Korean Journal of Geomatics
Vol. 1, No. 1 / December 2001
pp. 85~93

Software Analysis and Design of the Image Acquisition Subsystem Using the Unified Modeling Language

Jae-Hong Yom*

Abstract

Geospatial database, which is the basis for Geo-Spatial Information Systems, is produced by conventional mapping methods. Recently, with increased demand for digital forms of the geospatial database, studies are carried out to automate its production. The automated mapping system is composed of the image acquisition subsystem, positioning subsystem, point referencing subsytem and the visualization subsystem. The image acquisition subsystem is the most important part of the overall production line because it is the starting point and will affect all subsequent processes. This paper presents a software analysis and design of the image acquisition subsystem. The design was carried out using the Unified Modeling Language which is a modeling method used extensively in the software engineering field.

Keywords : automated mapping system, software design, unified modeling language, image acquisition subsystem, coordinate frame

1. Introduction

It is natural for expectations of the level of automation in mapping systems to grow as technology evolves and new possibilities emerge. Furthermore, as with several other disciplines, the mapping or geomatic (s) discipline (namely cartography, geodesy, photogrammetry, remote sensing and surveying) is obliged to cooperate with others. Examples of other disciplines with which geomatics cooperates include various sub-disciplines of computer science such as computer vision, digital image processing and artificial intelligence.

These expectations and obligations require rapid implementation of new technologies and concepts into working mapping systems, if end users are to benefit as soon as possible. However rapid implementation depends almost entirely upon such systems software design. For a system to be augmented with the latest technologies, it is crucial for any system to have a software design which is flexible in terms of extendibility, compatibility and reusability and also to be reliable in terms of meeting the requirements of the intended users.

The image acquisition subsystem of the mapping system, involves a human operated platform, such as a fixed wing airplane or a helicopter. The imaging sensors are a key component of the imaging subsystem. Some examples of imaging sensors are CCD digital cameras, video cameras, linear array scanners, laser profiling scanners and multispectral scanners.

Photographic processing is a task that will become unnecessary when digital images are acquired directly. It is expected that CCD cameras will be widely used for capturing digital images directly (Ohlhof *et al.*, 1994) (Peipe, 1994). At present though, because of currently available technology, it is still a practice in the mapping community to use film based cameras and to scan the film digitally.

2. Introduction to the Unified Modeling Language with Geomatics Examples

In the computer science discipline, various Object Oriented programming languages have been developed over the years as well as different approaches to analysis and design methodology, using Object Oriented concepts.

UML is, as the name suggests, a modeling language for the Object Oriented methodology. It began as a response to the Object Management Group's (OMG) request for a standard in the Object Oriented methodology. OMG (OMG,

---

*Director, GIS Technology Research Institute, Hanjin Information Systems and Telecommunication (E-mail : jhgeom@hist.co.kr)

The manuscript for this paper was submitted for review on December 10, 2001.

1999) states that UML "is defined as a language for specifying, visualising, constructing and documenting the artifacts of software systems, as well as for business modeling and other non-software systems". The OMG is an international organization supported by more than 800 various members of the computer science discipline. It was founded in 1989 to promote the theory and practice of Object Oriented technology. Later the OMG accepted the UML as the standard (Douglass, 1999). It is stated by some experts that UML will become in time a core skill for software engineers (Pooley *et al.*, 1999). The UML has been adopted as the design tool for this study, because it has been accepted as the standard by the main standardisation body for Object Oriented matters, the OMG.

Features of UML include the Object Model; Use Cases and Scenarios; and, Behavioural Modeling with Statecharts. These features are used in the analysis and design stages of the system to capture domain characteristics accurately and represent them in a standard way. This will serve to link and communicate information to later stages of the implementation, testing and maintenance of the developed system. UML uses various types of diagrams to document and relay views of a model, including :

- Class Diagram
- Use Case Diagram
- Behaviour Diagram
- Statechart Diagram
- Activity Diagram
- Interaction Diagram
- Sequence Diagram
- Collaboration Diagram
- Implementation Diagrams
- Component Diagram
- Deployment Diagram

The following paragraphs will give examples of the basic diagrams in the UML, namely the Class Diagram and the Use Case Diagram.

The Class Diagram is the basic diagram that shows the classes in the system domain. This is shown in Fig. 1. A Class Diagram represents the class as a rectangular box. The box has three compartments. The top compartment labels the name of the class and the second shows the data members. The third shows the member functions or the operations of the class. Templates or parameterised classes, have a dotted rectangle on the right corner to show the arguments the template is taking. Relationships between classes are shown as lines joining the two classes. An arrowline with the closed arrowhead represents an inheritance relationship with the arrowhead at the base class. A line with a diamondhead represents aggregation, with the head at the containing class. A dotted line with the open arrowhead shows a dependency relationship.

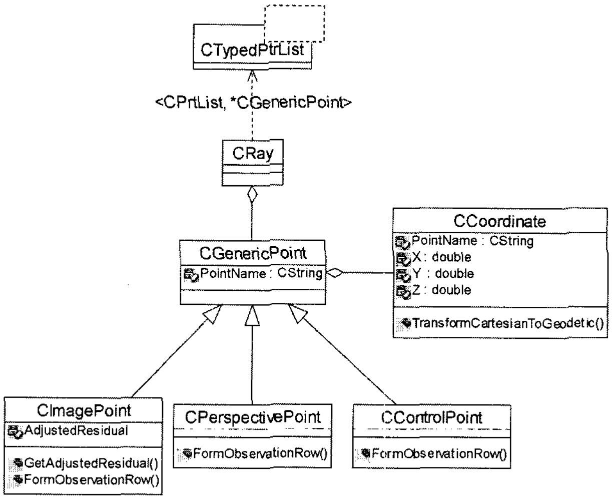

Fig. 1. Class Diagram Example.

There are many kinds of dependency, but the one shown on Fig. 1 is a bind dependency. It explains that the CRay class is a specialization of the template class, i.e. derived from CPtrList class, but because the template actually binds the arguments rather than derives the child class, it uses this special symbol to show such a relationship.

As can be seen from the diagram above, this is a very compact and efficient way of documenting.

Another key feature of the UML is the application of a Use Case Diagram to define what the system would do from a user's point of view. The Use Case Diagrams show how an actor, i.e. object outside the context of the system, would use the system without showing the design structure of the functions of the system itself. Use Case Diagrams are important to communicate the key requirements of the system as well as defining the boundaries of the system to be developed. They are also useful in guiding the direction of the development and to check if development is on track for meeting the user's requirements. The Use Case Diagram of the automated mapping system at its top level is shown in Fig. 2.

Actors are the figures who use the system and each elliptical object is a Use Case, i.e. a case of how the system would be used. A line depicts a relationship of 'uses' or 'extends' or even 'is-a' as in the inheritance relationship. In the figure, there are five actors, i.e. the pilot, the imaging sensor specialist, the positioning sensor specialist, the geospatial database specialist and the GIS visualisation specialist. For example, the pilot uses the automated mapping system to monitor the flight. So Monitor Flight is a Use

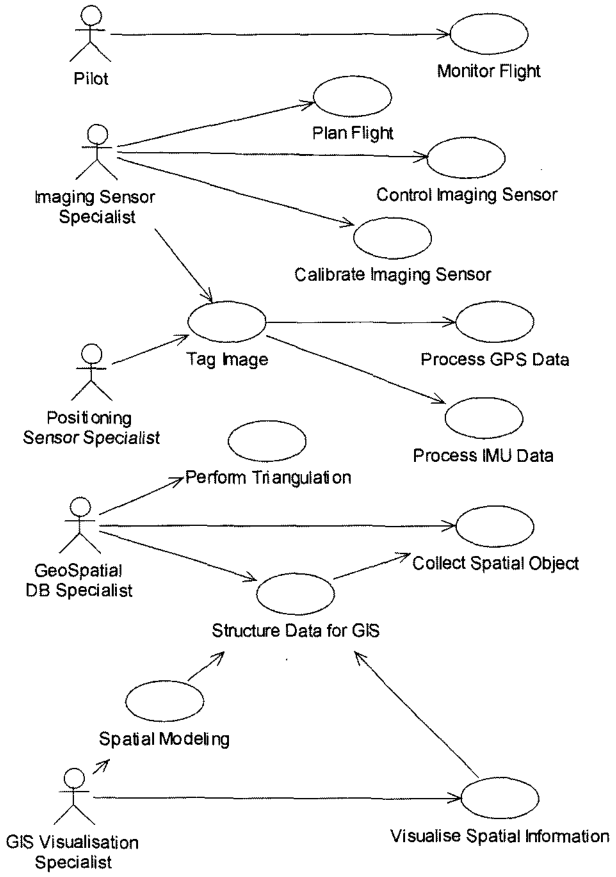

Fig. 2. Use Case Diagram Example.

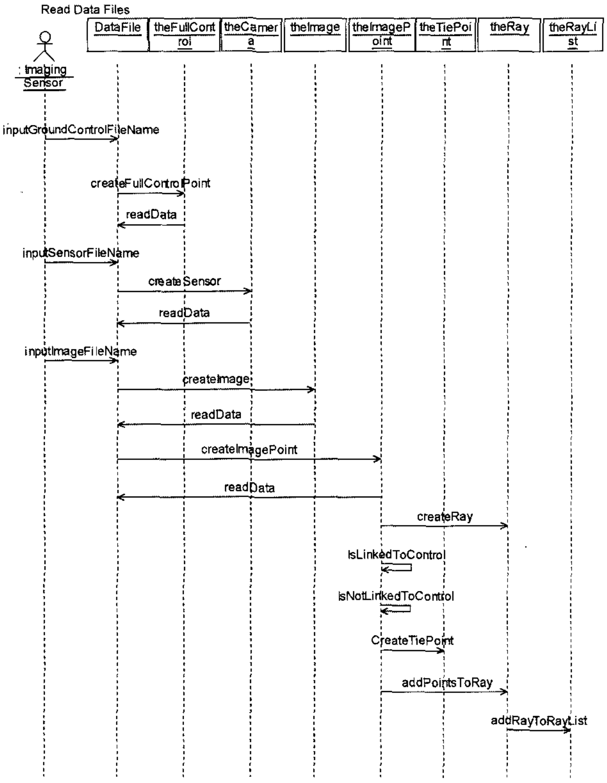

Fig. 3. Sequence Diagram of Read Data File.

Case of the system.

By way of further explanation, in the case of the positioning sensor specialist, he or she will use the automated mapping system to tag the image with position and rotation information. The Tag Image Use Case will in turn use other Use Case modules which will process GPS data and IMU data, namely the Process GPS Data Use Case and the Process INS Data Use Case.

A Sequence Diagram models the dynamic aspects of the system and emphasizes the time ordering of messages that are exchanged between objects. In a Sequence Diagram (for example Fig. 3), objects are represented by rectangles and these are arranged as a row on top of the page. Messages between two objects are represented by a horizontal arrow linking the two objects. The vertical positions of these arrows shows the time ordering of messages.

The Sequence Diagram of Fig. 3 shows the sequence of events between various objects when data files are read during the initial phase of camera calibration. These data files are the sensor data file, image data file and ground control data file. When the file names are keyed in by the user, the data from these files will populate the attributes of objects with corresponding values. These objects are then grouped to form the bundles of rays.

Further details of UML are described in OMG's publication "OMG Unified Modeling Language Specification" (OMG, 1999), but they have not been found essential to this work. The application of UML in analysis and design does not necessarily imply that a better design will be produced compared to when it is not applied. It is a method of documenting what has been perceived by the designer as a result of observation and investigation.

The result of UML could be used as a useful means of discussing and confirming what has been perceived. This will of course improve the understanding of the problem and probably lead to a better design. But in all cases, the initial phase of being able to perceive the problem clearly will be a prerequisite to a good design. It is important to note that software design, or any design, is not a trivial activity and using a methodology or a tool will help the situation but it will not solve the problem completely.

In the analysis stage, the main roles and the processes of the image acquisition subsystem will be explained. The main roles of the image acquisition subsystem are classified as:

- calibration of the imaging sensor;

· control of the image capturing process; and,
· flight planning.

A Use Case Diagram of the image acquisition subsystem is produced as the result of the analysis. After investigation of each role, candidate classes are identified. The identified classes are then further explained with regards to their purpose and some background information will be provided in the photogrammetric sense. Sequence Diagrams will show the actions that are followed in the processes of selected Use Cases.

In the design stage, the Use Case Diagrams and the Sequence Diagrams are then used to produce the Class Diagrams which show classes with all the relevant data members and member functions as well as their relationships with other classes. The design of an automated mapping system involves disciplines including photogrammetry and geodesy, and thus some choices had to be made regarding terminology. In photogrammetry and many other disciplines, 'coordinate system' usually refers to the reference system to which the values of the coordinates of points are referenced, such as 'photo coordinate system' and 'pixel coordinate system'. But in geodesy it is conventional to use 'coordinate frame' for such purpose. 'Coordinate system' in geodesy refers to the coordinate frame as well as the physical theories and their approximations that are used to define the coordinate axes (Jekeli, 1998). For consistency purposes, the term 'coordinate frame' will be used in this thesis to refer to what would be called 'coordinate system' in photogrammetry. Therefore, terms such as 'photo coordinate frame', 'pixel coordinate frame' and 'local coordinate frame' will be used.

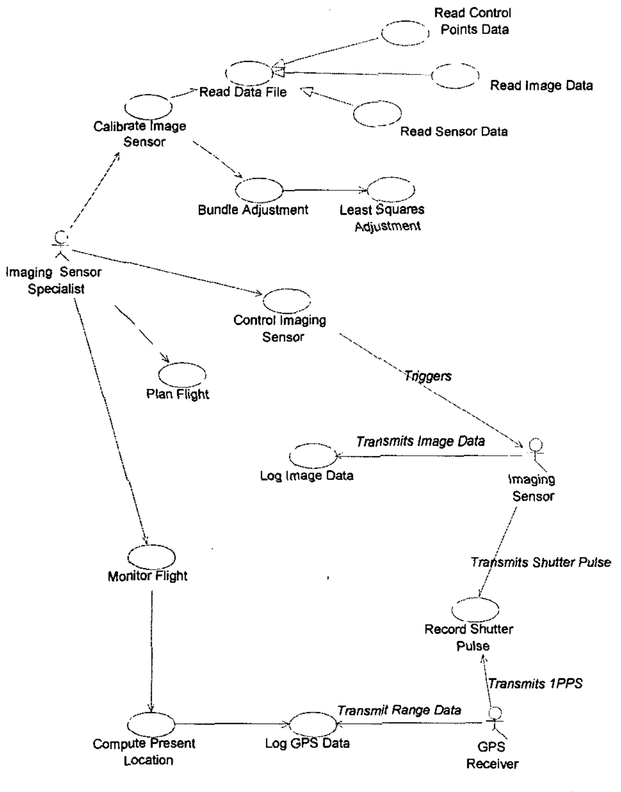

Fig. 4. Use Case Diagram of the Image Acquisition Sub-System.

The description of the roles and the processes of the image acquisition subsystem given above can be summarised by the Use Case Diagram as shown in Fig. 4. This diagram shows how different actors (i.e. human operators, imaging sensors and GPS receivers) would interact with the image acquisition subsystem. It also shows that the system would be used to calibrate image sensors, control imaging sensor and to plan and monitor the flight.

3. Analysis and the Design of the Imaging Subsystem

The key elements of the Imaging Subsystem are the imaging sensor, the image, the points and the coordinate frames. These 'key elements' most probably will later be identified as classes together with many other related classes. The following paragraphs will provide an overview and the status of these important objects. This information will be used in the design of the related classes.

3.1 The Imaging Sensor

Basically sensors can be seen as devices which convert a form of energy to an electrical signal. In a CCD camera the photo-site performs this transformation from the incident light to a number representing the light intensity at certain location of the scene. In a GPS receiver, the signals transmitted from the GPS satellites are transformed to pseudoranges and phases. In an IMU, the inertia created by movement is transformed to change in rotation and acceleration.

An imaging sensor is here defined as a type of sensor which produces images of the 3D scenes which include the topographic features of the earth. There are many types of imaging sensors which are used for mapping purpose. Some, which are currently used or are being researched are listed below.

· Linear CCD Array camera
· Matrix camera
· Traditional frame camera
· Matrix CCD camera
· Active Pulse sensors
· Imaging radar system
· Laser Scanner

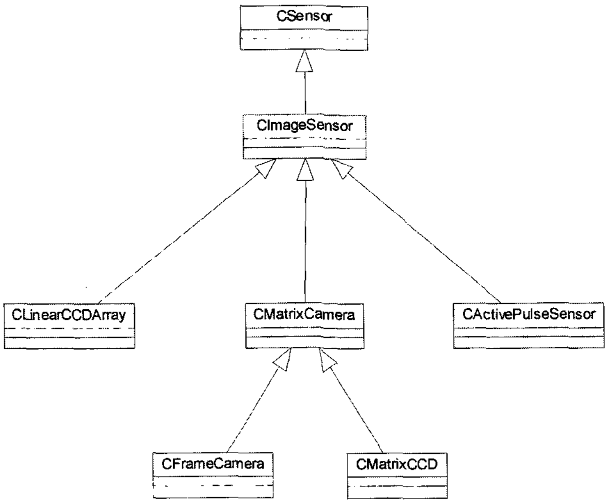

Fig. 5. Class Diagram of Imaging Sensor.

All these sensors use different mathematical models and have different processing methods. But they all share the common characteristic that they produce an image data set. This image data set is the result of a transformation by the sensor of the 3D scene to an image representing the scene. It is not attempted to model all these sensors in this research, and only the traditional metric camera and matrix type CCD camera will be modeled.

It is anticipated, however, that addition of a new sensor to the mapping system will involve software development tasks which will be simpler by using the inheritance and the encapsulation facilities of the object oriented design. For example a new sensor class could be derived from the generic sensor base class.

A class diagram of the imaging sensor showing only the class names and their relationships with other classes is shown in Fig. 5.

The traditional film based camera is still widely used in the mapping community. Images are reproduced on films and then transformed into a digital image using a digital scanner. The main reasons for this are the problem of digital cameras with regard to accuracy and the speed of image capture (Beyer, 1992) (King, 1995) (Mills *et al.*, 1996).

The geometric accuracy of the digital camera is inferior to the film based camera mainly because of the size of the imaging area, the sensor size. A normal aerial film is about 23x23 cm. A typical CCD sensor of a digital camera could have 1548x1032 pixel's with each pixels dimension being 4.85 micronsx4.85 microns. This would mean that the sensor size is about 7.5x5 mm. The poor geometry of rays due to the small imaging area results in a decreased performance in accuracy. To produce a digital CCD sensor with size similar to that of an aerial film introduces many problems such as irregularity in the flatness of the sensor plane, difficulty in management of the sensors, i.e. locating and exchanging faulty photo-site chips, and the speed of image capture, which is a major problem.

For a digital camera to capture two consecutive images, the photo-sites have to be refreshed and the quantized values transferred to a recording media. This is a time consuming task with the current technology. This limits the frame rate, i.e. the time necessary between two consecutive image exposures, effecting the flying speed of the aeroplane and the overall cost of the image capturing process.

However, where frame rate is not a major concern, and the distance of the object is relatively close, such as with most non-topographic photogrammetry, the problems mentioned above have little effect and the digital camera is suitable in applications such as industrial monitoring, architectural and historical monument preservation and restoration, robot vision and medical imaging (Sheffer *et al.*, 1989) (Waldhausl, 1992) (Streilein, 1995) (Peipe, 1995).

3.2 The Image Class

Images are produced by an imaging sensor. An image is composed of grids of data with a spatial component and its corresponding thematic component. The thematic component can be light intensity as in a camera or height computed from travel distance of sensor signals as in the radar and laser sensors.

If the grid is two dimensional, the image is a two dimensional image. If the grid is three dimensional, it is a three dimensional image. Each grid point or pixel in the case of an image from a digital camera will have a set of coordinates in the grid coordinate frame as its spatial component and a corresponding thematic value.

Two types of two dimensional images have been considered, i.e. CMappingImage and CDisplayingImage. Although both refer to one image of a scene taken by an imaging sensor, they differ in their responsibilities and roles. CMappingImage is mostly concerned with the geometric aspects of the image and is mostly used at the initial stage of photogrammetric data processing. CDisplayingImage is used mostly to display, zoom and pan images. It is also used for changing image formats. It is most likely that CDisplayImage will use the end product of CMappingImage in the visualisation subsystem.

Images captured by the imaging sensor can have a reference point and an attitude. In the case of a frame camera the reference point is the perspective point and the attitude is the attitude of the camera at instant of capture.

In the case of a laser profiler, it is difficult to determine the optimum design for the signal of the laser sensor.

In the first case, the signal itself could be defined as an image, in which case the image will be a single point with X, Y, Z and attitude and the travelled distance. This means it could be defined as a one dimensional image. This will be advantageous for the process of calibrating the image acquisition subsystem, because the computation of the reference point and the attitude is common for all images. This implies that a function for the computation of camera attitude computation can be defined at the generic class of image and it will be applicable for camera as well as the laser profiling scanner.

In the second case, the constructed two dimensional image from the laser signals could be defined as an image and the signals as a list of embedded objects. This will be advantageous in the subsequent process of extracting spatial objects from various types of images, because most of the images at this stage are two dimensional. However, the member function for computing the reference point and attitude of an image class is repeated in another class, the signal class. This means that if a change occurs in the computation of the reference point and the attitude, then the code in both the image class as well as the signal class should be updated. This inconsistency is something that the object oriented method aims to minimise.

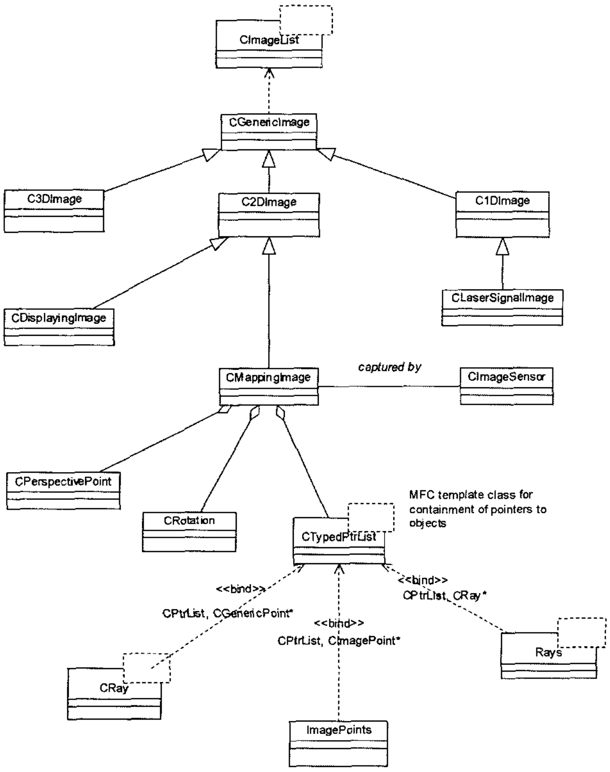

Fig. 6. Class Diagram of the Image Class.

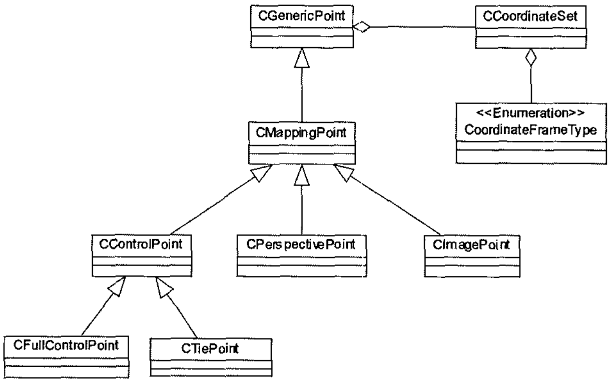

Fig. 7. Class Diagram of Point Class.

In this research the first case will be selected for purposes of consistency, but more research into the design for incorporating various sensors is needed to come up with an ideal solution. A class diagram of the image class and its relationship with other classes is shown in Fig. 6.

3.3 The Point Class

A point is defined by its coordinates in a coordinate frame. The coordinates can be transformed between coordinate frames with the transformation parameters. The coordinates are usually computed values from observations and these coordinates usually have a set of associated numbers representing their accuracies. Also points are usually given a unique identifier to distinguish one from another.

Some examples of points used in the mapping systems are image points, control point, tie points, and perspective points as shown in Fig. 7.

4. Coordinate Frames

The transformation between different coordinate frames is a common problem encountered in many mapping applications. The most important functions involved in coordinate transformation are scaling, rotation and translation.

Although the coordinate frame itself is important in understanding many of the theories in mapping applications, in software implementation, the transformation between different coordinate frames is of great concern. The transformation itself is incorporated into various mapping processes. For example, in the bundle adjustment of photogrammetric observations, the rotation matrix is formed and its elements are used in the forming of the collinearity equations. In IMU data processing, rotation matrices are also formed to transform observations from a body coordinate frame to a local coordinate frame, or from an earth-centred-earth-fixed coordinate frame to a body

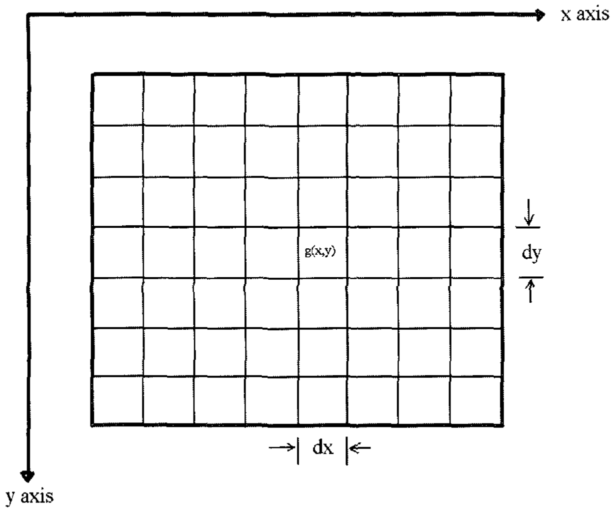

Fig. 8. Pixel Coordinate Frame.

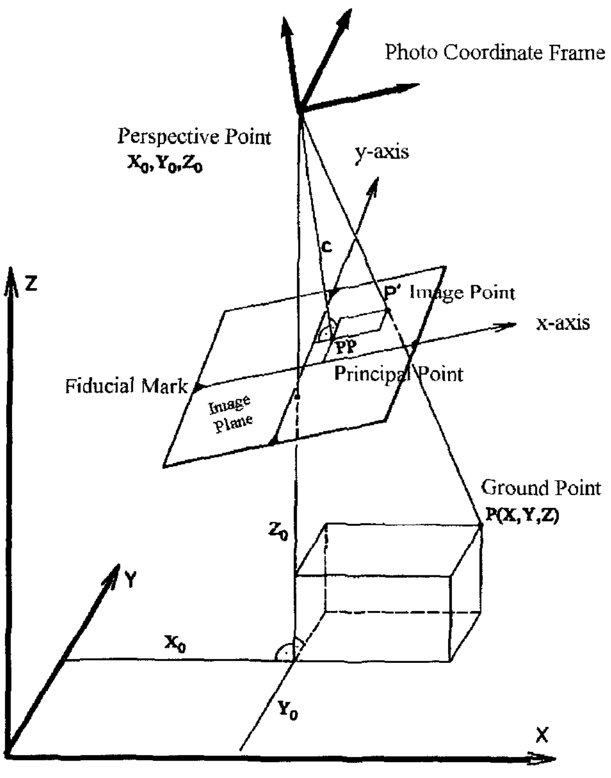

Fig. 9. Photo Coordinate Frame.

coordinate frame.

A separate class of coordinate frame will not be defined in this design, as the transformation itself is realised in other classes in the form of observation equations and rotation matrices. Instead a coordinate frame type will be associated with the coordinate set to signify its coordinate frame type.

The pixel coordinate frame, shown in Fig. 8, is a 2 dimensional coordinate frame usually defined with it origins on the top left corner of the image and its x axis extending right and the y axis extending down. The photo coordinate frame is a 3 dimensional right hand Cartesian coordinate frame and has its origin at the perspective point.

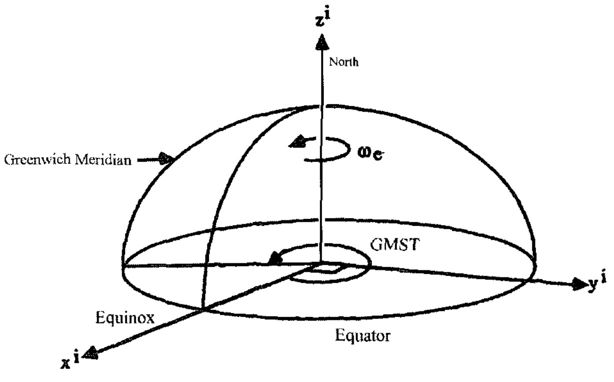

Fig. 10. Inertial Coordinate Frame.

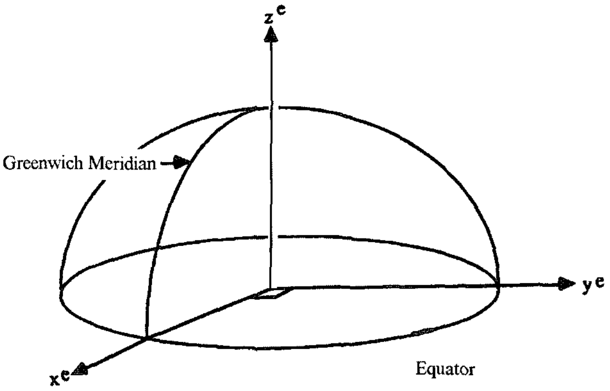

Fig. 11. Earth-Centred-Earth-Fixed (ECEF) Coordinate Frame.

The photo coordinate frame as shown in Fig. 9, has its origin at the perspective point. The x-axis is usually taken to coincide with flight direction and the y-axis is perpendicular to the x-axis in the same horizontal plane. The principal point is the point on an image plane where an image would be formed if the image plane was perpendicular to a direct axial ray coming through the centre of the lens. The principal distance is the perpendicular distance from the perspective point of the lens to the image plane. The fiducial centre is the point on an image plane where lines from opposite fiducial marks intersect.

The following coordinate frames are mostly used in the computation of rotation and position from IMU observations (Schwarz *et al.*, 1994).

The inertial coordinate frame, or i-frame, is a frame which does not rotate or accelerate (Fig. 10). The origin is at the mass centre of the earth; the x-axis points towards the mean vernal equinox; the z-axis points towards the north

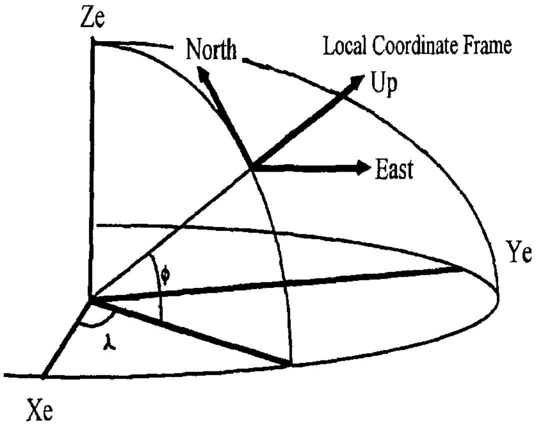

Fig. 12. Local Coordinate Frame.

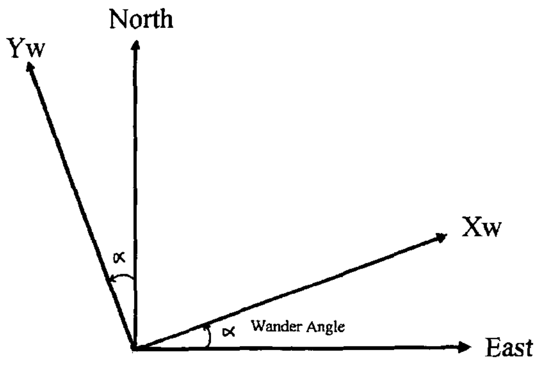

Fig. 13. Wander Coordinate Frame.

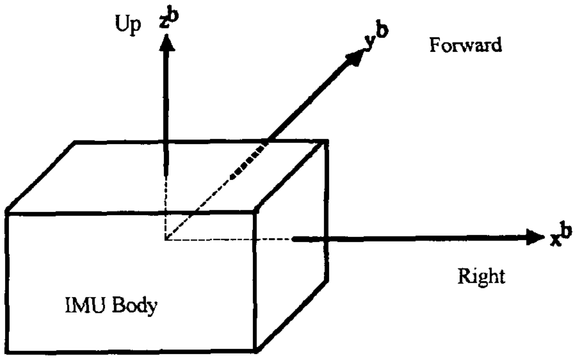

Fig. 14. Body Coordinate Frame.

celestial pole; and, the y-axis completes a right-handed system. The International Earth Rotation Service (IERS) continually establishes the inertial system through geodetic observations of quasars.

The Earth-Centred-Earth-Fixed (ECEF) coordinate frame, or e-frame, also has its origin at the mass centre of the earth as in i-frame, as shown in Fig. 11. The x-axis points towards the Greenwich meridian; the y-axis makes a 90° with the x-axis in the equatorial plane; and, the z-axis is taken as the axis of rotation of the reference ellipsoid. The e-frame is also the frame that is used by GPS.

The local coordinate frame, or l-frame is shown in Fig. 12. It has its origin at the topocentre; the y-axis or the North axis points north, tangent to the geodetic meridian; the x-axis or the East axis points to the East; and, the z-axis or the Up axis points upwards along the ellipsoidal normal.

The wander coordinate frame, or w-frame, is used in geographic areas of high latitudes to complement the l-frame (Fig. 13). The l-frame needs always to orient itself to the north, which involves large rotations in areas of high latitude. The wander coordinate frame resolves this by setting the y-axis at an arbitrary direction. The angle between the y-axis and the north axis is called the wander angle.

The body coordinate frame, or b-frame, is shown in Fig. 14. It is the coordinate frame to which the IMU observations are referenced. It has its origin at the centre of the IMU, the exact location of which should be given by a specification of the manufacturer. This position is later needed when offset distances between different sensors are surveyed. The x-axis points towards the right side of the IMU; the y-axis points towards the front of the IMU; and the z-axis points upwards.

4. Conclusions

This study has been undertaken to produce a software design for the image acquisition subsystem from a Geomatics point of view.

The processes involved in the acquisition of images have been analysed. Concepts used by domain experts (i.e. photographers and photogrammetrists) have been investigated.

As the result of the investigation, a Use Case Diagram of the Image Acquisition Subsystem was produced. Some the important Use Cases of this subsystem that were identified include Calibrate Image Sensor, Control Imaging Sensor, Plan Flight, Monitor Flight, Log Image Data and Record Shutter Pulse. These Use Cases give a good overall view the various situations that this subsystem will be used for.

Based on this Use Case Diagram, further analysis for each Use Case was carried out.

The image sensor class was designed with the consideration that various types of image sensors exist today and that more will be produced in the future. To accommodate this, three types of image sensor classes were derived from the base class CImageSensor, namely, the CLinearCCDArray class, the CMatrixCamera class and the CActivePulseSensor class. The CMatrixCamera was then reclassified into CFrameCamera and the CMatrixCCD camera. Other new types of image sensors can be derived from the CImageSensor in the future.

The image class was also designed with the fact that the automated mapping system will have to deal with various types of images in the future. A very polymorphic CGenericImage is at the highest level of abstraction. It is the base class of all forms of images. This generic image is classified into C1DImage, C2DImage and C3DImage classes. The normal images that we use such as the aerial image would be an object of the C2DImage class, whereas the output data from the CActivePulseSensor would belong to the C1DImage class. A three dimensional perspective view generated from stereo images would be an example of the C3DImage class. However for this thesis, the CMappingImage class is fully developed. The CMappingImage is derived from the C2DImage class and contains data members and member functions to perform the camera calibration and the bundle adjustment.

The point class too has a CGenericPoint class which has CCoordinateSet as its data member. The CMappingPoint is derived from this CGenericPoint base class and is used to represent all points used for mapping purposes. CControlPoint, CPerspectivePoint and the CImagePoint classes are derived from the CMappingPoint.

It is the objective of this research that the software design will lay a foundation for the development of automated mapping software. It is expected that this effort will benefit both the geomatics researchers who wish to program and implement their ideas and the software engineer who wishes to acquire geomatics domain knowledge. It is also expected that through refinement of the design and the addition of more classes and functions to the design, the common starting point of the development of an automated mapping system will be the software design that has emerged from this study, thereby enabling maximum software reuse and compatibility which will in turn provide quick implementation of new technologies for mapping purposes.

References

1. Ohlhof, T., and Kornus, W. (1994). "Geometric Calibration of Digital Three-Line CCD Cameras". *ISPRS Commission I Symposium*, 30(1), 71-81.
2. Peipe, J. (1994). Photogrammetric Calibration and Performance Test of Still Video Cameras". *ISPRS Commission I Symposium*, 30(1), 108-113.
3. OMG. (1999). OMG Unified Modeling Language Specification. 23-23. Ref Type: Report.
4. Douglass, B. P. (1999). "Introduction to Objects and the Unified Modeling Language". *Doing Hard Times - Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns.* Addison Wesley Longman, 3-55.
5. Pooley, R.J., and Stevens, P. (1999). *Using UML : Software Engineering with Objects and Components.* Addison-Wesley.
6. Jekeli, C. (1998). "Coordinate Frames and Transformations". *Inertial Navigations Systems with Geodetic Applications-Course Notes.* Dept. of Civil and Environmental Engineering and Geodetic Science, The Ohio State University, 1-1-1-23.
7. Beyer, H.A. (1992). Geometric and Radiometric Analysis of a CCD-Camera Based Photogrammetric Close-Range System. 18-42. Institut fur Geodasie und Photogrammetrie. Ref Type: Thesis/Dissertation.
8. King, D.J. (1995). "Airborne Multispectral Digital Camera and Video Sensors: A Critical Review of System Designs and Applications". *Canadian Journal of Remote Sensing*, 1995, 21 (3), 245-273.
9. Mills, J.P., Newton, I., and Graham, R.W. (1996). "Aerial Photography for Survey Purposes with a High Resolution, Small Format, Digital Camera". Photogrammetric Record, 15 575-587.
10. Sheffer, D.B., and Herron, R.E. (1989). "Biosterometrics", Karara, H.M., Non-Topographic Photogrammetry. American Society for Photogrammetry and Remote Sensing, 359-366.
11. Waldhausl, P. (1992). Defining the Future of Architectural Photogrammetry. 29(B5), 767-770. International Archives of Photogrammetry and Remote Sensing. Ref Type: Conference Proceeding.
12. Streilein, A. (1995). Videogrammetry and CAAD for Architectural Restitution of the Otto-Wagner-Pavillon in Vienna. "Optical 3-D Measurement Techniques III-Applications in Inspection, Quality Control and Robotics". Editors Gruen, A. and Kahmen, H. 305-314. Vienna, Wichmann. Ref Type: Conference Proceeding.
13. Peipe, J., and Schneider, T. (1995). "High Resolution Still Video Camera for Industrial Photogrammetry". Photogrammetric Record, 15 135-139.

14 Schwarz, K.P., and Wei, M. (1994). ENSU 623 Lecture Notes. Department of Geomatics Engineering, The University of Calgary.