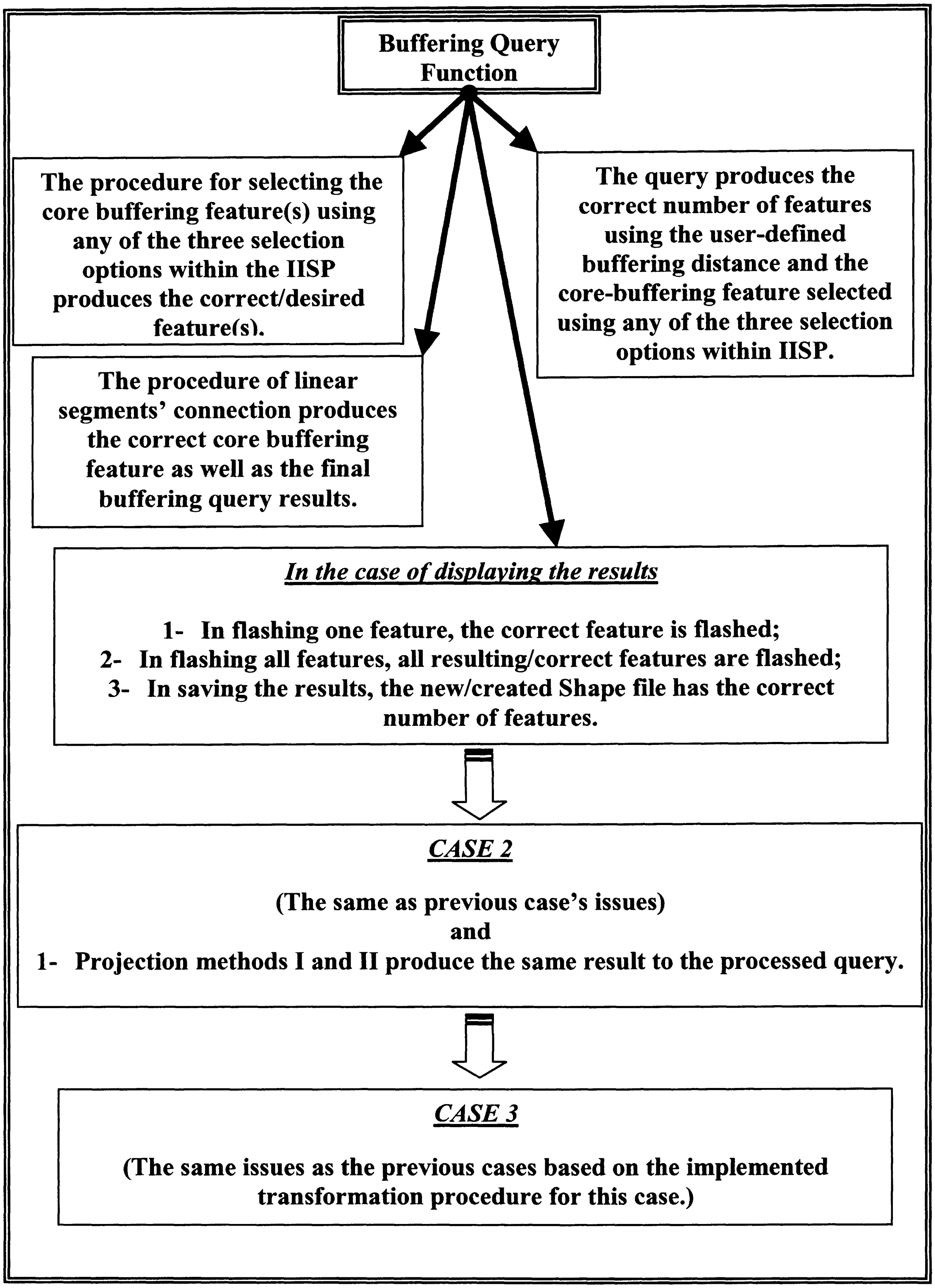


Figure 4.1 d.
IISP testing stage 2.

4.3 IISP: POSSIBLE MODIFICATIONS

The IISP developed within this research is a system prototype to demonstrate the concept of relying on remote sensing imagery, internally connected to GIS vector databases, as a reference layer to directly and on-the-fly query the real world and, consequently, perform exploratory types of spatial analysis. Within the system development, the author dealt mainly with the system interface issues, process work flow, required software components for the system, embedding on-the-fly projection and coordinate system transformation procedures, developing an orthorectification procedure for partially rectified and unrectifed imagery, and embedding these procedures internally within the process so that they can be performed when needed and executed simultaneously within the query process.

The selected functionality and components within the system were limited to just demonstrate the idea and the main required techniques and implementation methodology for this concept and definitely can be extended to handle more functions and to add more components. The following subsections represent the author's view on how the system can be expanded and/or integrated with existing infrastructure components to handle more functions and to serve different users more efficiently.

4.3.1 Imagery Types

As shown in this research project, the IISP dealt with 3 types of imagery, namely, orthoprectified imagery, IKONOS GOECARTERRA™ partially/geo rectified imagery, and unrectified airborne photography and video imagery. While theoretical as well as implementation solutions were given to the first two types, the third one has been examined and one possible theoretical solution was demonstrated. Further research should consider the implementation aspect of the third type of imagery. In addition to that, other types of imagery should be examined and implemented within the system. SPOT panchromatic and multispectral and Landsat MSS imagery should be closely examined and a comparison of the various developed and well-documented orthorectification models with those types of imagery should be conducted in order to decide on the most appropriate and computationally economical model to be implemented within the IISP. The system should also include other types of satellite imagery such as the Indian Remote Sensing satellite (IRS), the Canadian RadarSat, etc.

4.3.2 Vector Format

Within a spatial data infrastructure, files from different sources most probably will have different formats. The IISP was developed based upon using the ESRI Shape files as the main structure of vector data. Currently, MapObjects 2.0a supports other types of vector data formats such as ARC/INFO coverage files, CAD drawing, Vector Product Format (VPF) files, and Spatial Data Engine SDE layers. Other data file formats should be involved within the systems. More accurately, a software component should be added that can handle those formats and allow the system to access and retrieve information from layers with different formats even within the same query process. This is not a distant task. Global Geomatics Inc. is one of the leading companies in this discipline [Global Geomatics, n.d.]. Several components and adapters exist within their line of products that can handle such a task. A thorough research of these components as well as other components from other vendors should be conducted to decide on the best product that can handle a wide range of formats and that is appropriate for implementation within the system.

4.3.3 System Functionality

As mentioned previously, the developed IISP is a general-purpose system and not an application-specific one. If the system should be expanded in order to match the exact requirements for a certain application, several other components and functions may be added to the system. For example, in the Pointing Query function and when querying a linear feature layer/Shape file, the system will only search for the segment that matches the selected position. If the application requires whole road segments to be displayed in that type of query, a road-segments connection procedure should be accomplished in order to identify the entire road segments and the road can be displayed completely, as explained in the IISP Buffering Query function. Another example may relate to performing a Buffering Query function using a user-defined drawn shape (polygon shape, for example). In such a case users may not need to identify/draw the polygon themselves but rather they may need to input the polygon's vertices coordinates directly and the system creates the polygon based on the entered coordinates. This situation may happen because the user may have a previously prepared plan/procedure to be followed. Furthermore, the user may need to identify multiple features as a core buffering feature, that are not already stored in a file, in performing the buffering query. In such a case, a multiple pointing selection procedure may be developed similar to the one demonstrated in the IISP Pointing Query function, with the appropriate user-image interaction process.

Then the system can make use of the "UNION" method to allow the buffering operation around all the selected features to be created simultaneously.

These are some examples to demonstrate that each application and each function will have its own requirements, and the main procedures and methods to perform those functions are demonstrated within the developed IISP.

4.3.4 Integration With Other Spatial Data Infrastructure Elements

Within the present spatial data infrastructure age, the concept of having applications/elements/components integrated is an important one. One example that can benefit the IISP and the users is the CCRS project "CEONet". Within this component the user can, for example, search the available GCPs for a desired area, select some or all of them, and download them as a text file [CEONet, n.d.]. Not only GCPs but also DTM, imagery, maps, etc., can be searched, explored, and downloaded with the associated metadata information. Such applications can be useful when, for example, we need to identify GCPs or to provide a DTM for better and more accurate results, as explained within the IKONOS GEOCARTERRA™ imagery case, Case 3.

One possible method of implementation directly within the IISP process flow is to add a menu item, "USE CEONet" for example, or to add an option of downloading from the CEONet within the GCP's Identification Message or DTM File Identification Message. The user selects that option, the system opens a new browser window, the user browses the CEONet performing the required search and downloads the required file(s), and the system recognizes directly the downloaded file(s) and connects or uses them as appropriate within the IISP.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

In the preceding chapters, the development of the Intelligent Imagery System Prototype (IISP) was presented. This system allows for the direct on-the-fly usage of remote sensing imagery, with various levels of rectification, supported internally by hidden GIS vector database layers, with different parameters, and functionality to query the real world and, consequently, to perform an exploratory type of spatial analysis. These query procedures are accomplished without the need for superimposition, simultaneous display, or data conversion and as a result they were based upon an undisturbed synoptic view of the high-resolution remotely sensed data, the primary data source and the almost identical representation of the real world.

In addition to identifying the system requirements, the user-image interaction requirements, the required software components, and the technical problems (as well as system designing, implementing, testing, and performing an Internet-migration procedure), several contributions were made during this research particularly:

1- It provided a research overview for remote sensing and GIS integration. The reported problems within the current integration approaches are identified. Emphasis was given to the roles of remote sensing imagery within a GIS that lead to possible improvements within these roles and introduces the intelligent imagery concept in general.

2- It laid out a framework of possible integration cases that handle different levels of image rectification and vector layers characteristics. It consequently designed, built, and tested a system to handle the intelligent imagery concept in several of these cases with an increasing level of complexity.

3- A detailed flow chart mapping the system process was developed which provides the opportunity to track down required modifications for any possible expansions as more software components and/or toolkits are released.

4- A new simplified pixel-based orthorectification approach was developed dealing with one of the most recent commercial high-resolution satellite imagery, namely IKONOS. The developed approach was tested using the georectified, lowest rectification level, IKONOS GEOCARTERRA™, for the Fredericton area. The results showed that the developed method can on-the-fly orthorectify the image pixels' position, when needed within the query process, and obtain the correct/ground position with an accuracy as high as for the expensive orthorectified IKONOS imagery.

5- Two on-the fly projection utilities have been developed and implemented within the IISP. While the first one projects all desired layers internally at the beginning of the process, i.e., during the layers' identification process, the second method keeps the desired layer(s) unprojected and projects the required query input and results when needed during the query/results display process. Although both methods proved to obtain the same spatial results, the second one will be more flexible if the desired query layers are very large and if the user has limited computer resources to process such layers at once. It is also important to note that the second method involves several transformations within the query process, which may affect overall performance.

6- A theoretical solution as well as the anticipated process flow for using unrectified scanned aerial photos and/or captured video imagery frames for performing the query were demonstrated and presented.

7- As a common situation for many data production and distribution organizations, one possible solution to the problem of having linear features (such as roads) stored as separate segments within non-topologically structured layers was developed. This allows the system to automatically recognize the whole feature (road for example) and internally process either an individual segment or the whole feature when performing the query based upon the user selection.

8- The technical problems of using ESRI MapObjects 2.0a components within an Internet-based application were identified and a partial solution was implemented.

9- The author's vision, supported by an example, on the role of the software component technology in the future of the so-called GIS and remote sensing "Total Integration" was presented.

5.1 LIMITATIONS OF THE RESEARCH

This research project was subjected to several limitations; namely:

1- Image Data

The remote sensing data used in this project was obtained from the data sets available within the Department of Geodesy and Geomatics Engineering at the University of New Brunswick (UNB) or through SNB. The imagery covers different portions of the province of New Brunswick and was used in this project as obtained with no special enhancements in order to meet the same general situations that may face average users.

2- Vector Layers Format

Vector data layers covering the corresponding extent of the used imagery were obtained from the UNB digital library map collections, SNB, and Statistics Canada with different projections, coordinate systems, scales, and format. Some of the files were in Environmental System Research Institute (ESRI) Shape file format and others were in the Universal Systems Limited CARIS NTX file format. The ESRI Shape files format was the selected file format used to implement and test the developed prototype. Accordingly, file format conversion utilities were used to convert the CARIS NTX files to ESRI Shape files.

3- Research Cases

During the course of this research, a framework representing possible integration cases was developed, but not all of the cases were implemented. Selected research cases representing an increasing level of complexity were implemented.

4- Functionality:

Within the intelligent imagery concept in general and IISP in particular, a wide range of spatial operations/functions can be implemented. In this research, selected functions covering only four major categories of spatial operations were implemented.

5- *Development Environment*

The system was developed within the Windows NT 4.0 operating system and Visual Basic 6.0 programming environment. Furthermore, it was migrated to the Web using ActiveX Document technology. Unfortunately, ActiveX technology works only on a PC platform with Internet Explorer as the Web browser. In other words, Macintosh and UNIX users will not be able to access the application. Although recent plug-ins were available for the Netscape browser to handle ActiveX technology, it is still not working properly.

The following two sections represent the main conclusions reached during the course of this study and recommendations regarding the further developments of the system and the concept.

5.2 CONCLUSIONS

Based upon the course of this research, the following can be concluded:

1- General system

a- When used to perform the defined spatial queries, IISP provides the same results as those obtained by querying pre-processed vector layers even if the image used is not orthorectified and the vector layers have different parameters. This was done by linking the desired image with different vector layers through transformed/corrected features' image coordinates without the need for superimposition and vector-to-raster or raster-to-vector data conversion.

b- Within IISP, time consuming *pre-processing* tasks such as transformation, conversion, and rectification of vector layers and remote sensing imagery are no longer mandatory requirements for querying the real world and performing exploratory spatial analysis using both data sets.

c- As implemented and shown in the design stage, it is possible to embed the developed transformation/orthorectification procedures on-the-fly within the query process flow by performing it when required and only using the required image features/pixels.

d- IISP is an expandable system. As more functionality's components will be released on the future, IISP can be modified and expanded to handle other spatial tasks and integration cases.

2- *Software components technology*

a- In order to rely on the software components technology to reach a very advanced level in GIS and remote sensing integration, namely total integration, GIS vendors should add more functionality within their released software components and remote sensing vendors should consider releasing and maintaining the components that hold their technology.

b- When using MapObjects 2.0a in developing Internet-based applications, the packaging and deployment process may fail. A partial solution to this problem was attempted, as described in section 3.6.2 and 3.6.3, and the application Internet package was created.

c- Although Visual Basic provided a flexible and easy to use visual environment to create the prototype, the programming mode is still procedural rather than Object Oriented. In addition, not all available components are compatible with Visual Basic.

3- *System Internet migration using ActiveX Document technology*

a- ActiveX document technology proved to be an appropriate method for migrating developed application/software to the Internet. It consequently provides a faster and more flexible distribution approach with minor internal software modifications.

b- ActiveX document technology provides an automatic/on-the-fly updating procedure to the desired software while the user is accessing it through his/her browser.

c- ActiveX technology will not work properly or at all unless the user is using Microsoft Internet Explorer browser and a PC-based Windows environment.

4- *Remote sensing and GIS total integration*

a- As discussed, the software component technology provides a reasonable and productive solution towards the remote sensing and GIS integration main goal, namely, "Total Integration". Software components technology will provide the flexibility and the power to integrate the objects of the functionalities internally based upon pre-defined process flow and, consequently, perform closer remote sensing and GIS integration, which will lead to a shorter/more direct approach to reach the final goal of total integration. This technology will also effectively accommodate the dynamic application requirements for many organizations and deciplines.

b- The so-called "Total Integration" system should be project/application-oriented rather than a general-purpose system. In other words, the application related requirements should be identified first, the required software components that can handle the applications' functionality should be provided, and the required integrated solution/application can be developed to meet the needs of the previously defined application's requirements.

5.3 RECOMMENDATIONS

Based on this research, further research and development issues can be extracted as follows:

1- Candidates for short-term development and refinement

a- The developed orthorectification procedure for IKONOS GEOCARTERRA™ imagery pixel coordinates was tested using the Fredericton imagery, which represents a terrain with 120-metre elevation variations. This procedure should be tested using other IKONOS imagery representing different terrain elevations (possibly with larger terrain variations, and different elevation and azimuth angles).

b- Other types of high-resolution remotely-sensed data, such as SPOT and Landsat, should be examined and the appropriate modeling techniques should be implemented within the IISP.

c- The IISP should be extended to include data adapter components in order to deal with different vector layer file formats, such as CARIS format.

d- ESRI should consider expanding the objects and functionality that exist within MapObjects 2.0a in order to flexibly handle more application requirements. For example, surface interpolation techniques should be handled, which will facilitate the requirements of the IKONOS GEOCARTERRA™ imagery case and of other types of imagery that may need such functionality. In addition, direct reading of image file projection and coordinate system parameters will be helpful.

e- The developed theoretical solution for the unrectified scanned aerial photos and captured video imagery frames should be implemented and tested to examine the usage feasibility in terms of system performance and resulting accuracy within the Intelligent Imagery concept.

f- When developing an application using the software component technology and to be distributed/used across the Internet using the ActiveX technology, the developer should check whether any of the components, especially the spatial ones, are developed for Internet packaging. This can be done by creating a simple application containing all the desired components, deploy it, and monitor the behavior of the resulting package within the intended usage environment.

g- Further study should be made in order to identify and solve the technical problem, encountered within this research project, in using ESRI MapObjects components to build Web-based applications. Furthermore, other spatial components (such as MapInfo MapX and INTERGRAPH GEOMEDIA) should be tested against the same problem.

2- Items for longer-term research

a- Remote sensing vendors should consider the release and maintenance of software components including their line of products' objects and functionality.

b- IISP is an extendable system. Other application-related spatial query functionalities should be implemented within the system, such as the "Polygon-Based Image Analysis" procedure discussed in section 2.4.2. Required application-specific components, process flow modifications and functionality should be identified and implemented within the IISP to handle the desired application.

c- Reliable plug-ins and extensions should be developed in order to handle the existing problems in using ActiveX technology within the Netscape browser and Macintosh/Unix-based systems.

d- A classification scheme should be developed to identify the range of appropriate scale factors of vector layers that can be used with different image types having different geometrical and spectral characteristics. This will provide a guide in obtaining the correct vector layers to be involved in the query based upon the Intelligent Imagery concept.

e- Further research should be accomplished in order to identify the requirements and the necessary improvements within the software component technology to handle and probably reach the goal of GIS and remote sensing total integration in different applications.

LIST OF REFERENCES

Abdelrahim, M. (1994). "Aerotriangulation Using Bundle Block Adjustments: Merits and New Trends.", M.Sc.Eng, Alexandria University, Alexandria, Egypt.

Abdelrahim, M., D. Coleman, R. Casonguay, and D. Raymond (2000 a). "Compression and Distribution of SNB Orthophoto Map Database." *Proceedings, XIXth Congress of theInternational Society for Photogrammetry and Remote Sensing (ISPRS)*, Amsterdam,The Netherlands Volume XXXIII, Part B2 (CD), June 16-23, pp. 17 – 29.

Abdelrahim, M., D. Coleman, and W. Faig (2000 b). "Intelligent Imagery System: A Proposed Approach." *Proceedings, XIXth Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Amsterdam,The Netherlands, Volume XXXIII, Part B2 (CD), June 16-23, pp. 11 – 24.

Airborne Videography Ltd. (n.d.) "Using Video with GPS and GIS." http;//www.avxs.demon.co.uk, September, 2000.

Ambrosia, V. G., S. W. Buechel, J. A. Brass, J. R. Peterson, R. H. Davis, R. J. Kane, and S. Spain (1998). "An Integration of Remote Sensing, GIS, and Information Distribution for Wildfire Detection and Management." *Photogrammetric Engineering and Remote Sensing*, Vol. 64, No. 10, pp. 977 – 985.

Appleman, D. (n.d.) "Dan Appleman's Developing ActiveX Components Using Visual Basic 5 - Chapter7: ActiveX Components: What's in a Name?" http://www.tts.tomsk.su/personal/~miha/books/VB5ativeX/ch7.htm, Mar. 2001.

Armenakis C., E. Siekierska, F. Hegyi, P. Pilon (1999) "Production and 'Intelligence' of Raster Maps", *ICA '99 International Conference*, CD-ROM Proceedings, Section C3-B.

Aronoff, S. (1996). *Geographic Information Systems: A Management Prespective.* Ottawa, Canada: WDL Publications.

Australian National Mapping Agency (AUSLIG). (2000). "Topographic Map and Data Index." http://www.auslig.gov.au/mapping/index/mapind.htm, March 2001.

Baker, J. R. and J. E. Drummond (1984). "Environmental Monitoring and Map Revision Using Integrated Landsat and Digital Cartographic Data." *ITC Journal,* Vol. 15, No.1, pp. 10-19.

Baltsavias, E. (1996). "Digital Ortho-Images – A Powerful Tool for the Extraction of Spatial- and Geo-Information." *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 51, pp.63 – 77.

Barker, C. R. (1988). "Remote Sensing: The Unheralded Component of Geographic Information Systems." *Photogrammetric Engineering and Remote Sensing*, Vol. 54, No. 2, pp. 195-199.

Barnes, G. and A. Vonderohe (1985). " An Analytical Technique to Account for Photographic Distortion in Natural Resources Records." Annual Conference of the Urban and Regional Information Systems Association (URISA), Ottawa, Ontario, Canada, July 28 - August 1, pp. 171 – 180.

Baumgartner, M. F. and G. Apfl (1994). "Towards an Integrated Geographic Information System With Remote Sensing, GIS and Consecutive Modeling for Snow Cover Monitoring." *International Journal of Remote Sensing*, Vol. 15, pp.1507 -1517.

Buiten, J. H. and J. G. P. W. Clevers (1993). *Land Observation by Remote Sensing:Theory and Applications*. Amsterdam, Netherland: Gordon and Breach Science Publications.

Campbell, J. B. (1995). *Introduction To Remote Sensing*. New York, U.S.A: Guilford Publications.

Castek, D. (2000) "Visual Basic (Component Based) Applications." http://www.cbd-hq.com/articles/2000/001016db_vba4cbd.asp, Oct. 2000.

CEONet. (n.d.) "Search For Data." http://ceonet.ccrs.nrcan.gc.ca/cs/en/index.html, Mar, 2001.

Cheng, P. and Th. Toutin (1994). "Generation of Orthorectified Satellite Images and Air Photos Using Stereoscopic Images." *Proceedings of The Canadian Conference on GIS,* Ottawa, June 6-10, Canadian Institute of Geomatics, Vol. 2, pp.1431 – 1439.

Chuvieco, E. and R. G. Congalton (1989). "Application of Remote Sensing and Geographic Information Systems to Forest Fire Hazard Mapping." *Remote Sensing For Environment*, Vol. 29, pp. 147- 159.

Clark, W. A. S. (1974). "Orthophotography Applied to the Planning of Stonehouse New Town." *Photogrammetric Record*, Vol.8, No.44, pp. 154 – 166.

Cleynenbreugel, J. V., F. Fierens, P. Suetens, and A. Oosterlinck (1990). "Delineating Road Structures on Satellite Imagery by a GIS-Guided Technique." *Photogrammetric Engineering and Remote Sensing*, Vol. 56, No. 6, pp. 893 – 898.

Cooper, R. D., T. MaCarthy, and J. Raper (1995). "Airborne Videography and GPS." *Earth Obseravation Magazine*, Vol. 4, No.11, pp. 53 – 55.

CVIU (1998). "Special Issue on Automatic Building Extraction From Aerial Images." *Computer Vision and Image Understanding*, Vol. 72, No.2.

Derenyi, E. and D. Fraser (1996). "Using Images Within a GIS for Spatial Analysis." *International Archives of Photogrammetry and Remote Sensing*, ISPRS Volume XXXI, Part B4, Commission IV, Vienna, Austria, July 9-19, pp. 216 – 218.

Derenyi, E. and M. Turker (1996). "Polygon Based Image Analysis of Remotely Sensed Images in an Integrated Geographic Information System (IGIS)." *International Archives of Photogrammetry and Remote Sensing*, ISPRS Volume XXXI, Part B4, Commision IV, Vienna, Austria, July 9-19, pp.212 – 215.

DIST (2000). "Department of Information Systems and Telecommunications Montgomery County's Digital Orthophoto Image Server." http://gis.co.mo.md.us/gis/imgservr.htm, Dec. 2000.

DPR Consulting Ltd. (n.d.) "ActiveX Key Technology Introduction." http://www.dpr.co.uk/keytechactivex.htm, Mar. 2001.

Ehlers, M., G. Edwards, and Y. Bedard (1989). "Integration of Remote Sensing With Geographic Information Systems: A Necessary Evolution." *Photogrammetric Engineering and Remote Sensing,* Vol. 55, No. 11, pp. 1619 – 1627.

Ehlers, M., D. Grenlee, T. Smith, and T. Star (1991). "Integration of Remote Sensing and GIS: Data and Data Access." *Photogrammetric Engineering and Remote Sensing*, Vol. 57, No. 6, pp. 669 – 675.

Ehlers, M. (2000). "Integrated Geographic Information System: From Data Integration to Integrated Analysis." *Proceedings, XIX^th^ Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Volume XXXIII, Part B2, Amsterdam,The Netherlands, June 16-23, pp.172 – 178.

Eidahl, L. (1997). *Using Visual Basic 5: Platinum Edition*. 1st ed., QUE Corporation, Indianapolis, U.S.A.

El-Manadili, Y. (1994) "Reduction of Spot Imagery by The Direct Linear Transformation (DLT) Method." (ASPRS/ACSM 1994). http://www.odyssey.maine.edu/gisweb/spatdb/acsm/ac94019.html, Dec. 2000.

ERDAS (n.d.a). "ERDAS Products." http://www.erdas.com, September 1999.

ERDAS. (n.d.b). "ArcView IMAGE ANALYSIS." http://www.erdas.com, October 2000.

ERMAPPER (1999). "ER MAPPER C Library." http://www.ermapper.com/product/ermapper6/prodinfo/clibrary.htm, Dec. 2000.

ERSI (Earth Resource Surveying Inc.) (n.d.) "IKONOS Image Products" http://www.ersi.ca/ikonos.html, October 2000.

ESRI (1999a). "MapObjects and MapObjects LT Comparison." http://www.esri.com/software/mapobjectslt/mo_vs_lt.html, Dec, 2000.

ESRI (1999b). "Building Applications With MapObjects." Environmental System Research Institute, Redlands, California, U.S.A.

ESRI (2000a). "ARCVIEW Image Analysis Extension." http://www.esri.com/software/arcview/extensions/imageext.html , Dec. 2000.

ESRI (2000b). MapObjects 2.0a (On-Line-Help). Environmental System Research Institute, Halifax, Canada.

Gamba, P. and F. Casciati (1998). "GIS and Image Understanding for Near-Real-Time Earthquake Damage Assesment." *Photogrammetric Engineering and Remote Sensing*, Vol. 64, No. 10, pp. 987 – 994.

Gerlach, F. (2000). "Characteristics of Space Imaging's One-Meter Resolution Satellite Imagery Products." *Proceedings, XIXth Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Amsterdam,The Netherlands Volume XXXIII, Part B1 (CD), June 16-23, pp. 128-136.

Glenn, D., B. Compton, and P. Crosbie (1993). "Dynamic Integration of Spatially Referenced Data Independent of Data Source Format." *Proceedings, 7th Annual Symposium on Geographic Information System in Forestry, Environmental and Natural Resources Management*, Vancouver, British Colombia, Vol. 2, Feb. 15-18, pp. 1065 – 1071.

Global Geomatics. (n.d.) "OGDI Architecture." http://www.globalgeo.com/products/default.asp?profile=Tech, Mar. 2001.

Hartman,R. (1997). *Focus on GIS Component Software: Featuring ESRI's MapObjects*. OnWord Press, Santa Fe, New Meryland, U.S.A.

Henricsson, O., F. Begnone, W. Willuhn, F. Ade, O. Kubler, E. Balsavias, S. Mason, and A. Gruen (1996). "Project AMOBE: Strategies, Current Status and Future Work." *International Archives of Photogrammetry and Remote Sensing*, Volume XXXI, Part B3, Vienna, Austria, July 9-19, pp.321 – 330.

Hepner, G. F. (1988). "Spatial Query for Decision Support of Cross-Country Movement." *Photogrammetric Engineering and Remote Sensing*, Vol. 54, No. 11, pp. 1617 – 1621.

Hinton, J. C. (1996). "GIS and Remote Sensing Integration For Environmental Applications." *International Journal of Geographical Information Systems*, Vol. 10, No. 7, pp. 877 – 890.

Hodgson, M., J. R. Jensen, H. E. Mackey, and M. C. Coulter (1988). "Monitoring Wood Stork Foraging Habitat Using Remote Sensing and Geographic Information System." *Photogrammetric Engineering and Remote Sensing*, Vol. 54, No. 11, pp. 1601–1607.

INTERGRAPH (n.d.) "Architectural Overview." http://www.intergraph.com/gis/geomedia/geo_objects.asp, March 2001.

Jacobson, I., G. Booch, and J. Rumbagh (1999) *Unified Software Development Process.* Addison-Wesley Longman Inc., Upper Saddle River, New Jersey, U.S.A.

Jakubauskas, M. E., K. P. Lulla, and P. W. Mausel (1990). "Assesment of Vegetation Change in a Fire-Altered Forest Landscape." *Photogrammetric Engineering and Remote Sensing*, Vol. 56, No. 3, pp. 371 – 377.

Janssen, L. F., M. N. Jaarsma and E. T. M. Van der linden (1990). "Integrating Topographic Data With Remote Sensing for Land-Cover Classification." *Photogrammetric Engineering and Remote Se*nsing, Vol. 56, No. 11, pp. 1503-1509.

Jazouli, R., D. L. Verbyla, and D. L. Murphy (1994). "Evaluation of SPOT Panchromatic Digital Imagery for Updating Road Locations in a Harvested Forest Area." *Photogrammetric Engineering and Remote Sensing*, Vol. 60, pp. 1449 – 1452.

Jensen, L. M. (1990). "Knowledge-Based Classification of an Urban Area Using Texture and Context Information in Landsat- TM Imagery." *Photogrammetric Engineering and Remote Sensing*, Vol. 56, No. 6, pp. 899 – 904.

Johnston, J.B., M. Ehlers, D. Steiner, and A. Gomarasca, (eds.) (1997). Proceedings. ISPRS II/2 Workshop, *New Developments in Geographic Information Systems*, Environmental Research Institute of Michigan, Ann Arbor, MI., U.S.A.

Kim, T. and J. Muller (1998). "A Technique for 3-D Building Reconstruction." *Photogrammetric Engineering and Remote Sensing*, Vol. 64, No. 9, pp. 923 – 930.

King, D. (1991). "Determination and Reduction of Cover Type Brightness Variations With View Angle in Airborne Multispectral Video Imagery." *Photogrammetric Engineering and Remote Sensing,* Vol. 57, No. 12, pp. 1571 – 1577.

King, D. (1995). "Airborne Multispectral Digital Camera and Video Sensors: A Critical Review of System Design and Applications." *Canadian Journal of Remote Sensing,* Vol. 21, No. 3, pp. 245 - 273.

Koch, M. and F. El-baz (1998). "Identifying the Effect of the Gulf War on the Geomorphic Features of Kuwait by Remote Sensing and GIS." *Photogrammetric Engineering and Remote Sensing*, Vol. 64, No. 7, pp. 739– 747.

Kontoes, C. , G. G. Wilkinson, A. Burrill, S. Goffredo and J. Megier (1993). "An Experimental System for the Integration of GIS Data in Knowledge-Based Image Analysis for Remote Sensing of Agriculture." *International Journal Of Geographical Information Systems*, Vol. 7, pp. 247 – 262.

Kruck, E. (1988). "Mapping From SPOT Imagery on Zeiss PLANICOMP and ORTHOCOMP." *International Archives of Photogrammetry and Remote Sensing*, Part B3, pp. 127 – 141.

Laurer, D. T., J. Estes, R. Jensen, and D. Greenlee (1991). "Institutional Issues Affecting the Integration and Use of Remotely Sensed Data and Geographic Information Systems." *Photogrammetric Engineering and Remote Sensing*, Vol. 57, No. 6, pp. 647 – 654.

Lee, L. H., and L. C. Chen (1988). "Bundle Adjustment With Additional Parameters for SPOT Stereopairs." *International Archives of Photogrammetry and Remote Sensing*, Vol. 27, Commission III, Part B8, Kyoto, Japan, pp. 1-10.

Lillesand T. M., and R. W. Kiefer (1994), *Remote Sensing and Image Interpretation.* 3rd ed., John Wiley & Sons Inc., New York. U.S.A.

Maggio, R. and R. Baker (1988). "Techniques for Acquisition of Aerial Video Imagery." *Photogrammetric Engineering and Remote Sensing,* Vol. 54, No. 2, pp. 229 – 231.

MAPINFO (2000) "MAPINFO MapX Feature List." http://dynamo.mapinfo.com/products/web/Features.cfm?ProductID=41, Dec. 2000.

Marangozov, V. and L. Bellissard (1996) "Component-Based Programming of Distributed Applications." http://www.newcastle.research.ec.org/cabernet/research/radicals/1996/papers/comp-marangozov.html, Oct. 2000.

Marsh, S., L. James, L. Christopher, and G. Lee (1991). "Multitemporal Analysis of Hazardous Waste Sites Through the Use of a New Bi-Spectral Video Remote Sensing System and Standard Color-IR Photography." *Photogrammetric Engineering and Remote Sensing,* Vol. 57, No. 9, pp. 1221 – 1226.

Massachusetts Institute of Technology (MIT). (1998) "Tools to Facilitate Access to Digital Orthophotos." http://ortho.mit.edu/, Oct. 2000.

Mausel, P., J. Everitt, D. Escobar, and D. King (1992). "Airborne Videography: Current Status and Future Prespectives." *Photogrammetric Engineering and Remote Sensing,* Vol. 58, No. 8, pp. 1189 – 1195.

McKeown, D. M.(1984). "Knowledge-Based Aerial Photo Interpretation." *Photogrammetric Record,* Vol. 39, pp. 91 – 123.

McKeown, D. M. (1987). "The Role of Artificial Intelligence in the Integration of Remotely Sensed Data With Geographic Information Systems." *IEEE Transaction on Geoscience and Remote Sensing*, Vol. GE-25, No. 3, pp 330-348.

Meisner, D. (1986). "Fundamentals of Airborne Video Remote Sensing." *Remote Sensing Of Environment*, No. 19, pp. 63 – 79.

Meyer, B. (1999) "Promises and Challenges of Components." http://www.eiffel.com/doc/manuals/technology/bmarticles/computer/components/page.html, Dec. 2000.

Mickelson, J. G., D. L. Civco, and J. A. Silander (1998). "Delineating Forest Canopy Species in the Northeastern United States Using Multi-Temporal TM Imagery." *Photogrammetric Engineering and Remote Sensing*, Vol. 64, No. 9, pp. 819 – 907.

Microsoft Development Network (MSDN) (n.d.) "Design Consideration for ActiveX Documents." http://www.microsoft.com., Dec. 2000.

MODIS (2000). "IKONOS Image Tile." http://modis-land.gsfc.nasa.gov/val/ikonos_metadata/po_38589_metadata.txt., Dec. 2000.

Molenaar, M. and L. F. Janssen (1992). "Terrain Objects, Their Dynamics and the Integrated Processing of Remote Sensing Data." *Proceedings of the International Workshop IAPR TC7, Wageningen Agriculturs University,* Delft, The Netherlands, September 7-9, pp. 113 – 134.

Natural Resources Canada (NRCAN) (2001). "National Topographic Data base" http://www.cits.rncan.gc.ca/siteCITS/servlet/CITS?site_id=01&sessionid=&page_id=1-006-001, Jun 2001.

Niederost, M. (2000). "Reliable Reconstruction of Buildings for Digital Map Revision." *Proceedings, XIXth Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Amsterdam,The Netherlands Volume XXXIII, Part B2 (CD), June 16-23, pp.635 – 642.

PCI Geomatics (n.d.) "PCI Products". http://www.pcigeomatics.com/product_ind/product.html, October 1999.

Pharoah, A. and C. Brooke (2000). "Creating Commercial Components." http://componentsource.com/build/COMWhitePaper.asp, Dec. 2000.

Radhadevi, P. V., T. P. Sasikumar, and R. Ramachandran (1994). "Orbit Attitude Modeling and Derivation of Ground Coordinates From Spot Stereopairs." *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 49, No. 4, pp. 22-28.

Ravindran, K. (1998) "Developing Web Database Applications Using VB 5.0 & MDAC: Part II- Using ActiveX Documents to Implement The User Interface for VB Internet Application." http://www.vbwm.com/art_1998/mda2.htm, Dec. 2000.

Razavi, A. (1999). *Arcview GIS/Avenue Developer's Guide*. OnWard Press, New York, U.S.A.

Savopol, F., M. Chapman, and M. Boulianne (2000). "A Digital Multi CCD Camera For Near Real-Time Mapping." *Proceedings, XIXth Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Amsterdam,The Netherland, Volume XXXIII, Part B2 (CD), June 16-23, pp. 266-271.

Sharma, K. M. S. and A. Sarkar (1998). "A Modified Contextual Classification

Technique for Remote Sensing Data." *Photogrammetric Engineering and Remote Sensing*, Vol. 64, No. 4, pp. 273 – 280.

Shears, J. (1998). "Orthophoto Mapping: The Fast Way." http://www.erdas.com/before/articles/POB.html, March 2000.

Slaymaker, D., K. Johnes, C. Griffin, and J. Finn (n.d.) "Mapping Deciduous Forests in Southern New England Using Aerial Videography and Hyperspectral Multi-Temporal Landsat TM Imagery. http://bandersnatch.fnr.umass.edu/pub/video/vidveg3.html, September 2000.

Software Engineering Institute (SEI) (2000). "Component-Based Software Development/COTS Integration", http://www.sei.cmu.edu/str/descriptions/cbsd_body.html, Carnegie Mellon, Dec. 2000.

Space Imaging (1999). "IKONOS Satellite Launches Into Space." http://www.spaceimaging.com/newsroom/releases/1999/inorbit.htm, September 2000.

Space Imaging (n.d.) "Online Store." http://www.spaceimaging.com/level2/level2buy.htm, Nov. 2000.

Space Imaging Europe (n.d.)"IKONOS FAQ's." http://www.si-eu.com/faqs.asp, Dec. 2000.

Sprott, D. and L. Wilkes (2000). "Component Based Development in the Windows DNA Environment." http://msdn.microsoft.com/componentresources/html/articles/ta/ta_004.asp#section2, Dec. 2000.

Stefanidis, T. (1997). "Softcopy Column." *Photogrammetric Engineering and Remote*

Sensing, Vol. 63, No. 1, pp.7 – 8.

Stoney, W. E. and J. Hughes (1998). "A New Space Race Is On!" *GIS World*, Vol.11, No. 3, pp. 44 – 46.

Terra Surveys Ltd. (2000). "Aerial Mapping Using Video Map." http://www.terrasurveys.com/vmaerial.htm, Mar. 2001.

Toutin Th., and P. Cheng (2000). "Demystification of IKONOS." *Earth Observation Magazine*, Vol. 9, No. 7, pp. 17-21.

Trotter, C. M. (1991). "Remotely-Sensed Data as an Information Source for Geographical Information Systems in Natural Resources Management: A Review." *International Journal of Geographical Information Systems*, Vol. 5, No. 2, pp. 225 - 239.

Turker, M. (1997). "Polygon Based Image Analysis Within An Integrated RS/GIS Environment", Ph.D. dissertation, Department of Geodesy and Geomatics Engineering, University of New Brunswick, Fredericton, NB., Canada.

United States Geological Survey (USGS) (2000). "Geospatial Data Clearinghouse: National Mapping and Remotely Sensed Data." http://mapping.usgs.gov/nsdi/, Mar. 2001.

Universal System Ltd. (USL). (2001)"CARIS Raster Imagery Processing." http://www.caris.com/products/gis/raster_imagery_processing.html, Mar. 2001.

Universal System Limited (USL) (2000). "CARIS Spatial Framework Architecture." http://www.spatialcomponents.com/solutions/development/architecture.html, March 2001.

University of California Santa Barbara (UCSB) (n.d.) "IKONOS Imagery." http://www.geog.ucsb.edu/~jeff/115a/ikonos/ikonosdcpage.html, Mar. 2001.

Veryard, R. (2000) "Component-Based Business (Appendix 1: Components)." http://www.veryard.com, Dec. 2000.

Vigder M., W. Gentleman, and J. Dean (1996) "COTS Software Integration: State of the Art." http://wwwsel.iit.nrc.ca/abstracts/NRC39198.abs, Oct. 2000.

Ware, C. and R. C. Maggio (1990). "Integration of GIS and Remote Sensing for Monitoring Forest Tree Diseases in Central Texas" *Procedings GIS/LIS'90*, Anaheim, California, U.S.A., Nov. 7-16, American Society for Photogrammetry and Remote Sensing, pp. 724-732.

Welch, R. and M. Madden (1988). "Remote Sensing and Geographic Information System Techniques for Aquatic Resources Evaluation." *Photogrammetric Engineering and Remote Sensing*, Vol. 54, No. 2, pp. 177 – 185.

Westin, T. (1990). "Precision Rectification of SPOT Imagery." *Photogrammetric Engineering and Remote Sensing*, Vol. 56., No. 2, pp. 247 – 235.

Wilkinson, G. G. (1996). "A Review of Current Issues in the Integration of GIS and Remote Sensing Data." *International Journal of Geographical Information Systems*, Vol. 10, No. 1, pp. 85 – 101.

Wolf, R. P. and B. Dewitt (2000). *Elements of Photogrammetry With Application in GIS.* 3rd ed., McGraw-Hill, Boston, U.S.A.

Wright, R. and J. Um (n.d.) “Effect of Angular Field-of-View of a Video Sensor on the Information Requirement in a Strip Target: The Example of Revegetation Monitoring at A Pipeline Route.” http://Geo.knu.ac.kr/effect.htm, Dec. 2000.

Zhang, Ch., E. Baltsavias. (2000). “Knowledge-Based Image Analysis for 3D Edge Extraction and Road Reconstruction.” *Proceedings, XIXth Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Volume XXXIII, Part B2, Amsterdam,The Netherlands, June 16-23, pp.1008 – 1015.

APPENDIX A

IISP FUNCTIONAL REQUIREMENTS

This appendix is dedicated to list the functional/sub-functional requirements for the IISP cases. The following tables show all the required functions within the IISP in order to accommodate the cases considered in this research. The required controls, techniques and approaches to implement these functions have been discussed in the IISP design and implementation stages, sections 3.2 through 3.5.

Table A.1: IISP functional requirements.

Case Function	Case 1	Case 2	Case 3	Case 4
Image Loading and Display	**<u>Image Loading</u>** - Select a desired image from the local machine; - Download the image form an FTP site; - Display the image within a container (main container); - Display an overview window emphasizing the current image extent within the main container; - Display the image name; - Display the image coordinates; - Clear the image display; - When loading the image, warn the user in case of having an invalid image selection. **<u>Image Display Configuration</u>** (Required within all functions) - Zoom functions; - Pan function; - Full extent display function.	Functional Requirements as In Case 1 plus - Read or input the image coordinate system.	Functional Requirements as In Case 1, Case 2 plus <u>If the image is not georeferenced</u> - Image coordinate system establishment; - Conversion form screen to image coordinate system; - Relate image to ground, - Based on the established image-ground relationship, identify the image geographic/projected coordinate system.	Functional Requirements as In Case 1 plus - Image coordinate system establishment; - Conversion form screen to image coordinate system; - Relate image to ground; - Based on the established image-ground relationship, identify the image geographic/projected coordinate system.

Table A.1: IISP functional requirements (cont.)

Case / Function	Case 1	Case 2	Case 3	Case 4
Vector Layer Identification	**<u>Layers Selection, Downloading and Identification</u>** - Select/deselect desired vector layer(s) from the local machine; - Download the layer(s) form an FTP site; - Identify the name and the physical location of the layer(s); - Display the layer(s) name; **<u>Vector Layers Configuration</u>** - Select/deselect the layer(s) for query; - Clear/remove vector layer(s); - When identifying the layer, warn the user if the layer is already identified; - Indicate the extent of each vector layer identified within the overview window.	Functional Requirements as In Case 1 plus - Read or input the layer(s) coordinate system; - On-the-fly projection utilities from the layer coordinate system to the image coordinate system or vice versa when needed, based upon the design methodology.	Functional Requirements as In Case 1, Case 2 plus - Identify the layer content (whether or not it contains features above terrain height, buildings for example).	Functional Requirements as In Case 1, Case 2 plus - Identify the layer content (whether or not it contains features above terrain height, buildings for example).

Table A.1: IISP functional requirements (cont.)

Case **Function**	**Case1**	**Case2**	**Case3**	**Case4**
Pointing query function	**<u>Image-Vector Layer Linkage, Query Process and Attribute Display</u>** - Link the desired layer to the image; - Warn the user in case there is a connection/linkage error; - Change the cursor to a cross symbol for precise pointing to an image feature; - Search the database; - Display the image pixels related to that feature in a certain way; - obtain user confirmation; - Retrieve the attributes of the image feature; - In case of no results, due to geometric/georeferencing shift, warn the user and allow for search circle usage.	Functional Requirements as In Case 1 plus Based on the design methodology - Convert the point location from the image domain to the vector layer domain when required; - Convert the resulting feature(s) from the vector layer domain to the image domain when required; - Convert the user input search radius into degree in case of using geographic coordinate systems, when required.	Functional Requirements as In Case 1, Case 2 plus - Imaging geometry and relief displacement correction; - Correction for feature's height above ground.	Functional Requirements as In Case 1, Case 2 plus - Systematic error correction; - Tilt and relief displacement correction; - Correction for feature's height above ground.

Table A.1: IISP functional requirements (cont.)

Case Function	Case 1	Case 2	Case 3	Case 4
Pointing query function (continue)	**<u>Search Circle Usage</u>** - Search circle radius input; - Search the database within the search circle; - Indicate/display the resulting feature(s) and allow for user confirmation **<u>No Feature Found</u>** - Allow the user to expand the search circle; - In case of having void (null) results, warn the user of the need for updating the feature within the database/layer;			

Table A.1: IISP functional requirements (cont.)

Case **Function**	**Case 1**	**Case 2**	**Case 3**	**Case 4**
Polygon/Line Based Image Query	**<u>Layers Selection and Query Method</u>** - Select to localize the image query by a user-defined polygon/line; - Select the layers to be involved in the query. **<u>Linkage/Connection and Query Process</u>** - Connect/link to the desired/selected layers; - Warn the user in case of having connection/linkage error; - Allow the user to draw the desired polygon/line; - Allow the user to perform a search criterion within the drawn polygon/line; - Search the selected layers within the drawn polygon/line using the search criterion selected; - Allow the user to input a search expression within the results;	Functional Requirements as In Case 1 plus Based on the design methodology - Convert the drawn polygon/line from the image domain to the vector layer(s) domain when required; - Convert the resulting feature(s) from vector layer(s) domain to the image domain when required.	Functional Requirements as In Case 1, Case 2 plus - Imaging geometry and relief displacement correction.	Functional Requirements as In Case 1, Case 2 plus - Systematic error correction; - Tilt and relief displacement correction;

Table A.1: IISP functional requirements (cont.)

Case / Function	Case1	Case 2	Case 3	Case 4
Polygon/Line Based Image Query (continue)	**<u>Results Display</u>** - Display the result information; - Warn the user if any of the selected layers has no results; - Allow the user to display all the resulting features at once; - Allow the user to display a certain feature; - Zoom to a certain resulting feature; - Pan to a certain resulting feature; - Save the results as a shape file for subsequent usage.		Functional Requirements as In Case 1, Case 2 plus - In case of result displays, we have to consider the conversion from ground to screen coordinate systems and involve any required rectification.	Functional Requirements as In Case 1, Case 2 plus - In case of result displays, we have to consider the conversion from ground to screen coordinate systems and involve any required rectification.

Table A.1: IISP functional requirements (cont.)

Case / Function	Case 1	Case 2	Case 3	Case 4
Database Query	**<u>Vector Layers and Query Method Selection</u>** - Select the layer to be involved in the query; - Warn the user in case there are no layers or more than one layer is selected for the query. **<u>Linkage/Connection and Query</u>** - Connect/link to the desired/selected layer; - Warn the user in case of connection/linkage error; - Build a query tool allowing the user to input the query expression as well as to execute the query; - Display the selected layer attributes; - Allow the user to enter a search expression manually; - Allow the user to build a search expression;	Functional Requirements as In Case 1 and Based on the design methodology - Convert the resulting feature(s) from vector layer(s) domain to the image domain when required.	Functional Requirements as In Case 1, Case 2 plus	Functional Requirements as In Case 1, Case 2 plus

Table A.1: IISP functional requirements (cont.)

Case Function	Case 1	Case 2	Case 3	Case 4
Database Query (continue)	- Search the selected layer using the search expression; - Filter the results to obtain the features located within the image extent only. **<u>Results Display</u>** -The same as in the previous function.		- In case of result display, we have to consider the conversion from ground to screen coordinate systems and involve any required rectification.	- In case of result displays, we have to consider the conversion from ground to screen coordinate systems and involve any required rectification.

Table A.1: IISP functional requirements (cont.)

Case Function	Case 1	Case 2	Case 3	Case 4
Buffering Operation	**<u>Buffering Feature Creation</u>** - Create the buffering core shape using an existing image feature, a group of features stored in file, or user defined/drawn shape; - In case of a linear feature selected to be the buffering feature, connect the feature segments **<u>Pre-query parameters input, Layers Selection, Linkage/Connection and Query Process</u>** - Allow buffering distance input; - Allow the user to input a search expression and use it for a quick check for the existence of a certain feature satisfying that search expression; - Select the layers to be involved in the query; - Connect/link to the desired/selected layers;	Functional Requirements as In Case 1 and Based on the design methodology - Convert the buffering features from the layer domain to the image domain when required; - Convert the created buffering zone from the image domain to the vector layer(s) domain; - Convert the resulting feature(s) from the vector layer(s) domain to the image domain when required; - Convert the user input search radius and buffering distance into degree in case of using geographic coordinate systems.	Functional Requirements as In Case 1, Case 2 plus - Imaging geometry and relief displacement correction;	Functional Requirements as In Case 1, Case 2 plus - Systematic error correction; - Tilt and relief displacement correction;

Table A.1: IISP functional requirements (Cont.)

Case / Function	Case 1	Case 2	Case 3	Case 4
Buffering Operation (continue)	- Warn the user in case of connection/linkage error; - Warn the user if no buffering distance is specified; - Allow the user to search the layer within the buffering zone to retrieve all features or just features that match a specific search expression; **<u>Results Display</u>** The same as in the previous function.		- In case of result displays, we have to consider the conversion from ground to screen coordinate systems and involve any required rectification.	- In case of result displays, we have to consider the conversion from ground to screen coordinate systems and involve any required rectification.

APPENDIX B

IISP TESTING DATA

CASE 1

- Image Data

Image Original File Name: 46456470

Type: Orthophoto

Source: SNB Orthophotomap database

Format: TIFF image with TIFF WORLD FILE (.tfw)

Image Used: Sub-scene

Image Used Projection/Coordinate system: (Georeferenced to "ST/ATS 77")

Scale: 1:10 000

Pixel Size: 1 metre

- Vector Layers

Buildings, Water, Property parcels, Road net work vector layers were extracted from SNB topographic Sheet database and converted to ESRI Shape File

Sheet #: 46456470

Format: CARIS NTX FILE

Last updated: 1996

Scale: 1: 10 000

Figure B.1 shows the header file containing the original topographic sheet parameters.

```
==================== Header======================
1. Title  New Brunswick Digital Topographic Database, ETB96
[7MADHH]
2. File ID        46456470       3. Horizontal coord system
NEMR
4. Header length           198    5. Vertical coord system MR
6. Descriptor length        16    7. Sounding, Spot Ht units MR,M1
8. Coordinate resolutions         9. Coordinate shifts
XY            1.0000000000      X     0.000000000
Y             0.000000000
Z             0.1000000000      Z     0.000000000
10. Projection            ST    11. Central meridian   66-30-00.000W
12. Scale            10000.00   13. Scaling lat 1       46-30-00.000N
14. Scaling factor   0.999912   15. Scaling Lat 2  N/A
16. Ellipsoid            AT77   17. Vertical datum    MSL
18. N/A                      19. N/A
20. Graphic extent (429820,794548,440807,802184) (system)
21. Neatline corners (metres)     21. Neatline corners Lat,Long
E=    430599.000 N=    795847.000    46-26-59.988N   64-47-59.998W
E=    438281.000 N=    796018.000    46-27-00.015N   64-41-59.978W
E=    438154.000 N=    801574.000    46-29-59.991N   64-41-59.996W
E=    430479.000 N=    801404.000    46-29-59.993N   64-48-00.017W
22. Format ID                5    23. Last edited    29-SEP-1996
12:51
24. False North  800000.000       25. False East   300000.000
disphead completed

Copyright (c) 2000, Universal Systems Ltd., Fredericton, N.B. Canada.
```

Figure B.1
"46456470" SNB original topographic sheet parameters.

CASE 2

- Image Data

Image Original File Name: 46106470

Type: Orthophoto

Source: SNB Orthophotomap database

Format: MrSID Compressed format

Image Used: Full Image

Image Used Projection/Coordinate system: ST/NAD83

Scale: 1:10 000

Pixel Size: 1 metre

- Vector Layers

- 1:10 000 building layer with a Stereographic projection and ATS 77 datum;
- 1:10 000 river layer with a UTM zone 20N projection and ATS 77 datum;
- 1:10 000 land cover layer with a UTM zone 20N projection and NAD 83 datum;
- 1:1000 000 road network layer with a NAD 27 datum.

Buildings, River, and Land cover vector layers were extracted from SNB topographic Sheet database reprojectd in CARIS (if required) and converted to ESRI Shape Files

Sheet #: 46106470

Format: CARIS NTX FILE

Last updated: 1996

Scale: 1: 10 000

Figure B.2 shows the header file containing the original topographic sheet parameters.

The road network was obtained from Statistics Canada Digital Map Library. The original map scale is 1:1000 000.

```
========================Header========================
1. Title  New Brunswick Digital Topographic Database, ETB96 [7GS22W]
2. File ID         46106470        3. Horizontal coord system NEMR
4. Header length              198    5. Vertical coord system MR
6. Descriptor length          16    7. Sounding, Spot Ht units MR,M1
8. Coordinate resolutions          9. Coordinate shifts
   XY             1.0000000000        X                         0.000000000
    Y             0.00000000          Z                         0.1000000000
    Z            0.000000000
10. Projection                ST     11. Central meridian 66-30-00.000W
12. Scale              10000.00     13. Scaling lat 1        46-30-00.000N
14. Scaling factor      0.999912  15. Scaling Lat 2       N/A
16. Ellipsoid                AT77     17. Vertical datum    MSL
18. N/A                              19. N/A
20. Graphic extent (431512,756738,439489,761868) (system)
21. Neatline corners (metres)      21. Neatline corners Lat,Long
E=     431435.000 N=     756951.000     46-06-00.010N    64-47-59.986W
E=     439165.000 N=     757121.000     46-05-59.987N    64-42-00.021W
E=     439039.000 N=     762678.000     46-09-00.003N    64-42-00.009W
E=     431315.000 N=     762507.000     46-08-59.992N    64-48-00.022W
22. Format ID                5    23. Last edited    30-JAN-1997    14:17
24. False North    800000.000        25. False East   300000.000

Copyright (c) 2000, Universal Systems Ltd., Fredericton, N.B. Canada.
```

Figure B.2
"46106470" SNB original topographic sheet parameters.

CASE 3

- Image Data

Image Original File Name: po_43990_pan_0010100.tif

Type: IKONOS GEOCARTERRA™

Source: SPACE IMAGING

Format: GEOTIFF

Image Used: Sub-Scene

Image Used Projection/Coordinate system: UTM 19N/WGS84

Pixel Size: 1 metre

List B.1 shows the header file containing the original IKONOS imagery metadata file parameters.

- Vector Layers

- Two 1:10 000 building and road layers with NB. Stereographic projection and ATS 77 datum obtained from SNB;
- Two 1:10 000 building and road layers with a UTM zone 19N projection and WGS84 datum obtained from SNB;
- 1:250 000 land cover layer with a UTM NAD 27 coordinate system obtained from Statistics Canada on-line digital map library.

Buildings and road layers were extracted from SNB topographic Sheet database reprojectd, if needed, in CARIS and converted to ESRI Shape Files

Sheets #: 45906660, 45956660

Format: CARIS NTX FILE

Last updated: 1996

Scale: 1: 10 000

Figure (B.3) shows the header file containing the original "45956660" topographic sheet parameters. The land cover layer was obtained from Statistics Canada Digital Map Library. The original map scale is 1:250 000.

```
=================Header ====================
 1. Title  New Brunswick Digital Topographic Database, ETB96
[7EFHSE]
 2. File ID        45956660      3. Horizontal coord system      NEMR
 4. Header length          198   5. Vertical coord system        MR
 6. Descriptor length       16   7. Sounding, Spot Ht units      MR,M1
 8. Coordinate resolutions        9. Coordinate shifts
   XY          1.0000000000     X                0.000000000
    Y               0.000000000
    Z          0.1000000000     Z                0.000000000
 10. Projection             ST   11. Central meridian  66-30-00.000W
 12. Scale            10000.00   13. Scaling lat 1     46-30-00.000N
 14. Scaling factor   0.999912   15. Scaling Lat 2             N/A
 16. Ellipsoid            AT77   17. Vertical datum            MSL
 18. N/A                         19. N/A
 20. Graphic extent (283865,738180,293948,744851) (system)
 21. Neatline corners (metres)    21. Neatline corners Lat,Long
 E=    284494.000 N=    738889.000    45-57-00.005N   66-42-00.019W
 E=    292247.000 N=    738874.000    45-56-59.994N   66-36-00.009W
 E=    292254.000 N=    744431.000    45-59-59.989N   66-36-00.010W
 E=    284508.000 N=    744446.000    46-00-00.000N   66-42-00.020W
 22. Format ID               5   23. Last edited    30-JAN-1997 10:09
 24. False North   800000.000     25. False East   300000.000

Copyright (c) 2000, Universal Systems Ltd., Fredericton, N.B. Canada.
```

Figure B.3
"45956660" SNB original topographic sheet parameters.

==============================List B.1========================

Company Information
    Address
        Space Imaging
          12076 Grant Street
        Thornton, Colorado 80241
        U.S.A.
    Contact Information
          On the Web: http://www.spaceimaging.com
          Customer Service Phone (U.S.A.): 1.800.232.9037
          Customer Service Phone (World Wide): 301.552.0537
          Customer Service Fax (World Wide): 301.552.3762
          Customer Service Center hours of operation:
Monday - Friday, 7:00am - 11:00pm Eastern Standard Time

===

Product Order Metadata

Product Order Number: 43990
Customer Project Name: Fredericton
Product Order Area (Geographic Coordinates)
  Number of Coordinates: 4
  Coordinate: 1
    Latitude: 45.86800000 degrees
    Longitude: -66.78500000 degrees
  Coordinate: 2
    Latitude: 46.02300000 degrees
    Longitude: -66.78500000 degrees
  Coordinate: 3
    Latitude: 46.02300000 degrees
    Longitude: -66.54800000 degrees
  Coordinate: 4
    Latitude: 45.86800000 degrees
    Longitude: -66.54800000 degrees
Product Order Area (Map Coordinates)
  Coordinate: 1
    Map X (Easting): 671918.73 meters
    Map Y (Northing): 5081767.26 meters
  Coordinate: 2
    Map X (Easting): 671440.03 meters
    Map Y (Northing): 5098988.25 meters
  Coordinate: 3
    Map X (Easting): 689783.36 meters
    Map Y (Northing): 5099526.16 meters
  Coordinate: 4
    Map X (Easting): 690313.33 meters

Map Y (Northing): 5082305.27 meters
Sensor Type: Satellite
Processing Level: Standard Geometrically Corrected
Image Type: PAN
Interpolator Method: Bicubic
Multispectral Algorithm: None
Stereo: Mono
Mosaic: No
Map Projection: Universal Transverse Mercator
UTM Specific Parameters
Hemisphere: N
Zone Number: 19
Datum: WGS84
Product Order Pixel Size: 1.00 meters
MTFC Applied: Yes
DRA Applied: No
Media: CD
File Format: GeoTIFF
TIFF Tiled: No
Bits per Pixel per Band: 11 bits per pixel
Special Instructions: NA

==

Source Image Metadata

Number of Source Images: 2
Source Image ID: 2000072115010500000011630856
Product Image ID: 000
Sensor: IKONOS-2
Acquired Nominal GSD
Cross Scan: 0.94 meters
Along Scan: 0.88 meters
Scan Direction: 0 degrees
Nominal Collection Azimuth: 102.0119 degrees
Nominal Collection Elevation: 68.67448 degrees
Sun Angle Azimuth: 135.4853 degrees
Sun Angle Elevation: 58.25407 degrees
Acquisition Date/Time: 2000-07-21 15:01
Source Image ID: 2000072115005200000011630855
Product Image ID: 001
Sensor: IKONOS-2
Acquired Nominal GSD
Cross Scan: 0.95 meters
Along Scan: 0.88 meters
Scan Direction: 0 degrees

Nominal Collection Azimuth: 80.8307 degrees
Nominal Collection Elevation: 68.17849 degrees
Sun Angle Azimuth: 135.6210 degrees
Sun Angle Elevation: 58.29785 degrees
Acquisition Date/Time: 2000-07-21 15:00

==

Product Space Metadata

Number of Image Tiles: 3
  X Tiles: 2
  Y Tiles: 1
Product MBR Geographic Coordinates
  Number of Coordinates: 4
  Coordinate: 1
    Latitude: 45.86811842 degrees
    Longitude: -66.79116293 degrees
  Coordinate: 2
    Latitude: 46.02783771 degrees
    Longitude: -66.78480668 degrees
  Coordinate: 3
    Latitude: 46.02285247 degrees
    Longitude: -66.54113718 degrees
  Coordinate: 4
    Latitude: 45.86316073 degrees
    Longitude: -66.54819099 degrees
Product Map Coordinates
  UL Map X (Easting): 671440.03 meters
  UL Map Y (Northing): 5099526.16 meters
Pixel Size X: 1.00 meters
Pixel Size Y: 1.00 meters
Columns: 18876 pixels
Rows: 17760 pixels

==

Product Component Metadata

Number of Components: 3
Tile ID: 0000000
Product Image ID: 000
Tile File Name: po_43990_pan_0000000.tif
Tile Geographic Corner Coordinates
  Number of Coordinates: 4
  Coordinate: 1

Latitude: 45.86811842 degrees
Longitude: -66.79116293 degrees
Coordinate: 2
Latitude: 46.02783771 degrees
Longitude: -66.78480668 degrees
Coordinate: 3
Latitude: 46.02540113 degrees
Longitude: -66.66253143 degrees
Coordinate: 4
Latitude: 45.86569530 degrees
Longitude: -66.66923778 degrees
Tile Map Coordinates
UL Map X (Easting): 671440.03 meters
UL Map Y (Northing): 5099526.16 meters
Pixel Size X: 1.00 meters
Pixel Size Y: 1.00 meters
Columns: 9472 pixels
Rows: 17760 pixels
Tile ID: 0010000
Product Image ID: 001
Tile File Name: po_43990_pan_0010000.tif
Tile Geographic Corner Coordinates
Number of Coordinates: 4
Coordinate: 1
Latitude: 45.86811842 degrees
Longitude: -66.79116293 degrees
Coordinate: 2
Latitude: 46.02783771 degrees
Longitude: -66.78480668 degrees
Coordinate: 3
Latitude: 46.02540113 degrees
Longitude: -66.66253143 degrees
Coordinate: 4
Latitude: 45.86569530 degrees
Longitude: -66.66923778 degrees
Tile Map Coordinates
UL Map X (Easting): 671440.03 meters
UL Map Y (Northing): 5099526.16 meters
Pixel Size X: 1.00 meters
Pixel Size Y: 1.00 meters
Columns: 9472 pixels
Rows: 17760 pixels
Tile ID: 0010100
Product Image ID: 001
Tile File Name: po_43990_pan_0010100.tif
Tile Geographic Corner Coordinates

Number of Coordinates: 4
Coordinate: 1
Latitude: 45.86569504 degrees
Longitude: -66.66922490 degrees
Coordinate: 2
Latitude: 46.02540086 degrees
Longitude: -66.66251852 degrees
Coordinate: 3
Latitude: 46.02285247 degrees
Longitude: -66.54113718 degrees
Coordinate: 4
Latitude: 45.86316073 degrees
Longitude: -66.54819099 degrees
Tile Map Coordinates
UL Map X (Easting): 680912.03 meters
UL Map Y (Northing): 5099526.16 meters
Pixel Size X: 1.00 meters
Pixel Size Y: 1.00 meters
Columns: 9404 pixels
Rows: 17760 pixels

==============================END==========================

Appendix C

DATA AND RESULTS FOR THE IISP CASE 3 (IKONOS GEOCARTERRA™) TESTING

This appendix presents the data as well as the results obtained from the testing procedures for the different techniques and methods accomplished towards modeling the errors exist within the IKONOS GEOCARTERRA™ imagery and consequently perform the required on-the-fly orthorectification procedure within IISP Case 3.

TESTING PROCEDURE AND RESULTS

In order to test the developed method, the following datasets were used:

1- Fredericton IKONOS GEOCARETERRA image:

This image was ordered by the City of Fredericton to cover the area of Fredericton and its vicinity. The panchromatic image tile used has approximately 68.7849 degrees Nominal Collection Elevation angle and 80.8307 Nominal Collection Azimuth with GSD of 1 metre and has a UTM (19N) WGS84 projected coordinate system. The whole image tile is 320 MB. For the purpose of testing and implementation of the developed method, a sub-scene (3.7 x 6.5 Km) was extracted from this tile representing the extreme elevation variations within the Fredericton area (0 – 122 metres).

2- GCPs

Road intersections and building feature corners identifiable within the image were selected. Their image and ground coordinates were extracted from vector topographic sheets provided by SNB. The SNB data are in the NB. Sterographic ATS77 projected coordinate system. In order to identify the GCPs ground location to be used in transforming the image points, these topographic sheets were reprojected to UTM (19N) WGS84 using CARIS software. The Z coordinates of these GCPs were obtained by consulting the DTM provided by SNB for this area.

3- Check Points

In order to assess the accuracy of the developed method, 20 ground check points were used. These check points are selected to be well distributed across the entire image and represent the full range of elevations that exit within the area. The actual X and Y ground/check control point coordinates were obtained from the SNB topographic sheets as well.

The testing procedure at that stage was intended to measure the accuracy of the developed method by measuring the differences between the coordinates of ground features obtained by transforming and then applying relief correction (using the developed method) and their actual ground/map coordinates. Since the IISP system is basically concerned with how far the actual ground feature is from the transformed one, the direct distance between the actual and transformed point should be considered in assessing the accuracy. Consequently, the Horizontal Error (HE) was the basic concern in that testing stage. It is important to note that, from the IISP perspective, an accurate method should produce *high and consistent* accuracy across the whole image, so a small and consistent-radius search circle would retrieve the desired/correct feature.

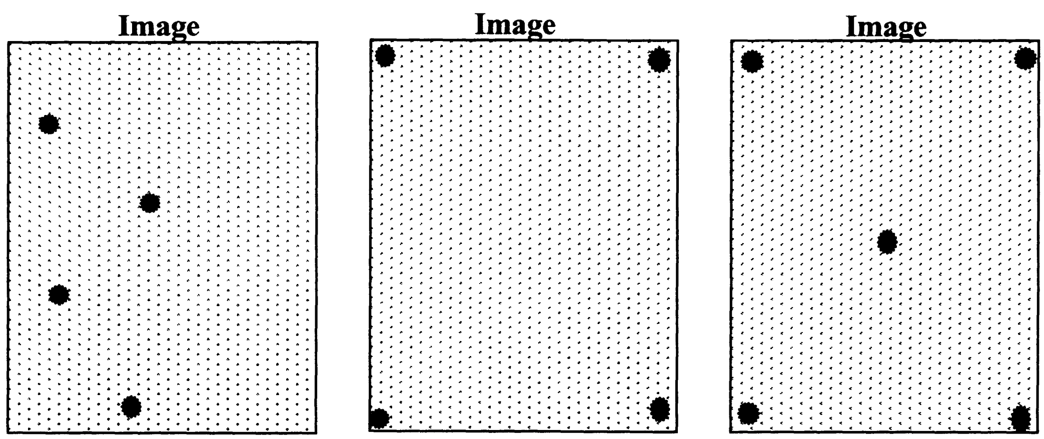
Testing: Step 1

The first step in this testing procedure was to examine the use of pure transformation to the image pixels. Affine transformation with a different collection of GCPs was used in this step. Three groups of GCPs were used to perform the transformation as shown in Figure C.1, namely:

1- 4 MIX. GCPs representing 4 different heights within the image.

2- The 4 image corners.

3- The 4 image corners and a middle point.



4 MIX. GCP's **4 Corners GCP's** **4 Corners and A Middle**

Figure C.1
GCP's distribution.

As shown in Figure C.2, the following results were obtained:

1- The accuracy obtained from the transformation was not consistent across the whole image. If a pure affine transformation is applied, in some areas a search circle with a 10 metre radius may be used while in other areas of the image a 60 metres search circle radius should be used to retrieve the correct/desired feature.

2- There is virtually no difference in the resulting accuracy then when using affine transformation with 4 corners or 4 corners plus a middle point.

3- If a pure affine transformation is the selected solution, the best/economic method (as tested within Fredericton area image) would be to use the 4 corners as the GCPs which provides a HE range of approximately (4-21 metre).

Testing: Step 2

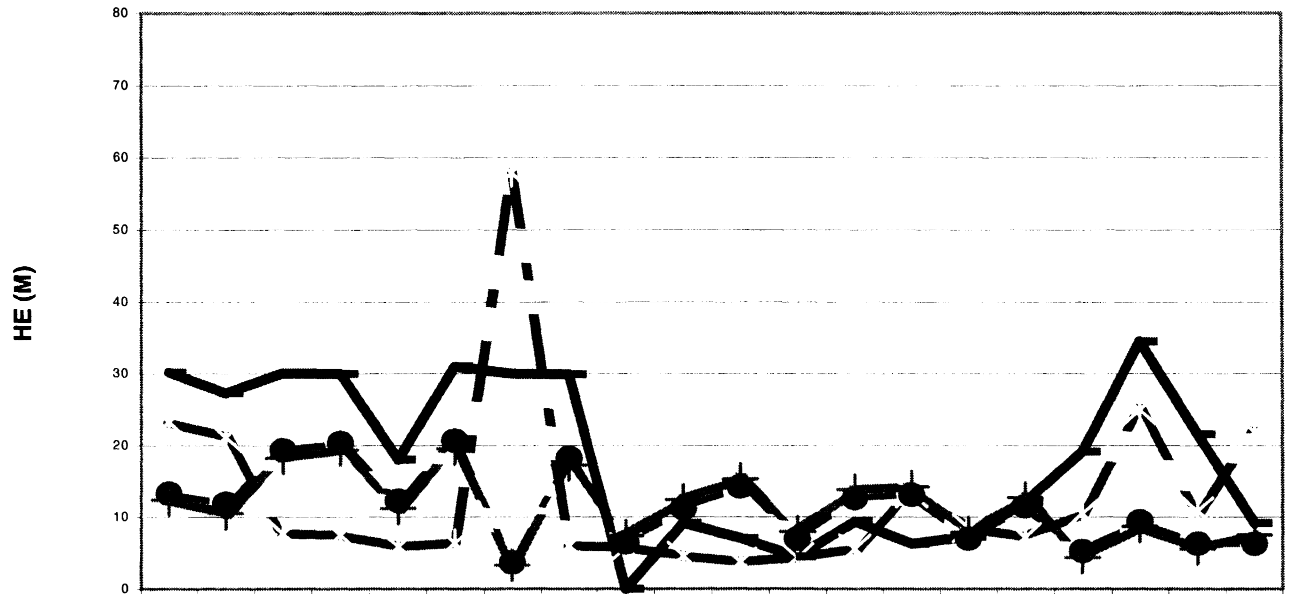
The next step in the testing procedure was to examine the developed method using affine transformation with 4 low elevation control points plus 3 methods for obtaining the height of the desired feature. The first method is to perform the transformation and use the transformed point to interpolate the height of the desired feature using DTM as outlined earlier. The second method is to use an average elevation for the whole area, where

$$\mathbf{h\ av.= [(h\ max. - h\ GCPs)/2]}$$

h max.: the maximum height within the area under considerations

h GCPs: average GCPs heights

The third method assumed that a limited number of full ground control points were available within the area under consideration. The desired point is first transformed and then an inverse distance interpolation can be used to interpolate the desired point elevation (approximate h) utilizing these existing GCPs. In this research, 15 full ground control points distributed across the image were used, as shown in Figure C.3.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
AFFINE 4 MIX POINTS	23.3363	21.1983	7.75692	7.50886	5.89428	6.45179	57.9009	6.06752	5.82667	4.62062	3.8491	4.4409	5.60063	13.3921	8.5053	7.25366	10.5073	24.9543	10.8539	21.9034
AFFINE 4 CORNERS	13.35	11.978	19.3339	20.4123	12.332	20.6431	3.75645	18.2868	6.51488	11.2698	14.2501	7.00903	12.6964	13.1481	6.97934	11.543	5.36451	9.46874	6.32084	6.30487
AFFINE 4 CORNERS & MIDDLE	12.4338	10.5298	18.3084	19.363	11.2004	19.6145	3.32515	17.279	7.46856	12.4346	15.3307	8.02701	13.8361	14.2264	7.93689	12.7552	4.43452	8.78083	5.61963	7.56227
Dx/Dy Original (IMAGE -MAP)	30.1862	27.3493	30.073	30.0195	18.0678	30.9892	29.978	29.9168	0	9.27573	7.15631	4.49862	9.45431	6.35446	7.32094	12.1228	19.1989	34.5669	21.6261	9.28218

CHECK POINT NUMBER

AFFINE 4 MIX POINTS — AFFINE 4 CORNERS — AFFINE 4 CORNERS & MIDDLE — Dx/Dy Original (IMAGE -MAP)

Figure C.2
Testing step 1: affine transformation only.

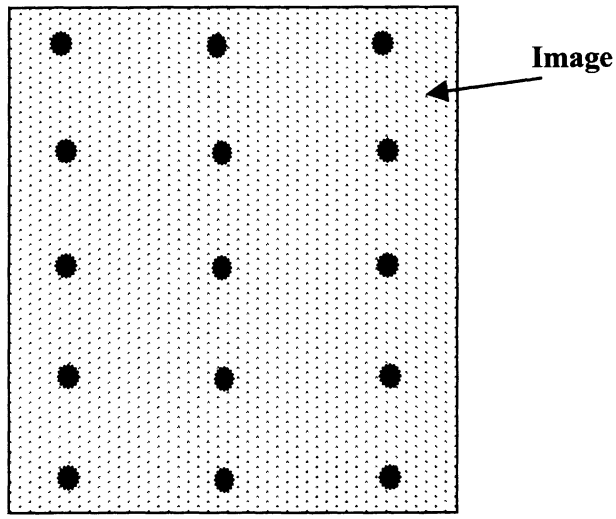
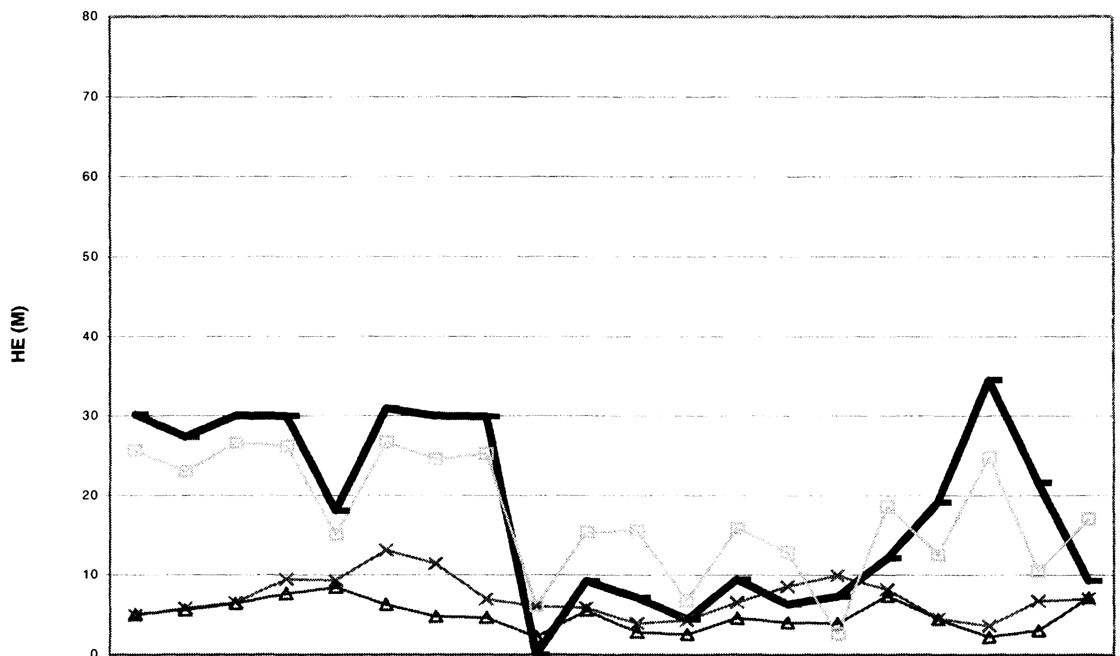


Figure C.3
Approximate interpolation GCP's distribution.

As shown in Figure C.4, when using the same control points (4 low GCPs), the method of elevation interpolation using DTM significantly affected the resulting accuracy across the entire image. The following can be observed:

1- Using a DTM elevation interpolation as outlined earlier, following an affine transformation with 4 low elevation GCPs produces a better and consistent accuracy across the entire image (HE 2 ~ 8.5 metres) as compared with the original accuracy (0 ~ 35 metres).

2- If no DTM is available, the inverse distance interpolation method for the desired point elevation using the available full GCPs, produces reasonably consistent



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
DEVELOPED METHOD WITH AFFINE (4 lOW POINTS WITH APPROX. h)	4.9524	5.808	6.5571	9.5095	9.358	13.129	11.448	6.9908	6.1291	5.9203	3.9045	4.3466	6.5554	8.6233	9.9789	8.2025	4.5492	3.6309	6.7545	7.0148
Dx/Dy Original (IMAGE -MAP)	30.186	27.349	30.073	30.02	18.068	30.989	29.978	29.917	0	9.2757	7.1563	4.4986	9.4543	6.3545	7.3209	12.123	19.199	34.567	21.626	9.2822
DEVELOPED METHOD WITH AFFINE (4 lOW POINTS WITH DTM)	5.0839	5.612	6.4264	7.6297	8.5007	6.302	4.8049	4.6747	2.2257	5.5479	2.8584	2.5529	4.6297	4.0816	3.9262	7.3683	4.4513	2.2142	3.0335	7.1561
DEVELOPED METHOD WITH AFFINE (4 lOW POINTS WITH AV. h)	25.688	23.003	26.61	26.311	15.108	26.697	24.557	25.265	6.0664	15.417	15.558	6.8209	15.898	12.863	2.6558	18.608	12.536	24.831	10.499	17.057

CHECK POINT NUMBER

DEVELOPED METHOD WITH AFFINE (4 lOW POINTS WITH APPROX. h) — Dx/Dy Original (IMAGE -MAP)
DEVELOPED METHOD WITH AFFINE (4 lOW POINTS WITH DTM) — DEVELOPED METHOD WITH AFFINE (4 lOW POINTS WITH AV. h)

Figure C.4
Testing step 2: developed method using affine transformation with 4 low GCPs and different elevation estimation method.

accuracy (HE 4 ~ 13 metres). It is important to note that a threshold has been used with this method. We should first identify the distance of the closest GCP to the desired point (d1) and then include all the other GCPs that have a distance (d) to the desired point within a specific range (d1 < d >= 2*d1) in the interpolation procedure.

3- The accuracy using an average elevation across the image produces accurate results within areas where this average elevation matches the actual ground elevation. For example, in the Fredericton image, using GCPs with a 10-metre average elevation and knowing the maximum elevation within the image to be 120 metre, then the average elevation can be calculated as:

h av. = [(120-10)/2]=55 metres

Consequently, the best accuracy produced with this method will be in areas where the actual elevation is close to 55 metres. The distribution of the GCPs used within the affine transformation may still have an effect on the final accuracy. The HE range produced using this method is approximately (3 ~ 26 metres).

The method of using the affine transformation with 4 low elevation GCPs followed by a DTM interpolation elevation provides a consistent accuracy of (HE 2 ~ 8.5 metres).

Testing: Step 3

In order to prove the hypothesis that using 4 low elevation GCPs in performing the transformation is a necessity to reach the desired accuracy, this step dealt with using 3 different groups of GCPs in performing the transformation process followed by the desired point's elevation interpolation using a DTM, namely, (4 LOW GCPs), (4 MIX. GCPs) representing 4 different elevation within the image, and (4 HIGH GCPs)

As shown in Figure C.5, a significant difference in the magnitude of the resulting accuracy (represented by HE) was clearly apparent. The 4 LOW GCPs was the best method among the 3 methods utilized and the other two provide even worse accuracy if compared with the original image accuracy

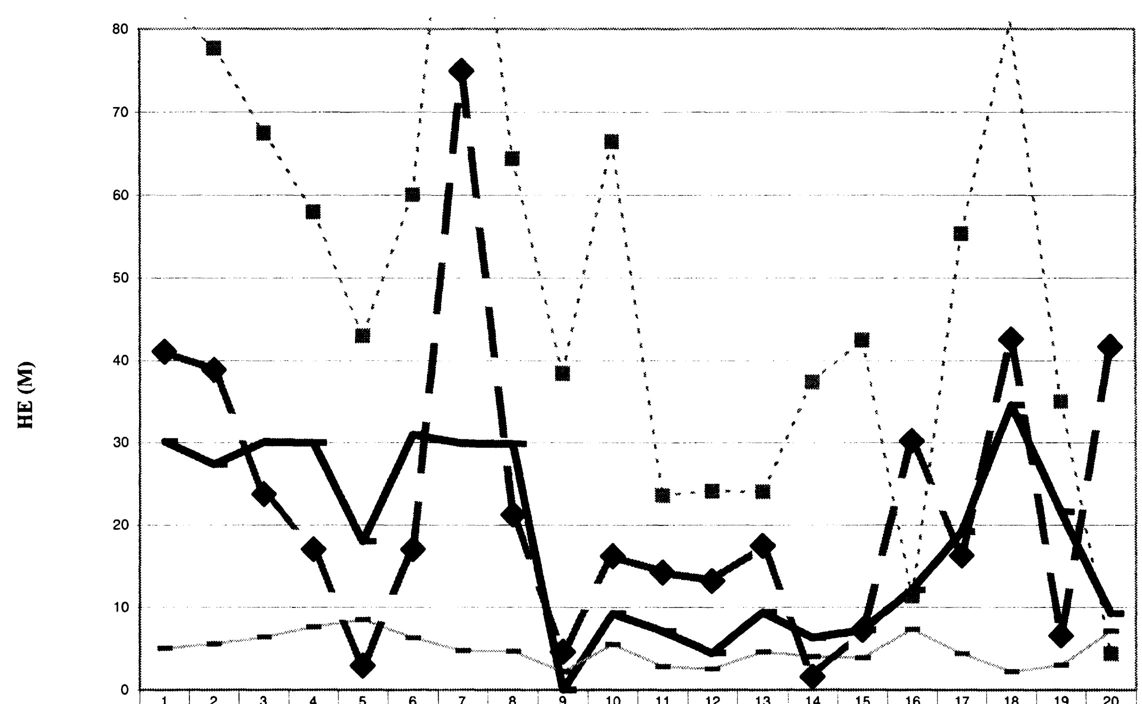
This test supports the previously mentioned hypothesis that using lower GCPs in eliminating the sensor error through an affine transformation will lead to an almost complete elimination of this error. Applying the affine transformation based upon higher elevation GCPs will eliminate the sensor error but introduce another error. This error comes mainly from two actions, namely:

1-By assuming that the new datum is an average of the GCPs elevation;

2-By trying to model and eliminate large magnitude of the relief displacements (GCPs existing relief) which is a function of "TAN (E)" and "SIN or COS (Φ)" using affine transformation.

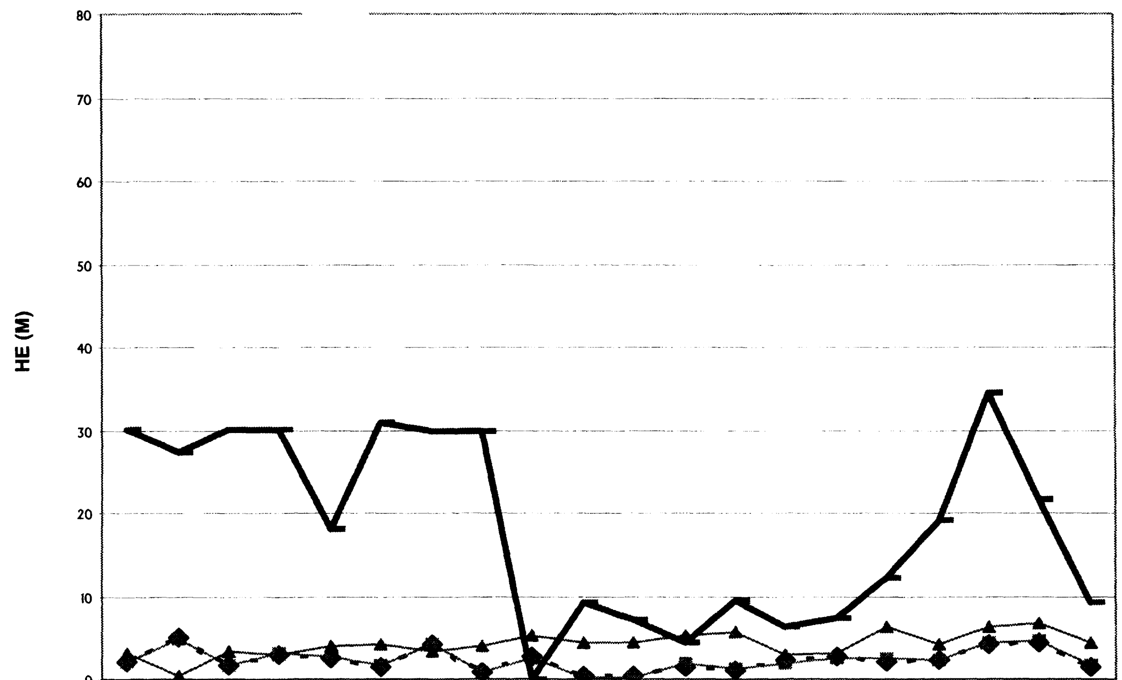
Testing: Step4

This step dealt with testing the consideration that the sensor error may be resolved into two main components, namely shifts in X and Y directions. Consequently, we can eliminate this error without applying any transformation and by adding or subtracting these two shift values from the feature's image coordinates. As shown in Figure C.6, this method was applied using only 3 LOW, MIX, and HIGH GCPs. The following can be observed:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Dx/Dy Original (IMAGE -MAP)	30.186	27.349	30.073	30.02	18.068	30.989	29.978	29.917	0	9.2757	7.1563	4.4986	9.4543	6.3545	7.3209	12.123	19.199	34.567	21.626	9.2822
DEVELOPED METHOD WITH AFFINE (4 IOW POINTS WITH DTM)	5.0839	5.612	6.4264	7.6297	8.5007	6.302	4.8049	4.6747	2.2257	5.5479	2.8584	2.5529	4.6297	4.0816	3.9262	7.3683	4.4513	2.2142	3.0335	7.1561
DEVELOPED METHOD WITH AFFINE (4 MIX POINTS DTM)	41.066	38.83	23.798	17.093	2.9468	17.082	75.027	21.263	4.6165	16.293	14.283	13.229	17.521	1.6366	7.25	30.212	16.356	42.576	6.5933	41.691
DEVELOPED METHOD WITH AFFINE (4 HIGH POINTS DTM)	84.041	77.72	67.5	58.027	42.992	60.026	116.83	64.384	38.39	66.476	23.621	24.155	24.098	37.401	42.45	11.433	55.381	80.767	35.061	4.4452

Figure C.5
Testing step 3: developed method using affine transformation and different groups of GCP's.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
DEVELOPED METHOD "AV. SHIFTxy" (3 MIX POINTS WITH DTM)	2.0656	5.1036	1.6183	3.0427	2.6262	1.4629	4.2202	0.8301	2.8169	0.4583	0.4729	1.5255	1.0387	2.3045	2.7928	2.1565	2.2275	4.1168	4.4332	1.551
DEVELOPED METHOD "AV. SHIFTxy" (3 HIGH POINTS WITH DTM)	2.0565	4.8779	1.5838	3.0578	2.7555	1.6358	4.2687	0.7802	2.534	0.244	0.1415	1.8351	1.2632	1.9875	2.4513	2.5072	2.3809	4.391	4.7238	1.7368
DEVELOPED METHOD "AV. SHIFTxy" (3 LOW POINTS WITH DTM)	3.2185	0.3981	3.3961	3.0318	4.0007	4.286	3.4601	3.9725	5.3022	4.3178	4.3992	5.187	5.7815	2.8897	3.0591	6.2369	4.1615	6.234	6.7076	4.3508
Dx/Dy Original (IMAGE-MAP)	30.186	27.349	30.073	30.02	18.068	30.989	29.978	29.917	0	9.2757	7.1563	4.4986	9.4543	6.3545	7.3209	12.123	19.199	34.567	21.626	9.2822

Figure C.6
Testing "step 4: developed average shift method"

1- Using this method with any of the 3 GCPs groups provided a high and consistent accuracy across the entire image.

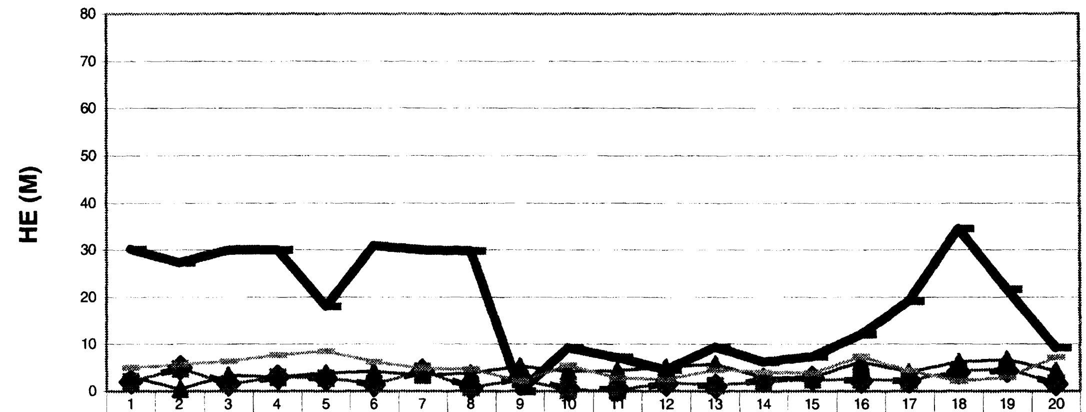
2- MIX and HIGH GCPs provided better accuracy when compared with LOW GCPs.

3- Using only 3 MIX GCPs combined with DTM elevation interpolation provided the best results.

4- As shown in Figure C.7, this method is more accurate than using 4 LOW GCPs affine transformation followed by DTM elevation interpolation mainly because the affine transformation parameters obtained are still affected by the position and the number of the selected GCPs. The transformed coordinates of any point are affected by the point position itself.

5- Using 3 MIX GCPs to calculate the average sensor X and Y shifts/errors followed by a DTM elevation interpolation of the desired point provides a consistent HE range of (0.5 – 5 metres). For standard comparison, this method will produce RMSEx of 1.73 metres, RMSEy of 2.13 meters, RMSE of 2.75 metres. This accuracy is approximately the same as the accuracy level obtained when purchasing the orthorectified IKONOS PRECISION products (RMSE of 1.9 metres). The latter cost approximately 4 times as much as the IKONOS GEOCARTERRA™ products.

For a global comparison, Figure C.8 shows the results of all the tested methods. The data and the processing results used for this testing are shown in Figures C.9 through C.16.



	1	2	3	4	5	6	7	8	9	10
DEVELOPED METHOD " AV. SHIFTxy" (3 MIX POINTS WITH DTM)	2.0656	5.1036	1.6183	3.0427	2.6262	1.4629	4.2202	0.8301	2.8169	0.4583
DEVELOPED METHOD " AV. SHIFTxy" (3 HIGH POINTS WITH DTM)	2.0565	4.8779	1.5838	3.0578	2.7555	1.6358	4.2687	0.7802	2.534	0.244
DEVELOPED METHOD " AV. SHIFTxy" (3 LOW POINTS WITH DTM)	3.2185	0.3981	3.3961	3.0318	4.0007	4.286	3.4601	3.9725	5.3022	4.3178
Dx/Dy Original (IMAGE -MAP)	30.186	27.349	30.073	30.02	18.068	30.989	29.978	29.917	0	9.2757
DEVELOPED METHOD WITH AFFINE (4 lOW POINTS WITH DTM)	5.0839	5.612	6.4264	7.6297	8.5007	6.302	4.8049	4.6747	2.2257	5.5479

	11	12	13	14	15	16	17	18	19	20
DEVELOPED METHOD " AV. SHIFTxy" (3 MIX POINTS WITH DTM)	0.4729	1.5255	1.0387	2.3045	2.7928	2.1565	2.2275	4.1168	4.4332	1.551
DEVELOPED METHOD " AV. SHIFTxy" (3 HIGH POINTS WITH DTM)	0.1415	1.8351	1.2632	1.9875	2.4513	2.5072	2.3809	4.391	4.7238	1.7368
DEVELOPED METHOD " AV. SHIFTxy" (3 LOW POINTS WITH DTM)	4.3992	5.187	5.7815	2.8897	3.0591	6.2369	4.1615	6.234	6.7076	4.3508
Dx/Dy Original (IMAGE -MAP)	7.1563	4.4986	9.4543	6.3545	7.3209	12.123	19.199	34.567	21.626	9.2822
DEVELOPED METHOD WITH AFFINE (4 lOW POINTS WITH DTM)	2.8584	2.5529	4.6297	4.0816	3.9262	7.3683	4.4513	2.2142	3.0335	7.1561

CHECK POINT NUMBER

DEVELOPED METHOD " AV. SHIFTxy" (3 MIX POINTS WITH DTM)
DEVELOPED METHOD " AV. SHIFTxy" (3 HIGH POINTS WITH DTM)
DEVELOPED METHOD " AV. SHIFTxy" (3 LOW POINTS WITH DTM)
Dx/Dy Original (IMAGE -MAP)
DEVELOPED METHOD WITH AFFINE (4 lOW POINTS WITH DTM)

Figure C.7
Accuracy comparison of using affine transformation vs. average shift in eliminating sensor error

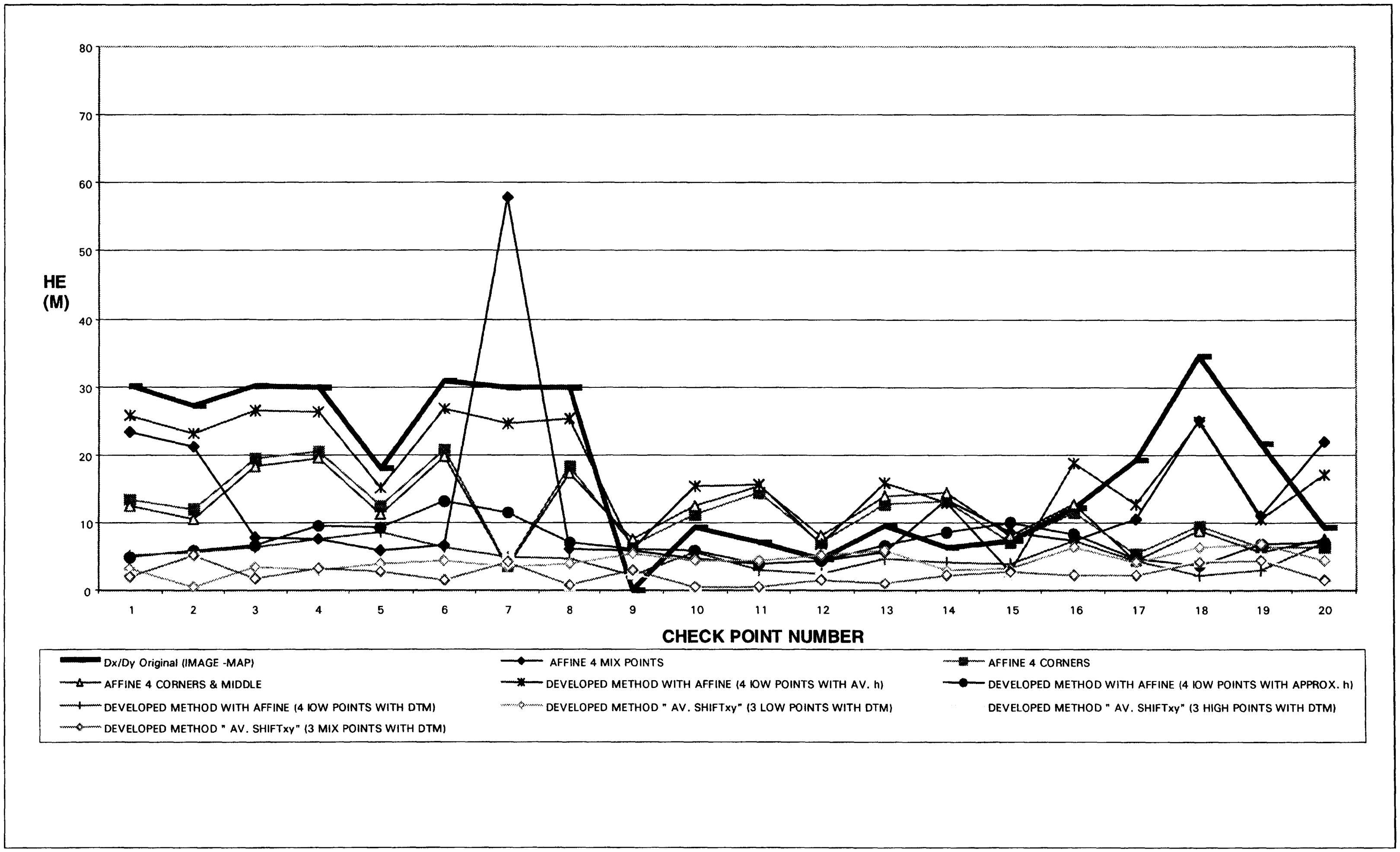


Figure C.8
IISP Case 3 testing results (all steps).

GCPs USED FOR TESTING		GCPs USED FOR AFFINE TRANSFORMATION				GCPs USED FOR AVERAGE SHIFT METHOD			
X/Y IMAGE	X/Y MAP	X/Y IMAGE	X/Y MAP	DTM INTERPOLATED HEIGHT		X/Y IMAGE	X/Y MAP	DTM INTERPOLATED HEIGHT	
682468.46	682439.51	681616.3	681590.85	10		682406.43	682381.94	8.5	
5092741.01	5092732.46	5092221.5	5092219.84			5092636.07	5092626.89		
682343.65	682318	684612.3	684583.57	11		682581.96	682554.39	9.5	3 LOW GCPs
5092530.31	5092520.82	5092964.11	5092954.5		4 LOW GCPs	5092535.08	5092525.85		
681804.57	681775.6	683048.49	683020.48	8		682343.19	682318	8.5	
5092422.11	5092414.04	5092127.61	5092120.83			5092530.91	5092520.77		
681815.49	681787.01	684682.46	684652.41	8.5					
5091998.34	5091988.85	5091667.18	5091658.07						
681515.58	681498.97								
5091462.58	5091455.47								
681986.32	681956.4								
5091900.59	5091892.52					683064.28	683067.06	89	
683372.76	683344.73	683064.28	683067.06	89		5089649.45	5089646.64		
5093773.99	5093763.36	5089649.45	5089646.64			683038.11	683041.38	90	3 HIGH GCPs
682159.9	682130.88	683038.11	683041.38	90		5089556.94	5089556.01		
5091921.19	5091913.92	5089556.94	5089556.01		4 HIGH GCPs	683125.95	683128.75	86	
681938.75	681938.75	681540.39	681536.41	80		5089567.22	5089563.95		
5090612.95	5090612.95	5091001.26	5090997.64						
681645.7	681654.96	683125.95	683128.75	86					
5090244.12	5090243.58	5089567.22	5089563.95						
682793.09	682800.21								
5089242	5089241.28								
683434.71	683431.61								
5088755.87	5088752.61								
681865.35	681874.79	681616.3	681590.85	10		683048.49	683020.48	8	
5090044.91	5090045.43	5092221.5	5092219.84			5092127.61	5092120.83		
682256.47	682262.63	681810.64	681793	40		681810.64	681793	40	3 MIX. GCPs
5090432.86	5090431.3	5091355.22	5091351		4 MIX. GCPs	5091355.22	5091351		
683525.6	683519.04	681540.39	681536.41	80		681540.39	681536.41	80	
5089589.01	5089585.76	5091001.26	5090997.67			5091001.26	5090997.67		
681523.23	681535.35	682660.47	682668.71	100					
5089750.1	5089749.84	5089158.39	5089155.31						
682861.36	682843.44								
5090951.1	5090944.21								
684423.22	684390.18								
5090990.58	5090980.42								
684141.32	684121.24	682008.8	681982.14						
5088818.99	5088810.96	5094186.19	5094177.86						
681675.6	681684.7	683966.4	683946.4	NO HEIGHT REQUIRED					
5088830.43	5088828.6	5093730.36	5093724.7		4 CORNERS GCPs				
		684628.81	684603.25						
		5088638.18	5088633.92						
		681675.6	681684.7						
		5088830.43	5088828.4						
		682008.8	681982.14						
		5094186.19	5094177.86						
		683966.4	683946.4						
		5093730.36	5093724.7						
		684628.81	684603.25	NO HEIGHT REQUIRED	4 CORNERS & MIDDLE GCPs				
		5088638.18	5088633.92						
		681675.6	681684.7						
		5088830.43	5088828.4						
		682859.35	682841.83						
		5090949.99	5090944.43						

Figure C.9
Testing Data.

AFFINE TRANSFORMATION (4MIX. GCPs)

	1	681616.3	5092221.5	0	0	0
	0	0	0	1	681616.3	5092221.5
	1	681810.64	5091355.22	0	0	0
A	0	0	0	1	681810.64	5091355.22
	1	681540.39	5091001.26	0	0	0
	0	0	0	1	681540.39	5091001.26
	1	682660.47	5089158.39	0	0	0
	0	0	0	1	682660.47	5089158.39

	1	0	1	0	1	0	1	0
	681616.3	0	681810.6	0	681540.39	0	682660.5	0
At	5092221.5	0	5091355	0	5091001.26	0	5089158	0
	0	1	0	1	0	1	0	1
	0	681616.3	0	681810.6	0	681540.4	0	682660.5
	0	5092221.5	0	5091355	0	5091001	0	5089158

	681590.85
	5092219.84
	681793
L	5091351
	681536.41
	5090997.67
	682668.71
	5089155.31

	4	2727627.8	20363736.37	0	0	0
	2727627.8	1.85999E+12	1.38862E+13	0	0	0
	20363736.37	1.38862E+13	1.0367E+14	0	0	0
At*A	0	0	0	4	2727628	20363736
	0	0	0	2727627.8	1.86E+12	1.39E+13
	0	0	0	20363736.37	1.39E+13	1.04E+14

	43333528.3	-15.2912194	-6.4637	0	0	0
(At*A)-1	-15.2912194	6.13027E-06	2.2E-06	0	0	0
	-6.46371347	2.1825E-06	9.8E-07	0	0	0
	0	0	0	43333528.31	-15.2912194	-6.463713472
	0	0	0	-15.29121941	6.13027E-06	2.1825E-06
	0	0	0	-6.46371347	2.1825E-06	9.77316E-07

	2727588.97		99751.3906
	1.85996E+12		0.98252817
	1.3886E+13	X	-0.0172556
At*L	20363723.82		-6721.25
	1.38862E+13		0.00215623
	1.0367E+14		1.0010308

	-1.39221
	-0.79416
	2.350465
V	1.291918
	-0.47999
	-0.28567
	-0.47008
	-0.28015

X/Y Image	X/Y Trans	X/Y MAP	Dx/DY
682468.46	682417.7646	682439.51	-21.7454399
5092741.01	5092740.929	5092732.46	8.468804268
682343.65	682298.771	682318	-19.229034
5092530.31	5092529.742	5092520.82	8.922495338
681804.57	681770.9767	681775.6	-4.62326921
5092422.11	5092420.269	5092414.04	6.228583187
681815.49	681789.0183	681787.01	2.008328485
5091998.34	5091996.085	5091988.85	7.235305923
681515.58	681503.5931	681498.97	4.623144879
5091462.58	5091459.126	5091455.47	3.656368767
681986.32	681958.5503	681956.4	2.150347494
5091900.59	5091898.603	5091892.52	6.082893356
683372.76	683288.4401	683344.73	-56.2898654
5093773.99	5093776.923	5093763.36	13.56347969
682159.9	682128.7421	682130.88	-2.13787694
5091921.19	5091919.598	5091913.92	5.678405919
681938.75	681934.0304	681938.75	-4.71956416
5090612.95	5090609.533	5090612.95	-3.41698122
681645.7	681652.4649	681654.96	-2.49507568
5090244.12	5090239.691	5090243.58	-3.88905476
682793.09	682797.1001	682800.21	-3.10993083
5089242	5089239.012	5089241.28	-2.26800854
683434.71	683435.8982	683431.61	4.28824215
5088755.87	5088753.764	5088752.61	1.154366228
681865.35	681871.7147	681874.79	-3.07528184
5090044.91	5090040.749	5090045.43	-4.68078553
682256.47	682249.3068	682262.63	-13.3231588
5090432.86	5090429.942	5090431.3	-1.35754178
683525.6	683510.8239	683519.04	-8.21607226
5089589.01	5089587.959	5089585.76	2.199148758
681523.23	681540.6593	681535.35	5.309292419
5089750.1	5089744.898	5089749.84	-4.94236514
682861.36	682834.6858	682843.44	-8.75421571
5090951.1	5090950.021	5090944.21	5.810942073
684423.22	684368.576	684390.18	-21.604014
5090990.58	5090992.909	5090980.42	12.48936416
684141.32	684129.0733	684121.24	7.833302919
5088818.99	5088818.473	5088810.96	7.513042641
681675.6	681706.2365	681684.7	21.53653394
5088830.43	5088824.608	5088828.6	-3.99181905

Figure C.10
Affine transformation with 4 MIX GCPs.

AFFINE TRANSFORMATION (4CORNERS GCPs)

A

1	682008.8	5094186	0	0	0
0	0	0	1	682008.8	5094186
1	683966.4	5093731	0	0	0
0	0	0	1	683966.4	5093731
1	684628.8	5088638	0	0	0
0	0	0	1	684628.8	5088638
1	681675.6	5088830	0	0	0
0	0	0	1	681675.6	5088830

At

1	0	1	0	1	0	1	0
682008.8	0	683966.4	0	684628.8	0	681675.6	0
5094186	0	5093731	0	5088638	0	5088830	0
0	1	0	1	0	1	0	1
0	682008.8	0	683966.4	0	684628.8	0	681675.6
0	5094186	0	5093731	0	5088638	0	5088830

L

681982.1
5094178
683946.4
5093725
684603.3
5088634
681684.7
5088828

At*A

4	2732280	20365386	0	0	0
2732280	1.87E+12	1.39E+13	0	0	0
20365386	1.39E+13	1.04E+14	0	0	0
0	0	0	4	2732280	20365386
0	0	0	2732280	1.87E+12	1.39E+13
0	0	0	20365386	1.39E+13	1.04E+14

(At*A)-1

1099492	-0.15747	-0.19483	0	0	0
-0.15747	1.61E-07	9.33E-09	0	0	0
-0.19483	9.33E-09	3.7E-08	0	0	0
0	0	0	1099492	-0.15747	-0.19483
0	0	0	-0.15747	1.61E-07	9.33E-09
0	0	0	-0.19483	9.33E-09	3.7E-08

AtL

2732216
1.87E+12
1.39E+13
20365365
1.39E+13
1.04E+14

X

21641.11
0.992491
-0.00325
4602.996
-0.00037
0.999144

V

9.629077
0.963108
-10.2531
-1.03255
6.867099
0.685879
-6.24268
-0.63

X/Y Image	X/Y Trans	X/Y MAP	Dx/DY
682468.46	682452.669	682439.51	13.159
5092741.01	5092734.71	5092732.46	2.25
682343.65	682329.48	682318	11.48
5092530.31	5092524.238	5092520.82	3.418
681804.57	681794.799	681775.6	19.199
5092422.11	5092416.32	5092414.04	2.28
681815.49	681807.013	681787.01	20.003
5091998.34	5091992.917	5091988.85	4.067
681515.58	681511.0941	681498.97	12.1241
5091462.58	5091457.725	5091455.47	2.255
681986.32	681976.87	681956.4	20.47
5091900.59	5091895.188	5091892.52	2.668
683372.76	683346.825	683344.73	2.095
5093773.99	5093766.478	5093763.36	3.118
682159.9	682149.08	682130.88	18.2
5091921.19	5091915.7	5091913.92	1.78
681938.75	681933.84	681938.75	-4.91
5090612.95	5090608.668	5090612.95	-4.282
681645.7	681644.19	681654.96	-10.77
5090244.12	5090240.261	5090243.58	-3.319
682793.09	682786.22	682800.21	-13.99
5089242	5089238.57	5089241.28	-2.71
683434.71	683424.601	683431.61	-7.009
5088755.87	5088752.63	5088752.61	0.02
681865.35	681862.84	681874.79	-11.95
5090044.91	5090041.141	5090045.43	-4.289
682256.47	682249.76	682262.63	-12.87
5090432.86	5090428.61	5090431.3	-2.69
683525.6	683512.1	683519.04	-6.94
5089589.01	5089585.02	5089585.76	-0.74
681523.23	681524.24	681535.35	-11.11
5089750.1	5089746.708	5089749.84	-3.132
682861.36	682848.4297	682843.44	4.9897
5090951.1	5090946.18	5090944.21	1.97
684423.22	684398.43	684390.18	8.25
5090990.58	5090985.067	5090980.42	4.647
684141.32	684125.7	684121.24	4.46
5088818.99	5088815.439	5088810.96	4.479
681675.6	681678.45	681684.7	-6.25
5088830.43	5088827.77	5088828.6	-0.83

Figure C.11
Affine transformation with 4 CORNERS GCPs.

AFFINE TRANSFORMATION (4CORNERS & Middle

	1	682008.8	5094186	0	0	0
	0	0	0	1	682008.8	5094186
	1	683966.4	5093731	0	0	0
A	0	0	0	1	683966.4	5093731
	1	684628.8	5088638	0	0	0
	0	0	0	1	684628.8	5088638
	1	681675.6	5088830	0	0	0
	0	0	0	1	681675.6	5088830
	1	682859.4	5090950	0	0	0
	0	0	0	1	682859.4	5090950

	1	0	1	0	1	0	1	0	1	0
	682008.8	0	683966.4	0	684628.8	0	681675.6	0	682859.4	0
At	5094186	0	5093731	0	5088638	0	5088830	0	5090950	0
	0	1	0	1	0	1	0	1	0	1
	0	682008.8	0	683966.4	0	684628.8	0	681675.6	0	682859.4
	0	5094186	0	5093731	0	5088638	0	5088830	0	5090950

	681982.1
	5094178
	683946.4
L	5093725
	684603.3
	5088634
	681684.7
	5088828
	682841.8
	5090944

	5	3415139	25456336	0	0	0
	3415139	2.33E+12	1.74E+13	0	0	0
	25456336	1.74E+13	1.3E+14	0	0	0
At*A	0	0	0	5	3415139	25456336
	0	0	0	3415139	2.33E+12	1.74E+13
	0	0	0	25456336	1.74E+13	1.3E+14

	1089807	-0.15418	-0.19337	0	0	0
	-0.15418	1.6E-07	8.84E-09	0	0	0
(At*A)-1	-0.19337	8.84E-09	3.68E-08	0	0	0
	0	0	0	1089807	-0.15418	-0.19337
	0	0	0	-0.15418	1.6E-07	8.84E-09
	0	0	0	-0.19337	8.84E-09	3.68E-08

	3415058
AtL	2.33E+12
	1.74E+13
	25456309
	1.74E+13
	1.3E+14

	21237.82
	0.992628
X	-0.00319
	4545.563
	-0.00035
	0.999153

	4.74568
	0.85005
	-1.0895
	-1.11133
V	6.006314
	0.576021
	-7.49642
	-0.79583
	3.645347
	0.528595

X/Y Image	X/Y Trans	X/Y MAP	Dx/DY
682468.46	682451.76	682439.51	12.25
5092741.01	5092734.59	5092732.46	2.13
682343.65	682328	682318	10
5092530.31	5092524.12	5092520.82	3.297999999
681804.57	681793.781	681775.6	18.181
5092422.11	5092416.2	5092414.04	2.156
681815.49	681805.97	681787.01	18.96
5091998.34	5091992.78	5091988.85	3.930000001
681515.58	681509.97	681498.97	11
5091462.58	5091457.58	5091455.47	2.109
681986.32	681975.85	681956.4	19.45
5091900.59	5091895.06	5091892.52	2.535
683372.76	683346.104	683344.73	1.374455413
5093773.99	5093766.39	5093763.36	3.027790675
682159.9	682148.08	682130.88	17.2
5091921.19	5091915.57	5091913.92	1.65
681938.75	681932.735	681938.75	-6.01472172
5090612.95	5090608.52	5090612.95	-4.42746951
681645.7	681643.02	681654.96	-11.9395293
5090244.12	5090240.11	5090243.58	-3.47356968
682793.09	682785.145	682800.21	-15.0651598
5089242	5089238.44	5089241.28	-2.84100248
683434.71	683423.584	683431.61	-8.02624872
5088755.87	5088752.5	5088752.61	-0.11085701
681865.35	681861.686	681874.79	-13.1040753
5090044.91	5090040.99	5090045.43	-4.44068703
682256.47	682248.687	682262.63	-13.9431043
5090432.86	5090428.48	5090431.3	-2.82469865
683525.6	683511.15	683519.04	-7.89033863
5089589.01	5089584.9	5089585.76	-0.85835445
681523.23	681523.027	681535.35	-12.322963
5089750.1	5089746.55	5089749.84	-3.29255184
682861.36	682847.467	682843.44	4.026994482
5090951.1	5090946.07	5090944.21	1.856959741
684423.22	684397.688	684390.18	7.507910814
5090990.58	5090984.97	5090980.42	4.553484722
684141.32	684124.784	684121.24	3.543868936
5088818.99	5088815.32	5088810.96	4.361337966
681675.6	681677.204	681684.7	-7.49641996
5088830.43	5088827.6	5088828.6	-0.99582935

Figure C.12
Affine transformation with 4 CORNERS and a MIDDLE GCPs.

DEVELOPED METHOD WITH 4 LOW GCPs AFFINE TRANSFORMATION (GCPs av.=9.5 metre)

	1	681616.3	5092221.5	0	0	0
	0	0	0	1	681616.3	5092221.5
	1	684612.3	5092964.1	0	0	0
A	0	0	0	1	684612.3	5092964.11
	1	683048.49	5092127.6	0	0	0
	0	0	0	1	683048.49	5092127.61
	1	684682.46	5091667.2	0	0	0
	0	0	0	1	684682.46	5091667.18

	1	0	1	0	1	0	1	0
	681616.3	0	684612	0	683048.49	0	684682.46	0
At	5092221.5	0	5E+06	0	5092127.61	0	5091667.2	0
	0	1	0	1	0	1	0	1
	0	681616.3	0	684612.3	0	683048.49	0	684682.5
	0	5092221.5	0	5092964.1	0	5092127.61	0	5091667

	681590.85
	5092219.84
	684583.57
L	5092954.5
	683020.48
	5092120.83
	684652.41
	5091658.07

	4	2733959.55	20368980.4	0	0	0
	3E+06	1.86864E+12	1.3922E+13	0	0	0
	2E+07	1.3922E+13	1.03724E+14	0	0	0
At*A	0	0	0	4	3E+06	20368980
	0	0	0	2733959.55	2E+12	1.392E+13
	0	0	0	20368980.4	1E+13	1.037E+14

	-0.172157571
	-0.376799332
	-0.011048614
V	0.064370241
	0.441831163
	1.304605057
	-0.096537759
	-0.169694151

	30018866.18	0.0908076	-5.907204275	0	0	0
(At*A)-1	0.090807634	1.579E-07	-3.90221E-08	0	0	0
	-5.907204275	-3.9E-08	1.16528E-06	0	0	0
	0	0	0	30018866.18	0.090807633	-5.907
	0	0	0	0.090807634	1.5787E-07	-4E-08
	0	0	0	-5.907204275	-3.90221E-08	1E-06

	2733847.31
AtL	1.86856E+12
	1.39214E+13
	20368953.24
	1.3922E+13
	1.03724E+14

	-4307.234375
	0.998707695
X	0.001013795
	3361.125
	-0.002422971
	0.999663875

Figure C.13
Developed method with 4 LOW GCPs.

X/Y Image	X/Y Trans	H DTM	NET Relief Displacements	X/Y MAP	Dx/DY	H Approximate	NET Relief Displacements	Dx/DY	H av.	Relief Displacements	Dx/DY
682468.46	682442.2633	9.5	0	682439.51	2.753268537	9	-0.197530213	2.555738324	55	21.72870894	24.48197748
5092741.01	5092736.734		0	5092732.46	4.273821385		-0.031884307	4.241937079	55	3.504884542	7.778705927
682343.65	682317.401	9	-0.197530213	682318	-0.796575698	12.5	1.185181278	0.586135793	55	21.72870894	21.12966346
5092530.31	5092526.407		-0.031884307	5092520.82	5.555169695		0.191305839	5.778359841	55	3.504884542	9.091938543
681804.57	681778.9079	9.5	0	681775.6	3.307917595	10	0.197530213	3.505447808	55	21.72870894	25.03662654
5092422.11	5092419.55		0	5092414.04	5.509597836		0.031884307	5.541482142	55	3.504884542	9.014482378
681815.49	681789.3842	10.5	0.395060426	681787.01	2.769250182	18	3.358013621	5.732203376	55	21.72870894	24.1028987
5091998.34	5091995.896		0.063768613	5091988.85	7.109347509		0.542033211	7.587612106	55	3.504884542	10.55046344
681515.58	681489.3186	43	13.23452427	681498.97	3.583138406	46.5	14.61723576	4.965849897	55	21.72870894	12.07732308
5091462.58	5091461.042		2.136248537	5091455.47	7.708583175		2.359438682	7.931773321	55	3.504884542	9.077219181
681986.32	681959.8943	9.5	0	681956.4	3.494326868	29.5	7.90120852	11.39553539	55	21.72870894	25.22303581
5091900.59	5091897.765		0	5091892.52	5.244519075		1.275372261	6.519891336	55	3.504884542	8.749403617
683372.76	683346.4419	12	0.987651065	683344.73	2.69951821	31	8.493799159	10.2056663	55	21.72870894	23.44057609
5093773.99	5093767.176		0.159421533	5093763.36	3.974939584		1.37102518	5.186543232	55	3.504884542	7.320402593
682159.9	682133.2709	9.5	0	682130.88	2.39089277	17	2.962953195	5.353845965	55	21.72870894	24.11960171
5091921.19	5091917.937		0	5091913.92	4.017015632		0.478264598	4.49528023	55	3.504884542	7.521900174
681938.75	681911.0804	80	27.85176003	681938.75	0.182158958	93	32.98754557	5.317944496	55	21.72870894	-5.94089213
5090612.95	5090610.673		4.495687219	5090612.95	2.218275641		5.324679188	3.04726761	55	3.504884542	1.227472964
681645.7	681618.0352	107	38.51839153	681654.96	1.593582413	93	32.98754557	-3.937263551	55	21.72870894	-15.1961002
5090244.12	5090242.677		6.217439771	5090243.58	5.314052928		5.324679188	4.421292346	55	3.504884542	2.6014977
682793.09	682762.9265	101.5	36.34555919	682800.21	-0.937971733	109.5	39.5060426	2.222511675	55	21.72870894	-15.554822
5089242	5089238.113		5.866712399	5089241.28	2.70007126		6.376861303	3.210220165	55	3.504884542	0.338243403
683434.71	683403.2245	79.5	27.65422982	683431.61	-0.731305854	71.5	24.49374641	-3.891789261	55	21.72870894	-6.65682673
5088755.87	5088750.592		4.463802912	5088752.61	2.445935987		3.953654008	1.935787083	55	3.504884542	1.487017617
681865.35	681837.1994	110.5	39.90110302	681874.79	2.310481063	89.5	31.60483408	-5.985787883	55	21.72870894	-15.861913
5090044.91	5090043.001		6.440629916	5090045.43	4.011997114		5.101489043	2.672856241	55	3.504884542	1.07625174
682256.47	682228.2072	94.5	33.58013621	682262.63	-0.842630283	76	26.27151833	-8.151248164	55	21.72870894	-12.6940575
5090432.86	5090429.873		5.420332108	5090431.3	3.993627113		4.240612767	2.813907772	55	3.504884542	2.078179548
683525.6	683494.8416	61	20.34561194	683519.04	-3.852748226	45.5	14.22217534	-9.976184829	55	21.72870894	-2.46965122
5089589.01	5089583.232		3.284083571	5089585.76	0.755953244		2.295670069	-0.232460259	55	3.504884542	0.976754215
681523.23	681495.2226	120.5	43.85170728	681535.35	3.724331774	94	33.382606	-6.744769515	55	21.72870894	-18.3986666
5089750.1	5089749.119		7.078316047	5089749.84	6.357723144		5.388447801	4.667854899	55	3.504884542	2.784291639
682861.36	682832.8409	40	12.04934299	682843.44	1.450263306	40.5	12.24687321	1.647793518	55	21.72870894	11.12962926
5090951.1	5090946.473		1.944942698	5090944.21	4.208413268		1.976827004	4.240297575	55	3.504884542	5.768355113
684423.22	684392.7225	7	-0.987651065	684390.18	1.554894643	11	0.592590639	3.135136347	55	21.72870894	24.27125465
5090990.58	5090982.156		-0.159421533	5090980.42	1.576437755		0.09565292	1.831512207	55	3.504884542	5.24074383
684141.32	684108.9853	41	12.44440342	684121.24	0.189702962	54.5	17.77771917	5.523018713	55	21.72870894	9.474008486
5088818.99	5088811.979		2.008711311	5088810.96	3.027532804		2.869587587	3.88840908	55	3.504884542	4.523706036
681675.6	681646.4634	108.5	39.11098217	681684.7	0.874341406	100	35.75296855	-2.483672215	55	21.72870894	-16.5079318
5088830.43	5088829.389		6.31309269	5088828.6	7.102436271		5.77105948	6.56040306	55	3.504884542	4.294228123

Figure C.13
Developed method with 4 LOW GCPs (cont.)

DEVELOPED METHOD WITH 4 HIGH GCPs AFFINE TRANSFORMATION (GCPs av.=86.25 metre)

A

1	683064.28	5089649	0	0	0
0	0	0	1	683064.3	5089649
1	683038.4	5089557	0	0	0
0	0	0	1	683038.4	5089557
1	681540.39	5091001	0	0	0
0	0	0	1	681540.4	5091001
1	683125.95	5089567	0	0	0
0	0	0	1	683126	5089567

At

1	0	1	0	1	0	1	0
683064.28	0	683038.4	0	681540.4	0	683126	0
5089649.45	0	5089556.94	0	5091001	0	5089567	0
0	1	0	1	0	1	0	1
0	683064.28	0	683038.4	0	681540.4	0	683126
0	5089649.45	0	5089556.9	0	5091001	0	5089567

L

683067.06
5089646.64
683041.38
5089556.01
681536.41
5090997.67
683128.75
5089563.95

At*A

4	2730769.02	20359774.9	0	0	0
2730769	1.86428E+12	1.3899E+13	0	0	0
20359775	1.38995E+13	1.0363E+14	0	0	0
0	0	0	4	2730769.02	20359775
0	0	0	2730769.02	1.86428E+12	1.39E+13
0	0	0	20359774.87	1.38995E+13	1.036E+14

(At*A)-1

4379711590	-702.528	-766.237	0	0	0
-702.527716	0.000113	0.000123	0	0	0
-766.236638	0.000123	0.000134	0	0	0
0	0	0	4.38E+09	-702.528	-766.23664
0	0	0	-702.528	0.000113	0.0001228
0	0	0	-766.237	0.000123	0.0001341

AtL

2730773.6
1.86428E+12
1.38995E+13
20359764.27
1.38995E+13
1.0363E+14

X

17816
1.001093
-0.00365
108272
-0.01669
0.980959

Figure C.14
Developed method with 4 HIGH GCPs.

X/Y Image	X/Y Trans	H DTM	NET Relief Displacements	X/Y MAP	Dx/DY
682468.46	682455.7686	9.5	-30.32088769	682439.51	-14.06227807
5092741.01	5092654.498		-4.89424105	5092732.46	-82.85633919
682343.65	682331.5906	9	-30.51841791	682318	-16.9277821
5092530.31	5092449.892		-4.926125357	5092520.82	-75.85375637
681804.57	681792.3159	10	-30.12335748	681775.6	-13.40750424
5092422.11	5092352.747		-4.862356744	5092414.04	-66.15493321
681815.49	681804.7934	11	-29.72829706	681787.01	-11.94487608
5091998.34	5091936.864		-4.798588131	5091988.85	-56.78433951
681515.58	681506.5096	43	-17.08636342	681498.97	-9.546768182
5091462.58	5091416.31		-2.757992514	5091455.47	-41.91816078
681986.32	681976.1667	11	-29.72829706	681956.4	-9.961566418
5091900.59	5091838.125		-4.798588131	5091892.52	-59.19345731
683372.76	683357.2897	10	-30.12335748	683344.73	-17.56361373
5093773.99	5093652.72		-4.862356744	5093763.36	-115.5021753
682159.9	682149.8614	9.5	-30.32088769	682130.88	-11.33950178
5091921.19	5091855.437		-4.89424105	5091913.92	-63.37762272
681938.75	681933.2412	80	-2.469127662	681938.75	-7.97796479
5090612.95	5090575.797		-0.398553831	5090612.95	-37.55166214
681645.7	681641.216	106	75.9503669	681654.96	62.2063563
5090244.12	5090218.879		1.259430107	5090243.58	-23.4410819
682793.09	682793.5156	101.5	6.024671496	682800.21	-0.669737335
5089242	5089216.696		0.972471349	5089241.28	-23.61137181
683434.71	683437.6102	79.5	-2.666657875	683431.61	3.333547928
5088755.87	5088729.117		-0.430438138	5088752.61	-23.92359154
681865.35	681861.8327	112.5	10.37033618	681874.79	-2.586927726
5090044.91	5090019.798		1.673926092	5090045.43	-23.95838304
682256.47	682251.9654	96.5	4.049369366	682262.63	-6.615229456
5090432.86	5090393.835		0.653628284	5090431.3	-36.81168784
683525.6	683525.5608	63	-9.185154904	683519.04	-2.664305471
5089589.01	5089544.876		-1.482620253	5089585.76	-42.36618736
681523.23	681520.4139	120.5	13.53081959	681535.35	-1.405245488
5089750.1	5089736.31		2.184074996	5089749.84	-11.34630041
682861.36	682855.6266	40	-18.2715447	682843.44	-6.084955261
5090951.1	5090892.114		-2.949298353	5090944.21	-55.04533767
684423.22	684419.0503	6.5	-31.50606897	684390.18	-2.635757655
5090990.58	5090904.782		-5.085546889	5090980.42	-80.72372022
684141.32	684144.7626	42	-17.48142385	684121.24	6.041161261
5088818.99	5088779.245		-2.821761127	5088810.96	-34.53691747
681675.6	681676.3049	108.5	8.790094478	681684.7	0.394966481
5088830.43	5088831.609		1.41885164	5088828.6	4.427601698

Figure C.14
Developed method with 4 HIGH GCPs (cont.)

DEVELOPED METHOD WITH 4 MIX GCPs AFFINE TRANSFORMATION (GCPs av.=57.5 metre)

	1	681616.3	5092222	0	0	0
	0	0	0	1	681616.3	5092222
	1	681810.6	5091355	0	0	0
A	0	0	0	1	681810.6	5091355
	1	681540.4	5091001	0	0	0
	0	0	0	1	681540.4	5091001
	1	682660.5	5089158	0	0	0
	0	0	0	1	682660.5	5089158

	1	0	1	0	1	0	1	0
	681616.3	0	681810.6	0	681540.4	0	682660.5	0
At	5092222	0	5091355	0	5091001	0	5089158	0
	0	1	0	1	0	1	0	1
	0	681616.3	0	681810.6	0	681540.4	0	682660.5
	0	5092221.5	0	5091355	0	5091001	0	5089158

	681590.9
	5092220
	681793
L	5091351
	681536.4
	5090998
	682668.7
	5089155

	4	2727627.8	20363736	0	0	0
	2727627.8	1.85999E+12	1.39E+13	0	0	0
	20363736.37	1.38862E+13	1.04E+14	0	0	0
At*A	0	0	0	4	2727628	20363736
	0	0	0	2727627.8	1.86E+12	1.39E+13
	0	0	0	20363736.37	1.39E+13	1.04E+14

	-1.39221
	-0.79416
	2.350465
V	1.291918
	-0.47999
	-0.28567
	-0.47008
	-0.28015

	43333528	-15.2912	-6.46371	0	0	0
(At*A)-1	-15.2912	6.13E-06	2.18E-06	0	0	0
	-6.46371	2.18E-06	9.77E-07	0	0	0
	0	0	0	43333528	-15.2912	-6.463713472
	0	0	0	-15.2912	6.13E-06	2.1825E-06
	0	0	0	-6.46371	2.18E-06	9.77316E-07

	2727588.97
AtL	1.85996E+12
	1.3886E+13
	20363723.82
	1.38862E+13
	1.0367E+14

	99751.39
	0.982528
X	-0.01726
	-6721.25
	0.002156
	1.001031

Figure C.15
Developed method with 4 MIX GCPs.

X/Y Image	X/Y Trans	H DTM	NET Relief Displacements	X/Y MAP	Dx/DY
682468.46	682417.765	9.5	-18.96290045	682439.51	-40.7083404
5092741.01	5092740.93		-3.060893426	5092732.46	5.407910842
682343.65	682298.771	9	-19.16043066	682318	-38.38946463
5092530.31	5092529.74		-3.092777732	5092520.82	5.829717606
681804.57	681770.977	9.5	-18.96290045	681775.6	-23.58616966
5092422.11	5092420.27		-3.060893426	5092414.04	3.167689761
681815.49	681789.018	10.5	-18.56784002	681787.01	-16.55951154
5091998.34	5091996.09		-2.997124813	5091988.85	4.23818111
681515.58	681503.593	43	-5.728376177	681498.97	-1.105231298
5091462.58	5091459.13		-0.924644889	5091455.47	2.731723878
681986.32	681958.55	9.5	-18.96290045	681956.4	-16.81255295
5091900.59	5091898.6		-3.060893426	5091892.52	3.02199993
683372.76	683288.44	12	-17.97524938	683344.73	-74.26511477
5093773.99	5093776.92		-2.901471893	5093763.36	10.6620078
682159.9	682128.742	9.5	-18.96290045	682130.88	-21.10077739
5091921.19	5091919.6		-3.060893426	5091913.92	2.617512493
681938.75	681934.03	80	8.888859585	681938.75	4.169295429
5090612.95	5090609.53		1.434793793	5090612.95	-1.982187431
681645.7	681652.465	105	18.76537023	681654.96	16.27029455
5090244.12	5090239.69		3.029009119	5090243.58	-0.860045641
682793.09	682797.1	101.5	17.38265874	682800.21	14.27272792
5089242	5089239.01		2.805818973	5089241.28	0.537810432
683434.71	683435.898	79.5	8.691329372	683431.61	12.97957152
5088755.87	5088753.76		1.402909487	5088752.61	2.557275714
681865.35	681871.715	109.5	20.54314215	681874.79	17.46786031
5090044.91	5090040.75		3.315967878	5090045.43	-1.36481765
682256.47	682249.307	94.5	14.61723576	682262.63	1.294076944
5090432.86	5090429.94		2.359438682	5090431.3	1.0018969
683525.6	683510.824	61	1.382711491	683519.04	-6.833360765
5089589.01	5089587.96		0.223190146	5089585.76	2.422338904
681523.23	681540.659	120.5	24.88880684	681535.35	30.19809926
5089750.1	5089744.9		4.017422621	5089749.84	-0.924942523
682861.36	682834.686	40	-6.913557455	682843.44	-15.66777317
5090951.1	5090950.02		-1.115950728	5090944.21	4.694991346
684423.22	684368.576	7	-19.95055151	684390.18	-41.55456547
5090990.58	5090992.91		-3.220314958	5090980.42	9.2690492
684141.32	684129.073	41	-6.518497029	684121.24	1.31480589
5088818.99	5088818.47		-1.052182115	5088810.96	6.460860526
681675.6	681706.237	108.5	20.14808173	681684.7	41.68461567
5088830.43	5088824.61		3.252199265	5088828.6	-0.739619789

Figure C.15
Developed method with 4 MIX GCPs (cont.)

DEVELOPED METHOD WITH AVERAGE SHIFT (DTM)

3LOW VERY CLOSE (Cx= - 29.63 & Cy = -10.01)

X/Y Image	X/Y MAP	H DTM	Relief Displacements	X/Y RESULTED	Dx/DY
682468.46	682439.51	9.5	3.753074047	682442.6131	3.103074
5092741.01	5092732.46		0.605801824	5092731.606	-0.8542
682343.65	682318	9	3.555543834	682317.6055	-0.39446
5092530.31	5092520.82		0.573917517	5092520.874	0.053918
681804.57	681775.6	9.5	3.753074047	681778.7231	3.123074
5092422.11	5092414.04		0.605801824	5092412.706	-1.3342
681815.49	681787.01	10.5	4.148134473	681790.0381	3.028134
5091998.34	5091988.85		0.669570437	5091989	0.14957
681515.58	681498.97	43	16.98759832	681502.9676	3.997598
5091462.58	5091455.47		2.74205036	5091455.312	-0.15795
681986.32	681956.4	9.5	3.753074047	681960.4731	4.073074
5091900.59	5091892.52		0.605801824	5091891.186	-1.3342
683372.76	683344.73	12	4.740725112	683347.9007	3.170725
5093773.99	5093763.36		0.765223356	5093764.745	1.385223
682159.9	682130.88	10	3.95060426	682134.2506	3.370604
5091921.19	5091913.92		0.63768613	5091911.818	-2.10231
681938.75	681938.75	80	31.60483408	681940.7548	2.004834
5090612.95	5090612.95		5.101489043	5090608.041	-4.90851
681645.7	681654.96	107	42.27146558	681658.3715	3.411466
5090244.12	5090243.58		6.823241595	5090240.933	-2.64676
682793.09	682800.21	101.5	40.09863324	682803.5886	3.378633
5089242	5089241.28		6.472514223	5089238.463	-2.81749
683434.71	683431.61	79.5	31.40730387	683436.5173	4.907304
5088755.87	5088752.61		5.069604736	5088750.93	-1.6804
681865.35	681874.79	110.5	43.65417707	681879.4042	4.614177
5090044.91	5090045.43		7.04643174	5090041.946	-3.48357
682256.47	682262.63	94.5	37.33321026	682264.2032	1.57321
5090432.86	5090431.3		6.026133932	5090428.876	-2.42387
683525.6	683519.04	61	24.09868599	683520.0987	1.058686
5089589.01	5089585.76		3.889885395	5089582.89	-2.87011
681523.23	681535.35	120.5	47.60478133	681541.2348	5.884781
5089750.1	5089749.84		7.684117871	5089747.774	-2.06588
682861.36	682843.44	40	15.80241704	682847.5624	4.122417
5090951.1	5090944.21		2.550744521	5090943.641	-0.56926
684423.22	684390.18	7	2.765422982	684396.3854	6.205423
5090990.58	5090980.42		0.446380291	5090981.016	0.59638
684141.32	684121.24	41	16.19747747	684127.9175	6.677477
5088818.99	5088810.96		2.614513134	5088811.595	0.634513
681675.6	681684.7	108.5	42.86405622	681688.8641	4.164056
5088830.43	5088828.6		6.918894514	5088827.339	-1.26111

3HIGH VERY CLOSE (Cx= - 33.11 & Cy = -7.14)

X/Y Image	X/Y MAP	H DTM	Relief Displacements	X/Y RESULTED	Dx/DY
682468.46	682439.51	9.5	3.753074047	682439.1031	-0.406925953
5092741.01	5092732.46		0.605801824	5092734.476	2.015801824
682343.65	682318	9	3.555543834	682314.0955	-3.904456166
5092530.31	5092520.82		0.573917517	5092523.744	2.923917517
681804.57	681775.6	9.5	3.753074047	681775.2131	-0.386925953
5092422.11	5092414.04		0.605801824	5092415.576	1.535801824
681815.49	681787.01	10.5	4.148134473	681786.5281	-0.481865527
5091998.34	5091988.85		0.669570437	5091991.87	3.019570437
681515.58	681498.97	43	16.98759832	681499.4576	0.487598317
5091462.58	5091455.47		2.74205036	5091458.182	2.712050362
681986.32	681956.4	9.5	3.753074047	681956.9631	0.563074047
5091900.59	5091892.52		0.605801824	5091894.056	1.535801824
683372.76	683344.73	12	4.740725112	683344.3907	-0.339274888
5093773.99	5093763.36		0.765223356	5093767.615	4.255223356
682159.9	682130.88	10	3.95060426	682130.7406	-0.13939574
5091921.19	5091913.92		0.63768613	5091914.688	0.767686131
681938.75	681938.75	80	31.60483408	681937.2448	-1.505165921
5090612.95	5090612.95		5.101489043	5090610.911	-2.038510957
681645.7	681654.96	107	42.27146558	681654.8615	-0.09853442
5090244.12	5090243.58		6.823241595	5090243.803	0.223241595
682793.09	682800.21	101.5	40.09863324	682800.0786	-0.131366762
5089242	5089241.28		6.472514223	5089241.333	0.052514223
683434.71	683431.61	79.5	31.40730387	683433.0073	1.397303866
5088755.87	5088752.61		5.069604736	5088753.8	1.189604736
681865.35	681874.79	110.5	43.65417707	681875.8942	1.104177071
5090044.91	5090045.43		7.04643174	5090044.816	-0.613568259
682256.470	682262.63000	94.5	37.33321026	682260.69	-1.936789744
5090432.86	5090431.3		6.026133932	5090431.746	0.446133932
683525.6	683519.04	61	24.09868599	683516.5887	-2.451314015
5089589.01	5089585.76		3.889885395	5089585.76	-0.000114605
681523.23	681535.35	120.5	47.60478133	681537.7248	2.374781331
5089750.1	5089749.84		7.684117871	5089750.644	0.804117871
682861.36	682843.44	40	15.80241704	682844.0524	0.612417039
5090951.1	5090944.21		2.550744521	5090946.511	2.300744521
684423.22	684390.18	7	2.765422982	684392.8754	2.695422982
5090990.58	5090980.42		0.446380291	5090983.886	3.466380292
684141.32	684121.24	41	16.19747747	684124.4075	3.167477465
5088818.99	5088810.96		2.614513134	5088814.465	3.504513135
681675.6	681684.7	108.5	42.86405622	681685.3541	0.65405622
5088830.43	5088828.6		6.918894514	5088830.209	1.608894515

3MIX VERY CLOSE)Cx = -33.45 & Cy= -7.23)

X/Y Image	X/Y MAP	H DTM	Relief Displacements	X/Y RESULTED	Dx/DY
682468.46	682439.51	9.5	3.753074047	682438.7631	-0.74693
5092741.01	5092732.46		0.605801824	5092734.386	1.925802
682343.65	682318	9	3.555543834	682313.7555	-4.24446
5092530.31	5092520.82		0.573917517	5092523.654	2.833918
681804.57	681775.6	9.5	3.753074047	681774.8731	-0.72693
5092422.11	5092414.04		0.605801824	5092415.486	1.445802
681815.49	681787.01	10.5	4.148134473	681786.1881	-0.82187
5091998.34	5091988.85		0.669570437	5091991.78	2.92957
681515.58	681498.97	43	16.98759832	681499.1176	0.147598
5091462.58	5091455.47		2.74205036	5091458.092	2.62205
681986.32	681956.4	9.5	3.753074047	681956.6231	0.223074
5091900.59	5091892.52		0.605801824	5091893.966	1.445802
683372.76	683344.73	12	4.740725112	683344.0507	-0.67927
5093773.99	5093763.36		0.765223356	5093767.525	4.165223
682159.9	682130.88	10	3.95060426	682130.4006	-0.4794
5091921.19	5091913.92		0.63768613	5091914.598	0.677686
681938.75	681938.75	80	31.60483408	681936.9048	-1.84517
5090612.95	5090612.95		5.101489043	5090610.821	-2.12851
681645.7	681654.96	107	42.27146558	681654.5215	-0.43853
5090244.12	5090243.58		6.823241595	5090243.713	0.133242
682793.09	682800.21	101.5	40.09863324	682799.7386	-0.47137
5089242	5089241.28		6.472514223	5089241.243	-0.03749
683434.71	683431.61	79.5	31.40730387	683432.6673	1.057304
5088755.87	5088752.61		5.069604736	5088753.71	1.099605
681865.35	681874.79	110.5	43.65417707	681875.5542	0.764177
5090044.91	5090045.43		7.04643174	5090044.726	-0.70357
682256.47	682262.63	94.5	37.33321026	682260.3532	-2.27679
5090432.86	5090431.3		6.026133932	5090431.656	0.356134
683525.6	683519.04	61	24.09868599	683516.2487	-2.79131
5089589.01	5089585.76		3.889885395	5089585.67	-0.09011
681523.23	681535.35	120.5	47.60478133	681537.3848	2.034781
5089750.1	5089749.84		7.684117871	5089750.554	0.714118
682861.36	682843.44	40	15.80241704	682843.7124	0.272417
5090951.1	5090944.21		2.550744521	5090946.421	2.210745
684423.22	684390.18	7	2.765422982	684392.5354	2.355423
5090990.58	5090980.42		0.446380291	5090983.796	3.37638
684141.32	684121.24	41	16.19747747	684124.0675	2.827477
5088818.99	5088810.96		2.614513134	5088814.375	3.414513
681675.6	681684.7	108.5	42.86405622	681685.0141	0.314056
5088830.43	5088828.6		6.918894514	5088830.119	1.518895

Figure C.16
Developed method with average shift.

APPENDIX D

IISP: CASE 3 BACK TRANSFORMATION EQUATIONS

This appendix represents the back transformation equations developed in order to bring the feature from its ground position to its position within the IKONOS imagery, as discussed in section 3.4.6.4.2.

Where:

Xg, Yg = Feature ground coordinates [$P_{g\,(x,\,y)}$].

xi, yi = Feature image coordinates [$P'_{(x,\,y)}$].

(a1…a6) = Transformation parameters obtained.

Φ = Nominal collection azimuth.

Rd = Calculated relief displacement value based upon the point interpolated or assumed height and the nominal collection elevation angle (E).

Φ < 90 Degrees

$$Xg = Rd * SIN(\Phi) + a1 + xi * a2 + yi * a3$$

$$Yg = Rd * COS(\Phi) + a4 + xi * a5 + yi * a6$$

Rearrange

$$xi * a2 = Xg - Rd * SIN(\Phi) - a1 - yi * a3 \qquad (1)$$

$$xi * a5 = Yg - Rd * COS(\Phi) - a4 - yi * a6 \qquad (2)$$

Divide (1) by (2)

$$Xg * a5 - Rd * SIN(\Phi) * a5 - a1 * a5 - yi * a3 * a5 =$$

$$Yg * a2 - Rd * COS(\Phi) * a2 - a4 * a2 - yi * a6 * a2$$

Rearrange

$$yi * (a5 * a3 - a6 * a2) = a5 * Xg - a5 * Rd * SIN(\Phi) - a5 * a1 - Yg * a2$$

$$+ Rd * COS(\Phi) * a2 + a2 * a4$$

$$yi = \frac{a5 * Xg - a5 * Rd * SIN(\Phi) - a5 * a1 - Yg * a2 + Rd * COS(\Phi) * a2 + a2 * a4}{(a5 * a3 - a6 * a2)}$$

$$xi = \frac{Xg - Rd * SIN(\Phi) - a1 - yi * a3}{a2}$$

<u>90 < Φ < 180 Degrees</u>

$$Xg = Rd * COS(\Phi - 90) + a1 + xi * a2 + yi * a3$$

$$Yg = - Rd * SIN(\Phi - 90) + a4 + xi * a5 + yi * a6$$

Rearrange

$$xi * a2 = Xg - Rd * COS(\Phi - 90) - a1 - yi * a3 \qquad (1)$$

$$xi * a5 = Yg + Rd * SIN(\Phi - 90) - a4 - yi * a6 \qquad (2)$$

Divide (1) by (2)

$$Xg * a5 - Rd * COS(\Phi - 90) * a5 - a1 * a5 - yi * a3 * a5 =$$

$$Yg * a2 + Rd * SIN(\Phi - 90) * a2 - a4 * a2 - yi * a6 * a2$$

Rearrange

$$yi * (a5 * a3 - a6 * a2) = a5 * Xg - a5 * Rd * COS(\Phi - 90) - a5 * a1 - Yg * a2$$

$$- Rd * SIN(\Phi - 90) * a2 + a2 * a4$$

$$yi = \frac{\left\{ \begin{array}{c} a5 * Xg - a5 * Rd * COS(\Phi - 90) - a5 * a1 - Yg * a2 \\ - Rd * SIN(\Phi - 90) * a2 + a2 * a4 \end{array} \right\}}{a5 * a3 - a6 * a2}$$

$$xi = \frac{Xg - Rd * COS(\Phi - 90) - a1 - yi * a3}{a2}$$

180 < Φ < 270 Degrees

$$Xg = - Rd * COS(270 - \Phi) + a1 + xi * a2 + yi * a3$$

$$Yg = - Rd * SIN(270 - \Phi) + a4 + xi * a5 + yi * a6$$

Rearrange

$$xi * a2 = Xg + Rd * COS(270 - \Phi) - a1 - yi * a3 \qquad (1)$$

$$xi * a5 = Yg + Rd * SIN(270 - \Phi) - a4 - yi * a6 \qquad (2)$$

Divide (1) by (2)

$$Yg * a2 + Rd * SIN(270 - \Phi) * a2 - a4 * a2 - yi * a6 * a2 = Xg * a5 + Rd * COS(270 - \Phi) * a5 - a1 * a5 - yi * a3 * a5$$

Rearrange

$$yi * (a5 * a3 - a6 * a2) = a5 * Xg + a5 * Rd * COS(270 - \Phi) - a5 * a1 - Yg * a2 - Rd * SIN(270 - \Phi) * a2 + a2 * a4$$

$$yi = \frac{\left\{ \begin{array}{c} a5 * Xg + a5 * Rd * COS(270 - \Phi) - a5 * a1 - Yg * a2 \\ - Rd * SIN(270 - \Phi) * a2 + a2 * a4 \end{array} \right\}}{a5 * a3 - a6 * a2}$$

$$xi = \frac{Xg + Rd * COS(270 - \Phi) - a1 - yi * a3}{a2}$$

270 < Φ < 360 Degrees

$$Xg = -Rd * SIN(360 - \Phi) + a1 + xi * a2 + yi * a3$$

$$Yg = Rd * COS(360 - \Phi) + a4 + xi * a5 + yi * a6$$

Rearrange

$$xi * a2 = Xg + Rd * SIN(360 - \Phi) - a1 - yi * a3 \qquad (1)$$

$$yi * a5 = Yg - Rd * COS(360 - \Phi) - a4 - yi * a6 \qquad (2)$$

Divide (1) by (2)

$$Yg * a2 - Rd * COS(360 - \Phi) * a2 - a4 * a2 - yi * a6 * a2 =$$
$$Xg * a5 + Rd * SIN(360 - \Phi) * a5 - a1 * a5 - yi * a3 * a5$$

Rearrange

$$yi * (a5 * a3 - a6 * a2) = a5 * Xg + a5 * Rd * SIN(360 - \Phi) - a5 * a1 - Yg * a2$$
$$+ Rd * COS(360 - \Phi) * a2 + a2 * a4$$

$$yi = \frac{\left\{ \begin{array}{c} a5 * Xg + a5 * Rd * SIN(360 - \Phi) - a5 * a1 - Yg * a2 \\ + Rd * COS(360 - \Phi) * a2 + a2 * a4 \end{array} \right\}}{a5 * a3 - a6 * a2}$$

$$xi = \frac{Xg + Rd * SIN(360 - \Phi) - a1 - yi * a3}{a2}$$

Appendix E

IISP TESTING STAGES RESULTS

As mentioned in section 4.2, testing stage's two steps were accomplished by comparing the results obtained by querying the image within the IISP context and by performing the same query process, when required, using the vector data either in one or multiple processing steps within ESRI ARCVIEW software packages. The results obtained from this testing stage's steps are represented in this appendix.

POINTING QUERY FUNCTION

In order to test this function, a feature on the image is selected for query by pointing. Then, the system searches this feature directly or using a search circle. The system retrieves the resulted feature(s) and the user has to confirm the results in order to retrieve the attributes related to the correct/desired feature. A manual comparison/inspection of the desired/selected feature and the actual one was accomplished in order to check whether or not the retrieved feature is representing the user desired feature. In other words, this testing stage answers two main questions:

1- Is the retrieved feature the correct/desired one?

2- Does the used search circle, if any, produce and consequently facilitate obtaining the correct/desired feature?

The results obtained from this step are presented on the following pages.

Case 1

Image ID	Vector Layer Used	Query Point Image Location	Queried Feature Type	Results IISP	Search Circle Needed	Correct Feature (Fid) and Attribute
464546470	Building	436358 798812	Polygon	Fid=31	NO	YES
464546470	Building	436962 799032	Polygon	Fid=15	NO	YES
464546470	Building	438370 798530	Polygon	Fid=23	NO	YES
464546470	Road	435789 798462	Line	Fid=499	NO	YES
464546470	Road	437294 799088	Line	Fid=294	NO	YES
464546470	Road	437644 797729	Line	Fid=296	NO	YES
464546470	Water	436809 798558	Line	Fid=197	NO	YES
464546470	Water	437131 797492	Line	Fid=192	NO	YES
464546470	Water	437892 797904	Line	Fid=187	NO	YES

Case 2

Projection Method I

Image ID	Vector Layer Used	Query Point Image Location	Queried Feature Type	Results IISP	Search Circle Needed	Correct Feature (Fid) and Attribute
46106470	Building	263365.9 7459719	Polygon	Fid=156	NO	YES
46106470	Building	2635091 7460309	Polygon	Fid=15	NO	YES
46106470	Building	2633858 7458187	Polygon	Fid=142	NO	YES
46106470	River	2634813 7459337	Line	Fid=45	NO	YES
46106470	River	2634087 7458195	Line	Fid=84	NO	YES
46106470	River	2636021 7461940	Line	Fid=21	NO	YES

Projection Method 2

When used with the same feature's image coordinates, projection Method II provided the same results as in Method I.

Case 3

The accuracy of the developed IKONOS GEOCARTERRA™ imagery orthorectification procedures was tested as shown in Appendix C. This testing stage is concerned with testing the implemented method of using Affine transformation with 4 LOW GCPs and *average terrain height* within the image area.

Image ID	Vector Layer Used	Query Point Image Location	Queried Feature Type	Results IISP	Expansion of Search Circle Needed	Correct Feature (Fid) and Attribute
IKONOS	Building Layer Part1	682399.75 5092817	Polygon	Fid=102	NO	YES
IKONOS	Building Layer Part1	682635 5092435	Polygon	Fid=115	NO	YES
IKONOS	Building Layer Part1	681539 509221.7	Polygon	Fid=150	NO	YES
IKONOS	Road Layer Part2	681344 5089574	Line	Fid=672	NO	YES
IKONOS	Road Layer Part2	682400 5089023	Line	Fid=1345	NO	YES
IKONOS	Road Layer Part2	682242 5091124	Line	Fid=981	NO	YES

Within the IKONOS imagery query and/or when using linear/point features' layers within any of the developed IISP cases, the usage of a search circle is performed automatically using a predefined radius, as explained within the design and implementation stage. As shown in the previous results, the default search circle used helped in obtaining the desired/correct query feature within these queries even without the need to expand the search circle radius.

One remaining issue here is that we have to consider testing the results obtained from using the search circle when having an image or a layer with a geographic coordinate system. In such case, as explained previously, the entered search radius has to be approximately converted from metre to decimal degree based upon a 6372 Km earth radius. In order to test that the conversion is producing acceptable results, Case 2 (Moncton image) was used along with the Road layer which has a geographic coordinate system. A location on the image has been selected for query and different search circles were used. The results obtained from the search circle were compared using IISP projection Method I and Method II. The results were as follows:

Query Point Image Location	Results Obtained Using Default Query Search Circle(30m)	Search Circle Radius (User-Defined) Metre	Results IISP Projection Method I (No conversion is necessary because the layer was converted to the image projected coordinate system at the beginning of the process)	Results IISP Projection Method II (Conversion of the entered search circle radius is required from Metre to Decimal Degree within the query)
2634029.6 2459689.9	NONE	100	NONE	NONE
	NONE	200	NONE	NONE
	NONE	300	NONE	562, 565, 568
	NONE	320	562, 565, 568	562, 565, 568
2635275.1 7460111.4	NONE	100	NONE	NONE
	NONE	200	555	555
	NONE	500	546, 555	546, 555, 557, 598
	NONE	525	546, 555, 557, 598	546, 555, 557, 598

As can be extracted from the previous results, the search circle's radius conversion process may produce a search circle radius that is larger than the desired one entered by the user. Since the user can increase or decrease the radius and he/she have to confirm the right feature resulted at the end of the query process, the difference presented within the results (@ 25 metres) is not significant to the pointing query function.

POLYGON/LINE BASED IMAGE QUERY FUNCTION

In order to test this function, ground features appearing on the image, were selected to represent the desired user-defined polygon/line vertices. Within IISP, the system processes the query based upon that polygon/line and the results obtained based on the pre-defined vector layers query. Within the ArcView, the same user-defined polygon/line connecting the pre-specified features' points in the processed vector layer domain is used to perform the same query as applied within the IISP. The results are shown below.

1- The query produces the correct number of features within the user-defined polygon/line for different search criteria

(NOT ALL SEARCH CRITERIA WERE USED FOR ALL CASES)
(The search criteria are shown in Figure 3.12)

Case 1

User-defined polygon
Coordinates (436324, 7988797), (436950.7, 798879), (437981.4, 798612.83), (436665.78, 798474.9)
Image Id: (464546470)
Vector layers used for query: Building- Road- Water

- Search Criterion # 1

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	13, 15, 16, 17, 18, 19, 20, 21, 23, 24, 31, 36	The same results	YES
Road	240, 269, 294, 298, 299, 300, 301, 302, 344, 370, 373, 376, 377, 378, 379, 380, 381, 382, 383, 446, 584	The same results	YES
Water	87, 197	The same results	YES

- Search Criterion # 4

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	20, 31	The same results	YES
Road	269, 294, 302, 378, 380, 446	The same results	YES
Water	197	The same results	YES

- Search Criterion # 5

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	13, 15, 16, 17, 18, 19, 20, 21, 31	The same results	YES
Road	269, 294, 298, 299, 300, 301, 302, 344, 378, 380, 381, 382, 383, 446	The same results	YES
Water	197	The same results	YES

- Search Criterion # 6

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	13, 15, 16, 17, 18, 19, 20, 21, 31	The same results	YES
Road	269, 294, 298, 299, 300, 301, 302, 344, 378, 380, 381, 383, 446	The same results	YES
Water	197	The same results	YES

- Search Criterion # 7

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	13, 16, 17, 18, 19, 20, 21	The same results	YES
Road	269, 298, 299, 344, 378, 380, 381, 383	The same results	YES
Water	NA	NA	NA

- Search Criterion # 8

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	13, 15, 16, 17, 18, 19, 20, 21	The same results	YES
Road	269, 294, 298, 299, 300, 301, 344, 378, 381, 382, 383	The same results	YES
Water	NA	NA	NA

User-defined line
Coordinates (435563, 799077), (436364, 798564), (436956, 799077)
Image Id: (464546470)
Vector layers used for query: Building- Property

- Search Criterion # 5

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Property	3, 6, 7, 24, 31, 32, 33, 502, 593, 594, 609, 593, 594, 609, 610, 611, 641, 650, 668, 669, 684, 688, 690, 706, 742, 753, 754, 755, 758, 760, 761	The same results	YES
Building	15, 16, 24	The same results	YES

- Search Criterion # 6

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Property	3, 6, 24, 31, 32, 33, 593, 609, 610, 611, 641, 649, 650, 668, 684, 688, 690, 706, 742, 753, 754, 755, 758, 760, 761	The same results	YES
Building	15, 16, 23, 24	The same results	YES

****

Case 2

User-defined polygon
Coordinates (2632632, 7460016), (2635130, 7460340), (2634733, 7457774)
Image Id: (46106470)
Vector layers used for query: Building- Road- Water

- Search Criterion # 4

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	2, 4, 10, 15, 39, 34, 162	The same results	YES
Road	555, 566, 598, 617, 631, 640	The same results	YES
Water	27, 45, 84, 91	The same results	YES

- Search Criterion # 5

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	2, 4, 10, 15, 16, 17, 30, 31, 32, 33, 34, 35, 36, 37, 38, 153, 154, 155, 156, 158, 162	The same results	YES
Road	555, 558, 559, 562, 565, 566, 568, 572, 573, 578, 598, 601, 603, 606, 608, 609, 616, 617, 622, 623, 624, 625, 631, 64	The same results	YES
Water	27, 45, 48, 57, 58, 84, 91	The same results	YES

The previous query was performed again (using search criterion # 5) but with switching the Building and Road layers in order to make sure that projection Method I and Method II are giving the same results within this function.

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Road	555, 558, 559, 562, 565, 566, 568, 572, 573, 578, 598, 601, 603, 606, 608, 609, 616, 617, 622, 623, 624, 625, 631, 64	The same results	YES
Building	2, 4, 10, 15, 16, 17, 30, 31, 32, 33, 34, 35, 36, 37, 38, 153, 154, 155, 156, 158, 162	The same results	YES
Water	27, 45, 48, 57, 58, 84, 91	The same results	YES

- Search Criterion # 8

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	2, 4, 15, 16, 17, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 43, 153, 154, 155, 156, 158, 162	The same results	YES
Road	558, 559, 562, 565, 568, 572, 573, 578, 601, 603, 606, 608, 609, 616, 622, 623, 624, 625, 641	The same results	YES
Water	27, 46, 57, 58, 84, 91	The same results	YES

User-defined line
Coordinates (2632632, 7460016), (2635130, 7460340), (2634733, 7457774)
Image Id: (46106470)
Vector layers used for query: Building- Road- Water

- Search Criterion # 5

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	3, 10, 15	The same results	YES
Road	555, 566, 598	The same results	YES
Water	27, 45, 91	The same results	YES

- Search Criterion # 6

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	3, 10, 15	The same results	YES
Road	555, 566, 598	The same results	YES
Water	27, 45, 91	The same results	YES

The previous query was performed again (using search criterion # 6)but with switching the Building and Road layers in order to make sure that projection Method I and Method II are giving the same results.

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Road	555, 566, 598	The same results	YES
Building	3, 10, 15	The same results	YES
Water	27, 45, 91	The same results	YES

Case 3

User-defined polygon
Coordinates (682382, 5092839), (682990, 5092446), (682153,5091922), (682691, 5092204)
Image Id: (IKONOS)
Vector layers used for query: Land Cover-Building Part I- Road Part I-Road Part II

- Search Criterion # 4

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Land Cover	NA	NA	NA
Building Part I	102, 113, 139, 205	The same results	YES
Road Part I	46, 828, 834, 837, 847, 857, 871, 874, 879, 887, 888, 904, 907, 1028, 1229, 1239, 1248, 1271, 1279, 1333, 1493, 1494, 1654, 1699, 1703, 1754, 1939, 1973	The same results	YES
Building Part II	NA	NA	NA

- Search Criterion # 9

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Land Cover	137	The same results	YES
Building Part I	NA	NA	NA
Road Part I	NA	NA	NA
Building Part II	NA	NA	NA

User-defined line
Coordinates (681938, 5090612), (682468.46, 5092741)
Image Id: (IKONOS)
Vector layers used for query: Building Part I- Road Part I-Building Part II-Road Part II

- Search Criterion # 4

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Difference (Fid)	Correct Attribute
Building Part I	133, 134, 135, 136, 137, 147, 153	133, 135, 136, 137, 147, 153	134	YES
Road Part I	807, 830, 836, 859, 865, 872, 876, 884, 970, 977, 982, 983, 989, 1154, 1277, 1493, 1494, 1561, 1568, 1669	46, 807, 830, 836, 859, 865, 872, 876, 884, 970, 977, 982, 983, 989, 1154, 1245,1277, 1493, 1494, 1561, 1568,	46, 1245, 1669	YES
Building Part II	NA	NA		YES
Road Part II	181, 547, 551, 560, 613	181, 446, 547, 551, 560, 613	446	YES

- Search Criterion # 6

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Difference (Fid)	Correct Attribute
Building Part I	133, 134, 135, 136, 137, 147, 153	133, 135, 136, 137, 147, 153	134	YES
Road Part I	807, 830, 836, 859, 865, 872, 876, 884, 970, 977, 982, 983, 989, 1154, 1277, 1493, 1494, 1561, 1568, 1669	46, 807, 830, 836, 859, 865, 872, 876, 884, 970, 977, 982, 983, 989, 1154, 1245,1277, 1493, 1494, 1561, 1568,	46, 1245, 1669	YES
Building Part II	NA	NA		YES
Road Part II	181, 547, 551, 560, 613	181, 446, 547, 551, 560, 613	446	YES

It should be noted here that the differences in some of the results obtained in Case 3 are due to the implemented method within this case, as discussed in section 3.4.6. This method is considering the use of 4 Low GCPs Affine transformation and an average height for the area under consideration not the interpolated one. Please, refer to Appendix C for a comparison of the accuracy obtained in dealing with IKONOS GEOCARTERRA™ imagery using different methods.

In order to verify the accuracy that may be obtained if the developed accurate methods for the IKONOS imagery case were implemented, the user-defined line query with the two points (681938, 5090612) and (682468.46, 5092741), as shown above, was performed another time. Within the second query process, the line's two vertices were transformed/orthorectified using the accurate developed method (Average Shift with 3 Mix GCPs and DTM interpolated heights). The resulting transformed point coordinates were used (simulated internally within the program) as the input for the user-defined line vertices to query the vector layers. The query results obtained from this method within IISP matched exactly the ones obtained from ArcView. This result was expected because, as shown previously in Appendix C, the Average Shift with 3 Mix GCPs and DTM interpolated heights method would produce orthorectified/transformed coordinates that are within approximately 5 metres from the actual map/ground coordinates.

2- Results Display:

The results Display functions (Flash all features, Flash one feature, Save results) were tested, by manual and visual inspection, using all the previously listed test cases and provided the correct position and number of the resulting features within the image and within the created ESRI Shape files.

DATA BASE QUERY FUNCTION

1- *The query process retrieves the correct feature based upon the user-defined search expression.*

Case 1

Image ID	Vector Layer Used	Expression	IISP results	ArcView results
464546470	Building	FeatureId>=15 AND AREA/Perimeter > 10	15, 23, 31, 42	The Same Results
464546470	Building	FeatureId>15 AND NOT AREA<800	16, 20, 23, 24, 31, 35, 36, 38, 42, 43, 44	The Same Results
464546470	Road	LENGTH>=500 OR ROAD_NAME>='B'	111, 112, 199, 287, 294, 308, 330, 325, 343, 419, 425, 495, 498, 499, 529	The Same Results

****

Case 2

Projection Method I

Image ID	Vector Layer Used	Expression	IISP results	ArcView results
46106470	River	FeatureId>=175 AND NOT LENGTH<=26.249	175, 176, 177, 178	The Same Results
46106470	Land Cover	AREA/PERIMETRE<2 OR FeatureId<5	1, 2, 3, 4, 182, 97, 177, 191, 217, 218, 219, 220	The Same Results
46106470	Building	FeatureId<=32 AND AREA>=704	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 27, 30	The Same Results

The same search expressions were used with the projection Method II and the same results were obtained proving that both methods will produce the same and correct results for this function

****

Case 3

Image ID	Vector Layer Used	Expression	IISP results	ArcView results
IKONOS	Land Cover	FeatureId>=1598	NONE	NONE
IKONOS	Building	(FeatureId=18 OR AREA<=543) AND NOT PERIMETER<=100	54, 132, 146, 147, 152, 186, 203, 204, 213	The Same Results
IKONOS	Road	ROAD_NAME LIKE 'Q%'	45, 46, 1269, 1270, 1271, 1272, 1273, 1274, 1669, 1940	The Same Results

It should be noted here that:

1- The previously mentioned query expressions correctly produced the resulting features that exist within *only the image extent*, as required by the design requirements for all cases. This has been examined by performing the query within ArcView and inspect the resulting features that exist within the image extent.

2- The system displays the correct attribute categories and values to the selected layer and these attributes are sorted based on the FeatureId.

2- Results Display:

The results Display functions (Flash all features, Flash one feature, Save results) were tested, by manual and visual inspection, using all the previously listed test cases and provided the correct position and number of the resulting features within the image and within the created ESRI Shape files. As concerned with this function, the results obtained were especially tested against the issue of having the query performed within the image extent only and consequently the results were flashed and saved correctly as required.

BUFFERING QUERY FUNCTION

1- The query process retrieves the correct feature based upon the selected IISP case, user-defined buffering core feature(s), and the buffering distance.

Case 1

Image Id: (46456470)
Buffering core feature: Building layer (FID=31)
Buffering Distance: 120 metre

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Road	241, 242, 245, 344, 378, 380, 381, 382, 556, 557	The same results	YES
Building	31, 32	The same results	YES

Image Id: (464546470)
Buffering core feature: Drawn Line (435552.59, 799077 & 436973.3, 798947.44)
Buffering Distance: 75 metre

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Road	242, 301, 302, 330, 344, 382, 495, 497, 529, 530	The same results	YES
Building	15, 16, 17, 29, 30, 36	The same results	YES
Property	24, 31, 641, 671, 688, 717, 725, 732, 733, 734, 735, 736, 737, 738, 739, 742, 743, 744, 745, 746, 747, 755, 763, 764, 769, 772, 773, 774	The same results	YES

Image Id: (464546470)
Buffering core feature: Linear feature with feature's segments connection procedure
Segments' Id: (308, 330, 495, 498, 499)
Buffering Distance: 75 metre

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Road	103, 104, 105, 106, 107, 108, 111, 112, 114, 115, 116, 117, 118, 119, 210, 211, 308, 309, 330, 419, 420, 494, 495, 497, 498, 499, 529, 530	The same results	YES
Property	24, 275, 309, 310, 311, 318, 319, 321, 322, 323, 326, 392, 394,443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 671	The same results	YES

Image Id: (464546470)
Buffering core feature: Group of features stored as an ESRI Shape File
Group Features' Id: Building(20, 21, 22, 23, 24, 34, 35)
Buffering Distance: 50 metre

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Road	269, 270, 271, 272, 273, 298, 334, 345, 353, 368, 370, 373, 374, 377, 378, 379, 446, 584	The same results	YES
Building	20, 21, 22, 23, 24, 34, 35	The same results	YES

****

Case 2

Image Id: (46106470)
Buffering core feature: Building layer (FID=156)
Buffering Distance: 150 metre

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	155, 156	The same results	YES
River	NA	NA	NA

Image Id: (46106470)
Buffering core feature: Drawn Polygon (2633977, 7459782 & 263512.9, 7460340.4 & 2635572, 7458608.6)
Buffering Distance: 120 metre

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	13, 14, 15, 16, 17, 155, 156	The same results	YES
River	45, 46, 57, 58, 91, 27	The same results	YES

Image Id: (46106470)
Buffering core feature: Group of features stored as an ESRI Shape File
Group Features' Id: Building (1, 2, 3, 4, 5, 6, 7, 17, 154, 155, 156)
Buffering Distance: 150 metre

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building	1, 2, 3, 4, 5, 6, 7, 8, 17, 18, 19, 20, 21, 22, 154, 155, 156	The same results	YES
River	91	The same results	YES

For the testing steps in Case 2, the procedure was repeated but with switching layer one and layer two. This was accomplished in order to make sure that both projection methods developed within IISP delivered the same results, which was positively true.

****

Case 3

Image Id: (IKONOS)
Buffering core feature: Building layer part I (FID=115)
Buffering Distance: 25 metre

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Road Part I	449, 843, 844, 865, 872, 873, 1029, 1030, 1031, 1541, 1700, 1701, 1702, 1969	The same results	YES
Building Part I	115	The same results	YES

Image Id: (IKONOS)
Buffering core feature: Drawn Line (682475.2, 5091096.5 & 682213, 5089847.9)
Buffering Distance: 50 metre

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Road Part I	NA	NA	YES
Building Part I	NA	NA	YES
Road Part II	76, 82, 184, 248, 398, 404, 405, 553, 554, 597, 681, 698, 736, 1027, 1509, 1484, 1058, 1544	75, 76, 82, 184, 248, 398, 401, 404, 405, 553, 554, 597, 681, 698, 736, 966, 967, 1027, 1058, 1484, 1509, 1544	YES

Image Id: (IKONOS)
Buffering core feature: Linear feature with feature's segments connection procedure
Segments' Id: (45, 46, 1269, 1270, 1271, 1272, 1273, 1274, 1669, 1940)
Buffering Distance: 100 metre

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building Part I	101, 102, 104, 106, 111, 113, 114, 116, 118, 119, 120, 122	The same results	YES

Image Id: (IKONOS)
Buffering core feature: Group of features stored as an ESRI Shape File
Group Features' Id: Building (102, 113, 114, 115)
Buffering Distance: 100 metre

Vector Layer Used	Results IISP Fid	Results ArcView Fid	Correct Attribute
Building Part I	101, 102, 104, 105, 106, 113, 114, 115, 116, 117, 205	The same results	YES

Finally, the results Display functions (Flash all features, Flash one feature, Save results) were tested using all the previously listed test cases and provided the correct position and number of the resulting features within the image and within the created ESRI Shape files.

As shown within the testing results of this function, Case 3 is the only case that provided different results when using IISP and ArcView. This is due to the fact that the implemented orthorectification technique within this case is based upon using 4 Low GCPs affine transformation and an average height for the area under consideration which is not as accurate as using the interpolated height, as mentioned earlier and as shown in the testing stages in Appendix C.

VITA

Candidate's full name : Mohamed Mahmoud Hosny Abdelrahim

Place and date of birth: Alexandria, Egypt.
26th September, 1970.

Universities attended :	Alexandria University, Alexandria, Egypt.	1987 - 1992 B.Sc.Eng.
	Alexandria University, Alexandria, Egypt.	1992 - 1994 M.Sc.Eng.
	University of New Brunswick, Fredericton, NB., Canada.	1997- 2001 Ph.D.

Publications:

Abdelrahim, M., D. Coleman, R. Casonguay, and D. Raymond (2000). "Compression and Distribution of SNB Orthophoto Map Database." *Proceedings, XIXth Congress of theInternational Society for Photogrammetry and Remote Sensing (ISPRS)*, Amsterdam,The Netherlands Volume XXXIII, Part B2 (CD), June 16-23, pp. 17 – 29.

Abdelrahim, M., D. Coleman, and W. Faig (2000). "Intelligent Imagery System: A Proposed Approach." *Proceedings, XIXth Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Amsterdam,The Netherlands, Volume XXXIII, Part B2 (CD), June 16-23, pp. 11 – 24.

El-Naghi, S. and M. Abdelrahim (1997) "Effect of Additional Parameters on Accuracy of Aerial Triangulation by Bundle Block Adjustment." *Alexandria Engineerig Journal*, Vol.36, No.2.

Hamed, M. and M. Abdelrahim (1994) *"Analytical Aerotriangulation and Its Perspectives." Alexandria Engineering Journal*, Vol.33, No.1.

Abdelrahim, M. (1994) *"Aerial Triangulation Using Bundle Block Adjustment – Merits and NewTrends."* M.Sc.Eng. Thesis, Alexandria University, Alexandria, Egypt.