CEng 491 Requirement Analysis Report

Fall 2007

pix‘r’us

Ebru Aydin 1394691

Berk Demir 1448588

Ender Erel 1395029

M.Ozan Kabak 1389568

1. INTRODUCTION

1.1. Background

Geographic Information Science is an emerging field that has found usage in various areas such as resource management, asset management, environmental impact assessment, urban planning, cartography, criminology, history, sales, marketing and logistics. This emerging field benefits from many enabling technologies; aerial imaging, satellite imaging and GPS (Global Positioning System) being the most remarkable examples.

Efficient usage of these enabling technologies has become an important engineering problem in the last two decades. In this report we present the requirements analysis of proposed photogrammetry software which processes the information generated by these enabling technologies to produce required outputs. Photogrammetry is concerned with obtaining reliable and accurate measurements from imaging.[1] Photogrammetry is a technology in which geometric properties are determined from photographic images.

1.2. Problem Definition

This project consists of the task of developing software which produces:

1. *Digital Elevation Map (DEM)* of an area.
2. *Ortho-rectified imagery (orthophoto)* of an area.
3. A complete map of the area by producing an *image mosaic*.

The software will use the following input to generate the mentioned output:

1. Aerial images of an area.
2. External / Internal camera calibration parameters. External camera calibration parameters consist of the position of the camera (in geographic coordinates)

and the orientation of the camera (roll, pitch, yaw angles). Internal camera calibration parameters consist of focal length, aspect ratio, image center coordinates and lens distortion coefficients.

In addition to the fundamental functionality presented above, the following extra capabilities may be included if the core functionality can be implemented ahead of the planned schedule:

1. Super resolution image generation from multiple images or videos.
2. Generation of a mosaic image using video frames.

1.3. Project Goals and Scope

The main goals of this project are given below:

1. The traditional approach in photogrammetry software is to provide a separate tool for each task. Instead of using this approach, all relevant photogrammetry functionality is aimed to be incorporated under one umbrella.
2. The software is planned to be compatible with a broad range of input (image) formats.
3. The software will come with a complete and descriptive documentation including examples for each feature.
4. An intuitive and user friendly GUI will be provided.

The scope of this project involves:

1. Analysis and documentation of various photogrammetry tools available in the market.
2. Development of a detailed user requirements specification.
3. Design of an appropriate photogrammetry tool satisfying the necessary requirements.
4. Implementation of the tool.
5. Documentation of the software product.

2. SOFTWARE DEVELOPMENT PROCESS

2.1. Software Development Process Model

In order to choose the optimal software development process model for this project, one has to consider the following fact: Modules of the software depend on each other in a linear fashion; most of the modules use other –less complicated– modules to produce their output. Hence, the development process is inherently not very suitable for parallelization. For example, in order to produce good quality orthophotos, a DEM needs to be constructed first.

Taking the facts presented above into account, we see that a linear model (such as the waterfall model) is most suitable for our project. However, it is a well established fact that rapid prototyping is a very effective model for software involving computer graphics and image processing. Thus, it is apparent that a hybrid model incorporating relevant aspects of both of these models is needed. Thus, we chose the **iterative waterfall model (iterative linear sequential model)** as our software development model.

2.2. Team Organization

The software involves solutions of 3 hard problems (DEM generation, orthophoto generation and image mosaic). All three of these hard problems involve literature research and implementation of complicated algorithms. Thus, one can conclude that although the software is not very complicated in terms of size, all constituent problems are hard. Given this context, team organization in this project is chosen to be **"Democratic Decentralized" (DD)**, as this organization scheme is most suitable for small groups working on hard problems.

2.3. Major Constraints

2.3.1. Project Deadline

The software must be delivered by the end of May 2008. Thus, the project schedule has to be fitted in 7 months. A detailed time schedule in the form of a Gantt chart is given in the appendix.

2.3.2. Programming Language Constraints

The contractor has required the programming language to be either C++ or Java. Java was discarded by our group as none of the group members are experienced in Java. In addition, the contractor required the GUI toolkit to be wxWidgets if C++ is to be used.

2.3.3. Data Constraints

There are well defined file formats specifying images containing geographical information. Digital elevation maps are also stored in files with well defined formats. Our aim is to support most of these file formats for both input and output purposes. The test inputs that will be used in the implementation and testing processes will be provided by the contractor.

2.3.4. Execution Speed Constraints

Execution speed is not a high priority issue for the fundamental features of the software. However, the optional video mosaic feature requires fast execution; so real time algorithms may need to be incorporated into the software.

2.3.5. User Interface Constraints

The contractor has requested some specific features (such as multiple document interface windows) to be implemented in the graphical user interface. The GUI toolkit to be used is wxWidgets (as requested by the contractor.)

3. LITERATURE SURVEY AND RELEVANT RESEARCH

3.1. Existing Photogrammetry Software

We decided to investigate existing third party software and the contractor has advised the following tools:

- Leica Photogrammetry Suite
- Regeemy
- Geometica
- IDL/ENVI

Regeemy is a tool for creating mosaic images. It has a very simple and easy to understand graphical user interface. We are planning to incorporate this feature into our final product. Other tools have very confusing interfaces, which is a consequence of having more functionality about geographical usage. On the other hand, they have other user interface functionalities that increase convenience of the interface.

Leica is a photogrammetry suite that supports many file formats used for representing images. This makes it one step ahead from others.

IDL/ENVI is a tool which is capable of a wide range of functionalities. It mainly has two parts; data analysis and data visualizations. Data analysis consists of mathematical and statistical analysis, image processing, signal processing and other various tools. Data visualization includes animations and graphic view of the environment using DEMs, image processing mappings etc among the others that our software will include.

3.2. Literature Survey

3.2.1. Digital Elevation Maps

DEM is a digital representation of ground surface topography or terrain. DEMs are used often in geographic information systems, and are the most common basis for digitally-produced relief maps. During our research we found that DEMs are most commonly generated from stereo images using calibration parameters.[2] Information about one point coming from two images are used to calculate the position of the point in the third dimension, namely elevation.[1] On the other hand, in our case, we cannot use conventional DEM generation algorithms due to lack of stereo images. We have to customize existing algorithms to suit our needs.

3.2.2. Image Registration

In the most general context, one can define image registration as the process of transforming the different sets of image data (each in its respective coordinate system) into one coordinate system. Registration is a necessary procedure in order to be able to integrate the data obtained from different measurements. In the context of photogrammetry, image registration can be more precisely defined as the process of transforming the image coordinates to world coordinates by the use of external parameters. This transformation is achieved by calculating a transformation matrix, which conveys the information of how a pixel coordinate in the image is transformed into a world coordinate. The transformation matrix is calculated using the external parameters and the input image itself.

The actual process of image registration depends on the external parameters available. If the images input images are orhtorectified and are supplied with worldfiles (worldfiles carry the world coordinate of a reference point on the image); the transformation matrix is just a linear transformation, whose calculation is trivial.

If the input consists of raw images with camera parameters (camera's real world coordinates and orientation), the process is more complicated. The transformation matrix is actually a back-projection matrix in this case. To be able to calculate this matrix, one needs altitude information. If this information is available (a DEM is present), then it can be used. If not, an average altitude is assumed for the calculation. Of course, the results obtained in the latter case have some error.

One other case is that only the raw images are available as input. Since there is no camera coordinates available, it is impossible to register the images according to world coordinates. Nevertheless, the images can be registered with respect to each other. This is achieved by detecting the overlapping areas in the images and rotating (and scaling) the images appropriately so that they are in the same coordinate space. This overlapping area detection process is actually performed in the following manner: Feature detection procedures (most commonly implementations of Harris corner detection algorithm) detect a number tie points in the images. Alternatively, the tie points can be manually designated as well. Then, the tie points are used to solve a system of linear equations to get the elements of the transformation matrix.

One interesting thing to note is that these steps also constitute the first steps of image mosaic construction.

3.2.3. Orthophoto Generation

An orthophoto is an aerial photograph that has been planimetrically corrected to remove distortion caused by camera optics, camera tilt, and differences in elevation, meaning that the photo can be considered equivalent to a map. In an orthophoto, all points seem to be perceived at a right angle, which can be considered to be an exact representation of the earth’s surface. Our research in this field revealed that best orthophotos are generated using digital elevation models. This supports our linear development model.

3.2.4. Mosaic Images

A mosaic image is an image which is constructed from a set of constituent images with nonempty overlapping areas. These constituent images are combined to make up the mosaic image which includes all relevant information in the constituent images.

In order to combine the images, the images are first registered. Then; by using the transformation matrices the images the images are superposed onto each other to get the mosaic image. One important issue to take into account is that various image parameters (brightness, contrast etc.) of the constituent images may be different from one another. To generate a mosaic without "stitch marks", the constituent images should be processed to address this issue. One other important thing to note is that, mosaic construction becomes much more successful when the constituent images are orthophotos.

3.3. Meeting With Milsoft

In our meetings with Milsoft (our contractor), they gave us a presentation about the features that will be implemented in this project. We were given detailed information about the requirements of the graphical user interface of the project. Also additional features like video mosaicking that can be added to the project were mentioned during this presentation. The concept of world files was introduced to us. World files are associated with orthorectified image files and they contain coordinate data about geographical information contained in the image file.

Regarding our contractors request, we are planning to implement our first release which will be a basic image viewer with additional features. World files are also to be included in this first release.

4. PROJECT REQUIREMENTS

4.1. User Interface Requirements

Software will use one user interface with a main window, which consists of an MDI (Multiple Document Interface) with a toolbar, menu bar and a status bar.

MDI will construct multiple child windows which will enable a user friendly interface for multi image operations such as image mosaic and DEM construction. MDI will offer a mechanism to bind relevant images so that it will be possible to perform operations on relevant images synchronously. Displaying thumbnails and managing auxiliary windows will also be handled by the MDI.

The toolbar will enable the user to perform various tasks on an image with a single click. The toolbar will include controls to handle the following:

- Brightness
- Contrast
- Sharpening
- Zoom
- Rotate
- Roaming by a predefined pattern

A toolbar may look like the one below in Figure 1;

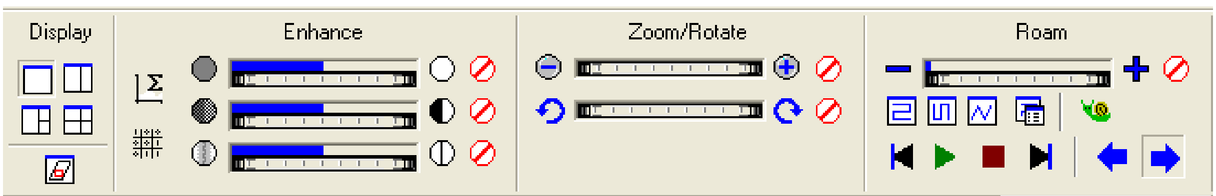


Figure 1

The menu bar will include various submenus for controls such as;

- File operations (Open / Close / Save / Edit / etc.)
- DEM operations (Creating DEMs / etc.)
- Orthophoto operations (Creating Orthophotos / etc.)
- Mosaic menu (Creating Mosaics / etc.)
- Options menu (Preferences / controlling options)
- Help menu

And optionally;

- Super resolution operations
- Video based photogrammetry operations

A menu bar may look like the example given below;

The status bar is used for;

- Displaying feedback messages to the user about:
  - Process status
  - Error messages
  - Estimated operation time
  - Time and date information
- Geographical information obtained from active images' world file.

A sample status bar may look like the one below in Figure 2:

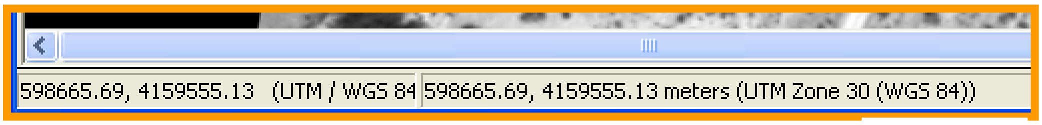


Figure 2

Screen layout will be kept simple and clean for fast and easy use of the software. Menu bar will be placed at the top of the main screen, and the toolbar will be below the menu bar. At the bottom of the screen status bar is placed. The region between the status bar and the toolbar is reserved for MDI.

For the commonly used operations keyboard shortcuts will be placed next to the toolbar; for fast use of the software by the experienced users.

General user interface screen may look like the one below:

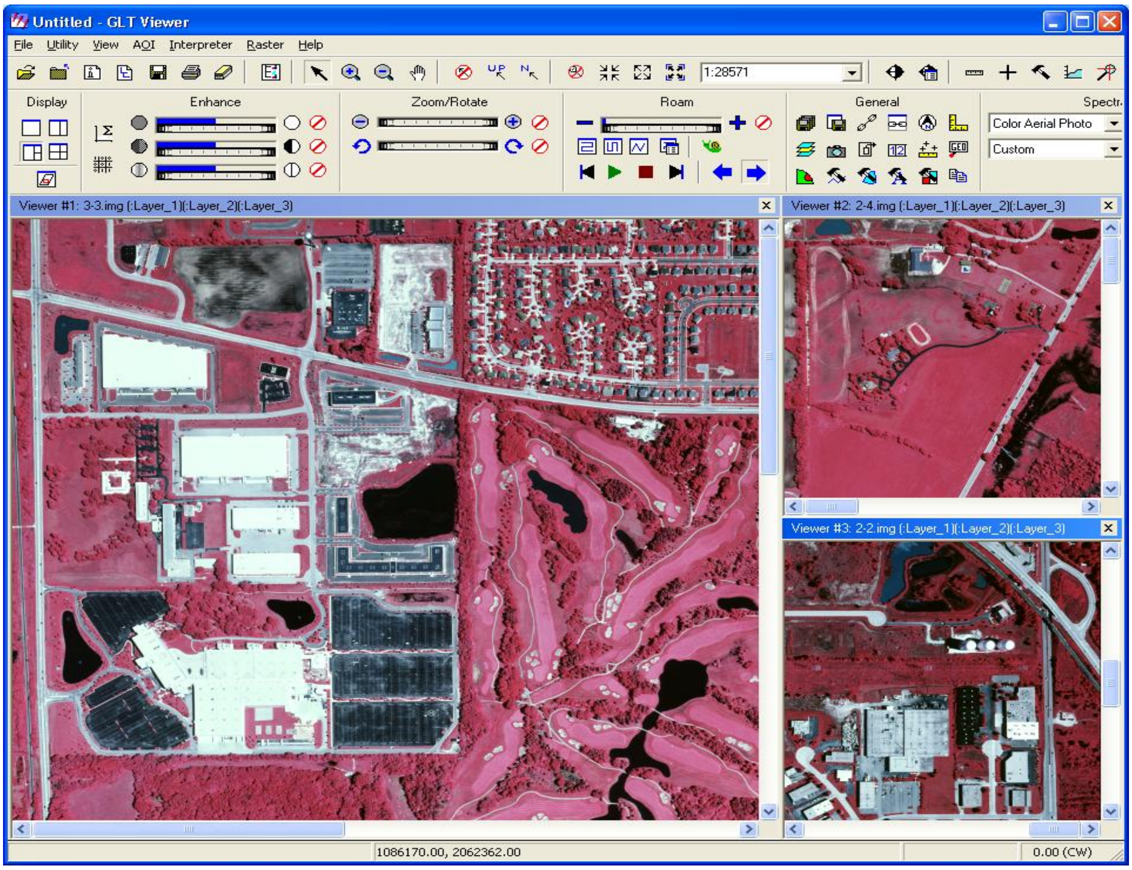


Figure 3

4.2. Input / Output Requirements

To maximize usability and compatibility our software will support the most common image file formats for input & output purposes. Currently, we are planning to support the following file formats:

Description	*Extension*
Microsoft Windows Device Independent Bitmap	.bmp
TIFF / GeoTIFF	.tif , .tiff
JPEG JFIF	.jpg
JPEG2000	.jp2 , .j2k
Netpbm	.ppm , .pgm
Portable Network Graphics	.png
X11 Pixmap	.xpm
USGS Dem format	.dem
Spatial data transfer standard SDTS	.ddf

4.3. Hardware & Software Requirements

Our software is aimed for users from both Linux and Windows platforms; therefore it will be cross-platform.

Any hardware that these operating systems run on will be sufficient.

5. SYSTEM ANALYSIS AND MODELLING

5.1. Structured Analysis – Function Model

In this section data flow diagrams will be introduced in two levels, Level 0 and Level 1. Level 0 represents major input output data. In Level 1 modules are described more detailed with input/output.

5.1.1. Level 0 of DFD

Level 0 of DFD is shown in Figure 4; it is an overview of the system in general.

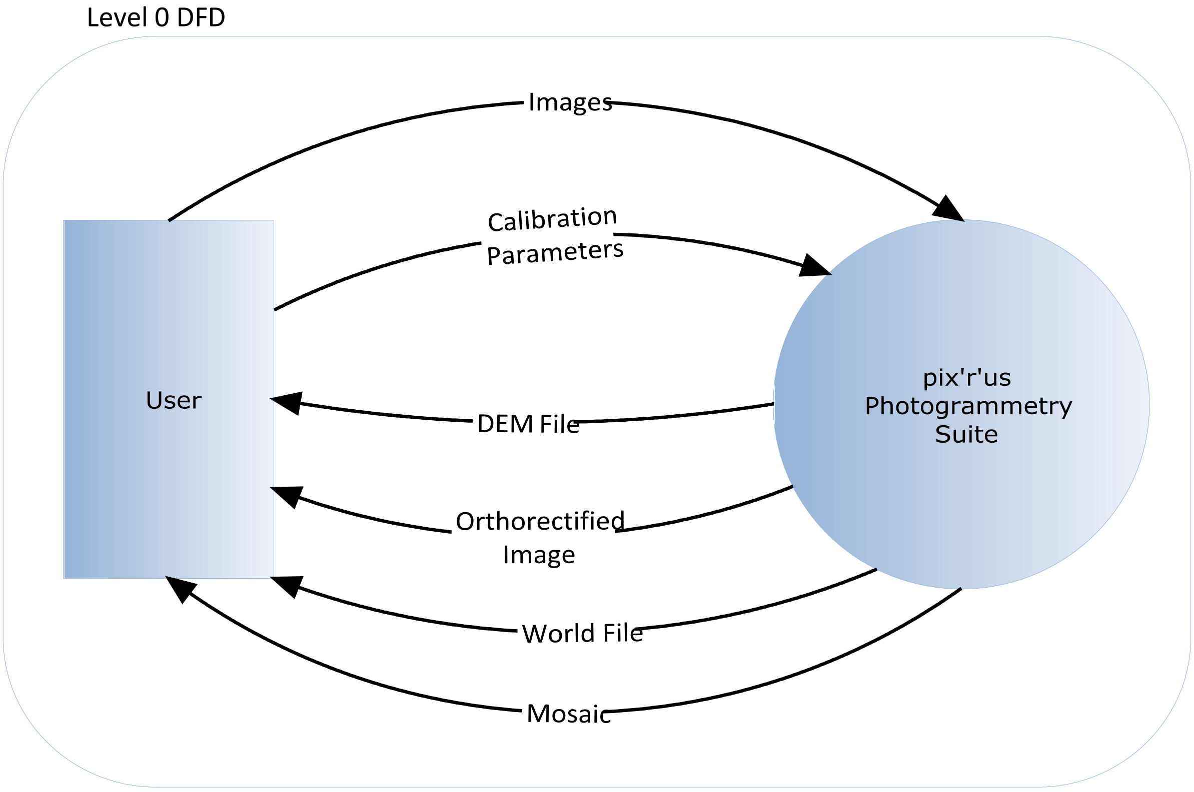


Figure 4

5.1.2. Level 1 of DFD

Level 1 of DFD is shown in Figure 5; it is a more detailed view of level 0 DFD. The main process is divided into five processes; image registration, DEM generation, ortophoto generation, mosaic construction and feature extraction.

Level 1 DFD

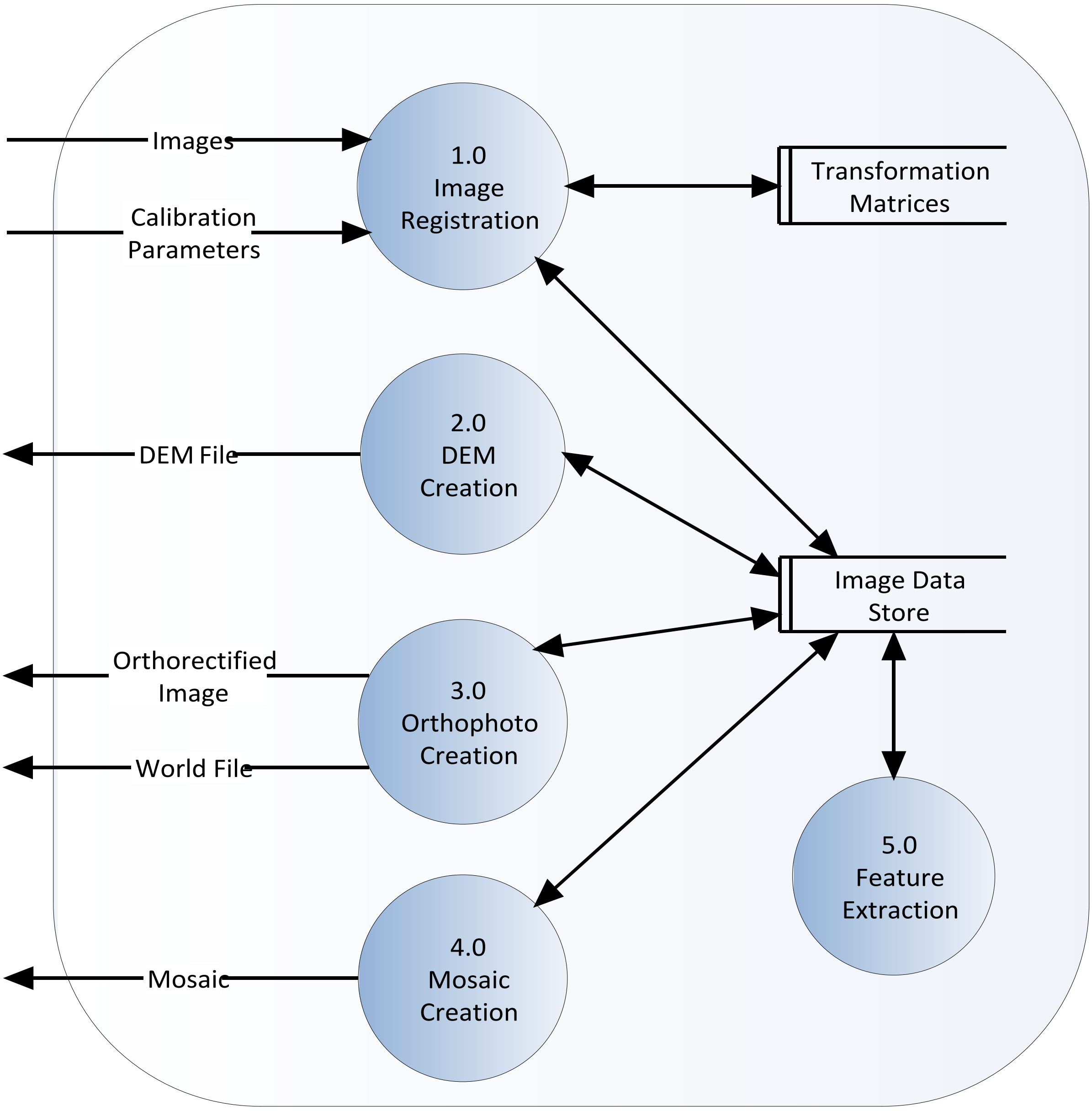


Figure 5

5.1.3. Explanation of DFD

<u>Image registration</u> Makes image to image and image to earth alignment, for this purpose constructs the transformation matrices.

<u>Dem Creation</u> Takes aligned input images and calibration parameters. Generates DEMs of overlapped areas and send this information to the data store.

<u>Orthophoto Creation</u> Uses DEM file and input images to construct orthorectified imagery. It also constructs world file using external calibration parameters.

Mosaic Creation Uses orthorectified images and constructs mosaic images. As a second choice uses tie points and orthographic images/images to create mosaic image.

Feature Extraction When other modules need, extract features of images, such as finding corners.

5.2. Use Case Analysis

In the Figure 6 capabilities of user are defined.

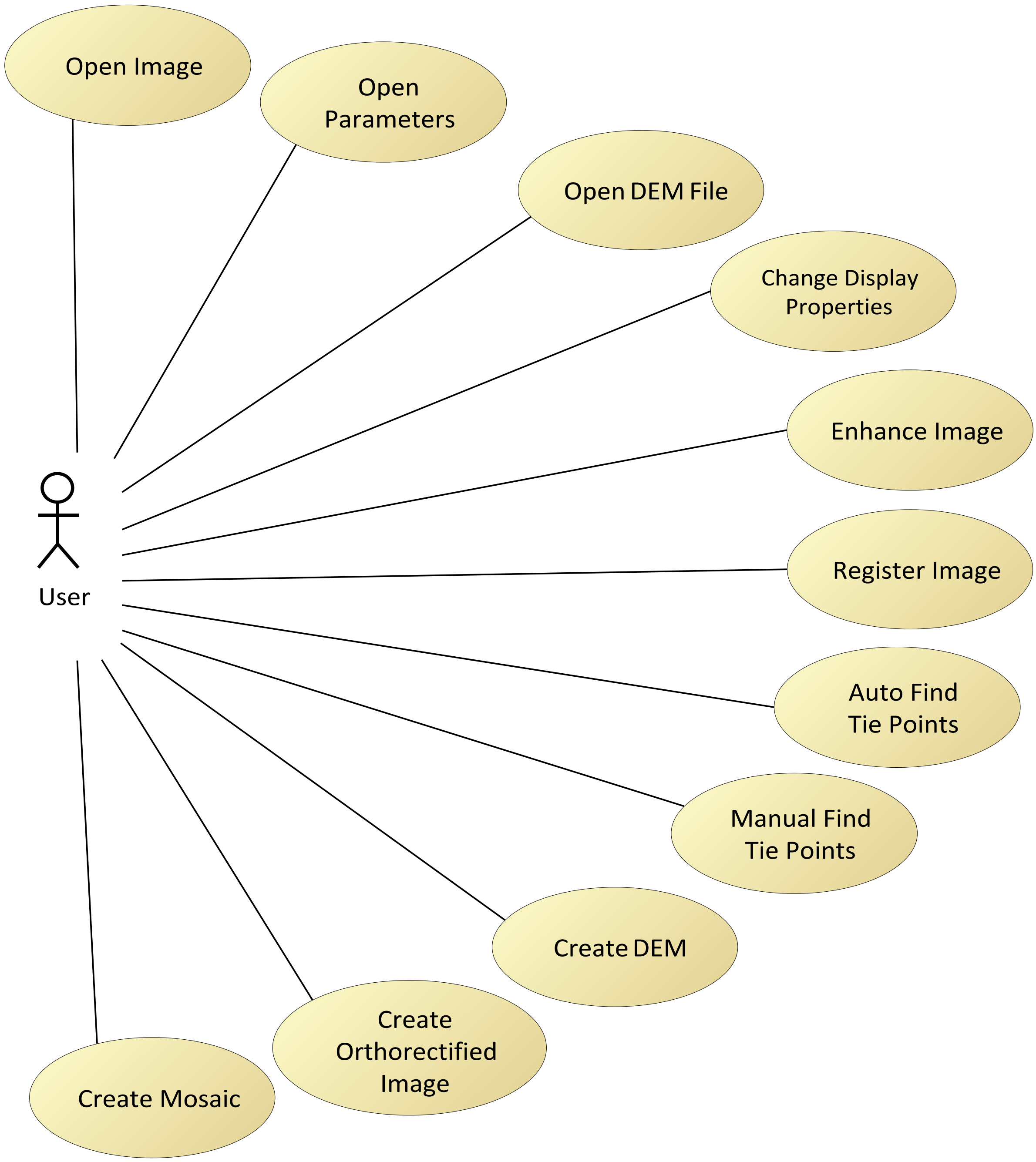


Figure 6

6. PROJECT SCHEDULE

Since we have limited time, we have to determine the schedule and the tasks of our project precisely. Being aware of the nature and algorithmic complexity of the project, we decided to develop the design by implementing several prototypes. Details are stated in the Gantt Chart (Appendix-A).

7. RISK MANAGEMENT

7.1. Explanation of Categories

7.1.1. Team Management

Throughout the project, members of the team should be aware of both their own and other member’s responsibilities. This is a requirement because; any unavailable team member may have a marginal impact on the flow of project. Unbalanced work distribution is an expected risk which can be solved with intra-help in the team.

In the coming stages of the project, roles and responsibilities of team members may vary because of the changes in expectancies by the contactor. With the dynamic role assignments and responsibility sharing these risks’ impact can be pulled down to a minimum level.

7.1.2. Project Management

Since the project will be under an iterative design, problems caused by ill-defined requirements can be realized before it reaches a catastrophic level. But this iterative design requires meeting deadlines very strictly, because any delay occurring in any level of the project will affect the succeeding levels catastrophically.

7.1.3. Contractor Related

Since the changing demands of the contractor, the project requires frequent feedback from the contractor to meet expectations.

7.1.4. Design Problems

The project contains various modules which require complex design and deep knowledge about the subject. For minimization of this risk, researches and further readings are expected from each member.

Also there will be development standards followed in the documentation and implementation according to the request of the contractor. To warrant these expectations, properties and specifications of expected standards should be learned by each team member.

7.1.5. Technical Problems

Finding proper libraries and tools for implementation is a risk to be taken care of since it would be time consuming to recreate these libraries and tools. Also the integration of these libraries and tools are considered as a technical problem, which may create marginal impacts on the project.

Risks	Category	Probability	Impact
Lack of Roles and Responsibilities	TM	%20	Negligible
Unavailable Team Members	TM	%15	Marginal
Unbalanced Work Distribution	TM	%20	Marginal
Incomplete or Ill-defined Requirements	PM	%15	Marginal
Scheduling Problems or Missing Deadlines	PM	%5	Catastrophic
Algorithmic Complexity of Project Modules	DP	%30	Marginal
Following Specific Development Standards	DP	%10	Negligible
Lack of Feedback from Contractor	CR	%5	Critical
Changing Demands of the Contractor	CR	%30	Marginal
Finding Proper Libraries and Tools	TP	%20	Critical
Integrating Proper Libraries and Tools	TP	%20	Critical

8. TESTING AND QUALITY ASSURANCE

8.1. Testing

To assure the required quality of the end product, extensive testing will be performed during the development process. Since an iterative linear sequential model will be employed for the software development process, there will be testing phase at each iteration.

To make the testing process as effective as possible, the linear nature (requirements section elaborates on this) of the project will be exploited. The key observation one has to make is that the final software itself is a pipeline of various modules. Although the modules are interconnected in a cascade fashion (hence the linear nature), designs and implementations of the modules themselves are decoupled. These constituent modules are:

1. DEM generation module
2. Image Mosaic construction module
3. Orthophoto generation module

Here, one has to note a few facts:

1. The modules are cascaded in the given order to make up the final software. Since there is a certain ordering of the modules, a well defined testing hierarchy can be used.

2. All these modules essentially consist of implementations of well defined (though very complex!) algorithms. Thus each module has well defined inputs and outputs.

3. Replacements for each of these modules are available in the market.

Taking all these facts into account, it becomes apparent that the testing procedure that it most suitable for this project is **black box testing**. At the end of each iteration in the development process, the developed module will be tested against numerous inputs. These inputs will include both synthetic inputs (generated by a testing software that will be developed) and sample real-world inputs.

In addition to the black box testing of the modules, release candidates will also be tested in the integration procedure. This testing process involves less automation and more tests involving real-world cases.

8.2. Quality Assurance

Since both the constituent modules and the final software has well defined inputs and outputs; the “quality” of the software can be measured more quantitatively than other common cases. Although various quality factors affect the overall quality of software, the following factors are the most important for a photogrammetry tool:

1. Correctness of results
2. Efficiency (both time and memory) of the tool
3. Usability of the GUI
4. Input error tolerance
5. Completeness of documentation

Although not an easy to measure quantity in general, measuring “correctness” is a much easier (and more well-defined) task in the context of a photogrammetry tool. This property of the project can be illustrated with the following concrete example: Suppose that an image mosaic is constructed from two input images. The result cannot be half right / half wrong: Either the mosaic is correctly constructed or it is completely wrong. The only degree of freedom on correctness lies in the “stitching contours" of the mosaic. The transitions around these contours are smoother for high quality software and sharper for lower quality software.

One other important property is that the outputs of the software (and its modules) can be displayed graphically either as 2-D images (for the orthophoto module and the mosaic module) or projected images of 3-D surfaces (DEM module). This fact makes the measurement of "correctness" by a human inspector convenient. Unfortunately, the same facts also imply that automating the correctness assurance task is a hard one. Thus, the most convenient quality assurance for our project is to use human inspectors to examine the outputs of the test cases.

Efficiency of the software is also a very important quality factor for photogrammetry tools. There are many uses of photogrammetry applications where there is a certain tolerance for errors in the output, but the speed of operation is critical. Following the assurance of correctness, assuring high efficiency will be considered second in the priority queue. Since memory measurement and execution time are measurable quantities, statistical methods will be employed for this part.

Since the contractor has given strict requirements on the GUI, we don't have a lot degrees of freedom as far as the GUI design is concerned. However, it is a well known fact that many applications do not catch up because of their complex (and counter intuitive) GUIs. Great care will be taken to ensure that PPS will not have the same fate.

Though not very common in most software projects, input error tolerance is a quality factor that cannot be neglected in the case of photogrammetry tools. Since the input images of photogrammetry tools are essentially photographs shot from planes (or satellites) with non-ideal cameras, errors in the input images are not very infrequent. Assuring robust performance even for these inputs is definitely a plus for a photogrammetry tool. Although not demanded exclusively by the contractor, we plan to add some input error tolerance capabilities (example: reducing lens distortion) if the timing constraints and the project schedule permits.

Finally, a good documentation is a must for photogrammetry tools like all other software. We plan to produce some documentation at each iteration so that a complete documentation of the overall product is assured.

9. REFERENCES

[1] Trucco T. , Verri A. (1998). Introductory Techniques for 3D Computer Vision. New York: Prentice Hall.

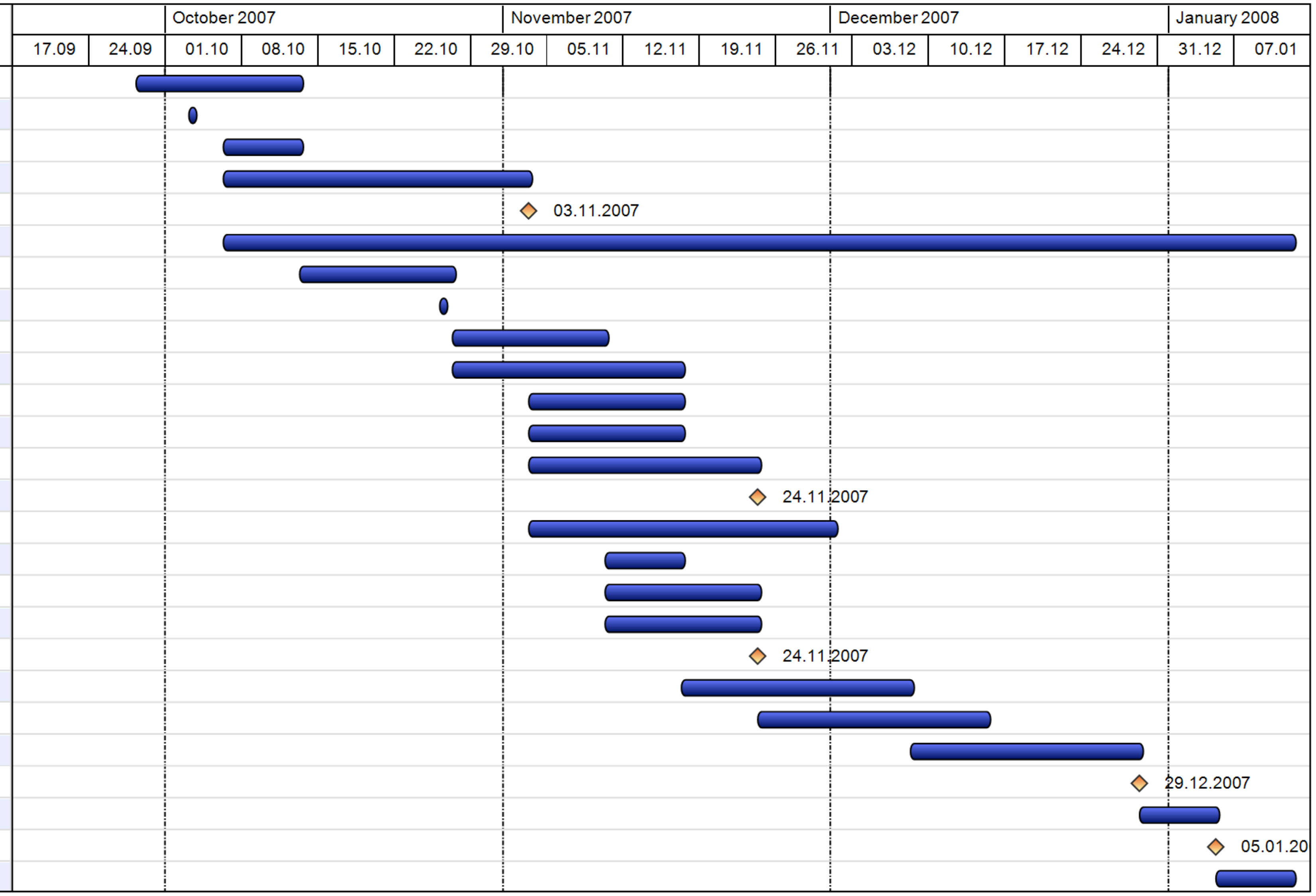
[2] Krauß T., Reinartz P., Lehner M., Schroeder M., Stilla U. Dem Generation from Very High Resolution Stereo Data in Urban Areas. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences VOLUME XXXVI.* Retrieved March 14, 2005 from http://www.isprs.org/commission8/workshop_urban/krauss.pdf

10. APPENDIX

10.1. APPENDIX-A GANTT CHART

gantt chart : Gantt Chart : Project

#	Name	Duration	Start	Finish
1	Project Proposal	16.0 d	28.09.2007	13.10.2007
2	First Technical Meeting	1.0 d	03.10.2007	03.10.2007
3	Wiki and Web Page Construction	8.0 d	06.10.2007	13.10.2007
4	Requirements Analysis	29.0 d	06.10.2007	03.11.2007
5	Requirement Analysis Report	0.0 d	03.11.2007	03.11.2007
6	Literature Survey on Photogrammetry	99.0 d	06.10.2007	12.01.2008
7	Investigation of Existing Photogrammetry Tools	15.0 d	13.10.2007	27.10.2007
8	Second Technical Meeting	1.0 d	26.10.2007	26.10.2007
9	Architectural Design	15.0 d	27.10.2007	10.11.2007
10	Basic GUI Design	22.0 d	27.10.2007	17.11.2007
11	Investigation on Input/Output Image Formats	15.0 d	03.11.2007	17.11.2007
12	Behavioural Modelling	15.0 d	03.11.2007	17.11.2007
13	Implementation of First Prototype (Basic Image	22.0 d	03.11.2007	24.11.2007
14	First Prototype	0.0 d	24.11.2007	24.11.2007
15	Detailed Literature Survey on Image Registration	29.0 d	03.11.2007	01.12.2007
16	Migration of Codebase to a Versioning System	8.0 d	10.11.2007	17.11.2007
17	Investigation of Feature Extraction Algorithms	15.0 d	10.11.2007	24.11.2007
18	Initial Design	15.0 d	10.11.2007	24.11.2007
19	Initial Design Report	0.0 d	24.11.2007	24.11.2007
20	Detailed Literature Survey on Image Mosaic	22.0 d	17.11.2007	08.12.2007
21	Revised GUI Design	22.0 d	24.11.2007	15.12.2007
22	Implementation of Second Prototype	22.0 d	08.12.2007	29.12.2007
23	Second Prototype	0.0 d	29.12.2007	29.12.2007
24	Final Design	8.0 d	29.12.2007	05.01.2008
25	Final Design Report	0.0 d	05.01.2008	05.01.2008
26	Prototype Demo	8.0 d	05.01.2008	12.01.2008



Phase
Task
Deadline
Milestone
Link