

# SENIOR PROJECT FALL 2007 REQUIREMENT ANALYSIS REPORT

---



**MIDDLE EAST TECHNICAL UNIVERSITY**



**DEPARTMENT OF COMPUTER ENGINEERING**



## Table of Contents

1	Introduction .....	4
1.1	Project Title .....	4
1.2	Problem Definition .....	4
1.3	Project Scope .....	4
2	Literature Survey .....	6
2.1	Image Registration .....	6
3.	Requirements Specification .....	7
3.1	Hardware Requirements .....	7
3.2	Software Requirements .....	7
3.3	Functional Requirements .....	7
3.3.1	GUI .....	7
3.3.2	Project Manager .....	8
3.3.3	Image Processing Tool .....	8
3.3.4	Digital Elevation Model Extractor .....	8
3.3.5	Orthophoto Generator .....	9
3.3.6	Mosaicker .....	9
4.	Process Model .....	9
4.1	Team Structure .....	9
4.2	Management Strategy .....	9
5.	Market Research .....	10
5.1	User Meetings .....	10
5.2	Market Analysis .....	11
5.2.1	Leica's Photogrammetry Suite .....	11
5.2.2	GRASS .....	11
5.2.3	Regeemy .....	12
5.2.4	Comparison Chart .....	13
6.	Constraints .....	14
6.1	Time Constraints .....	14
6.2	Experience and Skills of Member Constraints .....	15
6.3	Funding Constraints .....	15

6.4 Resource Constraints .....	15
7 Data Object Description .....	16
7.1 Entity Relationship Diagram.....	16
8 Process Description.....	16
8.1 Data Flow Diagrams .....	16
8.1.1 Level 0 Diagram.....	16
8.1.1 Level 1 Diagram.....	17
8.2 Data Dictionary .....	18
9 Use Case Diagrams.....	22
9.1 Using Photokan .....	22
10 Risk Management Plan .....	23
10.1 Necessity of a Risk Management Plan .....	23
10.2 Description of Risks.....	23
10.2.1 Team Management.....	23
10.2.2 Project Parameters .....	23
10.2.3 Development Process .....	23
10.2.4 Project Team .....	23
11 Task Plan .....	24
11.1 Releases .....	24
11.2 Reports.....	24
12 Quality Plan .....	24
Appendix .....	25
A.1 Gantt Chart .....	25
References.....	26

# **1 Introduction**

## **1.1 Project Title**

The title of the project is Photokan. The title is a derivation from the company name.

## **1.2 Problem Definition**

The advent of the computing technology and the research in the computer vision and image processing area opened up many possibilities for softcopy photogrammetry tools in which geometrical properties of objects are examined. These tools attract a well deserved attention since without the software assist, gathering information on geographical properties, such as terrain analysis, land survey becomes a heavy chore. Provided with online data, these tools can supply crucial information efficiently in case of emergencies. A disaster such as an earthquake is a good example of this. A quick analysis of the disaster's effects can speed up the life saving operations. Another area of application is the military tactical operations in which terrain topology forms the basis of the rationale. A before hand knowledge of the surface directly affects the decisions, such as the determination of necessary vehicles. The scope of softcopy photogrammetry actually covers a wider range of areas than mentioned, ranging from police investigation to archeology; anywhere geometric properties of objects are useful.

The application of these tools in critical issues (such as generating data concerning national security) makes national companies the first choice in obtaining licenses so even though a softcopy photogrammetry tool doesn't have a direct impact on human life, the interest in new tools don't cease. In Turkey, there is also a lack of software in this area. Another issue is the lack of flexible, user friendly software for linux platform even though some open source alternatives exist.

## **1.3 Project Scope**

Photokan will be a softcopy photogrammetry tool which will also employ geographical information processing facilities. The basic input target of Photokan is aerial images but it will be possible to use it on other types of images if deemed necessary. This will be a result of the flexibility of the program.

Basically, Photokan will process a set of aerial images with given camera parameters. The aim is using these images to create a model of the surface covered, and combining them to create one big picture which have the properties of a map. This can be achieved by a pipeline model. The process will work its

way sequentially but it will also be possible to directly jump to any stage since all phases are independent from another. The processing modules are:

- **Image Registration:** Image registration is performed to detect the overlapping areas of a given set of images. This process will also estimate the transformation relation between images. This data will be used in determining the missing camera parameters in case one is missing this data. Also the overlapped region will be combined in mosaicking process.
- **Digital Elevation Model Creation:** A digital elevation model (DEM) provides the height map of the surface. Digital elevation models will be created using multiple images of the scene. There will also be a 3-d visualization of the DEM.
- **Orthorectification:** This process will shift the perspective of the image so that It gains the basic property of a map.
- **Mosaicking:** Mosaicking yields the overall image which combines the overlapping images.

Photokan will provide a plugin mechanism to make the methods interchangeable and make it possible to develop and adapt better algorithms which can also be replaced in the future. All of these methods are prone to errors; this is a result of the problem. The problems are ill posed and the criterion of quality is mostly subjective. We will provide manual correction mechanisms in the methods to overcome the difficulty of reaching results which will be satisfactory.

The images that Photokan will work on are aerial images which possess geographical information. When a basic set of parameters are supplied, it will be possible compute the world coordinates under the mouse. The basic set of parameters will be provided in World file format. It will be possible to convert between various coordinate systems.

It will be possible to work on many file formats and convert between them.

Photokan will also employ basic image processing and manipulation methods so that preprocessing, enhancing, restoration will be possible by operator intervention. The basic capabilities are zooming, changing contrast and brightness values, rotating, sharpening, smoothing, contrast stretching, histogram equalization, and convolution by manually entered filters. It will be possible to work on a selected area of the image.

The scope of the project extends to bringing together qualified algorithms under a flexible interface, creating the necessary space to the user. In order to achieve these goals, fine tuning of algorithms will be as much as possible.

## **2. Literature Survey**

### **2.1 Image Registration**

Image registration is the process of geometrically aligning, and overlaying two images. After registration, it is possible to tell which point is mapped to which point in the second image, as long as the point is in the intersecting content. It is essential to the processes in which the final information is gathered from various sources. Many methods exist, each one applying to a specific class of problems usually. The most suitable methods for registration of aerial images are feature (such as edges, corners in the image) based ones. These methods try to relate the images based on the feature content. The research on this area is huge. Among possible solutions are the methods extracting region-like features which are projections of general high contrast closed-boundary regions of appropriate size (such as lakes, forest, buildings which are commonly found in aerial images). Once these features are detected, they will be matched according to an algorithm. For our case, methods employing invariant descriptors are feasible. When invariant descriptors are used, the corresponding features in image have the same description and two distinct features in the image have different descriptors. The sources are Refs. [1,2].

### **2.2 Digital Elevation Models**

A digital elevation model (DEM) is a matrix in which every element corresponds to the height of the surface model, or image in our case. DEM construction using an image set is quite straightforward if the intrinsic and extrinsic parameters of the cameras are known. This will be the case in our project. In case one of the camera's extrinsic parameters are unknown, it can be estimated from the registration data and other camera's parameters. The extrinsic parameters correspond to spatial properties, while the intrinsic parameters correspond to camera's internal properties. We can use the available registration data to construct the DEM. A Ray will be sent from the both cameras' coordinates (the points will be in one to one correspondence) to the corresponding coordinates on surface using georegistration data (which has the information on how image coordinates map to earth coordinates). The intersection gives the estimation of height. It should be noted the coordinates are not exact, so an approximation must be developed since the rays won't actually intersect in space. The source is Ref. [1].

## 3. Requirements Specification

### 3.1 Hardware Requirements

The hardware requirements fall in two groups, specifically for developers and for users. The requirements don't vary amongst these groups since hardware needed by the development environment doesn't exceed the hardware needed for the proper running of Photokan in computational power. Basic requirements are:

- Pentium 4 or equivalent AMD processor.
- 512MB of ram
- 1GB available hdd space

### 3.2 Software Requirements

The software requirements all compose of open source components. The software requirements for the developers are:

- Linux environment
- wxWidgets Library
- OpenCV Library
- ITK Library
- Gdal Library

### 3.3 Functional Requirements

#### 3.3.1 GUI

Photokan will compose of modules working independently. Every module will have its own graphical user interface (GUI) in order to exploit operator intervention and parameter tuning, that is GUIs of phases will be separated from each other. These GUIs will have controls to adjust parameters of the employed method. Main interface of the project will unify the modules, providing a flexible and user friendly access to every step. Main GUI will basically:

- Provide information about the progress and end result; with the option of intervention to and manipulation of the process if deemed necessary.
- Provide a viewer for the image set and the results of the operations.
- Provide menu bar and toolbar to quickly access to functions (creating a new project, opening an existing project, closing the current project, performing operations on the elements of the image set, processing and manipulating the elements of the image set by means of available image processing tools, converting between world coordinate formats).
- Provide a status bar to give quick information, such as the world coordinates under the mouse, result of the last process.

### 3.3.2 Project Manager

Project manager will import images into the workspace. This import will create a block with images having the given parameters. The block will be a virtual folder of images. It will be possible to enter new images into the block, or to remove from it. Project manager will register the images as groups of two as soon as they are entered into the block. The registration data obtained will be saved to be used in the following phases. Besides applying the basic operation of registration, it will create a new database for the block data in case a new project is created, or load an existing database in case a project is loaded. The main gui which is mentioned previously is the GUI of the project manager.

The total functionality of the project manager depends on the stability of other modules since it will be driving them.

### 3.3.3 Image Processing Tool

Image processing tool will provide contrast, brightness manipulation, zooming in, zooming out, histogram equalization, contrast stretching, convolution by manual filters functionalities. It will have to work on a predefined region of interest on the selected image. This module will be self contained and its functionality will depend on the project manager.

### 3.3.4 Digital Elevation Model Extractor

Digital Elevation Model (DEM) extractor will produce the DEM data of the image set. It will benefit from the registration data generated by project manager. It will have parameters with default values which can be manipulated on demand. The generated DEM data will be saved in the database for further use.



The functionality of DEM extractor is dependent on the project manager. DEM extractor will also offer 3-d representation of the generated surface model.

### **3.3.5 Orthophoto Generator**

Orthophoto generator will produce orthorectified versions of the images. It will use the associated DEMs. In case a DEM in database doesn't exist, orthophoto generator will use a predefined matrix of a flat surface. The functionality of orthophoto generator depends on the project manager.

### **3.3.6 Mosaicker**

Mosaicker will generate the combined image from the image set. It will be possible to manually correct the mosaick if it is not found satisfactory. Like other modules, its functionality depends on the project manager.

The reason that the functionality of modules of each stage depends on the project manager is that the project manager is the driver module.

## **4. Process Model**

### **4.1 Team Structure**

In our team structure the roles are interchangeable and communication between team members is horizontal. The reason for this is that new deadlines may occur since we are taking other courses, and a change in structure may be necessary. So we decided to adapt democratic decentralized model.

### **4.2 Management Strategy**

Photokan will be developed according to a hybrid combination of waterfall and prototyping development model. The certain steps of the project such as registration of images, DEM extraction, orthorectifying, and finally mosaicking are determined. And also fixed schedule of the reports is compelling us to use waterfall model. But the academic research part of the project is complicated and the risks can't be determined exactly. So the project requirements can't be specified totally in one step. In order to overcome this problem Milsoft staff advised us to release small prototypes after completing certain tasks to see whether our project is meeting the customer needs. This approach originates from spiral model, but it can not be called as spiral model because the whole process sequence will not be repeated as in the spiral model.

## 5. Market Research

### 5.1 User Meetings

Photokan project is to be developed for a specific client, and its userbase is already established. Since we don't have any potential users but the users, the process of gathering information about expectations and needs didn't require surveys. Two meetings were held with Milsoft in order to develop a view of the scope of the project and the requirements. The area of softcopy photogrammetry is vast and filled with algorithms, specifications and dense theoretical aspects. During the first meeting the first feedback came in the form of specifying the exact needs, narrowing the scope. The users will benefit from the software in land survey from the aerial images gathered from unmanned aerial vehicles. Actually every phase in Photokan is a separate area of study, but since the users have specific needs, they shared us their vision and guided in locating literature and market research about existing products. We have learned that a commercial product was already being used to perform the tasks, Leica Photogrammetry Suite and that the resulting product would resemble it in functionality. Since Milsoft is already developing the project in parallel, they have an experience on the subject. They provided pointers to open source libraries on computer vision and image processing. The process model was also a subject of the first meeting. Frequent releases were demanded. Creating a communication medium like a wiki and establishing a code repository were also other topics. The first desiderata gave us some insight about the necessary process model.

The second meeting was about mosaicking process, the user interface, and early technical issues. This meeting was an introduction to the details. They mentioned the desirability of additional useful image manipulation and processing features. It can be said that these meetings were also seminars on the subject.

The meetings are still ongoing and the results are directing us. Although we are reaching the users through Milsoft, they have firm specifications and we can get the necessary information through direct channels. An email address dedicated to the project exists. Being agile in obtaining necessary information really is very important for us because the theoretical foundations are complex, we usually get lost among details of which we can't determine their relevance to the project. Milsoft staff is also very kind and helpful.

## 5.2 Market Analysis

### 5.2.1 Leica's Photogrammetry Suite

This product is commercial software. It is really expensive, but it is a professional tool which is used by big companies after all. Its modular structure is very good and efficient. Since it is commercial software it has great support for its users and its documentation is very good. The weak spots of the LPS are that it is not cross-platform and it doesn't contain any 3-D properties like generating relief maps.

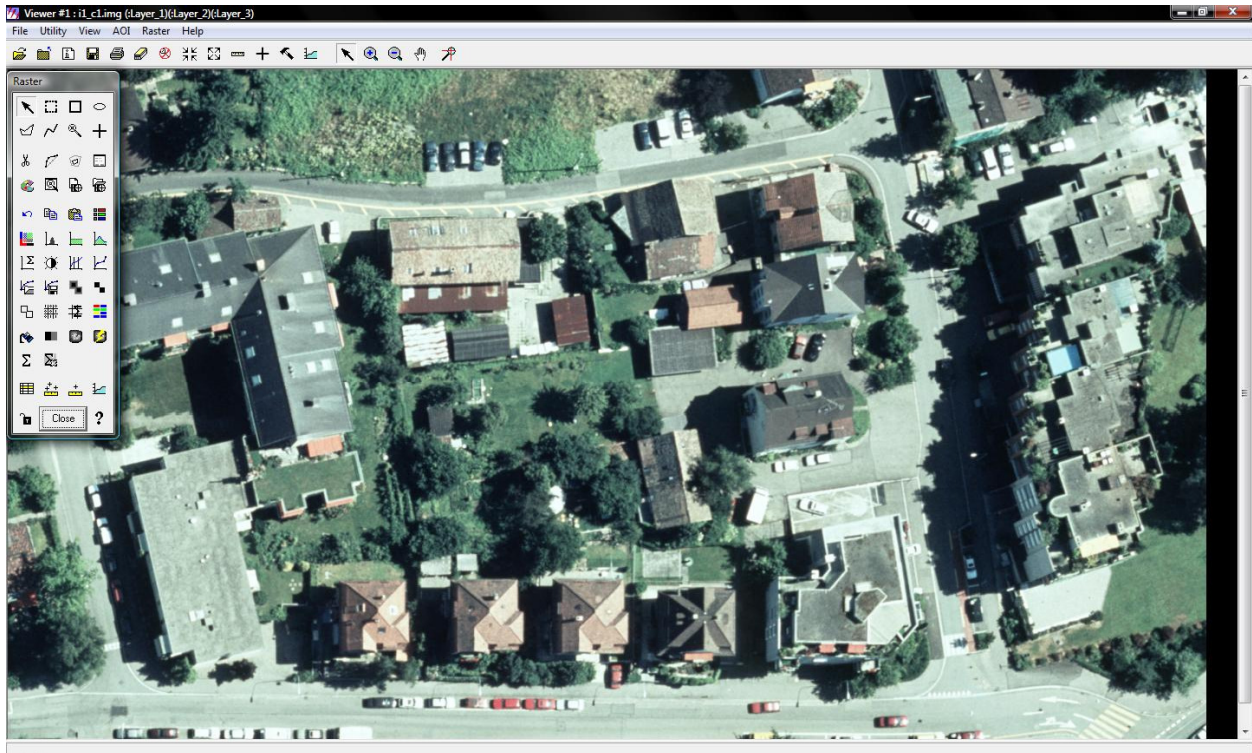


Figure.1 A screenshot of Leica Photogrammetry Suite

### 5.2.2 GRASS

This product is an open source project and it has GPL license. It can't be directly hosted on Windows but still a Cygwin version exists. It has a modular structure. It contains more than 350 modules. The documentation is poor; after all it is developed by voluntary people, not companies. For instance, last release's tutorial and help files can not be found easily. Another weak spot is that it is more complicated to use when compared to LPS.

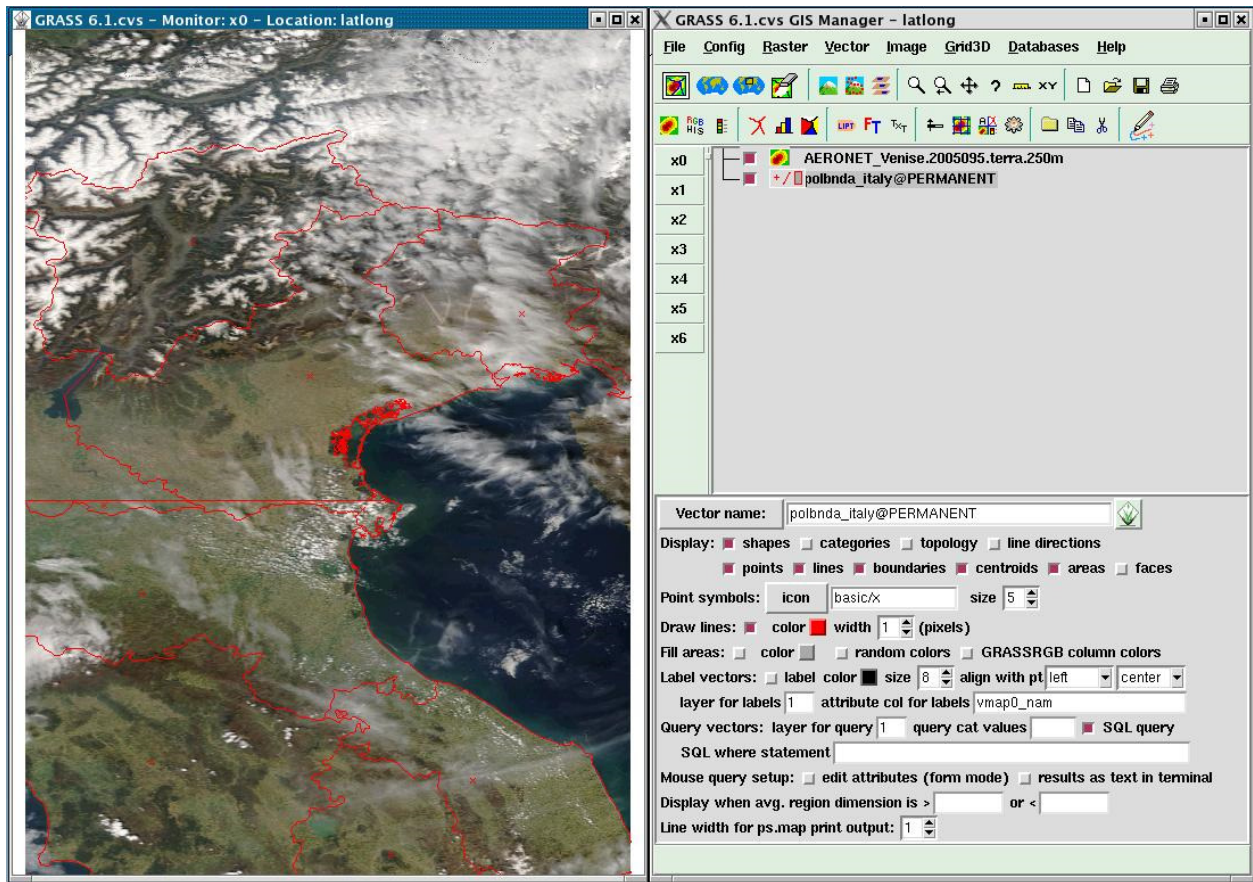


Figure.2 A screenshot of Grass v6.1

### 5.2.3 Regeemy

This product is a freeware which is developed for the purpose of academic research. The program is very simple. It only contains automatic point extraction and mosaic generation besides simple image processing operations like histogram equalization, adjusting contrast, etc. You can apply different algorithms for these purposes. It is a good tool for understanding the fundamentals of mosaicking but it can not be used at professional studies.



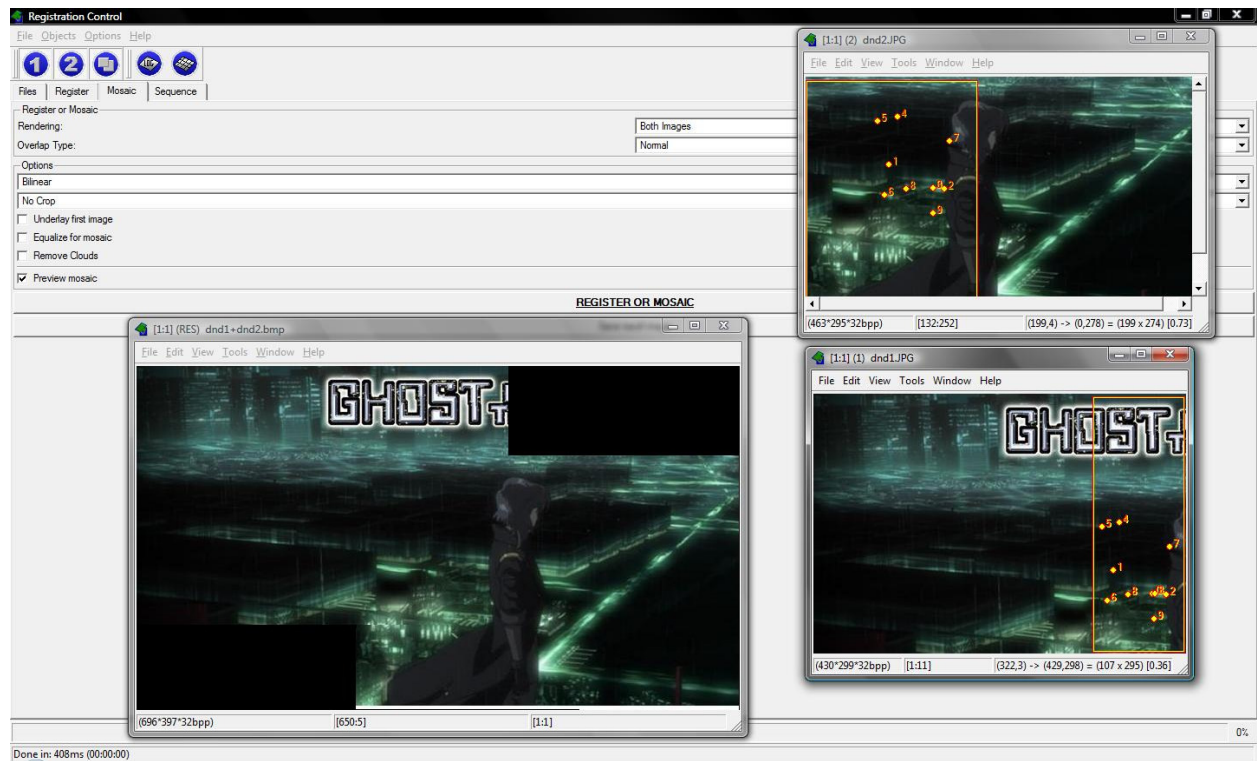


Figure.3 A screenshot of Regeemy

### 5.2.4 Comparison Chart

	GRASS GIS	REGEEMY	Leica Photogrammetry Suite	Photokan
Released By:	GRASS Development Team	Regeemy Development Group	Leica Geosystems	Okan
Modularity:	+	-	+	+
Operating Systems:	Linux, MS-Windows, Mac OS X, POSIX	Windows, UNIX	Windows	Linux
Hardware Requirements:				
Processor:			Intel Pentium 4 or Higher	Intel Pentium 4 or Eq.
RAM:			1GB or Higher RAM	512 or higher
Graphics Card:			Stereo Enabled Graphics Card	OpenGL enabled GC
Functionality:				
DEM	+	-	+	+
Mosaic	+	+	+	+
Orthophoto	+	-	+	+
Usability	Bad	Good	Bad	-
Complexity	High	Low	Too High	-
Documentation Support(1-5)	1	5	5	-
Price:	Free	Free for educational purposes	Depends on license type	Free
License Type	Open Source (GPL)	Closed Source	Commercial License	-

## 6. Constraints

### 6.1 Time Constraints

For the successful completion of the project at the end of the year and completion of the prototype at the end of this semester, time constraints are vital for our group. Minimizing potential risks that will harm the progression should only be avoided with a strict schedule. The time constraints are already predefined at the outline of the senior project.

## **6.2 Experience and Skills of Member Constraints**

Since none of the group members have worked on this kind of project before, our past experience is poor at this area. So we strongly need guidance which will help us to produce tangible results. Although the group members have past experience on developing software projects, the theoretical aspects and complexity of the project makes it a challenge.

## **6.3 Funding Constraints**

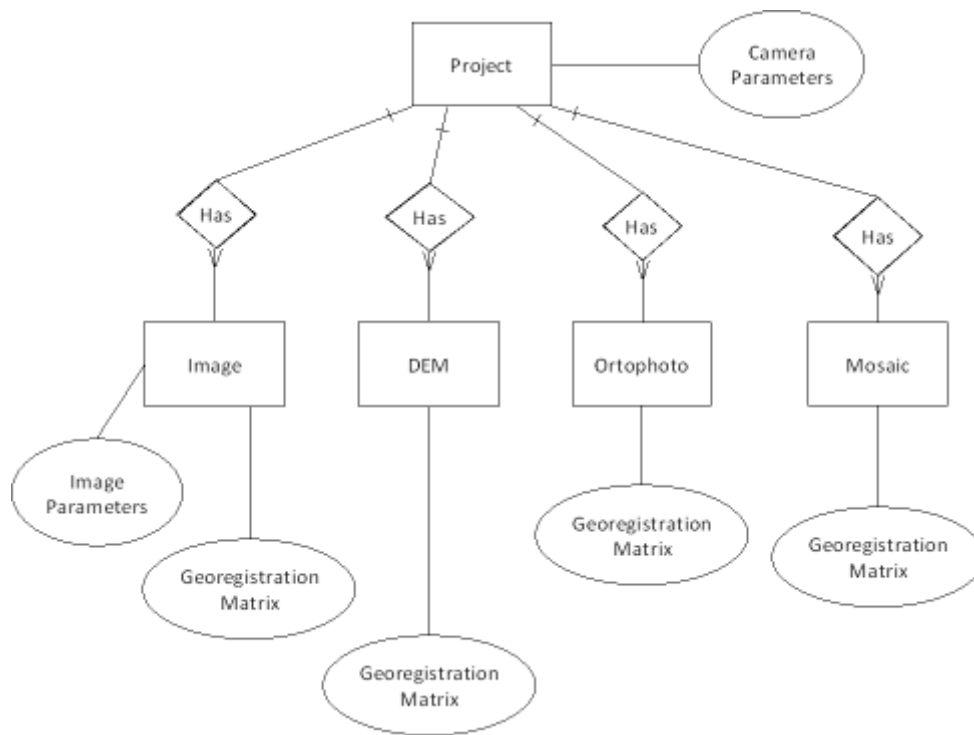
The external libraries and documents that we will need during the development process are mostly from open source environment, and we are not charged for their use. Furthermore, we also don't have the possibility of paying wages of programmers, so we don't have any funding constraint problems.

## **6.4 Resource Constraints**

The soft resources that we need for the project are mostly available on open source communities. So benefiting from them doesn't cause us problems. As well as we have the opportunity of finding them on the web environment, our department also provides them for us. The computers in our computer labs also offer us working individually and freely on the project. So we don't have any worries about resource constraint.

## 7 Data Object Description

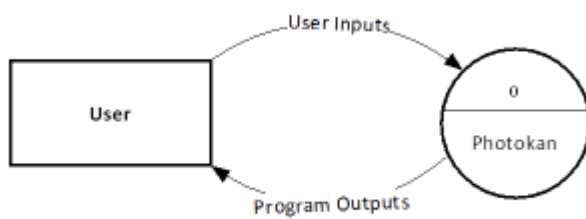
### 7.1 Entity Relationship Diagram



## 8 Process Description

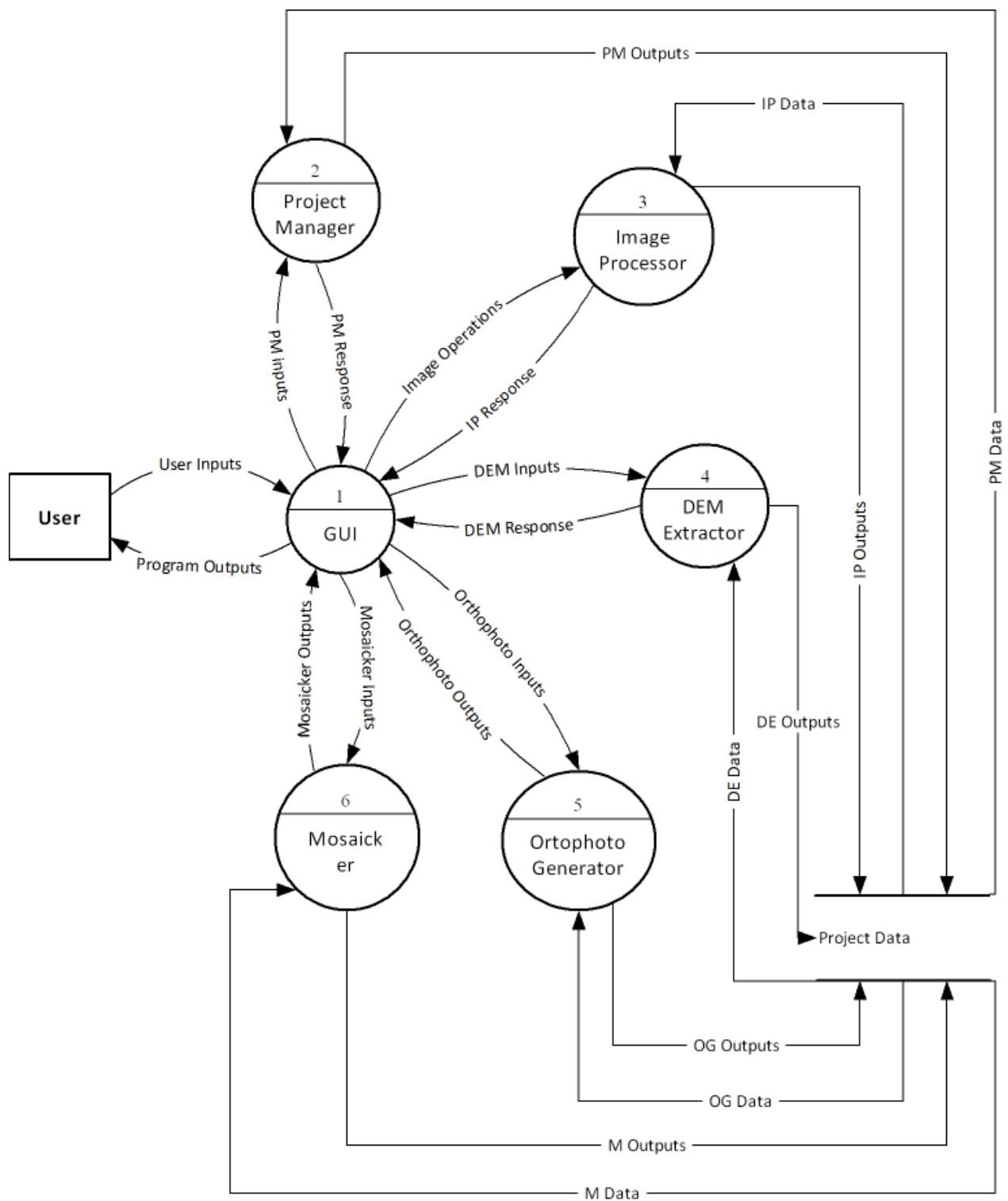
### 8.1 Data Flow Diagrams

#### 8.1.1 Level 0 Diagram





### 8.1.1 Level 1 Diagram



## 8.2 Data Dictionary

Name	User Inputs
Where/how used	Photokan (0) <i>input</i>
Description	Every command that the user issues and every input the user supplies

Name	Program Outputs
Where/how used	Photokan (0) <i>output</i>
Description	Every output that the program generates

Name	PM Outputs
Where/how used	Project Manager (1) <i>output</i>
Description	The registration data generated by project manager

Name	PM Data
Where/how used	Project Manager (1) <i>input</i>
Description	Every kind of information available in Project Data datastore

Name	IP Data
Where/how used	Image Processor (3) <i>input</i>
Description	Image information available in Project Data datastore

Name	IP Outputs
Where/how used	Image Processor (3) <i>output</i>
Description	Processed output image

Name	PM Response
Where/how used	Project Manager (2) <i>output</i> GUI (1) <i>input</i>

Description	The result of Project Manager operation
-------------	---

Name	PM inputs
Where/how used	Project Manager (2) <i>input</i> GUI (1) <i>output</i>
Description	The user command issued to the project manager

Name	Image Operations
Where/how used	Image Processor (3) <i>input</i> GUI (1) <i>output</i>
Description	The image to be processed along with the requested operation

Name	IP Response
Where/how used	Image Processor (3) <i>output</i> GUI (1) <i>input</i>
Description	The resulting image

Name	DEM Inputs
Where/how used	DEM Extractor (4) <i>input</i> GUI (1) <i>output</i>
Description	The user command issued to the DEM extractor

Name	DEM Response
Where/how used	DEM Extractor (4) <i>output</i> GUI (1) <i>input</i>
Description	The generated DEM representation

Name	Orthophoto Inputs
------	-------------------

Where/how used	Orthophoto Generator (5) <i>input</i> GUI (1) <i>output</i>
Description	The issued command to the Orthophoto generator

Name	Orthophoto Outputs
Where/how used	Orthophoto Generator (5) <i>output</i> GUI (1) <i>input</i>
Description	The resulting image

Name	Mosaicker Inputs
Where/how used	Mosaicker (6) <i>input</i> GUI (1) <i>output</i>
Description	The issued command to the mosaicker

Name	Mosaicker Outputs
Where/how used	Mosaicker (6) <i>output</i> GUI (1) <i>input</i>
Description	The resulting combined image

Name	M Outputs
Where/how used	Mosaicker (6) <i>output</i>
Description	The resulting mosaick which will be stored in the Project Data datastore

Name	M Data
Where/how used	Mosaicker (6) <i>input</i>
Description	The images stored in the Project Data datastore and their registration info

Name	OG Outputs
------	------------

Where/how used	Orthophoto Generator (5) <i>output</i>
Description	The Orthophoto image to be stored in the Project Data datastore

Name	OG Data
Where/how used	Orthophoto Generator (5) <i>input</i>
Description	The images stored in the Project Data datastore and their DEM data.

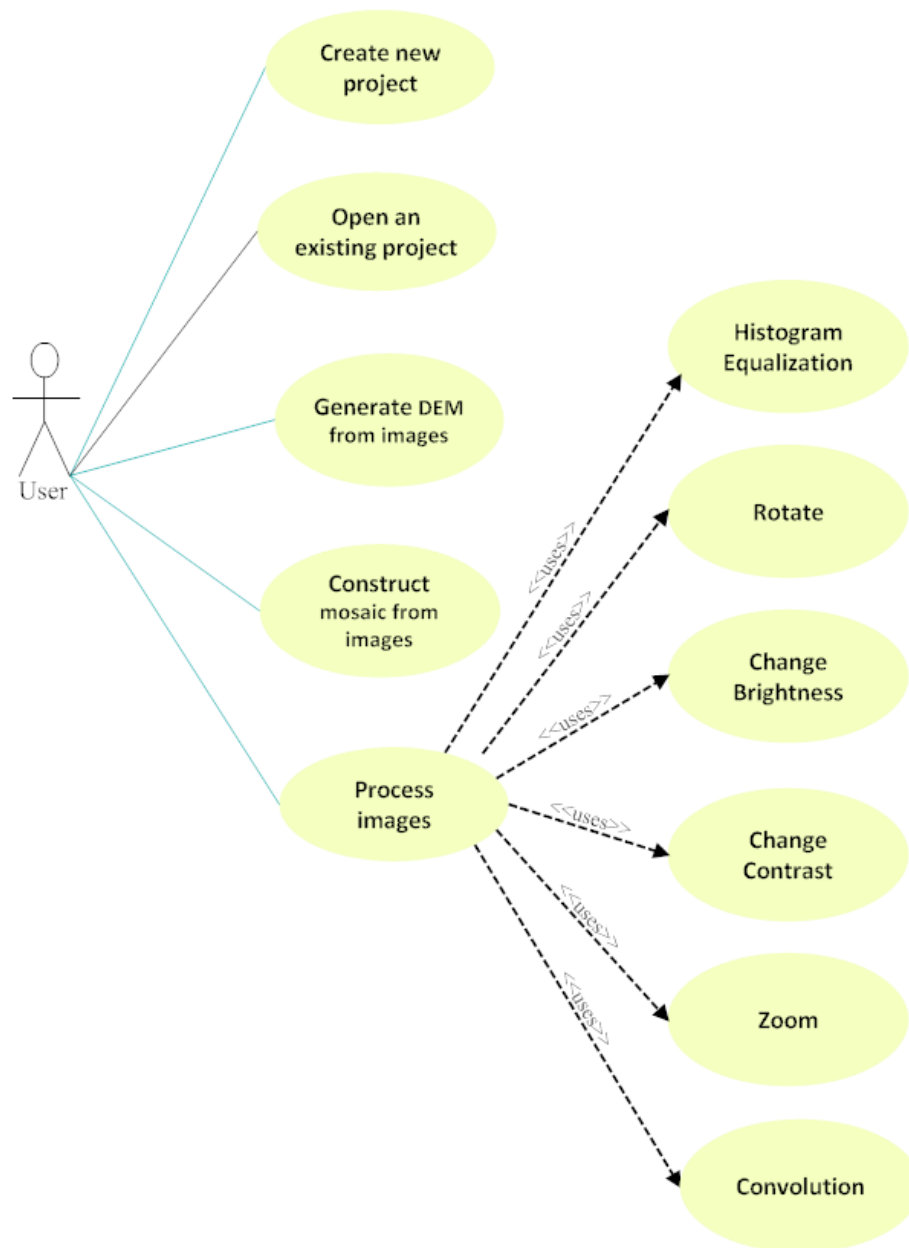
Name	DE Outputs
Where/how used	DEM Extractor (4) <i>output</i>
Description	The generated DEM data to be stored in the datastore

Name	DE Data
Where/how used	DEM Extractor (4) <i>input</i>
Description	The images stored in the datastore and their camera information

## 9 Use Case Diagrams

### 9.1 Using Photokan

#### *Use Case Diagram: Using Photokan*



## **10 Risk Management Plan**

### **10.1 Necessity of a Risk Management Plan**

Since the project that we work on is dependent to software, hardware and our efforts; it contains potential risks that may be faced in future. If any part of the system starts to malfunction, a sequence of problems which will put the future of the project at stake. So all of the factors that have the potential of harming the design and the development process should be considered and a risk management plan should be developed to prevent those defects or deal with them accordingly.

### **10.2 Description of Risks**

#### **10.2.1 Team Management**

Members of the group should be aware of their duties. Lack of knowledge about role of each member and lack of interaction between members can cause repetition or negligence of works. Also lack of members' willingness also causes the project to limp, if the situation continues project has the risk of failing.

#### **10.2.2 Project Parameters**

The ambiguity about project phases may seriously damage the schedule milestones. Since the timing is very important in the flow of the project, it should be planned carefully. Product Content

Since the user base of Photokan already exists and their demands about the product content are clearly specified, the potential risk of having to change the content is low.

#### **10.2.3 Development Process**

During the development process, testing plays an important role to prevent potential defects. Testing is also significant for a healthy final product to overcome several problems that users will face. In order to detect the defects in the software before they get too attached to the system, a testing plan should be developed to minimize the risks.

#### **10.2.4 Project Team**

Group members don't have required experience to work on such a serious and big project and it creates a major risk. Since the courses are also a burden on group members, it results in reduction of effort and time that can be allocated for the project.

## 11 Task Plan

### 11.1 Releases

Releases form one of the two types of tasks determined for the first semester. We will have 3 releases, starting from the second one; these releases will lead to milestones along with the reports.

The schedule for deliverables is:

- 9.11.2007: Release 0.0 - A package containing the software with image view coupled with world file formats support capability.
- 22.11.2007: Release 0.1- A package containing the software with added basic image registration capability.
- 20.12.2007: Release 0.2 – A package containing the software with added digital elevation model generation capability.

### 11.2 Reports

According to the task plan, the reports up to final design report are to be delivered by this semester.

The schedule for reports deliverables are:

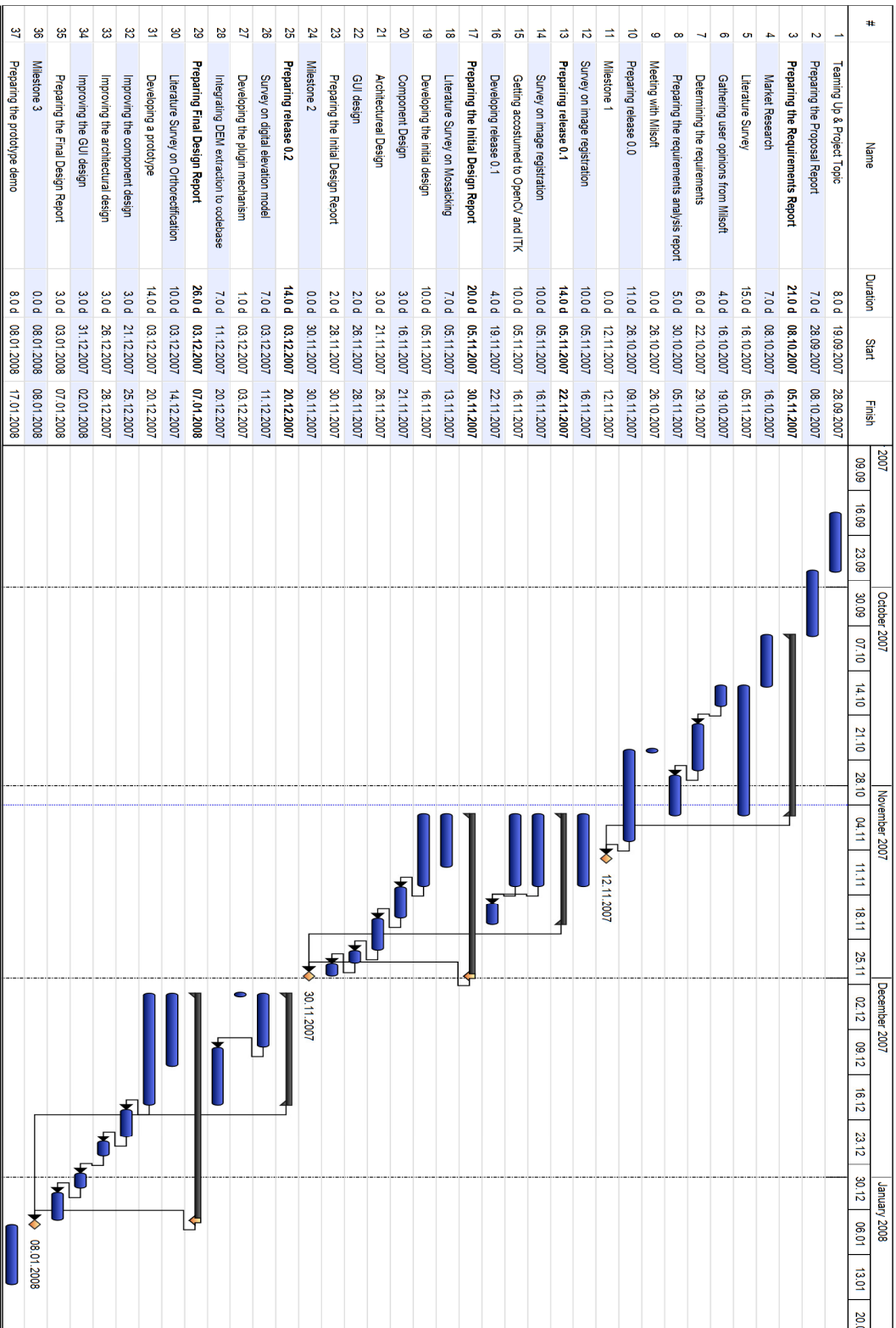
- 4.11.2007: Requirement Analysis Report
- 30.11.2007: Initial Design Report
- 7.1.2008: Final Design Report

## 12 Quality Plan

In order to provide quality assurance, we decided to develop a quality plan. The plan consists of our coding convention, ground rules, and example documents that will serve as the basis for comparison of our project against others.

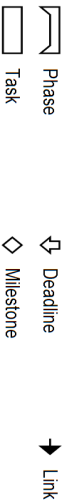


gantt : Gantt Chart : Project



## Appendix

### A.1 Gantt Chart



## References

- [1] Zitova, Barbara, and Jan Flusser. "Image registration methods: a survey." *Image and Vision Computing*, no. 21 (2003)
- [2] Brown, Lisa Gottesfeld. "A survey of image registration techniques." *ACM Computing Surveys*, no. 24 (1992)
- [3] Trucco, Alessandro Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.