İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

EXTRACTING SEMANTIC BUILDING MODELS FROM AERIAL STEREO IMAGES AND CONVERSION TO CITYGML

M.Sc. Thesis by
Ahmet Şengül

Department : Geomatic Engineering

Programme : Geomatic Engineering

Thesis Supervisors: Assoc. Prof Dr. Hande DEMİREL
Prof Dr. rer.nat. Thomas H. KOLBE

MAY 2010

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

EXTRACTING SEMANTIC BUILDING MODEL FROM AERIAL STEREO IMAGES AND CONVERSION TO CITYGML

M.Sc. Thesis by
Ahmet Şengül
(501061601)

Date of submission : 29 April 2010
Date of defence examination: 28 May 2010

Supervisors (Chairman) : Assoc. Prof. Dr. Hande DEMIREL (ITU)
Prof. Dr. Thomas H. KOLBE (TUB)
Members of the Examining Committee: Prof. Dr. Dursun Z. ŞEKER(ITU)
Prof.Dr. Necla ULUĞTEKIN(ITU)
Assist Prof. Ümit IŞIKDAĞ (BU)

MAY 2010

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

HAVA FOTOĞRAFLARINDAN SEMANTİK BİNA MODELLERİNİN ÇIKARTILMASI VE CİTYGML'E DÖNÜŞTÜRÜLMESİ

YÜKSEK LİSANS TEZİ

Ahmet Şengül
(501061601)

Tezin Enstitüye Verildiği Tarih : 29 Nisan 2010
Tezin Savunulduğu Tarih : 28 Mayıs 2010

Tez Danışmanı : Doç. Dr. Hande DEMIREL (İTÜ)
Tez Eşdanışmanı : Prof. Dr. Thomas H. KOLBE (TUB)
Diğer Jüri Üyeleri : Prof. Dr. Dursun Z. ŞEKER (ITÜ)
Prof. Dr. Necla ULUĞTEKIN (ITÜ)
Yrd. Doç. Dr. Ümit IŞIKDAĞ (BÜ)

MAYIS 2010

FOREWORD

This master thesis would not have been realized without the assistance, guidance and support of the people mentioned below.

First and foremost, I owe my greatest gratitude to my dedicated advisor, Prof. Dr. Thomas Kolbe who assisted, mentally and economicaly supported, and encouraged me to complete my master thesis. Without his instruction and constant encouragement, the completion of this thesis would not have been possible. Also I would like to express my appreciation and thanks for my supervisor Assoc. Prof. Hande Demirel.

I would like to thank the department of Institute for Geodesy and Geoinformation Science at Technical University Berlin for providing excellent research facilities, and for making my master thesis into a truly enjoyable experience. In addition, I am highly grateful to Claus Nagel who was always available for discussion on the FME conversion aspect of this research study and Alexandra Lorenz for her encouragement and many fruitful meetings and helpful discussions that we had, especially the data modelling aspects chapter of my thesis. Furthermore, my deep appreciation also expressed to technical responsible Dipl. Inform. Gerhard König who was helping to solve all my technical problems, reading my thesis thoroughly and providing me with insightful comments and valuable suggestions to refine my thesis.

As well, special thanks go to my dear classmates at Institute for Geodesy and Geoinformation Science TU Berlin, especially to Daniel Carrion, with whom I have had an endless number of discussions which have been invaluable to me, Nesrin Salepci, Andreas Pasewaldt, Aftab Ahmed Khan and Phd candidate Ahmed Alamouri for their assistance, friendship, and support on the days of study. Their support and encouragement made it possible for me to overcome the anxieties and difficulties in the process of thesis writing.

I would like to express my sincere appreciation to Christina Petersen who supported me mentally and provided me with a laptop to work on my thesis.

Many thanks also go to Alison Morton who helped me to improve my English writing skill through detailed readings of all my chapters, Alison's corrections were extremely useful and encourage me to write my master thesis.

Last but not least, I am greatly indebted to my parents whom supported me mentally and patiently during my research time in Berlin, Germany.

May 2010

Ahmet Şengül
Geomatic Engineer

TABLE OF CONTENTS

Page

ABBREVIATIONS

3D	**:** Three Dimensional
CityGML	**:** City Geography Mark-up Language
DTM	**:** Digital Terrain Models
DPW	**:** Digital Photogrammetric Workstations
FME	**:** Feature Manipulation Engine
GIS	**:** Geographic Information Systems
GCPs	**:** Ground Control Points
GDI NRW	**:** Initiative Geodata Infrastructure North-Rhine Westphalia
GML	**:** Geography Mark-up Language
XML	**:** eXtensible Mark-up Language
KML	**:** Keyhole Markup Language
LPS	**:** Leica Photogrammetric Suite
LIDAR	**:** Light Detection and Ranging
LCD	**:** Liquid Crystal Displays
LOD	**:** Levels of Detail
OGC	**:** Open Geospatial Consortium
SIG 3D	**:** Special Interest Group 3D
UML	**:** Unified Modeling Language
TIN	**:** Triangulated Irregular Network

LIST OF TABLES

LIST OF FIGURES

Page

EXTRACTING SEMANTIC BUILDING MODELS FROM AERIAL STEREO IMAGES AND CONVERSION TO CITYGML

SUMMARY

3D City Models are digital representations of the Earth's surface and related objects belonging to urban areas. In order to get information about a city it is necessary to collect data from different sources. There are several methods of collecting the data such as LIDAR, laser scanning, surveying measurements, aerial and satellite images…etc. The 3D GIS data collected using with 3D geographic imaging can be used for spatial modeling, GIS analysis, 3D visualization and simulation applications. The collection of geographic data is of primary importance for the creation and maintenance of a GIS. CityGML defines the classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantical and topological properties. 3D visualization and analysis of environmental properties is an efficient way of accessing the impact of urban projects.

The main purpose of the thesis is to fill the gap between photogrammetric methods and CityGML data model which is XML (eXtensible Markup Language) based format for the storage and exchanged of virtual 3D city models. The main problem is how to create a link between the photogrammetric methods and CityGML which is a common information model for the representation of 3D urban objects. In order to fıll the gap FME (The Feature Manipulation Engine) is used during the thesis study. Furthermore, data model of photogrammetric methods has defined using UML (Unified Modeling Language) diagram. Additionally, building data model of CityGML is minimized from the specification document depending on the requirements of the thesis work.

As a result of this study, the gap between photogrammetric methods and CityGML has been filled with FME the is conversion program. Moreover, the building model of photogrammetric methods has been created and also it is compared with CityGML building model. The differences and similarities of those models have also defined and also they mapped eachother with usıng UML diagram in order to understand better with the possible relations.

HAVA FOTOĞRAFLARINDAN SEMANTİK BİNA MODELLERİNİN ÇIKARTILMASI VE CİTYGML'E DÖNÜŞTÜRÜLMESİ

ÖZET

Üç boyutlu şehir modelleri, şehir detaylarının ve şehirlerde ki cisimlerin sayısal gosterimleridir. Bir şehir hakkında bilgi alabilmek için çeşitli kaynaklardan veri elde edilmesi gereklidir. Bu veriler LIDAR, lazer tarama, geleneksel ölçme yöntemleri, uydu ve hava fotoğraflarından yararlanılan çeşitli yöntemler yardımı ile elde edilir. Üç boyutlu Cografi Bilgi Sistemleri için toplanan veriler mekansal modelleme, coğrafi analizler, 3B görsellik ve simülasyon uygulamalarında kullanılır. Coğrafi verinin toplanması bir coğrafi bilgi sisteminin yaratılması ve onarımı için en önemli aşamalardan biridir. Elde edilen verilerin coğrafi, topolojik, mekansal ve görünüş özelliklerini amaca en uygun şekilde, arasında ki ilişkileri ve sınıfları ile birlikte CityGML veri modeli içerisinde tanımlanır. Üç boyutlu görselleştirme ve çeveresel özelliklerin analizi şehir projelerinin etkili olarak erişiminin önemli bir yoludur.

Yapılan projesinin ana amacı fotogrametrik methodlar kullanılarak olusturulan model ile ve XML veri depolama yöntemi temelli çalışan ve sanal 3D şehir modellerinin uzantısı olan CityGML veri modeli arasında bulunan boşluğu doldurmaktır. Ana problem, fotogrametrik yöntemler ve CıtyGML ki 3D şehir modellerinin sunumu için yaygın bir bilgi modeli arasındaki bağlantıyı nasıl yaratılacağıdır. Bu belirtilen boşluğu giderebilmek için FME adlı program tez çalışması sırasında kullanılmıştır. Bununla birlikte, photogrametrik yontemlere ait veri modeli UML diyagramı kullanılarak oluşturulmuştur. Ek olarak, CityGML bina modeli tez çalışması içinde gereksinimler dogrultusunda CityGML'in tanımlama dökümanında ki bina modelinden indirgenerek hazırlanmıştır.

Bu çalışmanın sonucunda, fotogrametrik yöntemler ve CityGML arasındakı boşluk bir dönüşüm programı olan FME tarafından doldurulmuştur. Ayrıca, fotogrametrik yöntemler ile oluşturulan bina modeli yaratılmış ve CityGML bina modeli ile karşılaştırılmıştır. Bu modeller arasındaki farklılıklar ve benzerlikler belirlenmiş ve daha kolay anlaşılması için bu modeller birbirileri ile ilişkilendirilmiştir.

1. INTRODUCTION

In recent years establishing three-dimensional (3D) city models and Geographic Information System (GIS) is getting more popular day by day. Since the recent developments, in the computer technology visualization is gaining more importance and getting more effective for the professionals who deal with the information systems. As a result of the developments, new technologies such as virtual reality, 3D GIS, urban modeling…etc, are currently in development. Also there are many projects related to these issues. For instance "Geo Data Management in the Administration of Berlin - 3D-VR-Model for Investors and Companies" is one of the project gathered 2D and 3D geo information of Berlin in an integrative and sustainable way - for planning, city information as well as location marketing. 3D GIS, in particular, is a very active research topic in the last few years within applications of city planning, tourism, noise maps …etc.

A 3D city model is a three-dimensional (3D) representation of a city or an urban environment, using data derived from multiple sources such as stereo aerial images, airborne LIDAR data and high resolution satellite data. It contains a large number of objects of different classes and different data models and structures.

1.1 Objective of The Thesis

The importance of by viewing the Earth’s surface in stereo view is that it can be interpreted, measured and delineated by using aerial images to obtain information about the building parameters. The set of parameters is divided into positional parameters on the one hand describing the position and orientation, on the other hand form parameters like width, height, length.

The thesis study has produced a building model including conversion from aerial images to CityGML model. For the test area namely F21C25C4B with 1/1000 scale map sheet is selected in the area at the historical area of Istanbul.

The initial steps have already been conducted, where a block building model was retrieved using a cadastral map. The cadastral map includes building information and number of floors. The project is completed using the ArcGIS program. After preparing the blocked building model using ArcGIS program, aerial images of this area are used by ERDAS Imagine Programs Stereo Analyst for ArcGIS in order to determine details of the building such as the roof type, chimney, dormer…etc.

The goal of the work is to develop a simple building block model of urban area based on stereoscopic measurements using aerial images to get detail information about structure of the buildings within an urban area. Starting from model of the buildings in the test area is consisting of simple building blocks and roof structures.

The thesis main aim is how to create a link between the photogrammetric methods and CityGML which is a common information model for the representation of cities with 3D buildings. The CityGML defines the classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantical and appearance properties. There is a gap between traditional photogrammetric measurement methods and CityGML. The objective is to fill the semantic gap between CityGML and photogrammetric methods.

1.2 Background

Most cities have information representing building footprints, building height information, aerial photography and terrain. These resources permit the development of a rough three-dimensional model of the city. However, if more detailed representations of some building details are needed, then building details such as roof structures should be accessible from satellite or aerial images. Nevertheless, there are some difficulties and problems related with the application work.[1]

When plotting in photogrammetric methods suc as ERDAS Stereo Analyst module difficulties occur in urban area with complex building structures. The main focus of this thesis is to create a link between photogrammetry and CityGML using Feature Manipulation Engine (FME). FME program is used in order to convert all information getting from different sources about the buildings in the working area. The buildings must be in terms of a boundary representation and modeled as several complex buildings, including details like roof structure, windows, doors, porches, chimneys, etc[1].

Within this scope, main component is a 3D spatial database which integrates the building model from aerial images. The integration of geometry will be provided by using aerial images with respect to visual representation. Also, the thesis work is to develop a 3D city model for representing and managing urban data, getting semantic information from aerial images and converting to City GML format and giving the city objects some semantic information from the test area.

Another reason getting semantic information about the city is necessary to set up 3D Geographical Information Systems. Definition of semantic information about 3D city models is another problem to solve.

The important thing is collecting the data from different sources in different application fields and different programs. After collecting the several data somehow they should be combined. There is an international standard for spatial data exchange issues is needed. When converting the data, need to be defining the main differences and similarities.

Converting the building model from aerial stereo images or from other possible sources such as semantic data from Municipality of the test area or extra input data ArcGIS shape format from other sources is one way to solve the problem creating a bridge between the photogrammetric methods and CityGML.

Choosing a storage format that supports semantic information is an important issue in city modeling. In order to solve storage problem every building part of the model should be saved separately. Building parts consist of a single solid and they are grouped depending on the building information.

1.3 Expectations

Without semantic information the 3D GIS does not make sense. Integrate data from different formats and data structures with complete control using the advanced data transformation capabilities offered by FME which is for spatial data conversion and distribution challenges is used in order to overcome such conversion problems. After getting building details from aerial images the building model is converted to CityGML, which is the first Open Geospatial Consortium (OGC) standard for the storage and exchange of virtual 3D city models.

A 3D view of the city is the key tool for increasing, understanding and improving communication related with urban issues. Several municipalities decide nowadays to build up 3D city models in order to clearly understand the cities real situations. They use those models for several purposes such as urban planning, emergency situations, pollution problems …etc. The most important thing is collecting correct information about city objects. City objects are located directly by coordinates and characterized by several properties such as height, type, usage… etc. House roofs show a wide variety of shapes, which make their classification challenging but necessary for establishment of a standard procedure[2].

Data of 2D map urban systems is available in some data layers and digital terrain model. Based on the 2D map, related data for 3D will be manipulated. The 3D geometric information of urban objects is often available from the project designers that produce the 3D CAD models[3].

Expectation of the thesis is to provide a 3D city models using aerial images and collect information about the city as accurate as possible. Different sources and formats already provide correct data related with urban issues. The critical point is creating a link between photogrammetry and CityGML and later on to give semantic information to the data coming from photogrammetric methods.

Transforming imagery into 3D GIS data involves several processes commonly associated with digital photogrammetry. The data and information required for building and maintaining a 3D GIS includes orthorectified imagery, Digital Terrain Models, 3D features, and non-spatial attribute information associated with the 3D features. Through various processing steps, 3D GIS data can be automatically extracted and collected from imagery[4].

In order to extract the data, there are some difficulties such as complex real 3D objects derived from image data which makes it difficult to define semantic information about an urban area. Working with image data also has some difficulties related to contrast or qualification of the image data which is important in order to collect the correct information about the city. When digitizing on three-dimensional aerial images, quality of the image data is important to get more accurate semantic information. Another point is classification of the roof structures and city objects that they should be defined and classified after drawing from aerial images. While establishing 3D GIS, the 3D data is obtained from different sources or different resolutions. In general difficulties arise when attempting to convert data from various sources caused by data coming from different sources have different data structures. When converting the data from one to another format attention has to be drawn to efficiently solve data conversion challenges.

The proposal workflow have been designed to define possible workflow during the thesis work. In figure 1.1 can be seen a simple workflow diagram of the conversions. Solutions of the transformation will be explained in the thesis work. The workflow is basically to obtain an overview of the stereo window, with the aid of this view and also using different sources will obtain detailed information about city objects and later on combining these data on FME to convert from photogrammetric tools to CityGML.

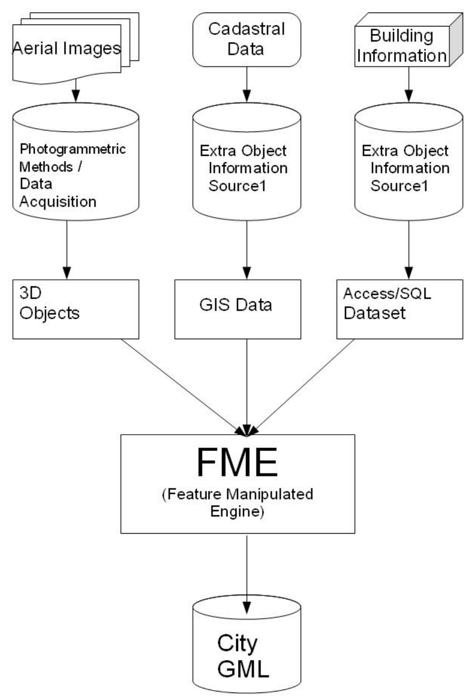


Figure 1.1 : Proposal Workflow of the conversions from ERDAS Stereo Analyst to CityGML

1.4 Hypothesis

The finding, which has provided the motivation for this master thesis is this: It is possible to employ classical photogrammetric tools possibly in combination with additional but limited amounts of extra information to create semantic 3D city models in CityGML. Also, it is possible to do conversion from the 3D city model to CityGML with integration of extra information by using a conversion program.

2. 3D CITY MODELLING

Nowadays it is possible to create 3D city models at a reasonable cost due to the rapid development of computer hardware, and possibility of data acquisition from stereo aerial images. This development has increased several application of three dimensional spatial information in a variety of fields including urban planning, telecommunications, ecology, tourism and entertainment [5].

A common understanding is that every 3D city model consists of a digital elevation model with ground height and 3D building data with building heights. Usually a 3D city model represents an existing city but in some applications, especially in gaming and entertainment, it may have no counterpart in the real world [5]. For instance, in the virtual 3D City Model of Berlin (Fig.2.1) several data sources has been used such as cadastral data, digital terrain model, aerial image, building models and different variants of city object colections [6].

Figure 2.1 : A view from 3D City Model of Berlin [6].

2.1 Semantic 3D City Models

An increasing amount of applications such as urban planning, navigation systems, facility management, disaster menagement, enviromental simulations are created mainly visualisation purposes as virtual 3D city models. Those applications are required additional information which is given in a standartdised representation about the cities by user and authority. Users and their applications expect the city models to be structured in a well defined way. Therefore, development of a city model is needed to exploit the semantic information and structure of the City Objects[10].

Objects are decomposed into parts due to logical criteria which are given or can be observed in the real world. To create a city model several researches are focused on the automatic process. Beside the automated process in 3D city models manuel efforts are needed to create and maintain the 3D city model [10].

Semantic 3D city models comprise besides the spatial and graphical aspects particularly the ontological structure including thematic classes, attributes, and their interrelationships. It follows structures that are given or can be observed in the real world. For example, a building can be decomposed into different building parts, if they have different roof types and their own entrances like a house and the garage[10].

The appropriate qualification of 3D data can be required to the semantic modelling of cities. The semantic modelling of cities can be used economically by different customers within multiple applications. For this reason a common information model is required by over the different users and applications. CityGML can be given as an informational data model. The semantic model of CityGML employs the ISO 19100 standards family framework for the modelling of geographic features. According to ISO 19109 geographic features are abstractions of real world objects. Geographic features may have an arbitrary number of spatial and non-spatial attributes. Object oriented modelling principles can be applied in order to create specialisation and aggregation hierarchies[10].

2.2 General Characteristic of CityGML

CityGML is an open data model, XML based format for the storage and exchange of virtual 3D city models. It is an application scheme based on the Open Geospatial Consortium's Geography Markup Language 3 (GML 3.1). The Geography Markup Language (GML) is a standard language for the modelling, storage and transport of geographic information data. GML bases on the eXtensible Markup Language (XML), the well distributed internet standard of the World-Wide-Web Consortium (W3C). The GML is an Implementation Specification of the OGC and also an international standard of the ISO. It realizes the abstract concepts of non spatial like e.g. ISO 19109, and spatial like e.g. ISO 19107, ISO 19123 and other standards of the ISO. CityGML is an open standard and therefore can be used free of charge. And it is originally developed by "Special Interest Group SIG3D" from Initiative Geodata Infrastructure North-Rhine Westphalia, Germany (GDI NRW) [9].

CityGML does not only represents the graphical appearance of city models but also takes care of the representation of the semantic thematic properties, taxonomies and aggregations of Digital Terrain Models (DTM), sites including buildings, bridges, tunnels, vegetation, water bodies, transportation facilities, and city furniture.

Current 3D city models are constructed from laser data such as LIDAR and terrestrial laser, photos such as terrestrial, satellite or aerial images, orthophotos, maps such as cadastral, city, soil, archives such as diachronic analysis of the urban, fabric, areas to preserve or investigate and databases containing location based information. These data become more commonly available as well as realtime visualization possibilities with free and three-dimensional viewers such as Google Earth. Therefore, the amount of 3D city models are increasing and many cities have been or are being modelled all around the world. However the generation and the maintenance of 3D city models are costly. Currently many works and researches are in progress such as EuroSDR related to the automatic generation of 3D city models from multiple data sources[8].

The aim of the CityGML is to reach a common definition of the basic entities, attributes and relations of virtual 3D city models that can be shared over different application fields.

The usage of 3D city models is wide and including urban planning and design, telecommunication planning, traffic regulation, disaster modeling, architecture, preservation of historical buildings, infrastructure and facility services, promotion of economic development, and homeland security or tourism. By using 3D city models, it is possible to visualize what a city will look like after a proposed change, or predict and visualize which parts of a city will be affected by a flood[8].

2.2.1 Levels of Detail (LOD)

CityGML supports five different consecutive Levels of Detail (LOD). LODs are required to reflect independent data collection processes with differing application requirements. Further, LODs facilitate efficient visualisation and data analysis. In a CityGML dataset, the same object may be represented in different LOD simultaneously, enabling the analysis and visualisation of the same object with regard to different degrees of resolution. Furthermore, two CityGML data sets containing the same object in different LOD may be combined and integrated.

CityGML files can contain multiple representations for each object in five different Levels of Detail (LOD) simultaneously[7].

- **LOD0** – regional, landscape

LOD0 is coarsest level and fundamentally a two and a half dimensional Digital Terrain Model. It may be draped by an aerial image or a map.

- **LOD1** – city, region

LOD1 is the well known model comprising prismatic buildings with flat roofs. Block model of the city can be given as an example.

- **LOD2** – city districts, projects

LOD 2 includes roof structure and outer building installations like dormer and chimney. LOD2 has differentiated roof structures and other building surfaces. Vegetation objects may also be represented.

- **LOD3** – architectural models -outside-, landmarks

LOD3 denotes architectural models with detailed wall and roof structures, balconies, bays and projections. High resolution textures can be mapped onto these structures. LOD3 model additionally can contains detailed vegetation and transportation objects.

- **LOD4** – architectural models -interior-

LOD 4 additionally contains the interior structures of buildings like rooms, furniture and interior installations LOD4 completes a LOD3 model by adding interior structures for 3D objects [9].

LODs are also characterised by differing accuracies and minimal dimensions of objects (Tab. 2.1). The five different LODs vary with respect to their accuracy requirements. Accuracy is described as standard deviation σ of the absolute 3D point coordinates. Relative 3D point accuracy is plannig to be added in a future version of CityGML and it is typically much higher than the absolute accuracy. In LOD1, absolute 3D point, the positional and height, accuracy must be 5m or less then 5m. The positional and height accuracy of LOD2 must be at least 2m and the object footprints may be represented 4m by 4m. In LOD3 the both accuracy are 0.5m, and the minimal footprint is 2m by 2m. LOD4 provides the positional and height accuracy less than 0.2m. The LOD categories make 3D city model datasets comparable and give an idea to provider and customer about the complexity of their integration [9].

Table 2.1: Levels of Detail (LOD) is taken by source [9].

	LOD0	**LOD1**	**LOD2**	**LOD3**	**LOD4**
Model scale description	regional, landscape	city, region	city districts, projects	architectural models (out-side), landmark	architectural models (interior)
Class of accuracy	lowest	low	middle	high	very high
Absolute 3D point accuracy (position / height)	lower than LOD1	5/5m	2/2m	0.5/0.5m	0.2/0.2m
Generalisation	maximal generalisation (classification of land use)	object blocks as generalised features; > 6*6m/3m	objects as generalised features; > 4*4m/2m	object as real features; > 2*2m/1m	constructive elements and openings are represented
Building installations	-	-	-	representative exterior effects	real object form
Roof form/structure	no	flat	roof type and orientation	real object form	real object form
Roof overhanging parts	-	-	n.a.	n.a.	Yes
CityFurniture	-	important objects	prototypes	real object form	real object form
SolitaryVegetationObject	-	important objects	prototypes, higher 6m	prototypes, higher 2m	prototypes, real object form
PlantCover	-	>50*50m	>5*5m	< LOD2	<LOD2
… to be continued for the other feature themes					

In CityGML each object can be represented differently for each LODs. Moreover different objects from the same LOD may be represented by an aggregate object in a lower LOD. Fig. 2.2 shows examples of 3D city models for each LOD [9].

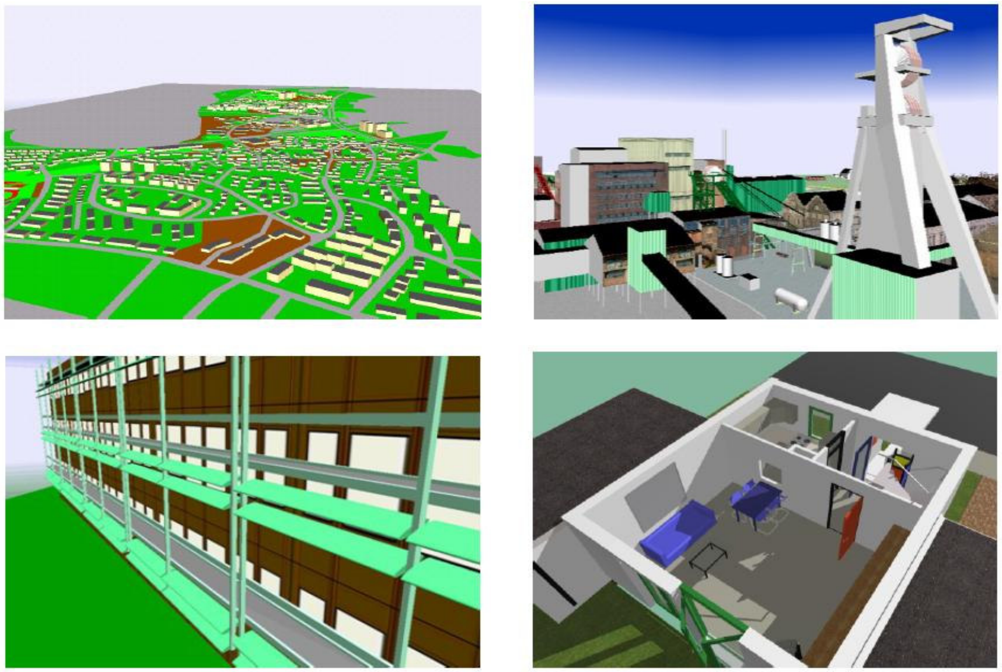


Figure 2.2 : Examples for city or building models in LOD1 (upper left), LOD2 (upper right), LOD3 (lower left), and LOD4 (lower right) [9].

3. PHOTOGRAMMETRIC OBJECT REGISTRATION

3.1 Basic Objectives

Modelling and 3D description of real world objects collected throught aerial imaging system has already become a hot topic of increasing importance as they are essential for a variety of applications. Several researches focus on 3D modelling as a topic of intensive investigations. After increasing availability of powerful hardware and development of computer technologies such as software, graphics researchers are focusing attention on the third dimensions. Many activities, companies, universities working on three dimensional city models, navigation systems, geographical information systems… etc. These fields require construction of 3D city models.

There are various methods currently is being used to create 3D city models. The simplest is to extrude building footprints from 2D maps to a given height based on building attributes such as the number of storeys. This method has been used in last years project namely “3D GIS Example in Historical Peninsula of Istanbul” which is presented by the author of the thesis as a poster presentation in XXI.ISPRS Congress 2008 in Beijing, China. Within this poster work all buildings in the historical peninsula of Istanbul are extruded given three meters height and multiplied based on how many storeys the building has coming from the cadastral data. Also a terrain model of the area has been used during the poster project[11].

A 3D terrain model is usually added to provide the landscape context for the buildings. Most Geographical Information Systems support this method but the 3D city model that is generated does not contain detailed information, such as facade geometry or textures, and the height of the buildings may not be very accurate. To overcome the problem of inaccurate building height data, photogrammetry or laserscanning methods have been developed to capture 3D city models from aerial images or airborne laserscan images.

Using photogrammetry, a 3D city model is produced from stereo aerial imagery in a semi-automatic way using special software tools. The 3D data can be exported into several commercial 3D formats for visualisation. In the thesis work, the method decided upon to create 3D city models was to use photogrammetric methods using aerial images which includes camera and orientations properties. To produce a 3D city model from stereo aerial photographs ERDAS Imagine Program LPS module, is a photogrammetry software , and was used in the thesis project.

3.2 Feature Extraction

Several methods have been reported which extract different image primitives to initiate the reconstruction of building objects in the scene. There are approaches which extract the building roof structures from stero aerial images such as semi automaticly or automaticly. The aproach is being used in the thesis project is that manually digitizing roof structures using with roof prototypes in the program namely ESRI ArcGIS Feature Assist extensions from ERDAS Imagine software.

3.3 Building Reconstruction

Using aerial images and laser scanning, the roof structure and often, building structure can be reconstructed. In general, however, it is not possible to acquire information about building facades due to steep observation angles and occlusions although in some cases with multiple overlapping images it might be feasible. A standard approach adopted by several researchers is to approximate building walls by vertical planes defined by the eaves lines of the reconstructed roof and extend them downwards to ground in digital terrain model (DTMs) or triangulated irregular network (TIN). Close range techniques can be used in order to provide a more detailed reconstruction for facades. A geodetic acquisition of the facade structure may be carried out, but in most case it is more appropriate to measure photogrammetrically and to use only geodetic measurement for control point determination [12].

Building reconstruction is done with photogrammetric measurements method. Aerial Images are use in order to capture the building roof structures.

3.4 Photogrammetric Measurement Methods

Photogrammetry is one of the technologies mostly using in 3D city modelling. 3D city modelling and urban visualization using the technology of photogrammetry is one of the most growing research topic in digital architecture. There are basicly two approaches for photogrammetry to get the information about a city. First one is to take photos from the ground with digital cameras. The photos should be taken from all around the building. Sometimes its hard to take the photos due to security and privacy issues. Another disadvantage of this method is that matching the photos needs more time to manage right model. On the other hand, it does not cost too much compared to the satellite images. The other method is to use aerial or satallite images. It costs more to take ground photos. Also, it needs to arrange flight plan, interrior and extrerior oriential parameters[14].

Reconstruction of man-made objects are targeted for several research efforts. Buildings generally have an amazing variety of architectural design.The rectangular shapes, vertical walls of some buildings often give a deceptive view that reconstruction of the building from its geometric primitives is an easy task. The lack of operational automatic methods for building reconstruction is an indication that the task is not as easy and simple as it seems especially in digital photogrammetric methods. The problem is mainly related to appropriate methods of collecting data about the third dimension of buildings for example their heights, roofs, facades, windows…etc. In general two different approaches are utilised to reconstruct buildings namely "top-down" and "extrusion". In the first approach, measured elements are upper parts of buildings, such as roof outlines, while in the second, the reconstruction starts from the footprints. Which method is better depends upon a number of considerations such as desired resolution such as complex roofs or rectangular boxes, available sources such as 2D GIS and/or aerial images, hardware and software, purpose of the city model[15].

3.5 Data Acquisition

Data acquisition in photogrammetry is concerned with obtaining reliable information about the properties of surfaces and objects. This is accomplished without physical contact with the objects which is the most obvious difference to surveying. Data acquisition was achieved by using stereo aerial images in the thesis project. It is the most important source in digital photogrammetry. The information has been captured using the stereo aerial images.

The remotely received information can be grouped into four categories;

- **geometric information** involves the spatial position and the shape of objects. It is the most important information source in photogrammetry.
- **physical information** refers to properties of electromagnetic radiation, e.g., radiant energy, wavelength, and polarization.
- **semantic information** is related to the meaning of an image. It is obtained by interpreting the recorded data.
- **temporal information** is related to the change of an object in time, usually obtained by comparing several images which were recorded at different times.

The semantic information is given after digitizing process on the stereo images for data acqusition in the thesis work. The generic name for data acquisition devices is sensor is mounted on a platform, consisting of an optical and detector system. The most typical sensors are cameras where photographic material serves as detectors. They are mounted on airplanes as the most common platforms geometric information involves the spatial position and the shape of objects[16].

There are three primary approaches to collecting 3D objects; namely image-based, point cloud-based, and the hybrid approach:

- *Image-based 3D data acquisition:* Use of images, such as close-range, aerial photographs, or satellite images, to collect information about 3D buildings, etc. 3D structural and dimensional information from imagery can be derived by using the approach. The process is well documented, but many components still have to be executed manually. This approach has been used for data acquisiton during the thesis study.
- *Point cloud-based 3D data acquisition*: mapping detailed structures of 3D objects apply active sensors, such as laser scanning devices. Either airborne and ground-based laser scanning, or a combination of the two, can produce very dense

and accurate 3D point clouds. Extraction of height information is largely automated, but textures from point clouds are often weak.

- *Hybrid approaches*: One of the technological trends is to combine optical images, point cloud data, and other data sources (e.g. maps or GIS/CAD databases). These approaches are generally more robust but require additional data sources[13].

4. DATA MODELLING ASPECTS

Data modelling is a method used to define and analyze data requirements which is needed to support the processes of a work. The data requirements are recorded as a conceptual data model with associated data definitions. Data are typically the results of measurements and can be the basis of graphs, images, or observations of a set of variables. Data modelling defines the relationships between data elements and structures. Data modelling techniques are used to model data in a standard, consistent, predictable manner in order to manage it as a resource[17].

This chapter is organized as follows; in the chapter 4.1 with general information about UML diagram and small explainations about the vocabulary of the UML. Geometric data model is discussed in the chapter 4.2 that the differences between mainly geometric models like obtained from ERDAS Imagine and models based on thematic features like modelled using CityGML model. Moreover in the chapter 4.3 is discussed mainly the geometric differences between Arc GIS Stereo Analyst and City GML and finilized on mapping of the data model in a UML diagram.

4.1 Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a standard language for writing software design which explains how something might be achieved. The UML could be used to visualize, specify, construct of a software intensive system. It is appropriate for modeling systems ranging from enterprise information systems to distributed Web-based applications and even to hard real time embedded systems. It is a very expressive language, addressing all the views needed to develop and then deploy such systems. Even though it is expressive, the UML is not difficult to understand or to use. Learning to apply the UML effectively starts with forming a conceptual model of the language, which requires learning three major elements: the UML's basic building blocks, the rules that dictate how these building blocks may be put together, and some common mechanisms that apply throughout the language. The

UML also provides a language for expressing requirements and for tests. Finally, the UML provides a language for modeling the activities of project planning and release management[18].

4.1.1 Terms and concepts of UML

UML class diagrams show the classes of the system, their interrelationships including inheritance, aggregation, association, the operations and also attributes of the classes. With the UML can be used class diagrams to visualize the static aspects of these building blocks and their relationships and to specify their details for construction as you can see in Figure 4.1. A class is a description of a set of objects that share the same attributes, operations, relationships and semantics[18].

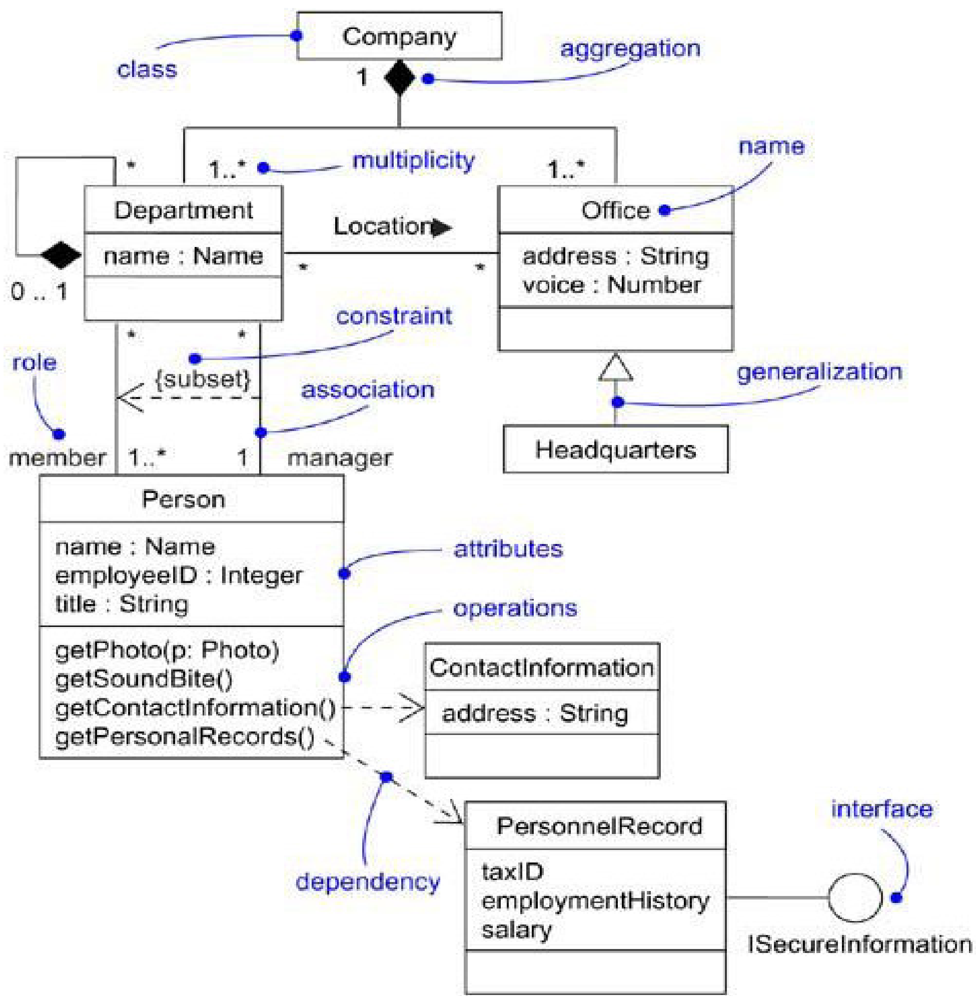


Figure 4.1 : UML Class Diagram definition from UML User Guide [18].

Defination of the important concepts of the UML diagram is explained with related figure 4.2 below. The UML notations used in the thesis are described.

Name: A name is a textual string. That name alone is known as a simple name is the class name prefixed by the name of the package in which that class lives.Also every class must have a name that distinguishes it from other classes[18].

Attributes: An attribute represents some property of the thing which is shared by all objects of that class. It is a named property of a class that describes a range of values that instances of the property may hold. A class may have any number of attributes or no attributes at all. For example, every wall can have height, width, and other specific attributes[18].

Operations: An operation is the implementation of a service that can be requested from any object of the class to affect behavior. A class may have any number of operations or no operations at all[18].

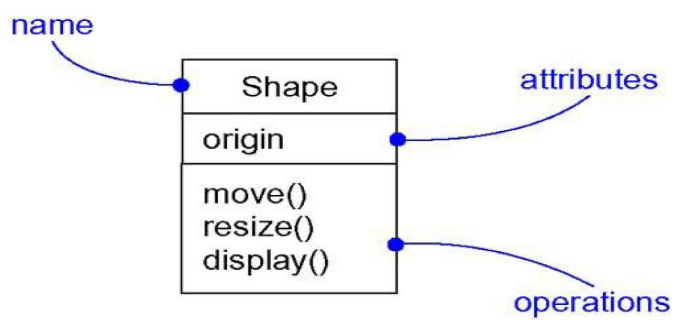


Figure 4.2 : Name, attributes and operations from UML User Guide [18]

Association: An association is a structural relationship that connecting two classes, can be navigated from an object of one class to an object of the other class, and vice versa. Associations can be used to show structural relationships[18].

Multiplicity : An association represents a structural relationship among objects. In many modeling situations, it's important for user to state how many objects may be connected across an instance of an association. This "how many" is called the multiplicity of an association's role, and is written as an expression that evaluates to a range of values or an explicit value as in Figure 4.3. A multiplicity can show of exactly one (1), zero or one (0..1), many (0..*), one or more (1..*) or even state an exact number[18].

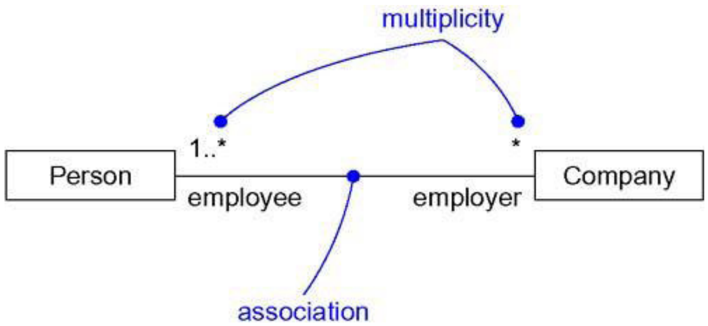


Figure 4.3 : Association and Multiplicity

Aggregation: An aggregation is special association where the involved classes are in a relation. Hierarchy can be described as "is part of" respectively "consists of". It is used to represent or specify ownership or a whole/part relationship. In an aggregation relationship the part may be independent of the whole requirements. Representing the UML symbol of aggregation in Figure 4.4 [19].

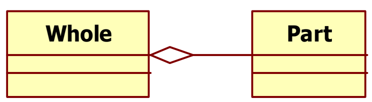


Figure 4.4 : The UML Diagram represents the aggregation

Composition : Strict form of aggregation, where the existence of the part depends on the existence of the whole. It is used to represent an even stronger form of ownership. With composition user can get part with whole. It means that a class can not exist by itself. Representing the UML symbol of composition (Fig.4.5) [19].

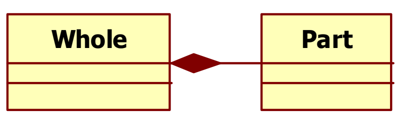


Figure 4.5 : The UML Diagram represents the composition

Inheritance: Inheritance refers to the ability of child class to inherit the identical functionality of super class, and then add new functionality of its own. To model inheritance in a class diagram, a solid line is drawn from the child class with a closed, unfilled arrowhead pointing to the super class (Fig4.6) [9].

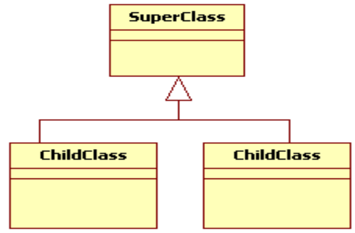


Figure 4.6 : UML Diagram represents the inheritance

4.2 Geometric Data Model

In this chapter the geometry model of ERDAS Stereo Analyst and CityGML are explained in detail in the following sections.

4.2.1 Multipatch geometry type

The multipatch data format, a geographic information system (GIS) industry standard developed by ESRI in 1997, is a geometry used as a boundary representation for 3D objects [20].

The multipatch geometry type was initially developed to address the need for a 3D polygon geometry type unconstrained by 2D validity rules. For example, representing extruded 2D lines and polygon footprints for 3D visualization would not be possible. Furthermore, multipatches allow for the storage of texture image, color, transparency, and lighting normal vector information within the geometry itself, making them the ideal data type for the representation of realistic-looking 3D features. A 3D geometry used to represent the outer surface, or shell, of features that occupy a discrete area or volume in three-dimensional space. Multipatches can be used to represent simple objects such as spheres and cubes or complex objects such as buildings, and trees [20].

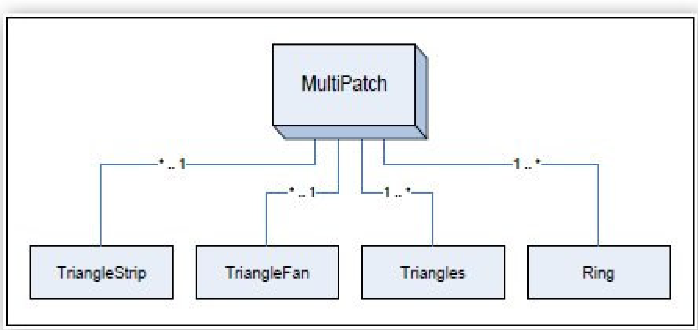


Figure 4.7 : Multipatch geometry construction is taken from source [20].

A multipatch can be viewed as a container for a collection of geometries that represent 3D surfaces. These geometries may be triangle strips, triangle fans, triangles, or groups of rings, and a single multipatch may comprise a combination of one or more of these geometries. (Fig4.7)

The geometries that a multipatch comprises are referred to as its parts or patches, and the type of part controls the interpretation of the order of its vertices. The parts of a multipatch can be of one the following geometry types: (Fig 4.8)

Constant	Value	Description
esriGeometryRing	11	An area bounded by one closed path
esriGeometryTriangleStrip	18	A surface patch of triangles defined by three consecutive points
esriGeometryTriangleFan	19	A surface patch of triangles defined by the first point and two consecutive points
esriGeometryTriangles	22	A surface patch of triangles defined by nonoverlapping sets of three consecutive points each

Figure 4.8 : Multipatch geometry types is taken from source [20].

4.2.2 City GML geometric model

Geometric modelling and description of 3D world objects collected through an imaging system has become an important topic, as they are essential for a variety of applications such as telecommunication, 3D city models, virtual tourist information system, etc.

CityGML is an open data model and XML-based format for storing and exchanging virtual 3D city models. CityGML uses a subset of the GML3 geometry model which is an implementation of the ISO 19107 standard. GML3, used with other OGC standards mainly the OpenGIS Web Feature Service (WFS) Specification provides a framework for exchange of simple and complex 3D models. It is implemented as an application schema of GML3, the extensible international standard for spatial data exchange developed within the Open Geospatial Consortium (OGC) and ISO TC211 [25].

The geometry model of GML3 consists of primitives, which may be combined to form complexes, composite geometries or aggregates. There is a geometrical primitive for each dimension: a zero-dimensional object is a *Point*, a one-dimensional is a _*Curve*, a two-dimensional is a _*Surface*, and a three-dimensional is a _*Solid* (Fig 4.9) [9].

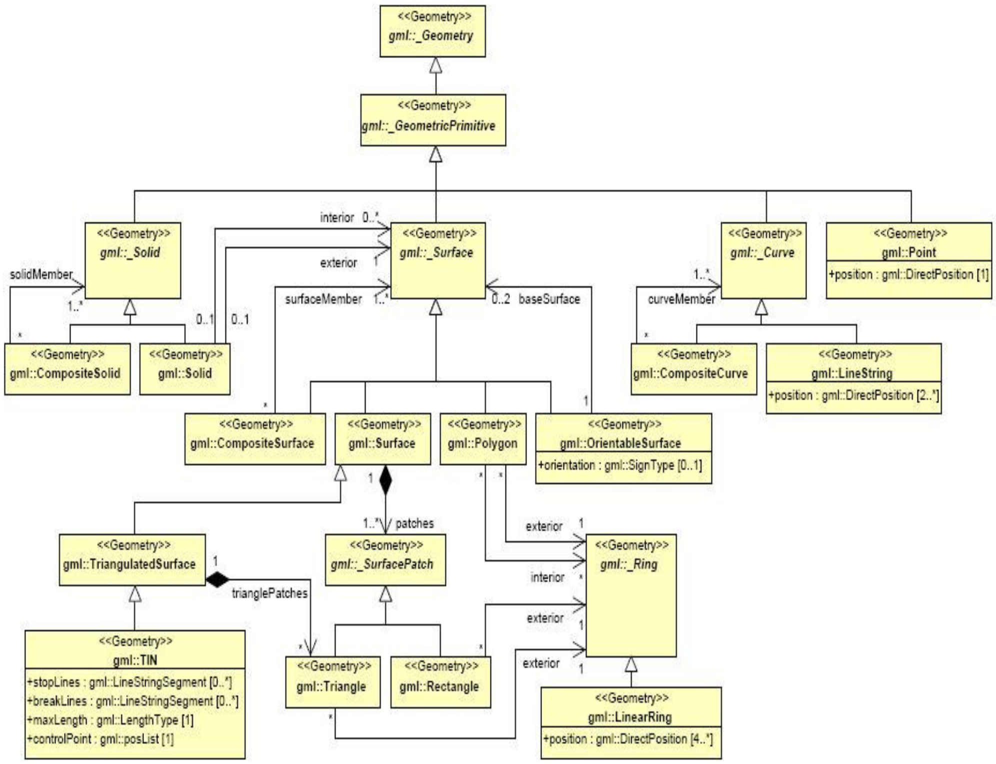


Figure 4.9 : UML diagram of City GML’s geometry model [9].

A solid is bounded by surfaces and a surface by curves. In CityGML, a curve is restricted to be a straight line, thus only the GML3 class *LineString* is used. Surfaces in CityGML are represented by *Polygons*, which define a planar geometry, i.e. the boundary and all interior points are required to be located in one single plane. Combined geometries can be aggregates, complexes or composites of primitives. In an *Aggregate*, the spatial relationship between components is not restricted. They may be disjoint, overlapping, touching, or disconnected. GML3 provides a special aggregate for each dimension, a *MultiPoint*, a *MultiCurve, a MultiSurface* or a *MultiSolid* (Fig.4.10). In contrast to aggregates, a *Complex* is topologically structured. A *Composite* is a special complex provided by GML3. Its elements must be disjoint as well, but they must be topologically connected along their boundaries. *A Composite* can be a *CompositeSolid,* a *CompositeSurface, or CompositeCurve.* Also each of the geometry can have its own coordinate reference system [9].

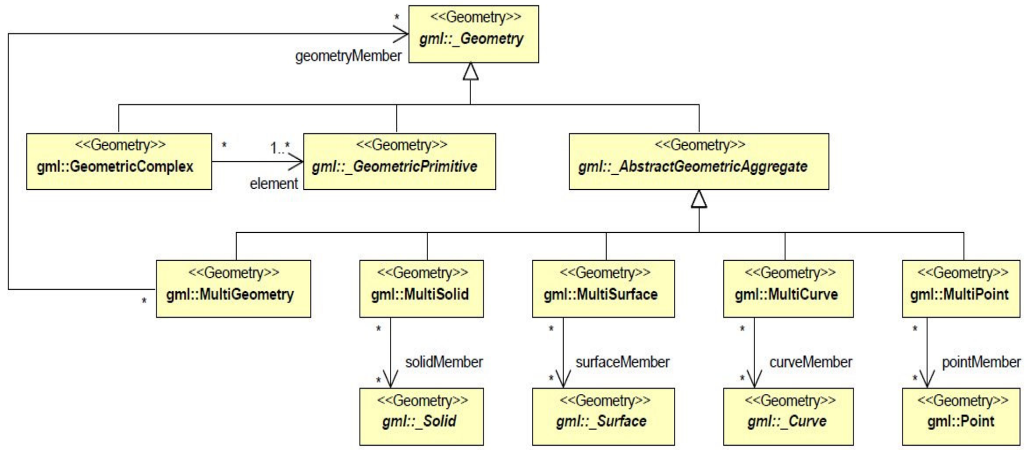


Figure 4.10 : UML diagram of City GML's geometry model: Complexes and Aggregates from OGC City GML implementation Specification [9].

Geometric modelling depends on the acceptance of the CityGML which is the target data format of the thesis work. According to ISO 19107 and GML3, geometries of geographic features are represented as objects having an identity and geometric substructures. GML3 provides classes for 0D to 3D geometric primitives, 1D-3D composite geometries, and 0D-3D geometry aggregates. Composite geometries like CompositeSurface must be topologically connected and isomorphic to a primitive of the same dimension (e.g. Surface). Surface in CityGML are represented by Polygons which define a planar geometry [10].

4.3 Data Model Analysis

In order to analyse the workflow, data model is used schematic representations of model structure using with UML diagram, geometric and semantic decomposition to visualise the discrepancies.

4.3.1 UML diagram of building model in photogrammetric tools

The photogrammetric tools are combined with ERDAS Imagine – LPS and ArcGIS Stereo Analyst in the thesis work. The building model of the ERDAS could not reach anywhere and it decided to build using UML. Modelling the UML diagram of the ERDAS Imagine LPS module and Arc GIS Stereo Analyst is another task to follow on the thesis project while working on the photogrammetric methods. The UML diagram will help the user to follow the thesis work. It is representing the details of the workflow. (Fig4.11)

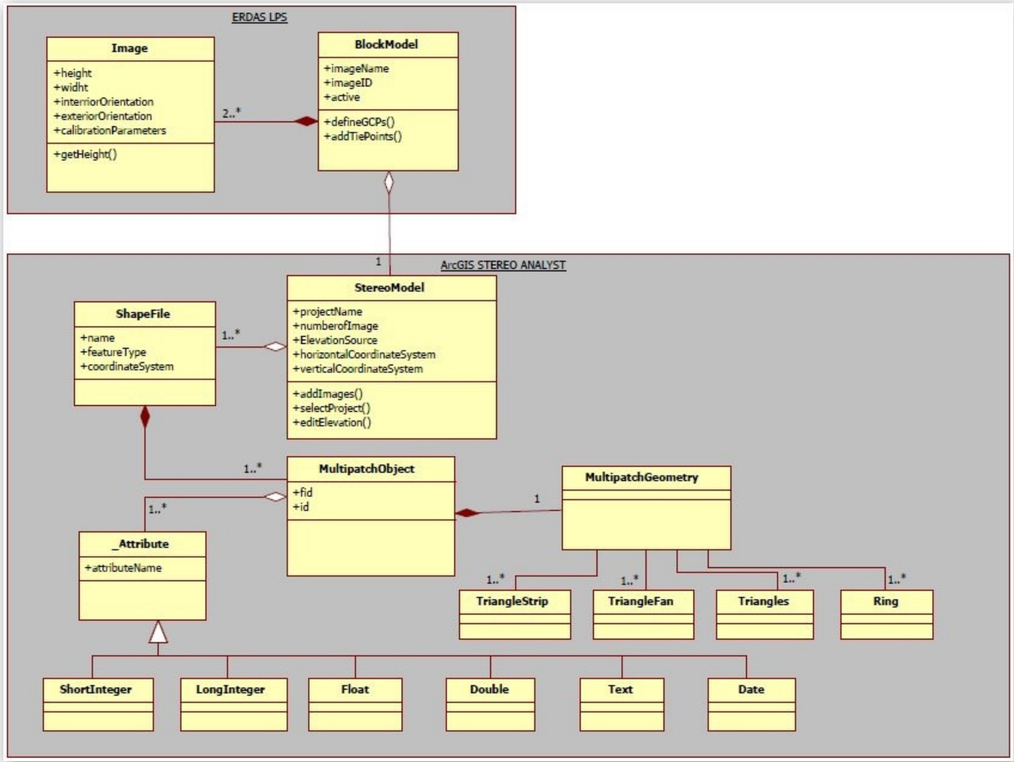


Figure 4.11 : Overview UML diagram of Photogrammetric Tool's Building Model

The Model is based on a geometric model which is obtained from Erdas imagine Leica Photogrammetric Suite (LPS). While building a stereo view on ERDAS Imagine LPS, generally at least two images are neccesarry. In the case study eight different raster images are used. those images have several attributes such as height, width, calibration parameters, interrior and exterior orientations. Each image has the "get height" method which shows the possibility to get height information. Block model is binary file which contains of processed images, Ground Control Points (GCPs), orientation parameters, image coordinates and projections. The block model is built using the images. As a relation between the block model and the images a composition is used. The composition means the images are part of the block model. The block model has attributes namely image name, image id, active. Also the block model has method such as define GCPs, add Tie Points. Therefore it can be defined GCPs and Tie Points inside the block model. Between the images and the block model has multiplicity which refers one block model needs two or more images. Similarly, it can be found an association between the Block Model and Stereo Model, specifying that the Stereo Model in Arc GIS Stereo Analyst is derived from the Block Model in ERDAS LPS.

When importing the Block Model into the Stereo Analyst for Arc GIS it comes with all information was defined in ERDAS LPS. Stero Model has attributes namely project name, number of image, elevation source, horizantal and vertical coordinate system. While importing the block model inside the ArcGIS Stereo Analyst, it is possible to have methods such as add images, select project and edit elevation. The multiplicity between Block model and Stereo Model has one to one relationship. After importing the block model, it is needed to create one Multipatch Shape file in the Arc GIS Stereo Analyst. Between the Stereo Model and the shape file there is an aggregation specifying that the stereo model requires at least one shape file. A multipatch shapefile allows a 3D model to be constructed and optionally textured for realistic scene generation. The aggregation relationship between the Multipatch Object and the shapefile is a composite aggregation. Every multipatch shape file has one Geometry class. A multipatch geometry may have triangle strips, triangle fans, triangles, or groups of rings. Moreover a multipatch shapefile itself has at least one attribute which user can use to enter the semantic information of the shape file. An attribute can be added in different type such as text, short integer, long integer, float, double, date.

4.3.2 UML diagram of building model in CityGML

Within the current version of the building model, an Abstract Building consists semantically of a Building and a Building Part classified in different LODs. Buildings can be represented in four level of details (LOD 1 to LOD4). For this project's case study, only LOD1 and LOD2 are chosen. The building model is the most detailed thematic concept of City GML. It allows for the representation of thematic and spatial aspects of buildings, building parts and installations.

The geometric representation and semantic structure of a Building model which defines as an *_AbstractBuilding* is shown in figure 4.12. The model is refined only from LOD1 to LOD2 due to the work purpose and requirement.

For intuitive understanding of the UML model, classes will also be shown in different colours such as blue, green, yellow. The UML diagram of the building model is depicted in figure 4.12.

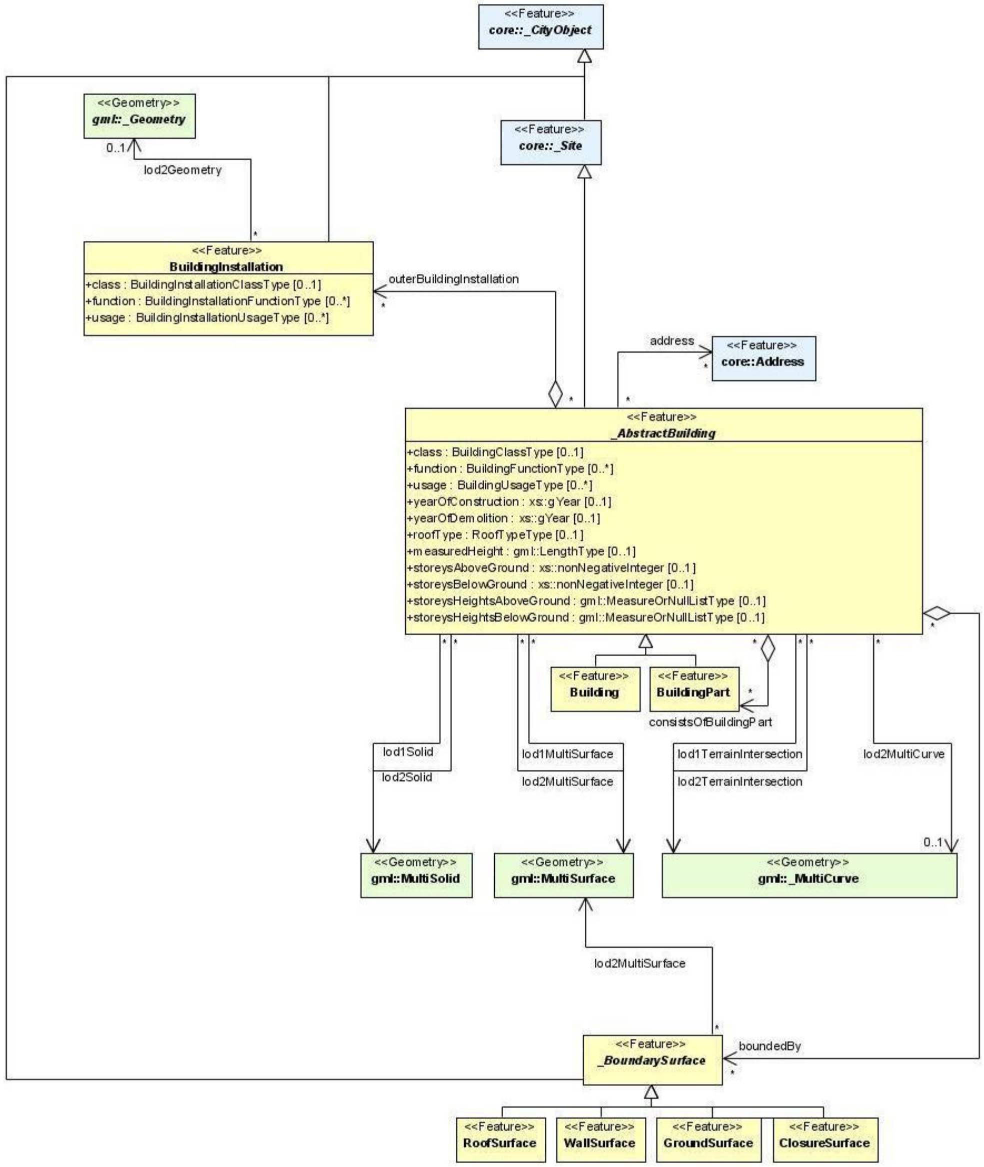


Figure 4.12 : Overview UML diagram of City GML's Building Model [9].

Therefore, not all aggregation levels are allowed in each LOD. In CityGML, an object can be represented simultaneously in different LODs by providing distinct geometries for the corresponding LODs. In LOD1, a building model consists of a geometric representation of the building volume. This geometric representation is refined in LOD2 by additional *MultiSurface* and *MultiCurve* geometries, used for modelling architectural details like a roof overhang, columns, or antennas. For the LOD 1-2 group the geometries *gml_Solid*, *gml_MultiSurface* and *gml_MultiCurve* are linked with the Abstract Building class. The City GML building model is minimized depending on the requirements of the thesis work [9].

4.3.3 Comparision of the data models

When we compare the UML diagrams deriving from the photogrammetric methods and ending with the City GML Building model it is possible to define differences and similarities. The differences of the UML diagrams are defined such as complexity structure, XML based, Level of Detail (LODs) definition, geometry and semantic …etc. In ERDAS LPS Building model can be defined with the UML classes in the workflow. So that the user who has aerial stereo images can be build a building model in the program. On the other hand the City GML Building Model is a standard for the representation and exchange of 3D city and landscape models issued by the Open Geospatial Consortium.

When comparing the geometry type of the models it can be seen that in City GML it's possible to distinguish surfaces of the building. It allows having multi surfaces. The City GML model has separate roof, wall, ground and building closure surfaces. On the other hand, in the Stereo Analyst model there is only one type of geometry a Multipatch geometry and with this geometry type one building has only one surface.

One of the other important difference is that solid form of the geometry. There is no way to define the object as Solid type in ERDAS Stereo Analyst. It is not allow to generate Solid type of geometry in ERDAS. On the other hand in CityGML data model the solid as MultiSolid is one type of defining the geometry. The solid geometry in CityGML allows the general composite design pattern which allows recursive structures and object hierarchies. Composite allows a group of objects to be treated in the same way as a single instance of an object. The composite composes objects into tree structures to represent part-whole hierarchies. The GML3 composite model realizes a recursive aggregation schema for every primitive type of the corresponding dimension. For example, a building geometry can be composed of the house geometry as *CompositeSolid*. On the other hand the garage geometry, the roof geometry and the geometry of the house body can be composed as *Solid* [9].

Predefined structure of the objects is not registered as specific type in ERDAS Stereo Analyst. It is often necessary to store and exchange attributes or even 3D objects which do not belong to any of the predefined classes in practical applications. The object is registered during the digitizing process is not allow to define one predefined object such as Building, Vegetation, Transportation. On the other hand in CityGML is well defined with detail.

Another difference is that in the CityGML model there are additional attributes deriving from the FME transformation like e.g. the Measured Height. When transform the shape format to city GML it is possible to calculate the height of the building wall surfaces and roof surfaces. Also in the CityGML classes are separated depending on semantic properties of the building such as Abstract Building, Building and Building Part. The abstract building class consists of building part. This meaning of semantic structure is not available in the photogrammetric building model.

It could be possible to mention that there are geometric differences between two models. In Arc GIS stereo analyst it must be mentioned that the only possible geometry type is the multipatch geometry. On the other hand in the CityGML building model has different geometry classes such as Multi solid, Multi surface and Multi curve. In the thesis case the geometry defined as multi surface.

In the CityGML Building Model is refined from LOD1 to LOD2 for thesis purposes. Therefore, all object classes are associated the LOD s with respect to the minimum acquisition criteria for each LOD. Take into consideration, the City GML Building Model allows the representation of thematic and spatial aspects of buildings, building parts and installations in four levels of detail, LOD1 to LOD4. On the other hand, the Photogrammetric Building Model does not include the LOD separation.

The CityGML building model is a XML based data model and model structure is complex. But in the photogrammetric model is not possible to mention about XML.

Although there are many of the differences in the UML diagrams, there are not many similarities due to complexity of the models and they are totally different from each other. The Photogrammetric model is more basic when compared to the CityGML building model. As an example of similarities, type of the Surface geometry as Multipatch in ArcGIS Stereo Analyst is very similarly structured with MultiSurface in CityGML.

The differences have listed into the Table4.1 between ERDAS LPS & ArcGIS Stereo Analyst and CityGML Data models.

Table 4.1: Shows the differences between ERDAS LPS & ArcGIS Stereo Analyst and CityGML Data models

Properties	**ERDAS LPS** **ArcGIS Stereo Analyst**	**City GML**
Complexity Structure	*No*	*Yes*
Semantic	*No*	*Yes*
XML based	*No*	*Yes*
Level of Details	*No*	*Yes*
Geometry	*Multipatch*	*Multi Surface*
Solid Type	*No*	*Multi Solid*
Predefined Structure	*No*	*Yes*

4.3.4 Mapping of the data models

The combinations of the ERDAS LPS , ArcGIS Stereo Analyst and CityGML UML diagrams is arranged in order to show the possible relations and related classes between the data models. The idea behind the combination is that to show similarities and possible connection in between CityGML data model and ArcGIS Stereo Analyst. There are several relationships between CityGML building model and ArcGIS Stereo Analyst Building model as associations in between several different classes. Those associations is listed as below:

- CityModel and StereoModel has one to one multiplicity association,
- AbstractBuilding and ShapeFile has many to one multiplicity association,
- BuildingInstallation and ShapeFile has many to one multiplicity association,
- BoundrySurface and ShapeFile has many to one multiplicity association,
- RoofSurface and ShapeFile has many to one multiplicity association,
- WallSurface and ShapeFile has many to one multiplicity association,
- GroundSurface and ShapeFile has many to one multiplicity association,
- ClosureSurface and ShapeFile has many to one multiplicity association,

There are associations which are coloured in different colour has different meaning such as;

- Green coloured association refers that building adress can added from external database and between *Address* and *Attribute* has one to many multiplicity association,
- Orange coloured association refers that attributes can be added during the digitizing process in ArcGIS Stereo Analyst and between *AbstractBuilding* and *Attribute* has one to many multiplicity association,

Considering to geometry in between the data models, there are two different possible associations;

- Geometry Model in CityGML and Multipatch Geometry Ring type has one to one-many multiplicity,
- Between MultiSurface geometry to Multipatch Geometry Ring type one to one-many multiplicity,

The Geometry MultiCurve and the feature ClosureSurface has coloured differently refers that both are not used in the thesis work.

All possible classes and relations can be seen in the figure 4.13

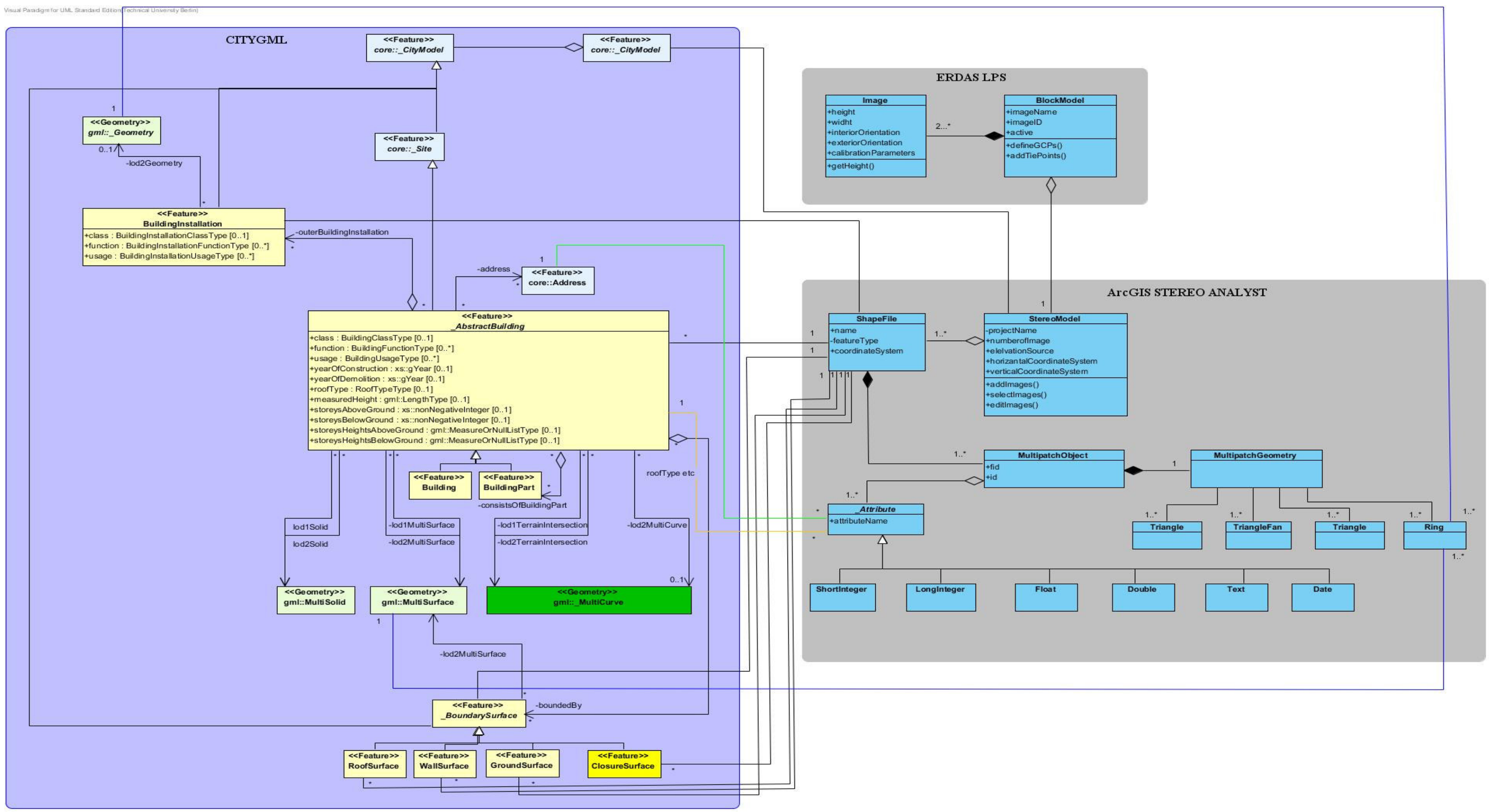


Figure 4.13 : Mapping between the UML diagram of City GML's Building Model & UML diagram of Photogrammetric Tool's Building Model

5. METHODOLOGY

5.1 Used Data

The data are eight aerial images and the digital terrain model of the test area which are explained in the following sub chapters.

5.1.1 Aerial images

The stero images are taken including camera paramaters from Istanbul Metropolitan Municipality. The number of the images is eight and from those images can be derived six stereo image pairs. Images coordinates are in Transverse Mercador Projection 30th longitude of central meridyen and 3rd sector for Turkey. In figure 5.1 below it is possible to see more information about the image coordinates are used in the thesis project. The resulation of the image is 50 cm for each image.

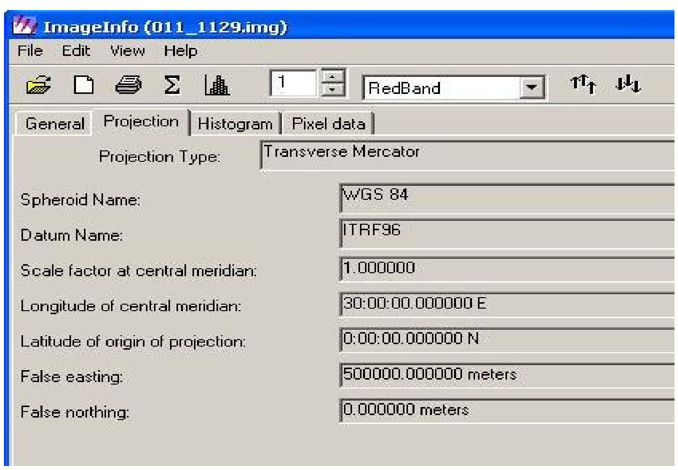


Figure 5.1 : The image projection information

5.1.2 Digital Terrain Model (DTM)

The DTM has also taken from the Municipality of Istanbul as ArcGIS shape format. Later on the geodatabase has created using ArcCatalog. The creation step can be read in the workflow chapter.

5.2 Applied Technologies

To use tecnologies which make the thesis work easier are Digital Photogrammetric Workstations, Erdas Stereo Analyst, Stereo Analyst and Feature Assist for ArcGIS, Feature Manipulate Engine (FME) and City GML as an accepted data model for 3D GIS.

5.2.1 Digital PhotogrammetricWorkstations (DPW)

Regarding to [21] Digital Photogrammetric Workstations (DPW) have been introduced in the market on a larger scale at the middle / end of the nineties and have become the standard for photogrammetric processing. The single most significant product of digital photogrammetry is the *digital photogrammetric workstation*. DPW have included in recent years means to deal with high resolution satellite imagery such as IKONOS or Quickbird together with aerial imagery. The aerial images has taken from the camera Zeiss LC 0030 including camera paremeters.

Recently new developments have been made in digital photogrammetry due to the availability of new hardware and software, such as powerful image processing workstations and enormous increased storage capacity which is very important issue for city modeling. Research and development efforts resulted in operational products that are increasingly being used by government organizations and private companies to solve practical photogrammetric problems. DPWs play a key role in the transition from conventional to digital photogrammetry [21].

An important aspect of analog or digital photogrammetric measuring system is the viewing component. Viewing and measuring is both performed stereoscopically and to achieve stereoscopic viewing is by displaying the two images of a stereopair on two separate monitors. For a human operator to see stereoscopically, the left and right image must be separated. Two solutions are available for viewing the stereo model. Firstly, a polarization screen is mounted in front of the display unit. It polarizes the light emitted from the display in synchronization with the monitor. An operator wearing polarized glasses will only see the left image with the left eye as the polarization blocks any visual input to the right eye. During the next display cycle, the situation is reversed and the left eye is prevented from seeing the right image. The second solution is more popular and less expensive to realize. It is based on active eyewear containing alternating shutters, realized by Liquid Crystal Displays

(LCD). The synchronization with the screen is achieved by an infrared emitter, usually mounted on top of the monitor. On the other hand, the polarizing screen and the monitor are a tightly coupled unit, offering less flexibility in the selection of monitors [21].

The two different solutions have been used in the thesis project. Mainly the LCD monitor with new technology 3D glasses and the LEICA Topo Mause which is created for the digitizing purposes has been used during digitizing works.

5.2.2 ERDAS - Leica Photogrammetry Suite (LPS)

While digitizing the roof structure in the test area using with stereo aerial images ESRI ArcGIS software is used with Stereo Analyst and Feature Assist extensions deriving from ERDAS Imagine Software. Stereo Analyst for ERDAS Imagine is an easy-to-use tool for collecting 3D features using stereo visualization. Stereo Analyst for ERDAS Imagine transforms your 2D GIS into real world dimensions by collecting 3D geographic information directly from imagery [22].

5.2.3 ESRI - Stereo Analyst / Feature Assist for ArcGIS

Stereo Analyst for ArcGIS is a stereo feature collection tool for collecting and updating feature data in ArcGIS Software. Stereo Analyst extends the standard tools of ArcGIS into the 3D world and provides an easy to use solution for the extraction of buildings from stereo images. ERDAS is launching a suite which provides tools for creating standard data products, including terrain and feature data layers of products to support a set of focused tasks. FeatureAssist for ArcGIS collects complex building roofs, with predifened roof types and with the option of extending these roofs to the ground to create real world 3D city models. Moreover using with ERDAS Terrain Editor for ArcGIS, users can edit Geodatabase terrains [23].

Regarding to the source [23] Stereo Analyst for Arc GIS can be used in such examples,

- Available for ArcGIS Desktop at any level
- Standard ArcMap Editor tools are used for creating and modifying features within the Stereo View
- Multi-user topological editing support natively through ArcGIS

- Supports the import of data from several industry standard triangulation package formats
- Supports the direct load of "oriented images" including NITF and .SUP files directly via the Add Data dialog in ArcMap
- Supports Vertical Datum transformations to eliminate the need to post process Z values
- Supports 3D Layer Based Snapping
- Feature Data Conversion Tools are provided to convert existing layers to 3D.

Stereo Analyst for ArcGIS adds several toolbars such as the Stereo View, the Stereo Enhancement and the Stereo Advanced Editing toolbar and features to ArcMap when it is installed. The Stereo Analyst toolbar provides access to importers and exporters as well as preference settings. The Stereo View toolbar provides tools to manipulate data in the Stereo window. The Stereo Enhancement toolbar controls the operation of image enhancement in the Stereo window. The Stereo Advanced Editing toolbar provides better control over the handling of Z values while editing features [24].

Another extension being used in the thesis project is FeatureAssist for ArcGIS. It is an add on to Stereo Analyst for ArcGIS for the collection of roof structures in the ESRI Multipatch format. ESRI Multipatch format supports a multipatch shapefile allows a 3D model to be constructed and optionally textured for realistic scene generation. Using templates, FeatureAssist for ArcGIS can quickly collect these features, handling varying degrees of complexity. In addition to the templates, manual construction and editing tools are provided for the creation or modification of any roof shape. To create a scene true to reality, roofs can be 'extended to the ground' or 'extended to roof base' into Feature Assist extension called as "Rooftop Tools".To create a 3D model that tools can be used in a visualization package. FeatureAssist for ArcGIS enables a user to collect complex buildings. Predefined templates and manual construction tools can be combined to collect any building shape into a single model. Many roofs have a common shape. Building templates within FeatureAssist provides an easy and useful way to collect common roof structures [23].

Regarding to the source [23] FeatureAssist for ArcGIS can have the properties such as,

- Templates include: Dome, Gabled, Steepled, Hipped, Cupola
- Manual construction tools supplied to allow for the construction of any shape
- Tools provided for edit tasks such as reshape and copying and pasting parts
- Optionally drop rooftop to existing elevation source or current elevation of the floating cursor for 3D model construction
- Snapping options provided including face, vertex, edge and guideline snapping
- Format support: only "ESRI Multipatch"

5.2.4 Feature Manipulated Engine (FME)

FME is a Spatial ETL (Extract, Transform and Load) application concept. FME provides unlimited flexibility in data model transformation and distribution and also format support for data translation and integration. FME is a central feature manipulated engine over 200 formats. User can read from any supported format and write to any supported format. FME's data model is designed to cover all possible geometry and attribute types.

User can accurately restructure the schema of the data as it moves from the source to the destination, without losing semantic information. FME is also smart enough to compensate automatically for limitations in a destination format to create a seamless translation process whenever possible. The detail information about FME can be found under the Chapter 6 namely is "Conversion with FME".

5.2.5 CityGML

CityGML is the international standard for the representation and exchange of 3D city models issued by the Open Geospatial Consortium (OGC). Also it is implemented as an application schema for the Geography Markup Language (GML) 3.1.1 of the OGC and the ISO TC211. Moreover, it is based on the international standards published by OGC and the ISO 19100 standards family. CityGML defines classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantic and appearance properties [9].

CityGML was developed within Special Interest Group 3D within initiative Geodata Infrastructure North-Rhine-Westphalia. The SIG 3D consists of 70 members from industry, academic institutions and public administrations.

CityGML defines classes and relations topographic objects in cities and regional models comprising built structures, elevation, vegetation, water bodies..etc on thematic level. Its information model covers the spatial, appearance and thematic aspects of 3D city models. By covering the thematic information, CityGML complements 3D graphics formats like KML and X3D/VRML. The thematic information supports the use of virtual 3D city models in different aplications such as simulations, urban data mining, facility management, planning...etc [31].

The semantic data model of CityGML is made up of representations for digital terrain models, buildings, vegetation, water bodies, transportation facilities and also city furniture. CityGML provides ontology for 3D virtual city models consisting of thematic objects, attributes and associations. City objects are decomposed according to their topological structure and not according to graphical considerations, e.g. Buildings can be decomposed into BuildingParts [25].

CityGML differentiates with five level of details (LODs) from LOD0 to LOD4. The analyst and visualization of the city object may be represented in differetn LODs with regard to different degrees of resolution [31].

Details about CityGML can be found in the implementation specification document [9] which can be freely downloaded from the "Best Practice Paper" section of the standards section of the OGC website.

5.3 Overview of The Workflow

An image of the Earth’s surface is a wealth of information. Images capture a permanent record of buildings, roads, rivers, trees, schools, mountains, and other features located on the Earth’s surface. But images go beyond simply recording features. Images are snapshots of geography, but they are also snapshots of reality. Images chronicle our Earth and everything associated with it; they record a specific place at a specific point in time. They are snapshots of our changing cities, rivers, and mountains. From snapshots to digital reality, images are pivotal in creating and maintaining the information infrastructure used by today's society. Today's geographic information systems have been carefully created with features, attributed behaviour, analyzed relationships, and modelled processes [24].

The computer which is used during the thesis project Dell Precision T3400 has the Graphic card is NVIDIA Quadro FX 4800 and Intel Core 2 Duo, 2.66 GHz, 8 GB RAM, 2 TeraByte hard disk. Leice Topo Mouse was used during the digitizing work in Arc GIS Stereo Analyst.

The images which are used in the project which is chosen in the historical peninsula of istanbul where is explained in the next session.

Figure 5.2 : Shows the six 1/1000 scaled map sheet in the test area

5.3.1 Project area

Istanbul is one of the most famous historical cities in the world and full of cultural history. During the centuries the city hosted as the capital of Rome, Byzantium and the Ottoman Empire and it is an interesting center for the world owing to its natural beauties. In addition to the historical structures which are the heritage of these cultures. The city was built as a highly important city of the Roman Empire in the year 330 and named Constantinople. After Rome was divided into two parts as the Eastern Rome and Western Rome, it was made the capital of the Eastern Roman Empire[11].

The reason for the choosing the historical peninsula of Istanbul, which is the district named Fatih, has been included in the capital of culture for 2010 and was chosen by UNESCO and constitutes one of the liveliest regions of the central area. The historic city of Istanbul is situated on a peninsula flanked on three sides by the Sea of Marmara, the Bosphorus and the Golden Horn.

This area was chosen as project working area for many reasons. it's attractiveness for tourism, historical backround, several different types of roof structures and complex area … etc.

5.3.2 Camera parameters

The camera type and all the camera information is in the table 1.

Table 5.1: Orientations Parameters of the images

BLOCK NO	COLON NO	IMAGE NO	PERSPECTIVE CENTERS (meters)			ROTATION VALUES (Degrees)		
			Xo	Yo	Zo	OMEGA	PHI	KAPPA
063	10	101134	414647.106	4542106.810	1376.276	0.621800	1.076800	178.325700
063	10	101135	414280.177	4542110.552	1379.001	-0.330400	0.932000	178.010500
063	10	101136	413910.166	4542116.072	1382.065	0.169600	1.145800	177.777400
063	10	101137	413537.564	4542122.471	1384.948	-0.441800	0.788100	175.847200
063	11	111126	413548.170	4542848.874	1381.619	1.595200	0.213500	3.267400
063	11	111127	413915.760	4542845.142	1378.395	0.818900	0.166900	3.236400
063	11	111128	414283.618	4542837.552	1379.145	-0.304000	-0.610400	1.808000
063	11	111129	414650.183	4542830.568	1379.239	-0.258400	-0.363500	2.037200

Rotation System: This field displays the rotation system associated with the block file. The rotation angles are used to form a 3 by 3 rotation matrix which is used within the aerial triangulation functional model as block bundle adjustment.(Fig.5.3)

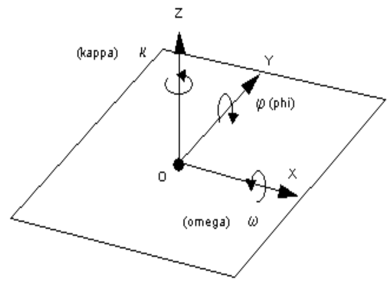


Figure 5.3 : Rotation System associated with the block file.

Omega, phi, and kappa define the orientation of the sensor as existed when the photograph or imagery was captured. The three rotation angles define the relationship between axis of the ground system and image coordinate system. Omega is a rotation about the photographic x axis, Phi is a rotation about the photographic y axis and Kappa is a rotation about the photographic z axis.

Camera Type is Zeiss LC 0030. The camera parameters can be seen in table 2.

Table 5.2: Camera Calibration Parameters

focal length (mm)	305.099															
ppac (x,y) (mm)	-0.014	0.008														
ppbs	0	0														
film_format (mm)	230	230														
fiducial (x,y) (mm)	1	110.008	0.002													
fiducial (x,y) (mm)	2	-110.009	-0.01													
fiducial (x,y) (mm)	3	0.004	110													
fiducial (x,y) (mm)	4	-0.008	-109.997													
lens_distortion_flag	on															
input_mode	linear															
distortion_spacing(mm)	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
distortion_deltas (mik. m)	0	-1	-1	-2	-2	-2	-1	-1	-1	0	1	1	1	0	0	1
distortions (mik. m)	0	-1	-1	-2	-2	-2	-1	-1	-1	0	1	1	1	0	0	1
camera_type	frame															
media_type	film															
focal_length calibration_flag	on															
calibrated_focal length_stddev	0.03															
ppac_calibration flag	on															
calibrated_ppac stddevs	0.003	0.003														
self_calibration enabled_params	4095															
antenna_offsets	0	0	0													
camera_parameters																

5.3.3 The processing workflow

First of all the workflow is described using different colours and in different boxes referring the different program which is used during the thesis work. Also on the workflow diagram the number refers to the steps which are in order of the thesis. (Fig.5.4)

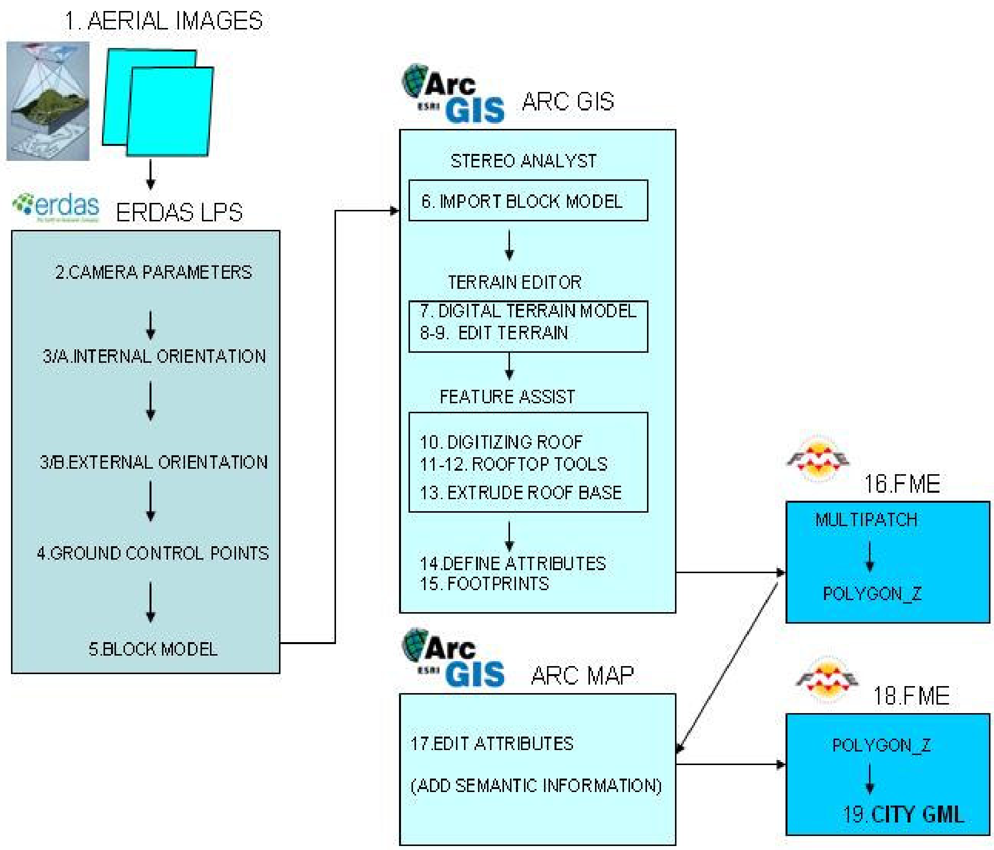


Figure 5.4 : Workflow Diagram

Starting the photogrammetric measurement with stereo images is not possible to without any aerial images. For this reason the aerial images has been taken from Istanbul Metropolitan Municipality (IMM). After that the images have been developed using in ERDAS Leica Photogrammetric Suite (LPS) and following the steps which are listed below:

1. *Adding Aerial Stereo images on ERDAS LPS and arranging to get 3D view*

The stereo aerial image are the most important data in this thesis work. Also one of the most important step is arranging the three dimensional stereo view from this aerial images. ERDAS imagine program is used for creating the stereo view.

After import the image on ERDAS LPS almost the first step is the defıning the camera parameters with the following three kinds of information – principal point, focal length and the coordinates of the fiducial marks.

Average Flight Height (units): The average height above ground at which the images were captured. This distance is not required for triangulation. The unit of measure here is the same as the vertical unit of measure you entered in block setup [26].

In order to calculate the average flight height (H), with Focal length and photo scale (1/S), user can use the following simple expression. For the thesis project used value is 1220 m with respect to the calculation:

$$\approx H = \text{Focal length} * S = 305.099 * 4000 = \sim 1220 \text{ m} \quad \textbf{(5.1)}$$

Of course, the flight height must be converted to *meter* units.

2. *Defining the camera properties associated with the camera parameters.*

This is the process of defining the internal geometry of the camera. Fiducial marks are measured on the image and then compared to the calibrated positions of the camera to derive a solution. Once the fiducial marks have been measured, a 2D affine transformation is used to determine the origin of the photo coordinate system. The origin of the photo or image space coordinate system defines the location on the image where the optical axis intersects with the image plane. Once the photo coordinate system has been defined, each subsequent image measurement is referenced to it. The fiducial marks is defined with respect to RMSE for the each image.

The image coordinate system is defined in the camera parameters as ITRF96 in TM30-3. It can be seen with more information about the coordinates in figure 5.5.

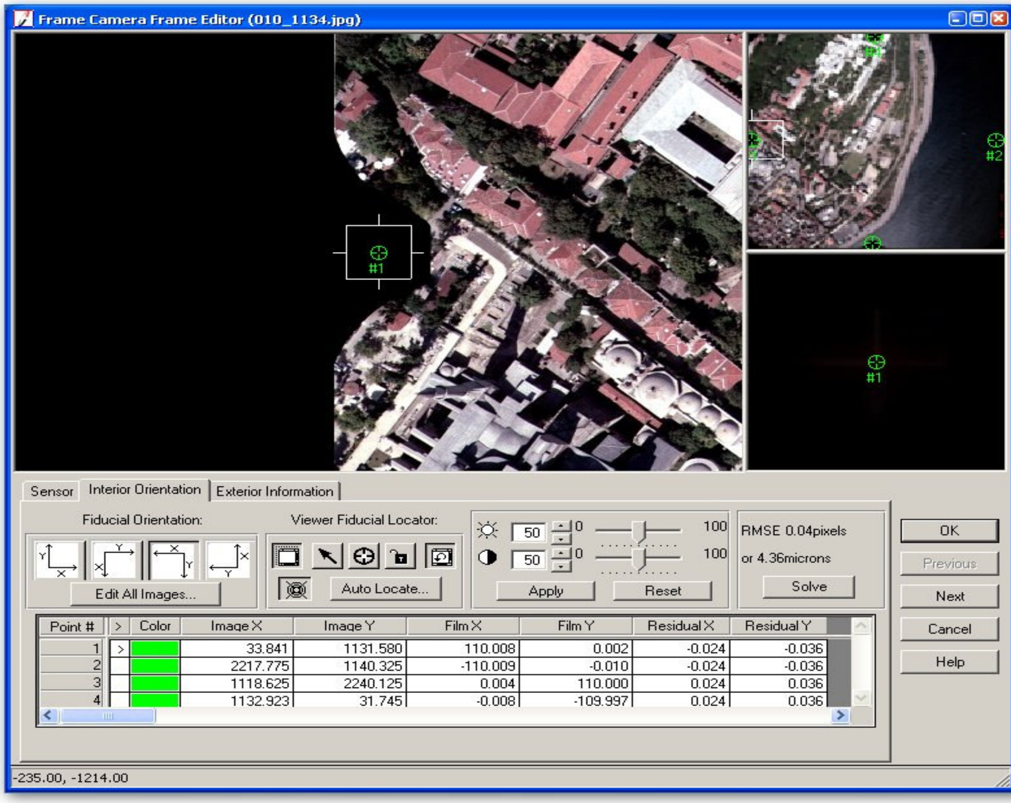


Figure 5.5 : Defining fiducial marks

In order to get correct stereo view the fiducial orientation is important to define in right orientation depending on flight way. In the project case it defines for the images name from 010_1134 to 1137 left –X and from 010_1126 to 010_1129 right –X direction as can be seen from the figure 5.6.

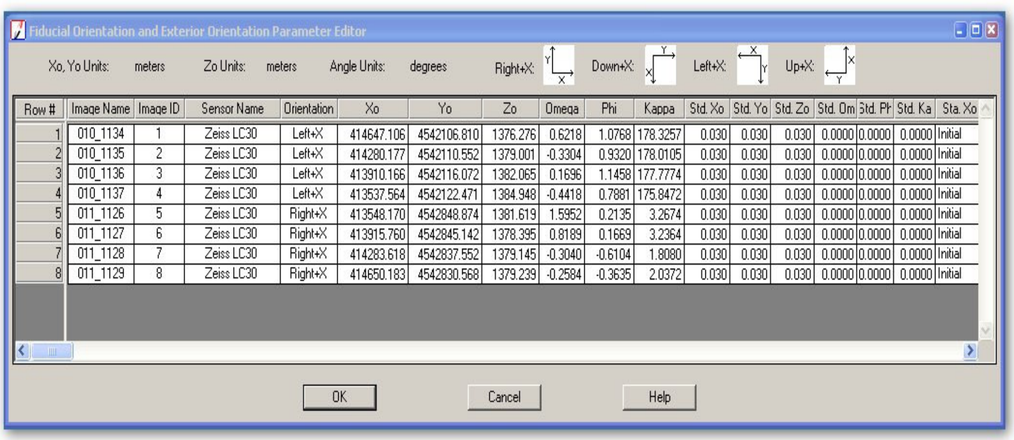


Figure 5.6 : Fiducial Orientation and Exterior Orientation Parameters

3. *Orientation Parameters*

A. Interior Orientation Parameters

Interior orientation establishes a link between the calibrated positions of the fiducial marks on the camera focal plane and the fiducial marks exposed on each of the aerial images. The interior orientation menu is accessed in the LPS Project Manager main screen and clicking on the interior orientation tab.

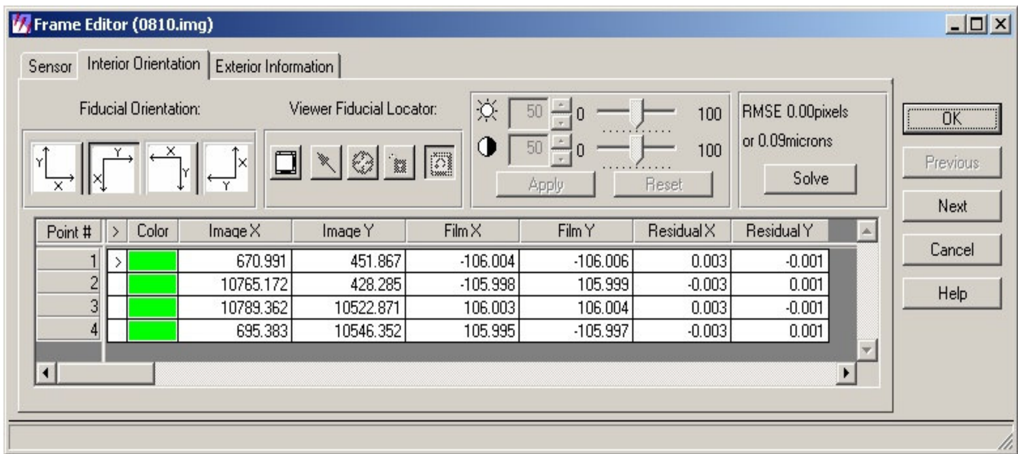


Figure 5.7 : The Frame editor for interior and exterior informations

In this menu there are two tasks to complete before you can have the crystal.
1. Set up the fiducial orientation

Fiducial orientation establishes the geographic orientation of each image. As there are just four models of fiducial oriantation, it is not uncommon for the fiducial layout to not fit any of them.

2. Locate the fiducial marks on each of the images

Clicking the viewer button enables the viewer fiducial locator. Within this window the image fiducial marks must be identified and marked in the same order as they were entered into the sensor fiducial calibration menu. Navigation to the fiducial mark in question is conducted by moving the viewer box around in the top right overview window. Once the correct fiducial is identified and positioned in the close up view, a point is placed on top of it using the point measurement tool. Inaccuracies can be corrected later on using the tool. (Fig 5.7).

Each fiducial mark covers an area of many pixels. Obtaining good precision, or overall RMSE, is likely to be an iterative process of moving all of the fiducial locating points around a few pixels at a time until individual residual errors are acceptably small. In the project, RMSE values are less then 0.05 pixels for each image.

B. Exterior Orientation Parameters

The exterior orientation describes the location and orientation of the bundle of rays in the object coordinate system with the 6 parameters: projection center coordinates (X0,Y0,Z0) and the rotations around the 3 axis (omega, phi and kappa). The definition of the rotations has to be respected the camera information as can be read in table 2, most often the successive rotations with the sequence omega, phi, kappa or phi, omega, kappa are used.

4. *Adding the Ground Control Points on ERDAS LPS*

LPS Project Manager uses the self-calibrating bundle block adjustment method in its triangulation process. By doing so, the internal geometry of each image and the relationships between overlapping images are determined. When multiple images are involved in a data block, such a modeling method can significantly ease the need of acquiring many GCPs.

In this step, we define ground control points (GCPs). In order to define the GCPs the public webpage of the Municipality has been used for X and Y coordinates and google earth has chosen to collect approximied Z coordinates. Collecting the GCPs with this method was only way to found out. It is admitted that the way is not accured enought. Figure 5.8 shows the collecting X, Y coordinates on the website of Istanbul Metropolitan Municipality (IMM) and figure 5.9 shows to collect Z coordinates on google earth.

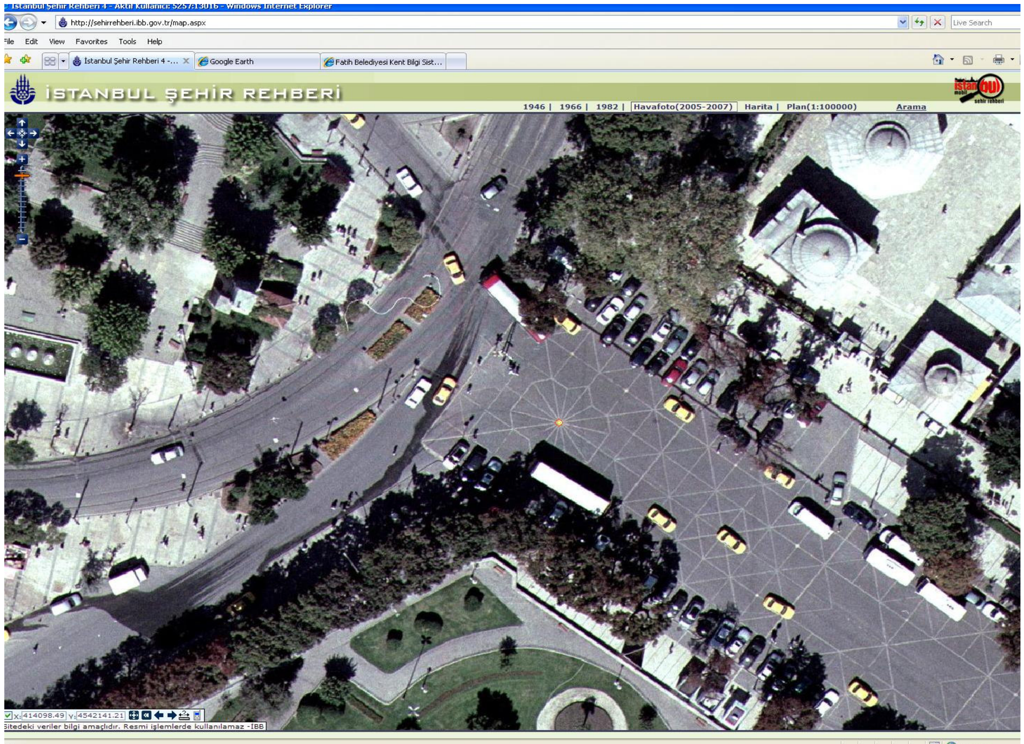


Figure 5.8 : X,Y coordinates are defined on IMM official website.

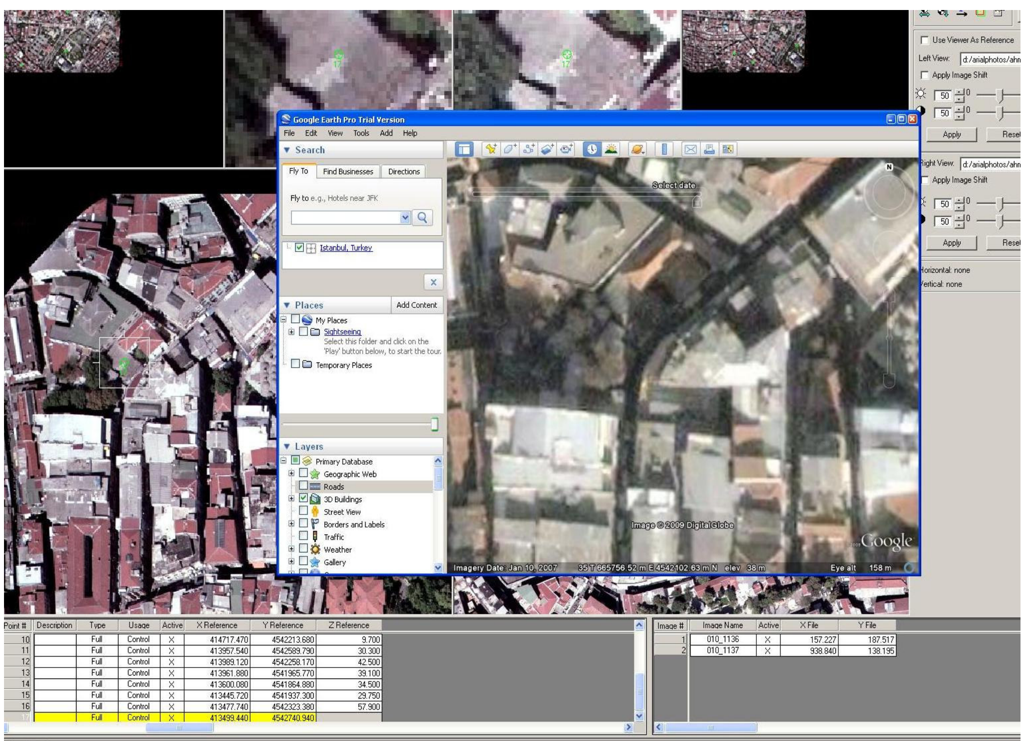


Figure 5.9 : Z coodinates are defined using Google Earth Program

- Performing automatic tie point collection

Image tie points are the common points in overlapping areas of two or more images. They connect the images in the block to each other and are necessary input for the triangulation. LPS Project Manager automates the identification and measurement of tie points so that your work and time for manual measurement are drastically reduced. Automatic tie point collection tools are used to measure the corresponding image positions of tie points on overlapping images. Aerial triangulation is performed to adjust the exterior orientation parameters and determine the XYZ coordinates of the tie points. Lastly, orthorectification is performed for each image sequentially using a constant elevation value.

5. *Building a Block Model in ERDAS LPS*

After having obtained control, check, and tie points, LPS Project Manager has all the information it needs to perform aerial triangulation. This step in the process establishes the mathematical relationship between the images that make up the block file and calculates exterior orientation.

From the Edit menu, select Triangulation Properties. The Aerial Triangulation dialog opens. The important point on the aerial triangulation is GCP Type and Standard Deviations section, click the Type dropdown list and select Same Weighted Values. Statistical weights can be assigned to the input observations including image coordinates, GCPs, exterior orientation, and interior orientation. The use of statistical weights assists in minimizing and distributing data error throughout the block of imagery, thereby ensuring highly accurate results. After defining all these steps when clicking the run button, a Triangulation Summary dialog generates and opens. All the triangulation result can be seen at the Appendix A.1. As can be seen in the triangulation summary report Total image unit-weight RMSE = 0.0353 after 4th iteration.

6. *Import the Block Model in the Arc GIS Stereo Analyst*

Stereo Analyst for ArcGIS supports importing LPS block files consisting of frame, digital, satellite imagery or aerial images. For a standard project, all camera or sensor information is contained in the block file structure itself and does not require repairing.

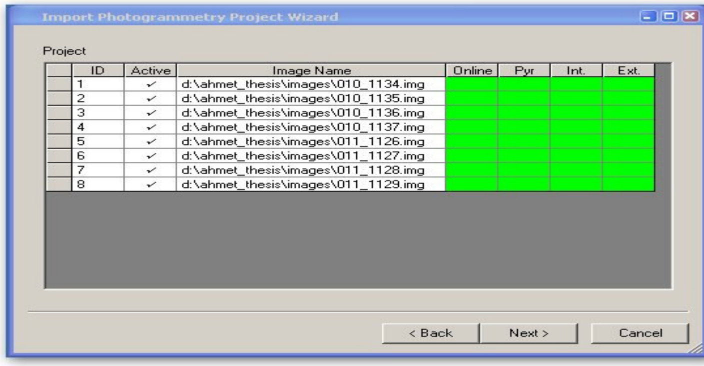


Figure 5.10 : Import the block model in the Arc GIS Program

Block files are binary files and have the .blk extension. They might contain information associated with one image, a strip of images that overlap or are adjacent to one another, or several strips of images. The block file contains all the dialog box information associated with the block including imagery locations on your system, camera/sensor information, fiducial mark measurements, ground control point information, image measurements, projection, spheroid, and datum information. [1]

As can be seen in figure 5.10 that the main cell array indicates the data path for each image in the project. The remaining cells indicate the status of pyramid layers and whether orientations were found for each image in the project and it shows with green colour.

In the Arc GIS stereo Analyst, the Import Photogrammetry Project wizard provides full control of all aspects of creating oriented images from photogrammetry projects. When import the block model which is build in Erdas Imagine LPS into the wizard (Fig 5.10), it can be followed the steps as select Project File, edit the mean ground elevation, the horizontal coordinate system and the vertical coordinate system for the oriented images. The horizantal coordinate system has been defined during the camera parameters. Digital cameras with integrated global positioning systems (GPS) and inertial navigation systems producing orientations based on the WGS84 datum.

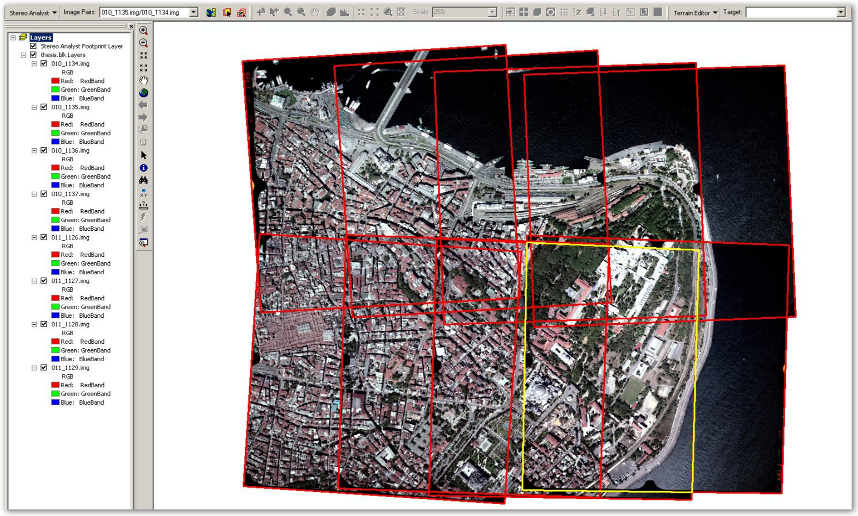


Figure 5.11 : View of the stereo images block in the Arc GIS Stero Analsyt.

During the collection of raw aerial photographs, flight plans are purposely coordinated so that resulting photographs along a strip properly overlap to create an optimum 3D digital representation of the interest area. The overlap percentage between two photographs in a strip is commonly 60 percent. It can be seen easly the overlapping area between the images. Figure 5. 11 indicates the image border with red line and the test area indicates with yellow line.

7. *Adding a Digital Terrain Model in ArcGIS*

Before begining to use Terrain Editor for ArcGIS, it must be added a geodatabase terrain and a stereo pair. A geodatabese terrain is created using with ArcCatalog Program. After creating the geodatabase, a feature dataset is needed for the next steps. The next step is creating a geo-database terrain using the digital terrain model which has been taken from the IMM. (Fig 5.12) The only possible way to extrude the roof to terrain on feature assist must be added a geo-database terrain as it explains on ninth step on the thesis workflow.

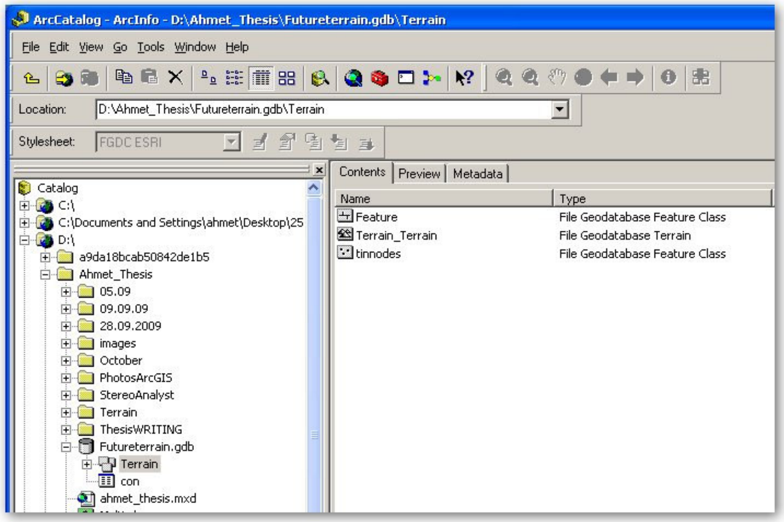


Figure 5.12 : Creating a terrain geodatabase on ArcCatalog

When a geodatabase terrain is added to the map document, Terrain Editor for ArcGIS obtains a reference to the terrain and constructs a terrain model based on the component feature classes of the geodatabase terrain. When a stereo pair is loaded in the Stereo window, the terrain features, points, and breaklines display in stereo along with triangles and contours.

8. *Editing the Terrain using the Terrain Editor*

Once these files are added, user could select Start Editing from the Terrain Editor drop down list and start using Terrain Editor for Arc GIS.

Terrain Editor is an ArcGIS extension that extends Stereo Analyst for ArcGIS to provide terrain editing capabilities for geodatabase terrains. Terrain Editor for ArcGIS provides you with point, line, and area editing tools for use with the stereo environment of Stereo Analyst for ArcGIS. The Terrain Editor for ArcGIS extension works by constructing a terrain model for each geodatabase terrain loaded. With one or more terrains loaded in the map document, Terrain Editor for ArcGIS lets the user select a terrain for editing. Each feature class that participates in the terrain is editable on a layer-by-layer basis. As features are modified, the terrain is dynamically updated in the Stereo window.

9. *Adding Terrain on Feature Assist*

After creating the terrain geodatabase which is needed on extrusion process, it can be added on Feature assist under the stereo analyst rooftop options.(Fig 13)

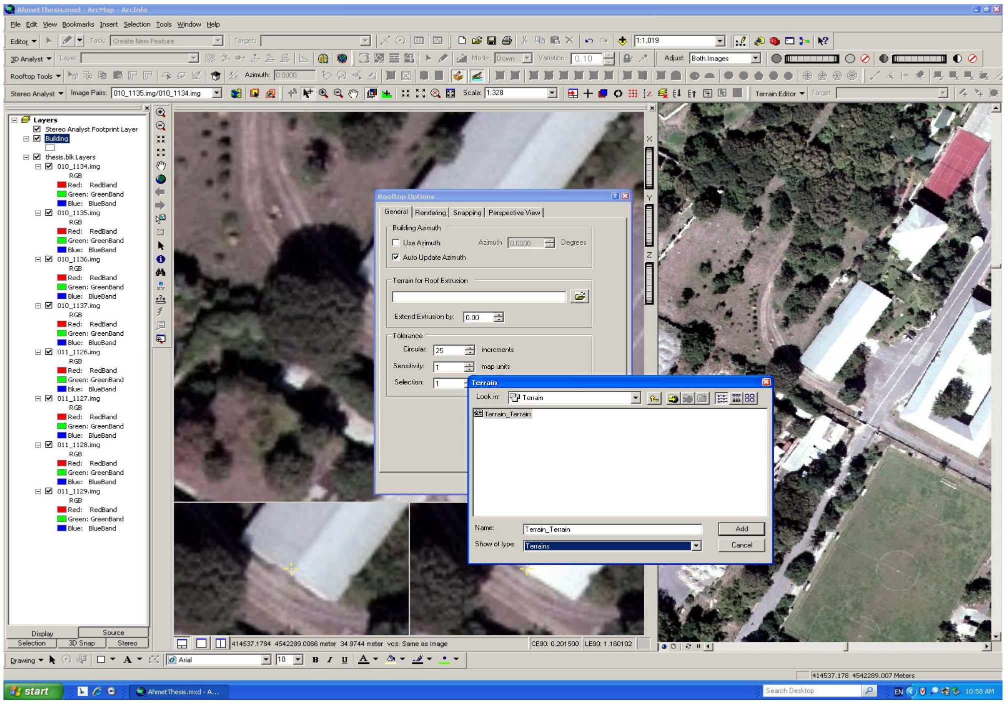


Figure 5.13 : Adding Terrain on Feature Assist

10. *Digitizing the roof structure from Aerial stereo images in 3D view using Feature Assist for ArcGIS*

In order to digitize the roof stucture of the buildings, FeatureAssist for ArcGIS is chosen during the digitizing process. FeatureAssist for ArcGIS is another optional extension to Stereo Analyst for ArcGIS. It allows you to collect roof and building structures in the ESRI multipatch format. Using templates, FeatureAssist for ArcGIS can quickly collect these features while handling varying degrees of complexity. In addition to the templates, manual construction and editing tools are provided for the creation or modification of any roof shape. To create a scene that is true to reality, roofs can be extended to the ground or to an existing terrain, creating a 3D model that can be used in a visualization package.

11. *Chosing the roof template in ArcGIS Stereo Analyst to digitize it*

The Stereo Rooftop toolbar is a standard dockable menu bar on Feature Assist extension. This toolbar provides access to all of the rooftop collection functionality implemented as standard ArcMap commands, tools, and options, as well as several rooftop templates. The templates are Flat Roof, Gabled, Hipped, or Steeple Roof…etc. (Fig 5.14).

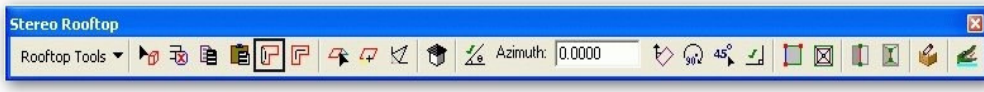


Figure 5.14 : Stereo Rooftop on Feature Assist

Ring Edit Tool button as shown on fig 5.14 on the Stereo Rooftop toolbar can be use to construct a rooftop that is irregularly shaped or not well represented by an existing template. Several different roof templated has been used during the thesis project. When more roof type templates necessary on the project, it can be found on the Stereo Rooftop Extended toolbar. (Fig 5.15)

Figure 5.15 : Stereo Rooftop Extended on Feature Assist

The Buttons on the Stereo Rooftop Extended toolbar are as follows: (Fig 5.16)

Figure 5.16 : Stereo Rooftop Roof type list

12. *Digitizing the building installations and merging with the building roof*

Before activating the Stereo Rooftop toolbars,it must be first added a multipatch shapefile and enable editing from the Editor toolbar. Digitizing process starts choosing right roof template on stereo rooftop toolbar. The user should be digitize carefully on correct height information using with the 3D glasses and the topo mouse. During the digitizing process every corner of the roof must be chosen by the user after activated the fixed cursor and the 3D floating cursor mode on the Stereo Analyst. After digitizing the roof structure of any building right click button and select Finish Sketch, or move the floating cursor to the height of the ground and then right-click and select Extrude Base. When finished the digitizing of any roof surface it can be combined any template with another template or a manual construction of a building by right-clicking and selecting Finish Part. As the details of each roof structure defined, Finish Part lets the user continue adding elements to the sketch.

Also it is possible to merge the building parts using Merge Selected Buildings from the Rooftop Tools drop-down list on the Stereo Rooftop toolbar. All of the buildings you selected are merged into one building. The user can select Finish Sketch the various parts of the feature to merge into a single building, after all features of the roof are collected. Once the template has been matched to the rooftop, the user can finish sketch to simply collect the roof or finish part and select another template or manual construction tool to ‘add’ to the existing sketch. After finishing the sketch and merge the buildings roof can be extrude to terrain using right click and select Extrude to Terrain. As it can be seen on Figure 5.17 all the surfaces belongs to the selected building is merged together including building instilations such as chimney and dormers.

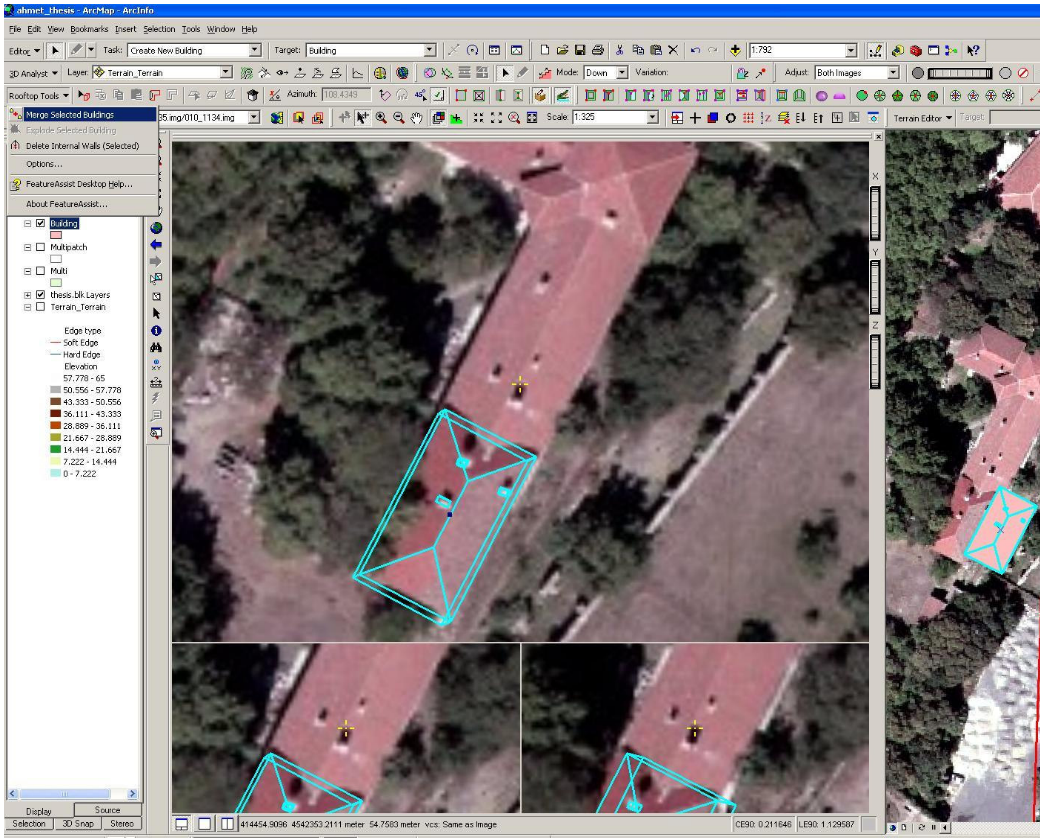


Figure 5.17 : Merge Selected Buildings on Rooftop Tools

13. *Extruding the roof structure with Extrude Roof Base based on the terrain model.*

Using templates, FeatureAssist for ArcGIS can quickly collect these features, handling varying degrees of complexity. In addition to the templates, manual construction and editing tools are provided for the creation or modification of any roof shape. To create a scene true to reality, roofs can be 'extended to the ground' or to an existing terrain, creating a 3D model that can be used in a visualization package.

While digitizing the roof structure the snapping toolbar could be active in order to better capturing of roof borders and make it together. In figure 5.18 can be seen the snapping options are activated for having better results.

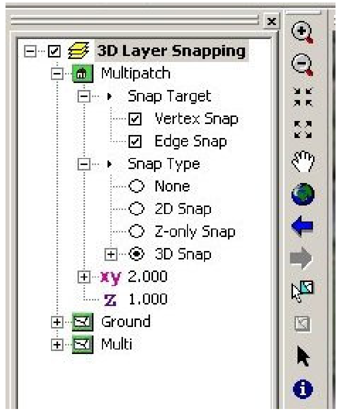


Figure 5.18 : Snapping options on Arc Map

When finishing the digitizing of any building`s roof structure on feature assist stereo rooftop it can be selected Finish Sketch, or move the floating cursor to the height of the ground and then right-click and select Extrude to Terrain or Extrude to Roof Base as it can be seen on figure 5.19. Extrude to roof based option is chosen in order to get flat ground surfaces. Otherwise the ground surfaces are depending on the digital terrain model.

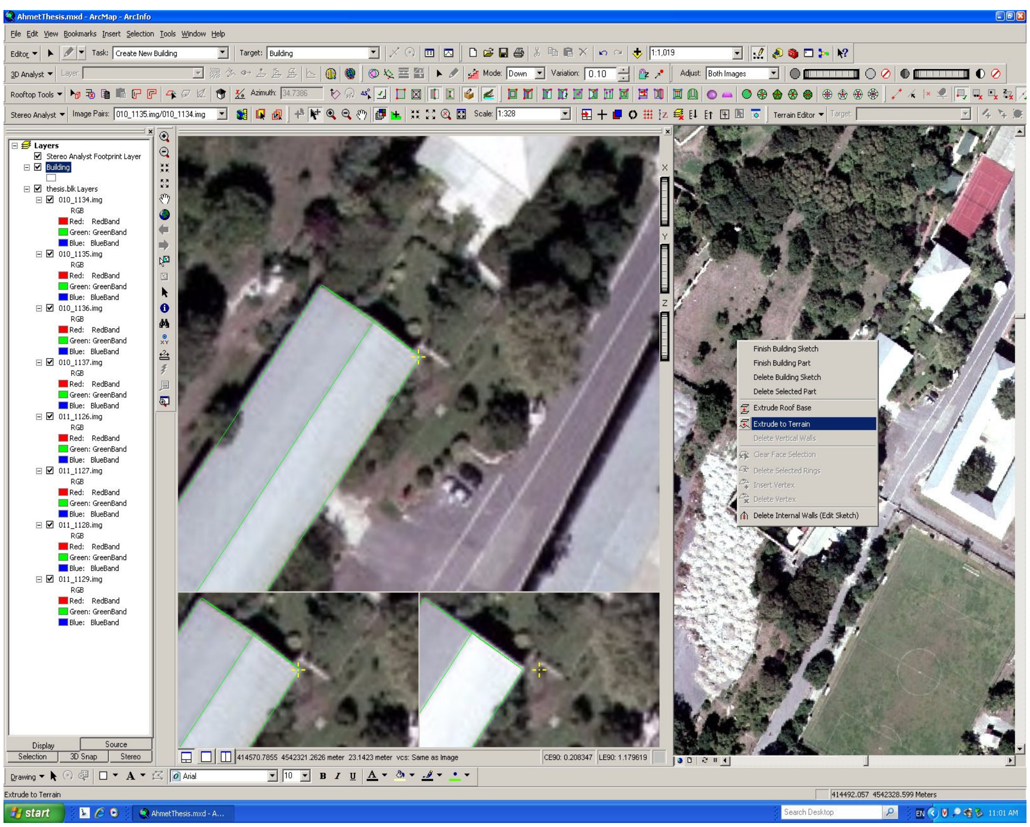


Figure 5.19 : Extrude to Terrain or Extrude to Roof Base

14. *Given the attributes of the building,*

One of the most important part on the worklow is the attributes of the buildings. After finish the building sketch the building is ready for adding attributes. The attributes has been added on the project are ID, TYPE, TYPE_N and ROOF (Fig 5.20). The explanation of the attributes: ID is the unique number of the building, TYPE is added in order to distinguish the building surfaces such as roof, wall and ground, TYPE_N refers the number of each different building surfaces and ROOF refers what kind of roof type the building has such as flat, gabled, hipped...etc.

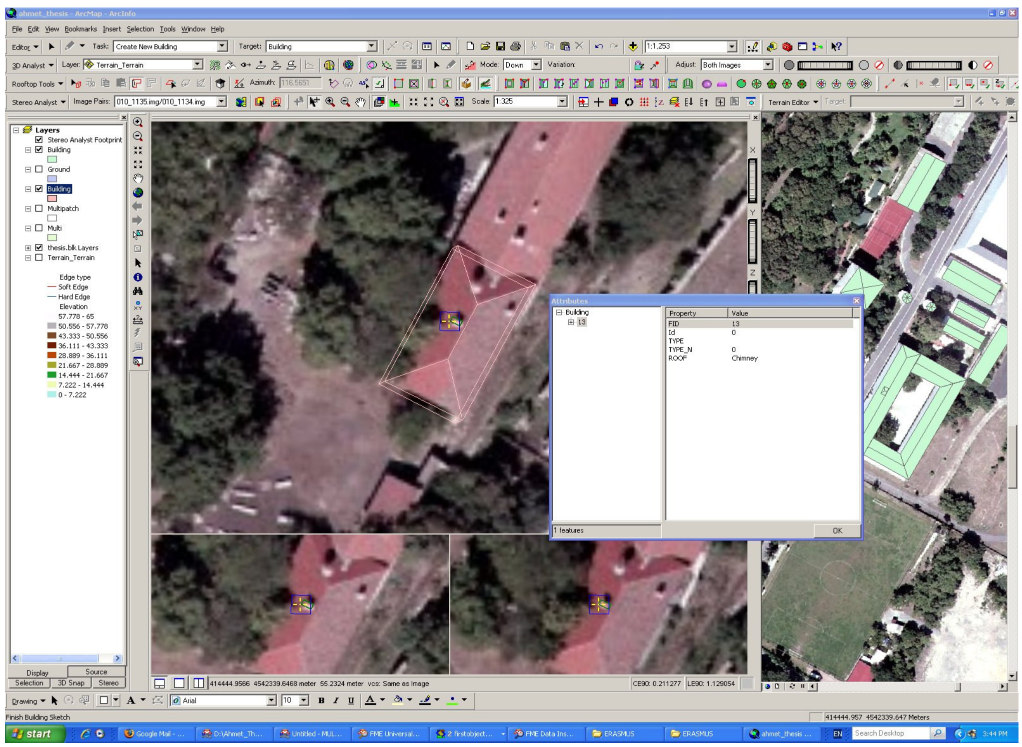


Figure 5.20 : Adding building attributes after finish the building sketch

15. *Footprints of the buildings from Multipatch Shape file*

In order to define the buildings ground surfaces for the project, multipatch footprints conversion toolbox is chosen on Arc Toolbox program. The convertion converts all multipatches into polygons. The output polygons represent the 2D area covered by the multipatches when viewed from directly above. After finishing all sketching process on Stereo Analyst the conversion can be used to get the "ground surfaces" of the buildings. Figure 5.21 shows the process of getting ground surfaces from building multipatch.

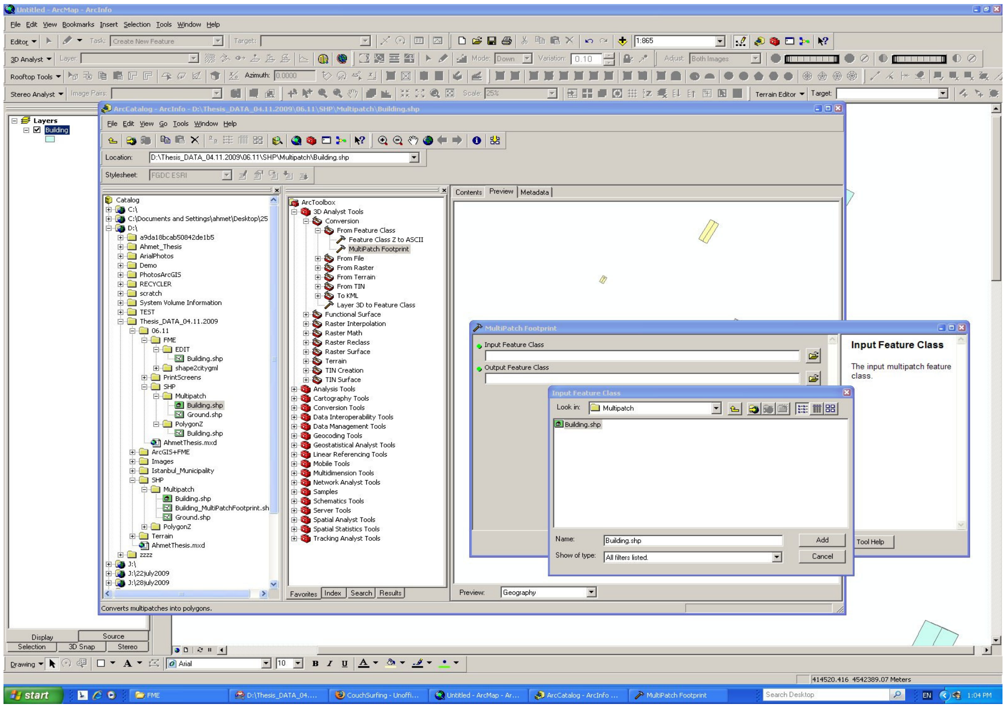


Figure 5.21 : Footprints of the buildings from Multipatch Shape file

16. *Converting from "Multipatch" to "Polygonz" on FME Workbench,*

In order to add semantic information of the buildings, the Multipatch shape file is converted to PolygonZ using FME Workbench program that contain source and destination type (data) and their attributes. During the convertion work on FME from Multipatch shape file to PolygonZ shape file some important setting properties is taken into the account. When destination source is chosen on FME workbench as ESRI Shape under the output settings properties the "Surface and Solid Storage" is defined as "PolygonZ" (Fig. 5.22).

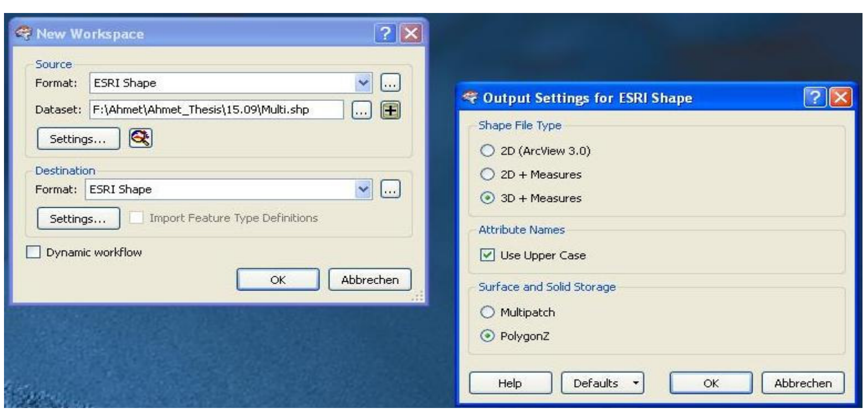


Figure 5.22 : Output setting properties are defined the destination source on FME.

After selected the destination sources and its properties the conversion process started on FME. On this program three different transformers namely bounds extractors, expresion evaluator and deaggregator has been chosen. BoundsExtractor extracts the minimum and maximum values of the feature's coordinates into new attributes. It is needed to calculated the mathematical function. The Expression Evaluator supports the following mathematical functions in expressions. (Fig 5.23)

$$HEIGHT = (@Value(\_zmax)-@Value(\_zmin)) \quad \textbf{(5.2)}$$

Deaggregator decomposes an aggregate feature into its components. Each component of the original feature is output via the port corresponding to its geometry type.

When the workbench ran the process on FME, all the building surfaces are seperately recorded. After FME conversion process the buildings has polygonZ shape file type and ready for giving the semantic information about building surfaces for each surface.

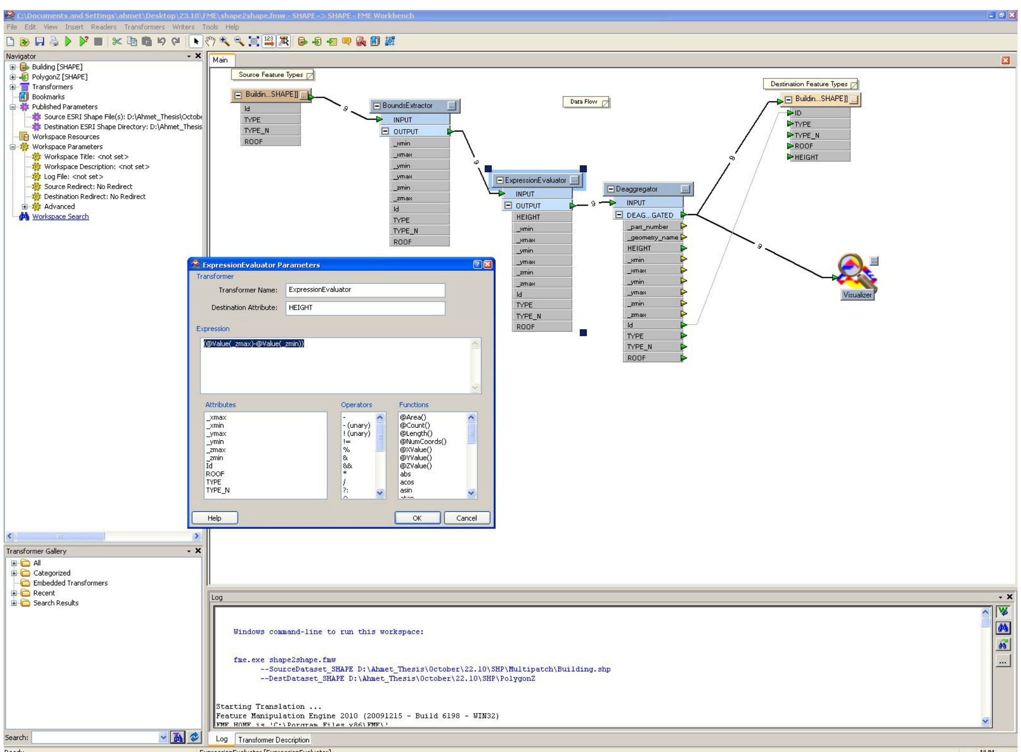
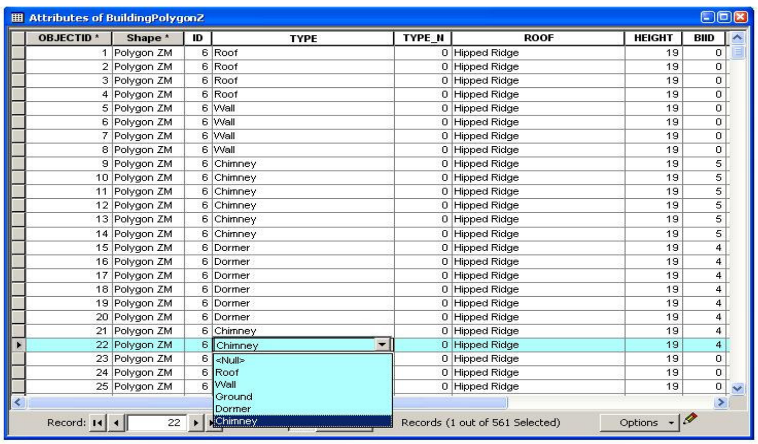


Figure 5.23 : Converting from "Multipatch" to "PolygonZ" on FME Workbench

17. *Giving the attributes to the building's surface*,

Deaggregating the building surfaces on FME using with Deaggregator transformer, all the surfaces are saved with a different row in the table concept. After that the building surface type semanticly can be added on ArcGIS program. The building surfaceis defined for Roof "1", Wall "2", Dormer "4" and Chimney "5".

The ground surfaces which are derived on Arc Toolbox multipatch footprint conversion, are edited and type number is defined as Ground "3".



OBJECTID *	Shape *	ID	TYPE	TYPE_N	ROOF	HEIGHT	BIID
1	Polygon ZM	6	Roof	0	Hipped Ridge	19	0
2	Polygon ZM	6	Roof	0	Hipped Ridge	19	0
3	Polygon ZM	6	Roof	0	Hipped Ridge	19	0
4	Polygon ZM	6	Roof	0	Hipped Ridge	19	0
5	Polygon ZM	6	Wall	0	Hipped Ridge	19	0
6	Polygon ZM	6	Wall	0	Hipped Ridge	19	0
7	Polygon ZM	6	Wall	0	Hipped Ridge	19	0
8	Polygon ZM	6	Wall	0	Hipped Ridge	19	0
9	Polygon ZM	6	Chimney	0	Hipped Ridge	19	5
10	Polygon ZM	6	Chimney	0	Hipped Ridge	19	5
11	Polygon ZM	6	Chimney	0	Hipped Ridge	19	5
12	Polygon ZM	6	Chimney	0	Hipped Ridge	19	5
13	Polygon ZM	6	Chimney	0	Hipped Ridge	19	5
14	Polygon ZM	6	Chimney	0	Hipped Ridge	19	5
15	Polygon ZM	6	Dormer	0	Hipped Ridge	19	4
16	Polygon ZM	6	Dormer	0	Hipped Ridge	19	4
17	Polygon ZM	6	Dormer	0	Hipped Ridge	19	4
18	Polygon ZM	6	Dormer	0	Hipped Ridge	19	4
19	Polygon ZM	6	Dormer	0	Hipped Ridge	19	4
20	Polygon ZM	6	Dormer	0	Hipped Ridge	19	4
21	Polygon ZM	6	Chimney	0	Hipped Ridge	19	4
22	Polygon ZM	6	Chimney	0	Hipped Ridge	19	4
23	Polygon ZM	6		0	Hipped Ridge	19	0
24	Polygon ZM	6		0	Hipped Ridge	19	0
25	Polygon ZM	6		0	Hipped Ridge	19	0

Figure 5.24 : Editing the building surface type using the Domain

In order to protect typing error when the attributes is given to the surfaces the domain which can be created by ArcCatalog Software. After creating the domain, the user can choose as predifened attributes in combo-box. The domain is arranged as below list: (Fig. 5.24)

1 Roof

2 Wall

3 Ground

4 Dormer

5 Chimney

Moreover, The type of building installation has defined on Arc Map while editing the building surfaces. Extra column has added as BIID and is given unique number such as for dormers 401, 402... , for chimney 501, 502... for each surface of them.

Also the building height value which is calculated on FME added as a new column to the attributes table. After the conversion on FME the building PolygonZ shape file can be added on Arc Map. (Fig. 5.25)

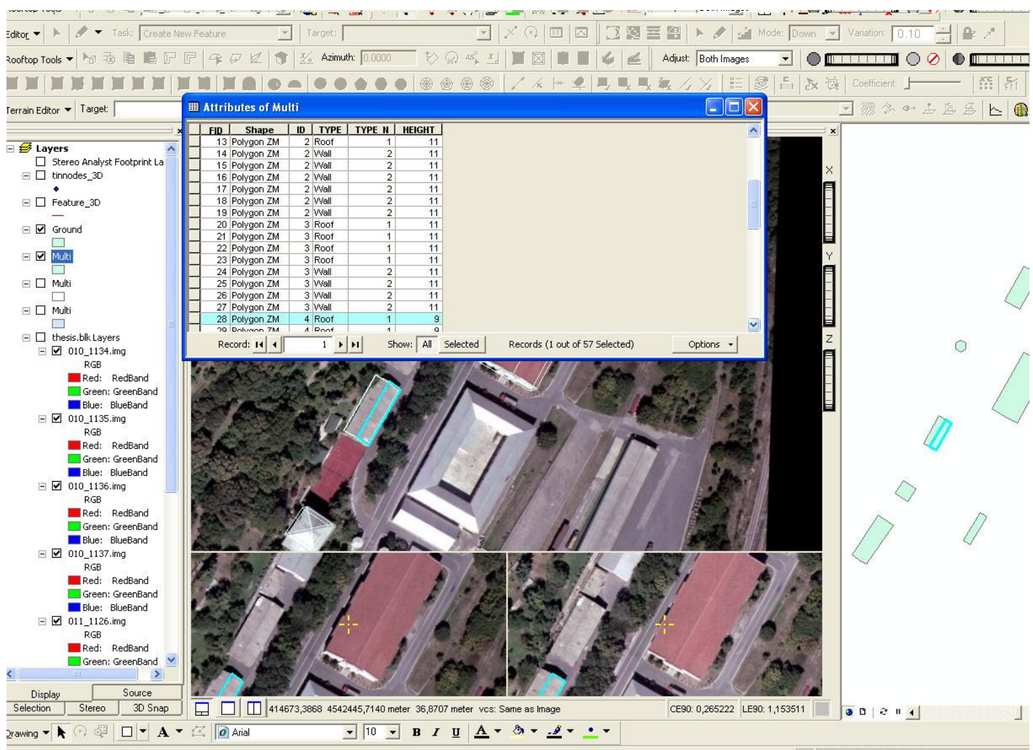


Figure 5.25 : Editing the building surface on Arc Map

18. *Converting the building data from "Shape" to "CityGML" on FME Workbench.*

After giving all informations regarding to the building surfaces, the buildings are ready semantically to convert to City GML. In order to convert from ESRI shape file – PolygonZ City GML data model, FME program was used for the conversion. The FME conversion will be explained in the chapter 6 with more detail. Basically, the conversion process was done with several transformers in the FME program. FME contains over 300 transformers, and it can sometimes be challenging to navigate the list to locate the ones is needed for the conversion. Figure 5.26 shows the FME interface with necessary transformers for the conversion.

Moreover, The type of building installation has defined on Arc Map while editing the building surfaces. Extra column has added as BIID and is given unique number such as 401, 501 for each surface of them.

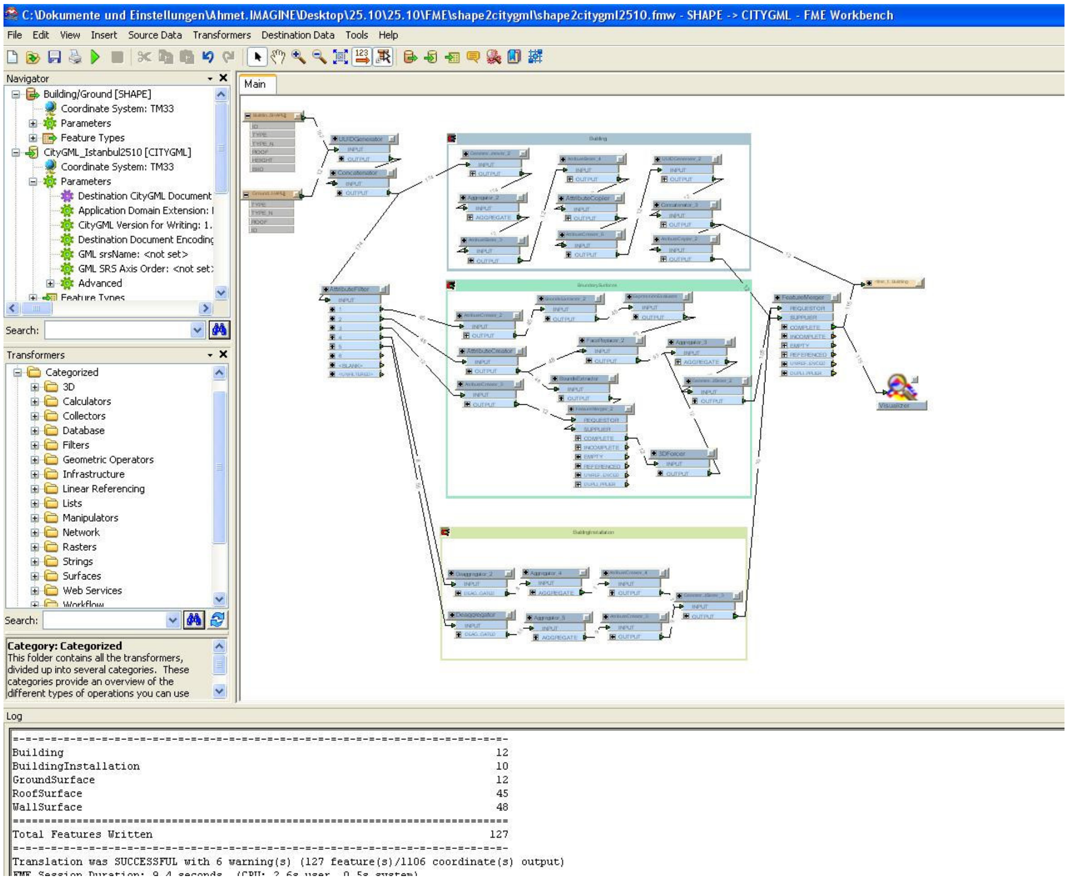


Figure 5.26 : Converting from "PolygonZ" to "CityGML" on FME Workbench

19. *Displaying the CityGML data model in Land Explorer or Aristotales.*

After following all the steps and getting the CityGML file, it is time to visualize it. There are several software which can be opened and visualized the CityGML file. Land Expolorer CityGML Viewer and Aristoteles3D Program is chose in order to visualize the CityGML data. Both program has advantages and disadvantages when to compare them. A building group is chosen in order to show the differences and capabilities of the programs. Working on Land Explorer Program is much more easy to navigate and also to see building attributes and surface information. (Fig. 5.27) On the other hand, Aristoteles3D is much more capable to see geometry tree view for features and semantic information. (Fig. 5.28) It is also possible to see geometry tree view even for the building installations. (Fig. 5.29)

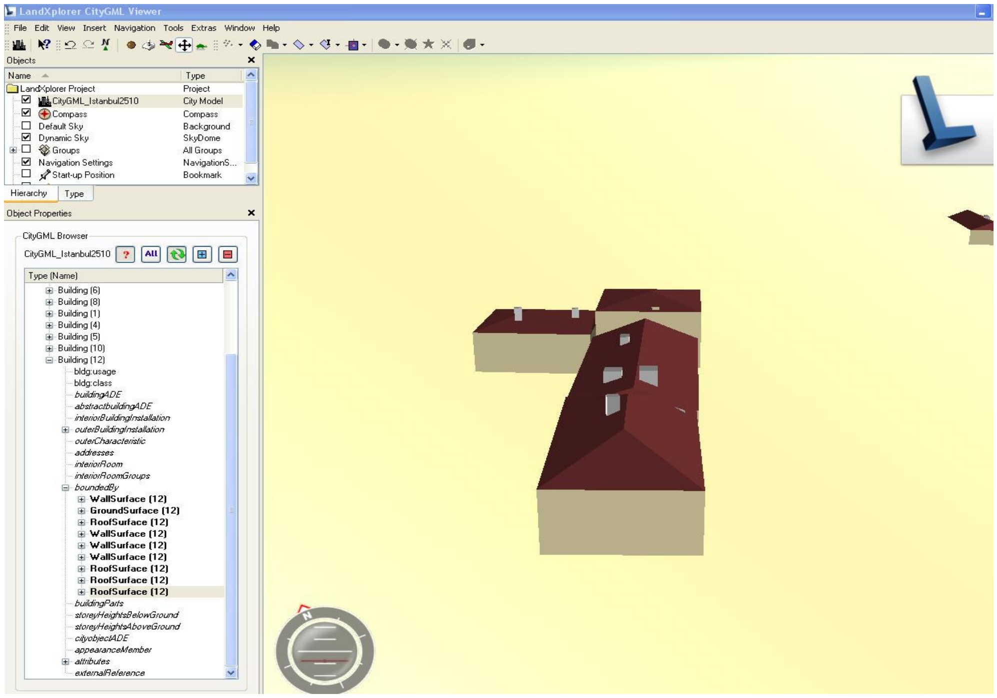


Figure 5.27 : Building model view on LandXplorer CityGML Viewer program

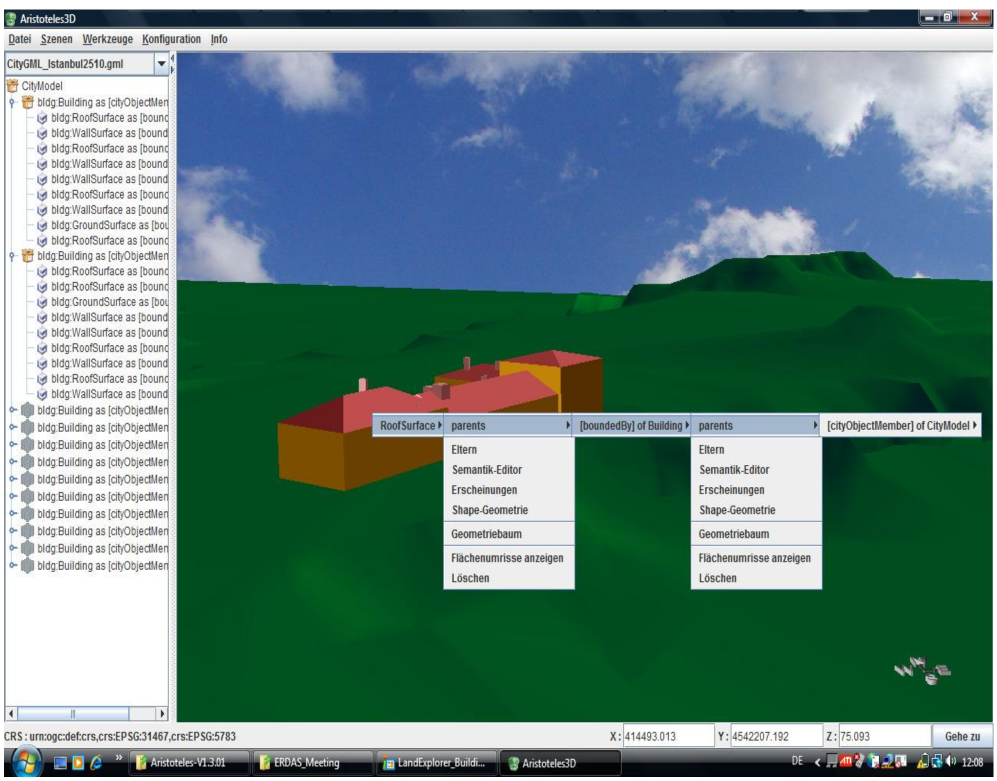


Figure 5.28 : Building model view on Aristoteles3D program with semantic info

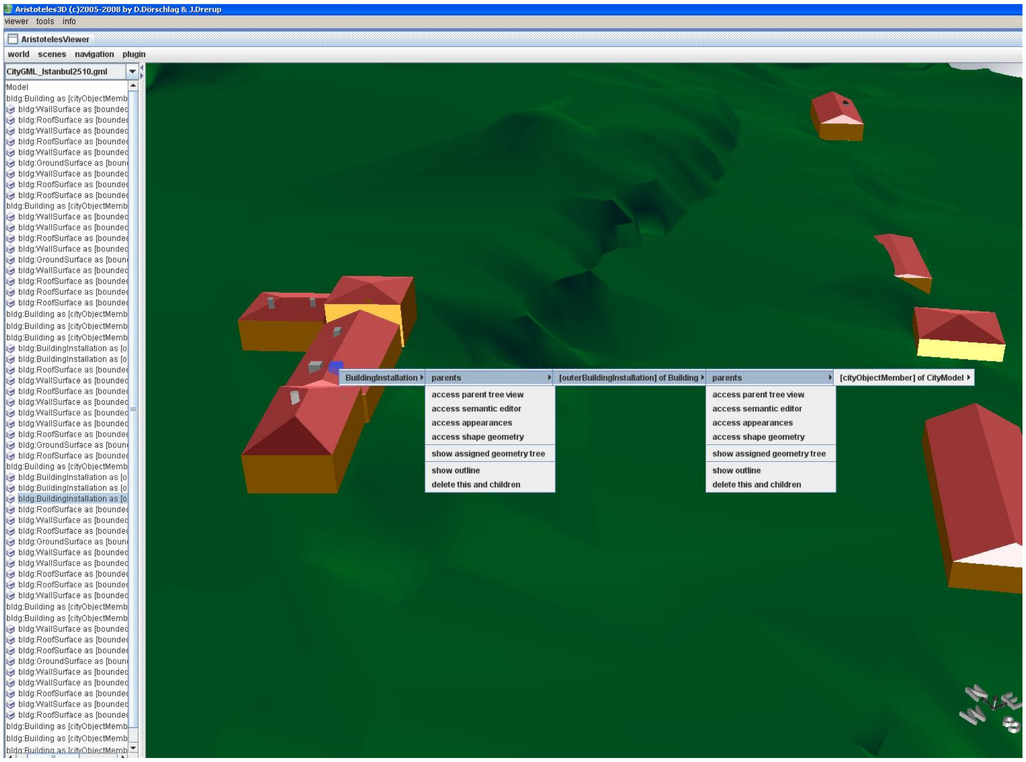


Figure 5.29 : Building Installation view on Aristoteles3D with semantic info

5.3.4 The problems and solutions

Several problems were faced during the thesis project but the most important problems is chosen with small explanation how to overcome the problem with possible solutions and is listed below:

- The main problem in the thesis is that how to digitize the building surfaces separately,

Firstly the problem is tried to solve in the ERDAS program, but it is realized that it is not possible to digitize the roof surfaces including each building surfaces as a seperate feature surface. In order to solve the problem, ArcGIS Stereo Analyst which is the ESRI ArcGIS Extensions developed by ERDAS is chosen for digitizing process. Stereo Analyst for ArcGIS transforms imagery into accurate and reliable real-world (3D) feature data stored in an intelligent multiuser Geodatabase. It is used with combination Feature Assist for ArcGIS whis is another extension developed by ERDAS.

- Another Problem is that having gap between building ground surface and wall surface,

It is solved with FME program using "3D Forcer" transformer. The transformer is explained with detail in the following chapter 6. Basicly it is for if the feature was two-dimensional, it becomes three-dimensional with a constant elevation. The transformer shifts the Ground to meet the lowest point of the corresponding building. The key to this is to shift the ground polygons before replacing with faces or aggregating.

- Digital Terrain Model is not meeting with Buildings Wall Surfaces

DTM not meeting for each buildings. There are some gaps in between the building and digital surface model and sometimes the DTM reachs till the building roof surface. It is suspection that the problem is caused by different Z vertical coordinate system defination between the building model which is created by user and DTM which is taken from Municipality. The solution of the problem could not be found during the thesis project time.

- Building surfaces normal vectors are not directed the outwards direction

After the conversion on FME the CityGML data can be seen on some specific program. When visualize the CityGML building model in the program, some of the building surface are not visible. The reason is that Normal vectors are not always shows to outwards direction. The solution is the problem can be solved using additional JAVA program.

6. CONVERSION WITH FME

In this chapter, the main aim of the thesis which is filling the gap between photogrammetric tool and CityGML data model has been solved by adding several transformers in the FME program.

The conversion is explained in detail in the following sections. Firstly the general information about FME program taking into account in the first section. Furthermore the conversion part is divided into two main part:

1. Conversion from ESRI *Multipatch* shape file format to ESRI *PolygonZ* shape file format with calculation of the building height,
2. Conversion from ESRI PolygonZ Shape file to CityGML data model which is the main conversion task of the thesis work. Also the convertion is divided into three part which is called bookmark in FME,
    a. *Building bookmark* which includes general information about buildings and necessary attributes in order to have correct CityGML dataset,
    b. *Boundry Surface* bookmark which includes the buildings surface informations, necessary attributes as well as extra attribute such as roof height,
    c. *Building Installations* bookmark which includes the building installation necessary attributes in order to open properly in the CityGML data model,

The conversion is done with several different Transformers in the FME program. The transformers are explained in the related conversion. Each Transformer needs necessary input data and should be in the right position in the FME workbench. Finally the conversion is reached the destination data as CityGML which is XML based format.

6.1 General Information about FME

FME (Feature Manipulation Engine) uses to translate geographic data to a different format had limited capabilities. The software overcomes many of the problems associated with traditional translation methods. Users can work with transformers to manipulate the data so user get the output after adding source dataset into the FME workbench. FME includes a gallery of over 300 transformers which may operate on individual features one at a time or on groups of features and it is resulting in output that can be much more useful than the sum of the inputs. [28] FME restructures the data through the use of the transformers. By combining transformers user can resolve a variety of translation and convert from one extension to others with limited amount of them [27].

Primary FME Components

FME comprises a number of spatial data handling components which explained below one by one [27].

- **FME Workbench**

FME Workbench is the primary tool for data translations in FME. It has enables the user to visualize the data flow. It offers powerful transformation and translation capabilities traditionally reserved for custom software solutions. Using Workbench, user can graphically adjust the way your data workflows from its source to the destination [27].

- **FME Universal Viewer**

FME Universal Viewer allows quick viewing of data in any of the FME supported formats. It is used primarily for data validation and quality assurance by allowing you to preview the data before translation, or review it after translation [27].

- **FME Universal Translator**

For simple translations that do not involve any customization, you can get quick results by using the FME Universal Translator. Advanced FME users can also run custom mapping files through this interface. FME Universal Translator was the first FME application to be developed. It required that users define translations using a scripting language rather than a graphic interface. For this reason, the Universal

Translator is largely superseded by FME Workbench, although it still has tremendous value in certain operations such as quick translation [27].

- **FME Data Inspector**

The FME Data Inspector is a Safe Software technology preview, intended to eventually replace the FME Universal Viewer. Although it maintains many of the same features as the FME Universal Viewer. It is used the latest display technology, and it supports 3D viewing in the thesis project [27].

- **What is a Transformer?**

A transformer is an FME Workbench object that carries out the restructuring of features. FME contains over 300 different transformers to carry out different types of restructuring. In the Workbench interface, transformers are stored in the Transformer Gallery and grouped in categories applicable to their associated functionality [29].

There are many ways to place a transformer on the Workbench canvas. However, user can simply double-click the transformer name and it will appear in the workspace. Every transformer has a Properties button on the right of the transformer. (Fig.6.1)

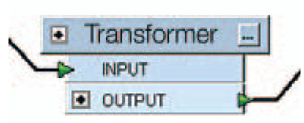


Figure 6.1 : Transformer on FME requests an input data [29]

There is an example of the transformer in FME.(Fig.6.2) The transformers allow the user to group results according to selected attributes. If there is any required fields it can be highlighted to show that it needs to fill in. After fill in the required fields the OK button can be seen enable. The defaults menu allows user to replace FME defaults for this transformer with own parameters [29].

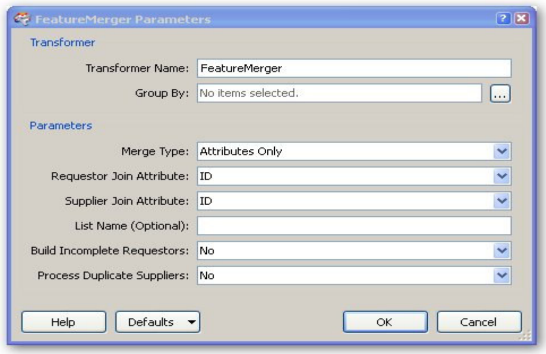


Figure 6.2 : An example transformer from FME Workbench

6.2 Shape-Multipatch to Shape-PolygonZ Conversion on FME

The conversion is needed in the project due to the Multipatch shapeformats capabilities. With the multipatch shape format is not possible to have each surfaces seperately. Every building is represented by one row in the shapefile. The geometry for each row is given as a set of polygonal surfaces with 3D coordinates. It is needed to represent every surface in different row. So that user could add the semantic information of the building surfaces such as wall, roof, building installation...etc. It is realised that the building surfaces are aggregated and should be deaggregated. For this purpose the multipatch shape type is converted to PolygonZ shape file using the FME workbench. The conversion steps is shown in the figure 6.3 with some explainations what exactly do the transformers.

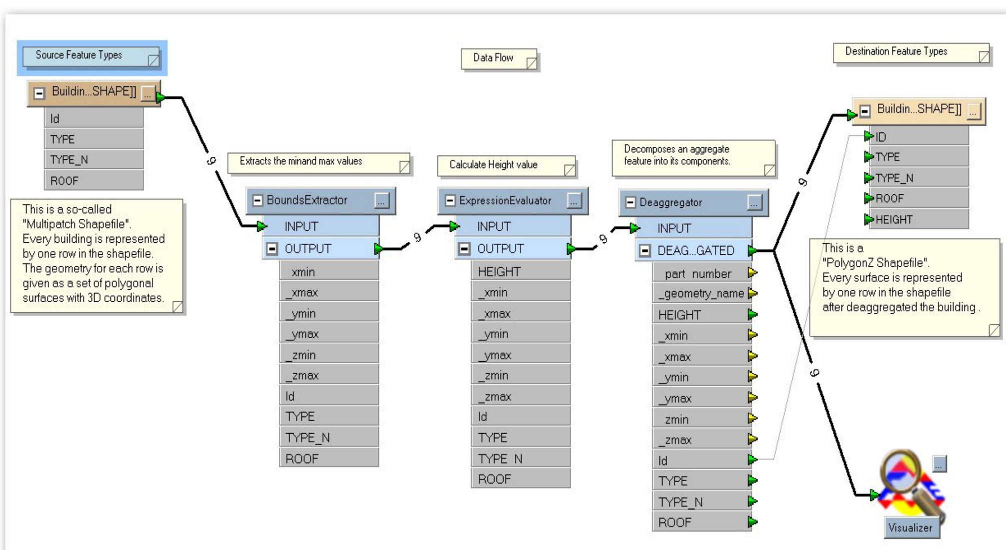


Figure 6.3 : Conversion from ESRI Multipatch to ESRI PolygonZ shape file

In order to convert the shape file from Multipatch format to PolygonZ format three transformers has used. The three transformers are explained shortly below.

- **BoundsExtractor**

This transformer extracts the minimum and maximum values of the feature's coordinates into new attributes. It is for calculation of XYZ coordinates for each surface. As a result parameters minimum and maximum values are calculated by the transformer and are saved in different attribute such as 'zmin' and 'zmax' [30]. It needs to calculate the height value of the buildings.

- **ExpressionEvaluator**

The Expression Evaluator evaluates an arbitrary expression and returns the result in a new attribute. The operators permitted in the expressions to be evaluated are a subset of the operators permitted in C expressions. Expressions usually yield numeric results, such as integer or floating-point values. The expression which has been calculated during the conversion is that :
(@Value(_zmax)-@Value(_zmin))
The result is returned the attribute called HEIGHT by the transformer. The height value represents buildings height including wall and roof surfaces [30].

- **Deaggregator**

This transformer decomposes an aggregate feature into its components. Each component of the original feature is output via the port corresponding to its geometry type. Each output feature has a complete, unaltered copy of the source feature's attributes. If a non-aggregate feature is input, it will be output untouched via the port corresponding to its geometry [30].

Recursive is used to deaggregate nested aggregates recursively. If Yes, the final output will contain no aggregates, only the underlying geometry components. Otherwise, any nested aggregates will not be broken apart. In order to deaggregate the surfaces "Yes" option has been selected [30]. It causes to have shape file which is represented by one row in the shape file for each surface.

The polygonZ shape file is achieved as a destination source. Also height information of the building is added as an extra attribute in the building model.

Shape to CityGML Conversion on FME WORKBENCH

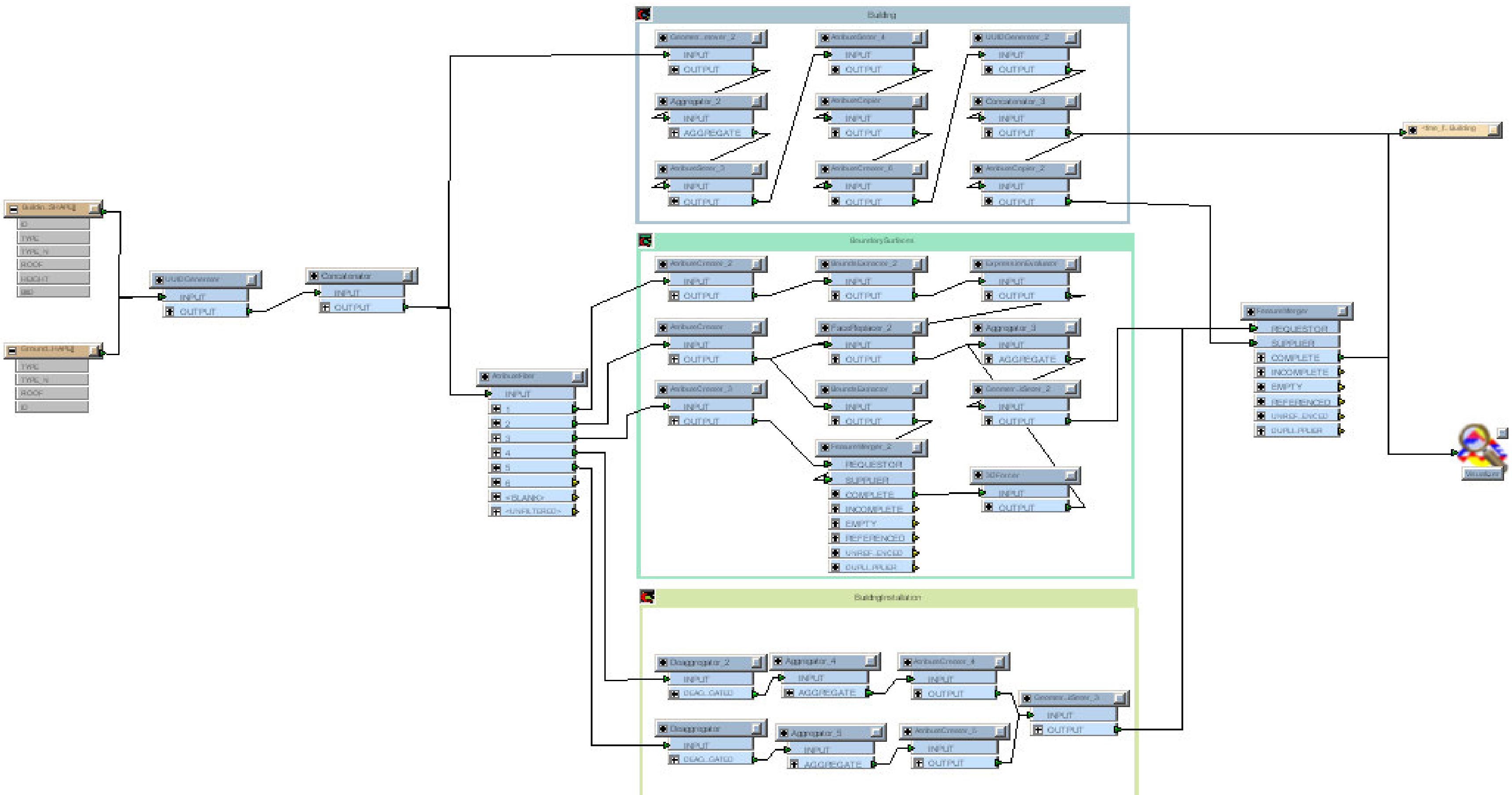
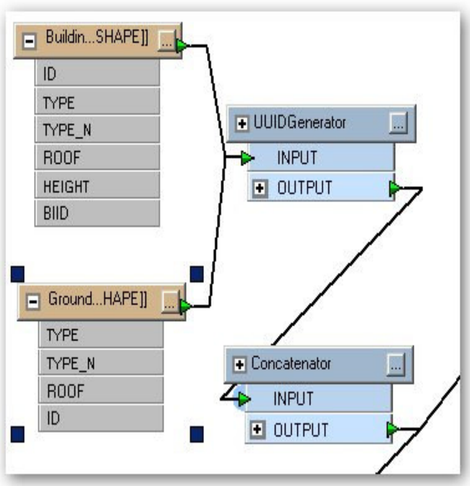


Figure 6.4 : Conversion from ESRI Polygon Z shape file to CityGML data model

6.3 Shape to CityGML Conversion on FME

The conversion between ESRI Shape file and CityGML was achieved using Feature Manipulation Engine. In this section, it is explained with detail information about the conversion.

FME considers a Shape dataset to be a collection of shape files in a single directory. The geometry type and attribute definitions of each shape file must be defined in the mapping file before being read or written. Because of this reason the shape files are defined regarding to their related surface information on Arc Map program after first conversion to polygonZ shape file type. Building surfaces and Ground surfaces which are produced using footprint of the buildings on Arc Catalog has added as source datasets. (Fig. 6.5)



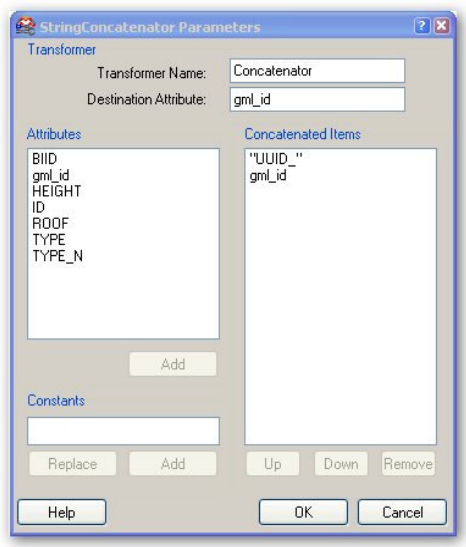


Figure 6.5 : Source of the data

Figure 6.6 : Concetenator transformer

After adding the source data on FME the conversion steps are started using with some transformers such as:

- **UUIDGenerator**

This transformer calculates a UUID (Universally Unique IDentifier) for each incoming feature, and adds it as a new attribute. The UUID is expressed as a string consisting of 8 hexadecimal digits, each followed by a hyphen, then three groups of 4 hexadecimal digits, each followed by a hyphen, then 12 hexadecimal digits. It is bytes in size. As an example of UUIDs from the result data looks like: [30] UUID_f594b3ad-018f-43df-9bfc-64856cd61210

The reason why it is used that to generate unique ID's of the buildings which can help to distinguish them. On the other hand, FME has flat data model structure when compare the CityGML data model. CityGML has complex data model with aggregation hierarchies. In order to bring the data from simple feature model (ERDAS/ArcGIS) into the CityGML deep feature model with nested objects, the gml id is created using the UUID Generator transformer. It is the first step and the next step has done using String Concetenator transformer which is explained below.

- **String Concatenator**

Concatenates the values of any number of attributes and constants, and stores the result in a new attribute. The StringConcatenator complements Workbench's fanout capability by allowing the user to fan out by more than one attribute simultaneously. [30] It is needed to combine with UUID number with gml id value.

Concetenated Items: "UUID_", gml_id (Fig. 6.6)

After Concetenator transformation output data has separated two different way. The first one goes to Geometry Remover transformer in order to remove geometry of the building. The second way goes to into the Attribute filter in order to filter the input attributes depending on building semantic information. (Fig. 6.7)

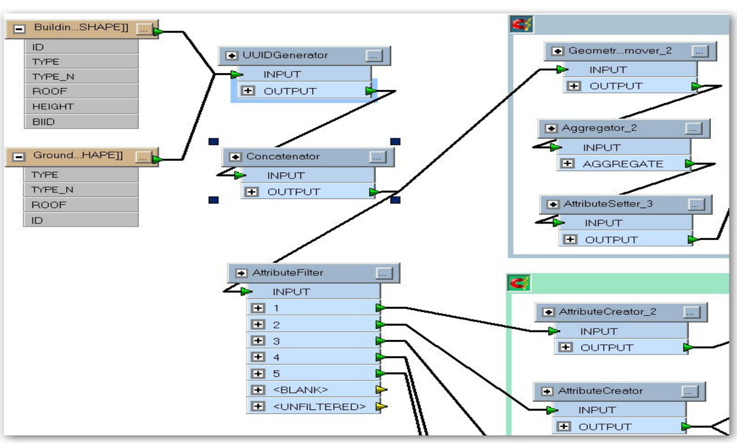


Figure 6.7 : Source data and first transformers on FME Workbench

- **AttributeFilter**

Routes features to different output ports depending on the value of an attribute. The set of possible attribute values can be entered manually, or can be extracted from an input source in the properties dialog. [30] In the project, the inputs have different number showing the related surface and it saved on the shape file as TYPE_N (Number of the Surface type). Attributes filtered by the TYPE_N attribute. There are five numbers from one to five and 1 refers "Roof", 2 refers "Wall", 3 refers "Ground", 4 refers " Chimney" and 5 refers "Dormer".

6.3.1 FME conversion - Building bookmark

In order to collect all transformers which is including general information about the building, a bookmark is used with blue color. Inside the building bookmark has eight different transformers and the number refers each of them.

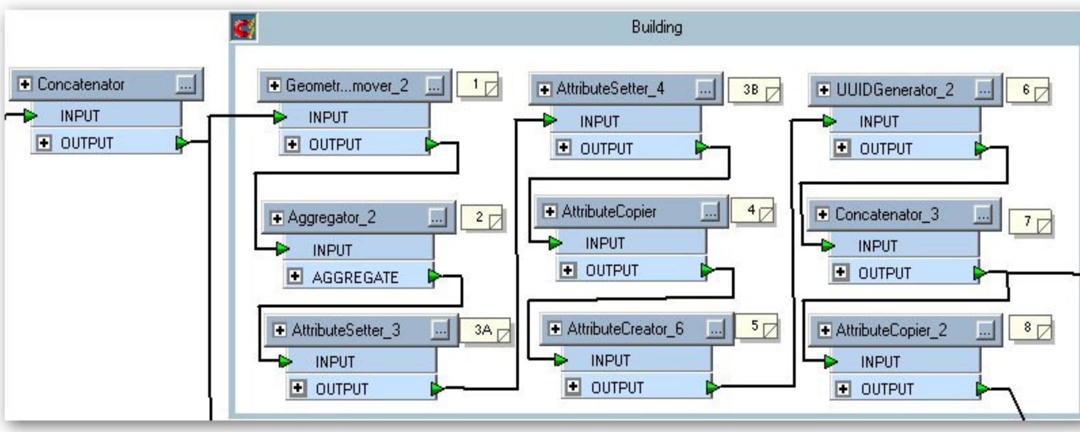


Figure 6.8 : Building Bookmark

The transformer which is used inside the building bookmark is explained step by step with the given numbers in figure 6.8 as below:

1. *GeometryRemover*

Completely removes the geometry of the feature, for example, turning spatial data into non-spatial data [30]. It removes the building`s geometry in order to aggregate the surfaces.

2. *Aggregator*

Combines feature geometries into aggregates. One aggregate feature is output for each unique combination of values of the attributes specified in the Group By parameter which is building ID in the thesis project data [30]. ID is a unique number for each building. The accumulate Attributes option is set to Yes in order to the attributes from the original features will be merged onto the output AGGREGATE features.

3. *AttributeSetter*

Sets an existing attribute to a constant value, or to the value of an another attribute [30]. Two new attributes has added as fme feature type and citygml feature role. Two different attribute setter transformers is following each other as named 3 A and 3 B in the figure 6.8. The new attributes and their values are:

A. Attribute: fme_feature_type

   Value: Building

B. Attribute : citygml_feature_role

   Value: cityObjectMember

The attributes are needed for definition of the buildings in CityGML data model.The structure of the CityGML data model "Building" as fme feature type and "cityObjectMember" as CityGML feature role.

4. *AttributeCopier*

Copies existing attributes to new attributes with the specified names. The existing attribute remains and a new attribute is created that has a different name, but the same value. [30] The reason why the transformer is used that "gml name" is needed for every citygml document as it is predefined.

Old Attribute: ID

New Attribute: gml_name

5. *AttributeCreator*

Adding a number of attributes to the feature and supplying them with constant values. Any feature that enters the transformer emerges with a new set of attributes as defined in the transformer's parameters dialog. The new attributes were created in order to add value to buildings function, usage and class. (Fig 6.9)

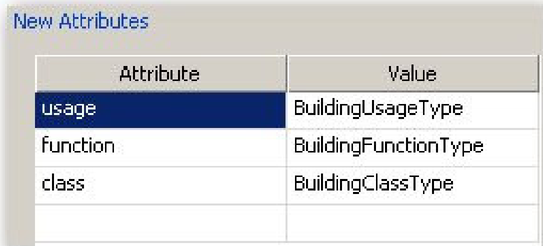


Figure 6.9 : New attributes are added in the Attribute creator transformer

It is important to arrange the attributes order such as class, function then usage. Otherwise the validation of the CityGML data will be not valid. The value of the new attributes has not given because of the not have any information about buildings which type are.

6. *UUIDGenerator*

The general explanation of the transformer has already given in the previous chapter. Additionally, it generates new UUID number for each building. The generated UUID number put in a new attribute namely gml id as a new parameter.

New UUID Attribute: gml_id

7. *Concatenator*

The transformer is explained also in the previous chapter and it is combining the UUID unique number and "UUID_". The concetenated items are including 'gml_id' from attributes of the building and 'UUID_' is added on constants section of the transformer.

Concetenated Items: "UUID_", gml_id

8. *AttributeCopier*

The transformer copies existing attributes to new attributes with the specified names. The existing attribute remains and a new attribute is created that has a different name, but the same value [30]. The changing atributes are:

Old Attribute: gml_id

New Attribute: gml_parent_id

After the last transformer is added, the building is ready with necessary information for citygml data model. The output of the building bookmark is directed into the FeatureMerger transformer as supplier input data.

6.3.2 FME conversion - BoundarySurface bookmark

In the Boundary Surface Bookmark is used to collect the trasformers which is given the building`s surface information. The starting input is derived from Attribute filter which is for filtering the attributes with specific attributes. The building surfaces are filtered by roof type and each of them put into the Attribute creator transformer.

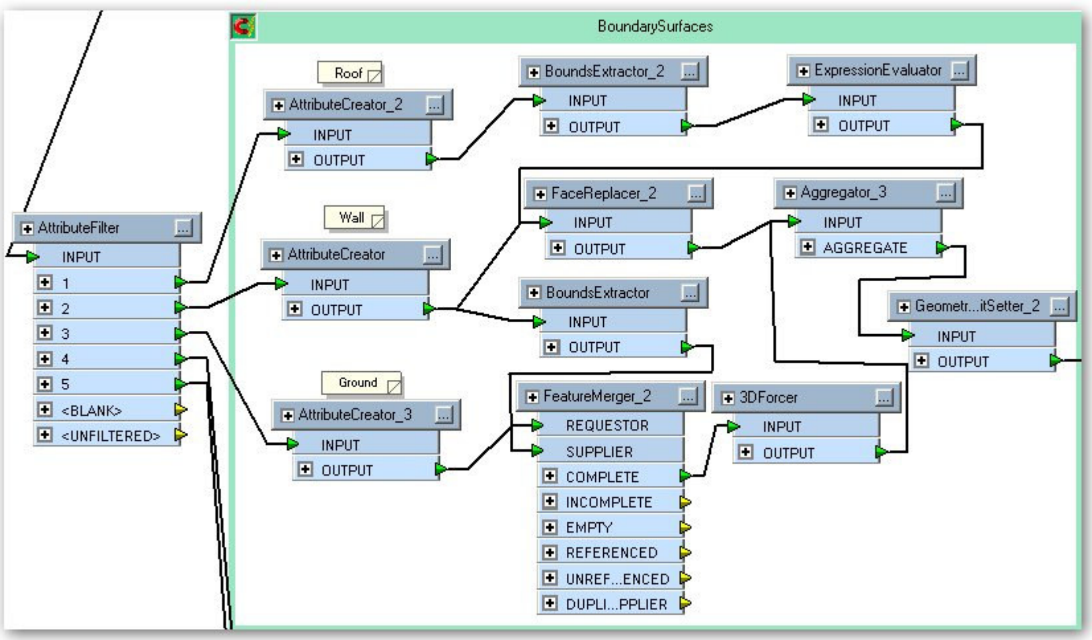


Figure 6.10 : Boundary Surfaces bookmark

- **AttributeCreator**

The general explanation of the transformer was already given in the previous chapter. Three different attribute creators were used in the boundry surface bookmark. The explanations about them can be found under the seperate heading as below such as Roof, Wall and Ground. In the following figure, the AttributeCreator is used to add attributes and values to a dataset for each surfaces. (Fig 6.10)

The attributes given are the most necessary part for the citygml data model requirements such as citygml level of detail (LOD), citygml feature role, fme feature type, xml type, citygml LOD name, gml geometry and surface dimension. Generally the values are similar for each surfaces and the only fme feature type is given differently depending on which surface belongs to as could be seen in the Attribute creator proporties.(Fig 6.11)

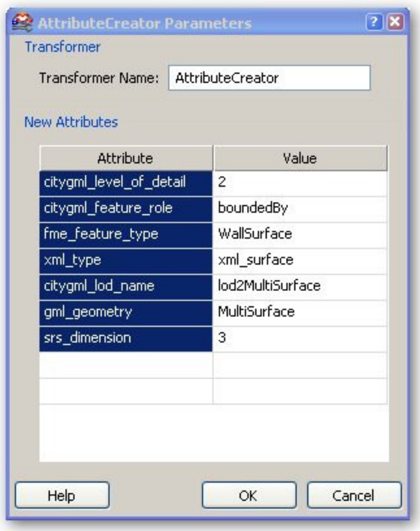


Figure 6.11 : Attribute Creator transformer

A. Roof Surface

After the necessary surface informations is given in the Attribute Creator transformer, the Bounds Extractor and Expression Eveluated are used in order as they are used in the Multipage shape to PolygonZ shape conversion. The reason is that using the both transformers is that to calculate the roof height information from the minimum and maximum Z coordinates. The more explanation how to get the height information could be found under the conversion chapter. It is exactly the same calculation has been done during the Multipage shape to PolygonZ shape conversion.

The output of the Expression Eveluated transformer and Attribute Creator which is derived from wall surface have together reached the Feature Replacer transformer.

- **FaceReplacer**

The transformer replaces the geometry of a feature from donut or polygon to face. A face is a planar area in 3D space. The planar structure can be a raster, polygon or a donut. The planar area has a concept of a surface normal, a vector that points outwards perpendicular from the area. The direction of the surface normal in a face is determined by using the right-hand rule: if the fingers of your right hand curl along the order of the vertices, the direction that the thumb points to is the direction of the surface normal [30].In the thesis project, the wall surface's normal vector is not always pointing outwards. It causes a problem to visualize the buildings.

B. Wall Surface

Wall surface information is given on the Attribute Creator transformer. The output of the wall surface has divided two different arrows. One of them goes into the Face Replacer transformer as input in order to connect the roof surfaces and the other arrow goes into the Bounds Extractor in order to extracts the minimum and maximum values of the feature's coordinates into new attribute which is needed for shifts the Ground to meet the lowest point of the corresponding building. The key to shift the ground polygons before replacing with faces or aggregating. The lowest point of the wall surfaces which is defined in the bounds extractor transformer is directed in the Feature Merger as supplier port input data.

C. Ground Surface

How to arrange the ground surfaces have been explained in the methodology chapter. Basically they are the footprint of the building shape file. After being derived from the Attribute Creator the related informations has been given as it is done for roof and wall surfaces. The output of the transformer is pointed the Feature Merger as reguestor port input data. In order to merge the ground surface and wall surface 3D Forcer transformer has been used with Feature Merger. More explanation about the both transformer is explained in below with details.

- **FeatureMerger**

The transformer moves the attributes or geometry from one feature to another feature. Features that contain the desired attributes/geometry are connected through the SUPPLIER port, and the features that will receive the attributes/geometry are connected through the REQUESTOR port [30]. Requestor features are joined to Supplier features when their respective Join Attributes have the same value. The join attributes have chosen as ID for both features. Also merge type has defined as "Attributes only" in the Feature Merger Parameters.

- **3DForcer**

The transformer turns two-dimensional data into three-dimensional data by adding a z-value to every coordinate with a constant elevation. If the feature was already three-dimensional, its previous elevations are wiped out and replaced with the value that held in the specified attribute [30]. The elevation parameter is defined as minimum Z coordinate which is derived from wall surface using bounds extractor transformer. The output of the 3DForcer and Face Replacer are pointed the Aggregator as input data. The problem, having gap between wall surface and ground surface, was solved using this transformer.

- **Aggregator**

It is already explained in the previous chapter. Basically it combines the attributes and has options to group them by specific attribute. The roof, wall and ground surfaces are aggregated grouped by gml id and accumulate attributes are chosen as yes in order to the features are merged onto the output AGGREGATE features. The all attributes within the aggregate goes to Geometry Trait Setter transformer as input data.

- **GeometryTraitSetter**

It copies attributes from a feature into geometry traits. Geometry traits are similar to attributes on a feature; they are defined as any kind of user-defined data that is stored onto a geometry. Each of the separate geometry can hold its own specific data. This capability is useful for situations where geometries are combined or split apart within features [30]. The transformer source attributes option three different attributes citygml level of detail (lod), citygml lod name and gml geometry has chosen. Hereby the new traits will have the same names as the selected source attributes.

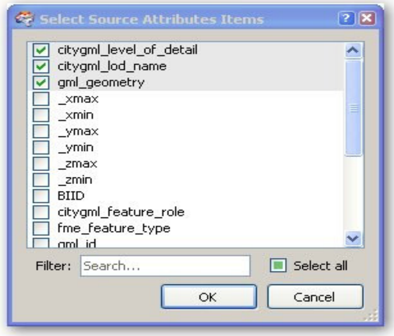


Figure 6.12 : Selection of the source attributes in GeometryTraitSetter Transformer

6.3.3 FME conversion - BuildingInstallation bookmark

In order to collect the Building Installations in the same group, a bookmark is created namely Building Installation. The starting input is derived from Attributes Filter as it is for Boundry Surfaces. The Building Installations are defined while editing the surfaces on the ArcMap as it is explained in the workflow chapter. The type of building installation has defined on Arc Map while editing the building surfaces. Extra column has added as BIID and is given unique number such as 401, 501 for each surface of them. In additionally, type number of the surfaces have been defined on Arc Map such as number 4 refers to Dormer and number 5 refers to Chimney. The information about building installations is given seperately after the Attribute Filter transformation using four different transformers. The transformers are used as below:

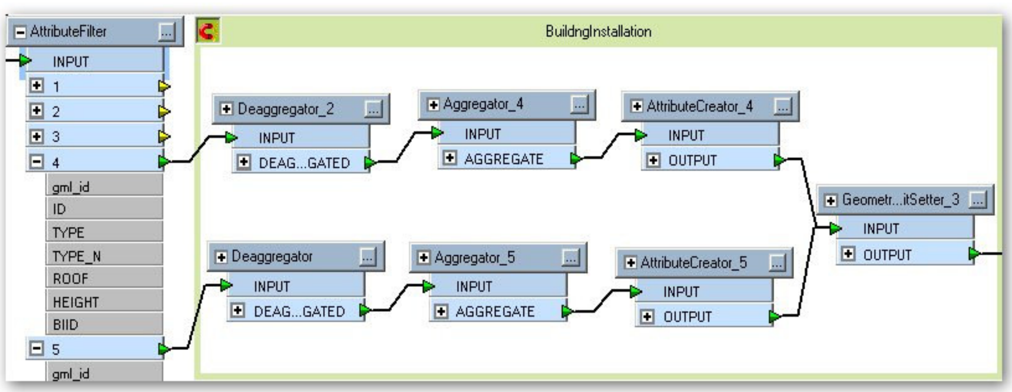


Figure 6.13 : Building Installations bookmark

- **Deaggregator**

The transformer decomposes an aggregate feature into its components. Each component of the original feature is output via the port corresponding to its geometry type. Each output feature has a complete, unaltered copy of the source feature's attributes. [30] The reason is to use the deaggregator that building installations have multi surfaces including six different surface member and somehow the surface can be transformed into the single surface. In order to transform the surfaces the Deaggregator has been used.

- **Aggregator**

It combines feature geometries into aggregates [30].

- **AttributeCreator**

Adds a number of attributes to the feature, supplying them with constant values [30]. The given attributes are almost the same for the building surfaces such as Wall, Roof surfaces. The differences are that CityGML LOD name is "lod2Geometry" instead of "lod2Multisurface" and Building installations function and class type are added and the values are given as defined in the CityGML encoding standart such as for function type 1030 refers "chimney (part of a building)" and class type 1000 refers "outer and inner characteristics". (Fig 6.14)

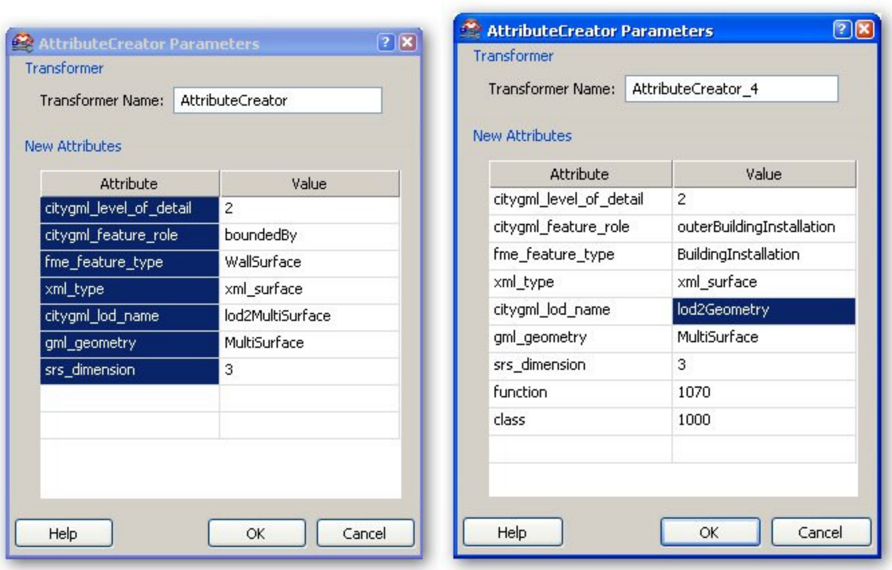


Figure 6.14 : Attribute Creator Parameters for Building Installations

- **GeometryTraitSetter**

The same steps which were defined for building surfaces can be followed. The output of the transformer goes to in the Feature Merger transformer as requestor which has done for the other building surfaces.

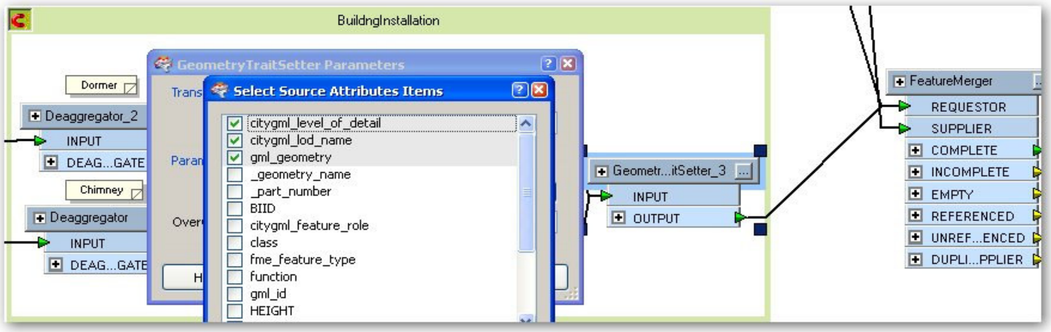


Figure 6.15 : Selection of the source attributes for the Building Installations

- **FeatureMerger**

Features that contain the desired attributes/geometry are derived from Boundary surfaces and Building Installation bookmarks through the REQUESTOR port, and the features that will receive the attributes/geometry are derived from Building bookmark through the SUPPLIER port. When a Requestor finds a Supplier, then the attributes from the Supplier are merged onto the Requestor. Requestor features are joined to Supplier features when their respective Join Attributes have the same value as ID [30]. The defination of the parameters can be seen in the following figure 6.16.

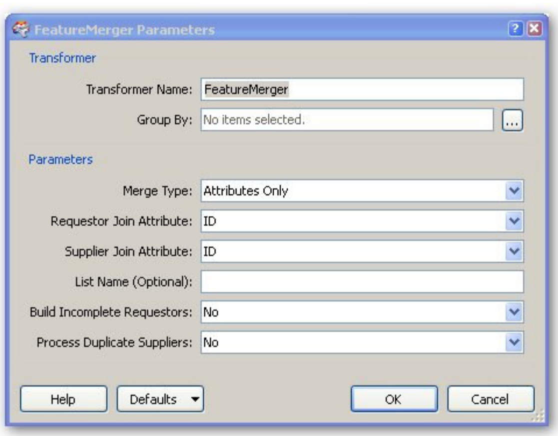


Figure 6.16 : Feature Merger Transformer parameters

The output of the Feature Merger transformer from Complete port pass to the destination data Feature type with the name Building.

6.3.4 CityGML data as a Result

In the last step of the conversion is deriving the destination data as CityGML is an XML-based format for the storage and exchange of 3D urban models. The output of the Concetenator together with Feature Merger complete port pass to CityGML dataset. The CityGML data includes all the necessary user and format attribues. In the Feature type properties the Fanout by attribute option has activated with fme feature type attribute. (Fig 6.17)

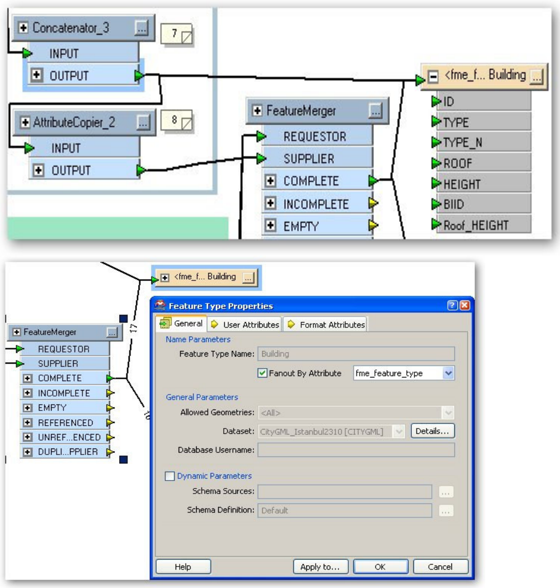


Figure 6.17 : Destination Data

The CityGML is based on XML format type and it represents of the city object attributes. XML schema is a rich language with many capabilities.

The city model is bounded by city object member and it includes Buildings and possibly Building parts. Each building has gml unique id, gml name and several attributes as well as building class, usage and function type. Nevertheless, each building is bound by related building surfaces such as wall, roof and ground. Generally every building has four different wall surfaces, at least four roof surfaces except not in flat type, and only one ground surface. Also every surface has own unique gml id and gml name. Each building surfaces have lod2 Multi Surfaces including exterior linear ring and coordinate list of each corner of the building surfaces. (Fig 6.18)

Figure 6.18 : Viewing the CityGML data in the XML editor

7. CONCLUSION AND RECOMMENDATIONS

The result of the thesis work is a valid citygml building model according to proposal hypothesis. In this thesis work the aerial images were only the data which have been working on the ERDAS LPS Program and the 3D city model have been created as a result. The following conclution can be mention based on the thesis practical and technical aspects:

I. Creating the 3D City Model: As it mentioned in the master thesis hypothesis, the 3D city model based on 3D buildings has been created with reference to CityGML encoding standard.

II. Conversion from phoptogrammetric tools to CityGML: The conversion has been done using with FME Workbench from ESRI Shape file to CityGML data file. It takes only minutes to convert the shape file within FME Workbench.

During the thesis work several steps have been done in different softwares. Mainly the gap between ERDAS Leica Photogrammetric Suite and CityGML is filled and conversion is completed between two different data model using FME software. After completing the workflow it is realized that there are still some future improvements in the thesis work. The possible improvements about the master thesis are listed below:

- Adding JAVA Program to check the orientation of the building's surfaces,

In order that all building's surfaces appear correctly within any CityGML viewer, the normal vector of the surface must look outwards direction. Some of the building's surfaces need to be reversed after the conversion on FME.

A " right hand rule " method to determine the *normal vector* of *polygons* in computer graphics applications. The rule is derived from the mathematical rule to determine the forward direction of screws. When the observer looks at the *polygon* from one side, and the points of the *polygon* are defined in counter clockwise sequence, then the normal vector points towards the observer. This can be symbolised by a "thumbs up" closed right hand fist. The four remaining fingers point into a counter clockwise direction (the sequence of the points), while the thumb shows the direction of the normal vector.

A Java program can be helpful to reverse the building surfaces. In order to program it citygml4j which is a Java class library for CityGML can be used[1].

The possible algorithm of the Java program:

- Checking each of the three different building surfaces seperately;

I. Ground Surface : If the normal vector has a negative Z value, then it points downwards which mean that the surface is oriented correctly. Otherwise the ordering of the points has to be reversed.

II. Roof Surface : If the normal vector has positive Z value, the surface points outward direction and it is correct

III. Wall Surface: Even or odd method can be use in order to define the direction of the wall surface as outward. Within the mothod the program calculates the normal vector of the wall surfaces. Moreover it checks the number of the intersection with any other wall surface of the building.

   - if the number of intersections are even (0,2,..n) means the orientation of the surface is correct and the normal vector points outwards..
   - if the number of intersections are odd (1,3,..n-1) means the orientation of the surface is correct and the normal vector points outwards

- Adding Texture to the Buildings Surfaces

The possibilty of adding texture of the building surfaces is another task to follow as future direction of the thesis practical work.

In order to add texture onto the building surface, Aristoteles Viewer[2] which is is developed for different applications which to be visualized, evaluate new features and attributes for 3D models and the version 1.0 built 20 of Aristoteles Viewer has option to add any no metric picture as texture on the building model. In this program user can follow the defined steps as below:[32].

1 http://opportunity.bv.tu-berlin.de/software/wiki/citygml4j/Citygml4j_-_A_Java_class_library_for_CityGML?version=84

2 http://www.ikg.uni-bonn.de/aristoteles/index.php/Download

- Using plugin in the main menu user can select picking custromization
  - Addition of the feature by TextureCreation
  - Once the surface can be respected to add the texture, but in this context the geometry of that surface should be taken into the account
  - The texture could be created based on an image selected with respect that the texture creation is associated with a rectification process of the image chosen within this process at least four points should be determined homogeneous on the image.
  - After selecting the points the image can be matched to the building's surfaces.

- Adding TIN in the CityGML Model,

The program has written by Prof. Dr. Thomas Kolbe and working on ESRI ArcScene Program. This small utility program allows to export an activated TIN layer from within ArcScene as a CityGML file. The TIN is exported as a CityGML TINRelief feature being the only member of a CityGML ReliefFeature feature.

It used to convert TIN data from ESRI shape format to CityGML data format. The program is used in order to present the building model more effective and realistic. It could be seen in the Appendix A.2.

- Managing to add the TIC Terrain Intersection Curve

During the thesis work, the main problem was that not every building was in contact with the Digital Terrain Model. There are some height different between buildings and DTM. If the problem has been solved the next step would have been the creation of a TIC in CityGML data model.

- External References and Additional database

The external reference denotes an external information system and the unique identifier such as *building id* of the buildings in the other databases.

While during conversion process in FME Workbench program the Joiner transformer can be used in order to reach the extra information and external database. Access MDB files, ODBC connections, Oracle 10g/8i/9i, Oracle 7 and 8 attribute tables, ESRI ArcSDE, dBase III files, comma separated value (CSV) files, PostgreSQL/PostGIS tables, Microsoft Excel (XLS) files, Microsoft SQL Server, MySQL, SQLite, and DB2 tables can all be used as the database.

Joiner Transformer

A Joiner joins attributes from an external database (most popular databases are supported) to other spatial (or non-spatial) features as they are being processed through a translation. Queries a database to retrieve attributes associated with a feature. One or more feature attributes (foreign keys) are joined to one or more columns (primary keys) in a table in the database, and the values from the matching table rows are added as feature attributes [30].

During the conversion on FME, unfortunately the trasformation could not be added due to could not reach any available and suitable external database in the test area.

- Conversion from City GML to 3D PDF , KML… etc

Conversion on FME Workbench have possiblilty to convert from CityGML dahat format to decades different format which 3D city models can be represented. As the standard laid down by the Open Geospatial Consortiums (OGC) by which 3D city models are exchanged, CityGML plays a central role in these activities. Moreover, FME supports OGC WMS, WFS, GML, VRML and KML along with CityGML. Some of the possible formats have been converted from CityGML file format and listed below:

1. 3D PDF

The cityGML building model can be converted to 3D PDF format using FME Workbench.

2. OpenGIS KML Encoding Standart

The cityGML building model can be converted to KML format using FME Workbench.

There are several different researches and projects about creating the 3D City Model in the field of 3D GIS. As an example 3D city model, Munich City Model which is one of the biggest related project can be given. When the future direction has been completed the possible view of the city model can be seen in figure 7.1.

Figure 7.1: Munich City Model as an example city model

Consequently, the gap between photogrammetric methods and CityGML has been filled with FME program. As a result data CityGML building model has been created with necessary semantic building information such as roof, wall, ground and building installations. Furthermore, the building model of photogrammetric methods has been created and also the building model of CityGML is minimized from the specification document depending on the requirements of the thesis work.Also those data models are compared with CityGML building model. The differences and similarities of those models have also been defined and also they mapped each other usıng a UML diagram.

REFERENCES

[1] **Kolbe T. H., Gröger, G.,** 2003: Towards Unified 3D city models, *Proceedings of the ISPRS Commission IV Joint Workshop on Challenges in Geospatial Analysis*, Integration and Visualization II, Stuttgart, Germany.

[2] **Zlatanova, S., Paintsil, J. and Tempfli, K.,** 1998: 3D object reconstruction from aerial stereo images. In: WSCG '98, *6th international conference in Central Europe on computer graphics and visualization:* February 9-13, 1998, Plzen-Bory, Czech Republic.

[3] **Trung T. Pham, Marjorie Musy, Daniel Siret, Jacques Teller**, 2007 Methodology for Integrating and Analyzing Environmental and Urban Data in 3D GIS In: *The 10th AGILE Conference on GIScience* , Aalborg, Denmark, May 2007

[4] **Fischer Andre, Kolbe Thomas H., Lang Felicitas, Cremers Armin B.,Förstner Wolfgang, Plümer Lutz, Steinhage Volker:** ,1998: Extracting Buildings from Aerial Images using Hierarchical Aggregation in 2D and 3D In: *Computer Vision & Image Understanding*, 72. Jg., Heft 2.

[5] **Url-1**<http://www.veps3d.org/site/249.asp >, accessed at 30.01.2010.

[6] **Döllner, J., Kolbe, T. H., Liecke, F., Sgouros, T. And Teichmann, K.,** 2006. The Virtual 3D City Model of Berlin - Managing, Integrating and Communicating Complex Urban Information, *25th International Symposium on Urban Data Management UDMS 2006* in Aalborg, Denmark.

[7] **Url-2** <http://www.veps3d.org/site/250.asp >, accessed at 30.01.2010.

[8] **TU0801**, 2008. Semantic Enrichment of 3D City Models for Sustainable Urban Development. Memorandum of Understanding, from http://www.igg.tu-berlin.de/uploads/tx_ikgproject/TU0801-e.pdf (accessed at Feb. 2010).

[9] **Gröger, G., Kolbe, T.H., Czerwinski, A., Nagel, C.,** 2008. OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 1.0.0, OGC Doc. No. 08-007r1, Open Geospatial Consortium

[10] **Kolbe, T.H**, 2009, Representing and Exchanging 3D City Models with CityGML, *3D Geo-Information Sciences* 10.1007/978-3-540-87395-2_2, ISBN: 978-3-540-87395-2, pp. 15-31.

[11] **Sengul A.,** 2008, 3D GIS Example in Historical peninsula of Istanbul, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XXXVII. Part B2. Beijing 2008, p 659-664.

[12] **Brenner, C**., 1999 Interactive Modelling Tools for 3D Building Reconstruction, In Proceedings of Photogrammetric Week '99'

[13]**Tao, C.V.**, 2006: 3D Data Acquisition and Object Reconstruction for AEC/CAD. In: Zlatanova, S. and Prosperi, D. (eds.): Large-scale 3D data integration – Challenges and Opportunities. Taylor&Francis, London, pp. 39-56.

[14] **Kobayashi, Y**. 2006. Photogrammetry and 3D City Modeling, In: ALI A, BREBBIA C.A., *Digital Architecture and Construction*, WIT Press: Cambridge

[15] **Zlatanova, S.; Paintsil, J. & Tempfli, K**., 1998: 3D Object Reconstruction from Aerial Stereo Images. From *http://www.gdmc.nl/zlatanova/thesis /html/refer/ps/ sz_jp _kt98.pdf* (accessed at Feb. 2010).

[16] **T. Schenk 2005**, Introduction to Photogrammetry, From *http://www.ins.itu.edu .tr/jeodezi/fotog/hdemirel/AnalyticalPhotog/schenk.pdf* (accessed at Sep. 2009).

[17] **John Vincent Carlis, Joseph D. Maguire** ,2001. Mastering Data Modeling: A User-driven Approach.

[18] **G. Booch, J. Rumbaugh and I. Jacobson**, 1999. The UML User Guide, Addison-Wesley, Reading, MA.

[19] **Url-3**<*http://www.ibm.com/developerworks/rational/library/content/Rational Edge/ sep04 /bell/* >, accessed at 30.01.2010.

[20] **The Multipatch Geometry Type An ESRI White Paper**, December 2008

[21] **Heipke C,** 1995, State-of-the-Art of Digital Photogrammetric Workstations for Topographic Applications, PE&RS, Vol. 61, No. 1, pp. 49-56.

[22] **ERDAS LPS Project Manager User's Guide**, March 2009

[23] **ERDAS Production Suite for ArcGIS Stereo Analyst**, FeatureAssist & ERDAS Terrain Editor *http://www.geosystems.pl/upload/zalaczniki/ERDAS Production SuiteforArcGIS.pdf* (accessed at Feb. 2010).

[24] **Introducing Stereo Analyst for ArcGIS**, December 2008

[25] **Kolbe, T., Bacharach, S. ,** 2006, CityGML: an open standard for 3D city models. From *http://www.directionsmag.com/article.php?article_id=2209* (accessed at Feb 2010).

[26] **Niwa, Y.** ,2002. ESRI Japan, Creating Orthorectified Aerial Photography Without A Camera Calibration Report

[27] **Getting Started with FME Desktop,** 2009. From *http://docs.safe.com/fme/pdf/FME GettingStarted.pdf* (accessed at Feb. 2010)

[28] **FME Workbench Help**, 2009. From *http://docs.safe.com/fme/html/FME_Workbench/ FME_Workbench_Help.htm* (accessed at Feb. 2010)

[29] **FME Workbench Transformers Quick-Reference Booklet December** 2009.From *http://downloads.safe.com/fme/brochures/transformers.pdf*, (accessed at Feb. 2010)

[30] **FME Workbench Transformers Help**, 2010

[31] **Kolbe, T.H.**, 2008. Final report on the EuroSDR **CityGML** Project

[32] **Url-4** <*http://www.ikg.uni-bonn.de/aristoteles/index.php/Aristoteles*>, accessed at April. 2010.

APPENDICES

APPENDIX A.1: Triangulation Report from the Block Model

APPENDIX A.2: ArcScene TIN to CityGML Exporter Written by Thomas H. Kolbe, Technical University of Berlin

APPENDIX A.1: Triangulation Report from the Block Model

The Triangulation Report With LPS

The output image x, y units: millimeters
The output angle unit: degrees
The output ground X, Y, Z units: meters

The Input Image Coordinates

image ID = 1

Point ID	x	y
1	-19.4135	-100.9316
2	-9.2172	83.5274
3	41.0798	88.7559
4	69.4324	113.4814
6	79.0094	-47.3088
7	88.0055	-101.8320
9	-23.0101	-56.0011
10	-14.4201	20.4980

Affine coefficients from file (pixels) to film (millimeters)

A0	A1	A2	B0	B1	B2
114.1889	-0.100761	-0.000656	-112.8052	-0.000393	0.099656

image ID = 2

Point ID	x	y
1	-100.7134	-108.7277
2	-90.3552	76.3443
3	-40.2496	81.7687
4	-12.0117	106.5057
6	-2.3831	-54.1556
7	8.5773	-109.1600
9	-104.3306	-63.4396
10	-95.6257	13.3656

Affine coefficients from file (pixels) to film (millimeters)

A0	A1	A2	B0	B1	B2
114.4520	-0.100727	-0.000576	-113.2180	-0.000290	0.099601

image ID = 3

Point ID	x	y
11	-6.5526	-64.2379
12	-16.5784	10.1124
13	-13.0111	76.6807
14	67.6756	101.9856

17	97.9186	-94.6462
18	9.6589	-99.8327

Affine coefficients from file (pixels) to film (millimeters)

A0	A1	A2	B0	B1	B2
113.8599	-0.100739	-0.000542	-113.2826	-0.000302	0.099635

image ID = 4

Point ID	x	y
11	-86.2113	-72.3688
12	-99.3936	1.8224
13	-98.1195	68.5177
14	-17.6330	96.4348
17	19.5221	-99.6401
18	-67.7948	-107.7174

Affine coefficients from file (pixels) to film (millimeters)

A0	A1	A2	B0	B1	B2
114.1897	-0.100738	-0.000656	-113.0798	-0.000348	0.099611

image ID = 5

Point ID	x	y
19	91.8630	-2.2088
20	106.4934	-72.8883
21	60.1617	-109.2967
22	16.1827	-102.1739
24	-17.0433	2.7438
25	4.0038	38.4077
26	46.9003	23.9341

Affine coefficients from file (pixels) to film (millimeters)

A0	A1	A2	B0	B1	B2
-119.8438	0.100758	0.000669	113.4028	0.000394	-0.099592

image ID = 6

Point ID	x	y
19	10.6679	7.3697
20	24.3007	-63.1858
21	-23.5636	-99.3134
22	-68.3051	-92.1288
24	-98.6814	12.3389
25	-77.1324	48.1386
26	-34.1110	33.5793

Affine coefficients from file (pixels) to film (millimeters)

A0	A1	A2	B0	B1	B2
-119.8860	0.100731	0.000581	113.8347	0.000308	-0.099604

image ID = 7

Point ID	x	y

27	91.6004	23.2094
28	110.1188	-10.3104
29	109.2005	-61.7402
30	92.8503	-104.5464
31	31.8408	-108.4762
32	4.1993	-95.8275
33	18.5138	-4.4632
34	18.0691	16.3068

Affine coefficients from file (pixels) to film (millimeters)

A0	A1	A2	B0	B1	B2
-119.6919	0.100711	0.000563	113.8607	0.000279	-0.099616

image ID = 8

Point ID	x	y
27	11.9251	27.0991
28	30.6338	-6.5704
29	29.6570	-58.2102
30	12.3368	-101.1058
31	-51.0700	-104.5408
32	-77.5847	-91.7455
33	-61.5637	-0.3561
34	-61.7630	20.4176

Affine coefficients from file (pixels) to film (millimeters)

A0	A1	A2	B0	B1	B2
-119.7885	0.100717	0.000594	113.3079	0.000314	-0.099633

THE OUTPUT OF SELF-CALIBRATING BUNDLE BLOCK ADJUSTMENT

the no. of iteration =1 the standard error = 0.0446
the maximal correction of the object points = 7.40489

the no. of iteration =2 the standard error = 0.0353
the maximal correction of the object points = 5.19691

the no. of iteration =3 the standard error = 0.0353
the maximal correction of the object points = 0.01908

the no. of iteration =4 the standard error = 0.0353
the maximal correction of the object points = 0.00004

The exterior orientation parameters

image ID	Xs	Ys	Zs	OMEGA	PHI	KAPPA
1	414701.5702	4542287.3001	1377.5527	0.8240	1.9239	178.3105
8	414700.9633	4543017.3146	1380.7572	-0.3106	0.3259	2.0067
2	414341.3333	4542287.1973	1387.3674	0.0292	2.0417	178.0117
7	414330.3549	4543021.3598	1384.7959	-0.2329	-0.0801	1.8315

3	413914.7732	4542293.1338	1388.2861	0.5148	-0.1188	177.8130
6	413940.8012	4543018.5189	1383.4545	1.3269	-0.2206	3.2153
4	413556.7818	4542291.0415	1383.9942	0.2357	0.1308	175.9112
5	413571.9616	4543043.3078	1382.1823	1.2515	-0.2187	3.2171

The residuals of exterior orientation parameters

image ID	rXs	rYs	rZs	rOMEGA	rPHI	rKAPPA
1	-0.0154	0.0174	0.0339	-0.0005	-0.0017	0.0025
8	-0.0586	-0.0337	0.0477	0.0008	-0.0033	-0.0020
2	-0.3225	-0.5283	-0.0527	0.0208	-0.0138	0.0043
7	0.1480	0.1086	-0.0147	-0.0037	0.0050	-0.0036
3	0.2788	-1.5639	-0.1456	0.0623	0.0105	0.0031
6	-0.2445	0.8148	-0.0384	-0.0342	-0.0100	-0.0020
4	0.0427	0.5921	-0.0227	-0.0220	0.0007	-0.0007
5	0.2958	-0.6969	0.1266	0.0289	0.0119	-0.0041

The accuracy of the exterior orientation parameters

image ID	mXs	mYs	mZs	mOMEGA	mPHI	mKAPPA
1	0.9164	0.8550	0.6774	0.0357	0.0402	0.0440
8	0.9510	0.9491	0.7307	0.0411	0.0414	0.0539
2	0.9131	0.8996	0.6810	0.0372	0.0400	0.0440
7	0.9293	0.9431	0.7528	0.0415	0.0406	0.0538
3	0.9258	0.8866	0.7023	0.0381	0.0415	0.0490
6	0.9538	0.9526	0.7325	0.0415	0.0417	0.0543
4	0.9235	0.9351	0.7080	0.0400	0.0413	0.0487
5	0.9411	0.9575	0.7393	0.0419	0.0413	0.0543

The interior orientation parameters of photos

image ID	f(mm)	xo(mm)	yo(mm)
1	305.0990	-0.0141	0.0065
8	305.0990	-0.0141	0.0065
2	305.0990	-0.0141	0.0065
7	305.0990	-0.0141	0.0065
3	305.0990	-0.0141	0.0065
6	305.0990	-0.0141	0.0065
4	305.0990	-0.0141	0.0065
5	305.0990	-0.0141	0.0065

The accuracy of the interior orientation parameters

image ID	mf(mm)	mxo(mm)	myo(mm)
all	0.0106	0.0106	0.0106

The values and accuracy of the additional parameters

No.	Ai	mAi	MaxX	MaxY
1	-8.0534E-008	1.9443E-008	-0.1096	-0.1414
2	3.4735E-004	5.5920E-004	0.0306	-0.0394
3	-1.7710E-004	1.0796E-003	-0.0000	-0.0156
Total	88.0055Mx	113.4814My	-0.0791	-0.1964

The residuals of the control points

Point ID	rX	rY	rZ
1	-0.2843	0.5393	-2.2544
2	-0.5404	-0.0031	1.0905
3	0.0176	-0.1016	-0.8695
4	0.1436	0.3272	0.4316
6	0.5193	-0.4209	0.7629
7	0.0959	0.1080	0.4518
9	0.4632	0.3036	0.3634
10	-0.0771	-0.2416	0.0425
11	0.1865	0.2368	1.4628
12	-0.3122	-0.4464	-0.3858
13	0.1651	-0.3382	2.4963
14	0.8789	0.6749	-0.4401
17	-0.3549	1.1772	-0.3613
18	-0.8850	-0.3324	-2.6035
19	0.5400	0.0941	0.8720
20	-0.4959	0.4943	-1.0303
21	-0.3010	-0.2350	1.5019
22	-0.3620	-0.3992	-0.7566
24	0.5435	0.0264	-1.5704
25	-0.1992	-0.1949	2.2952
26	0.2233	0.0964	-1.4000
27	0.1071	-0.0656	0.6043
28	-0.0207	0.0706	-0.4199
29	-0.3387	0.0981	-0.9372
30	0.0266	0.0731	0.9103
31	0.2079	-0.4352	1.2738
32	0.0403	0.2337	-1.8371
33	-0.1030	-0.2406	0.1836
34	-0.0089	0.1911	0.1891

aX	aY	aZ
-0.0043	0.0445	0.0023
mX	mY	mZ
0.3742	0.3689	1.2514

The coordinates of object points

Point ID	X	Y	Z	Overlap
1	414755.3257	4542753.8693	22.9456	2
2	414685.9196	4541931.0469	3.7405	2
3	414460.1576	4541914.9184	13.6505	2
4	414329.6836	4541807.1472	15.2816	2
6	414311.9593	4542526.8991	38.8629	2
7	414270.9159	4542779.7380	18.3518	2
9	414765.7232	4542553.8136	16.5634	2
10	414717.3929	4542213.4384	9.7425	2
11	413957.7265	4542590.0268	31.7628	2
12	413988.8078	4542257.7236	42.1142	2

13	413962.0451	4541965.4318	41.5963	2
14	413600.9589	4541865.5549	34.0599	2
17	413499.0851	4542742.1172	37.8387	2
18	413891.0850	4542756.4876	19.4965	2
19	413992.5300	4543086.6141	4.5720	2
20	414070.0241	4542774.7543	17.1697	2
21	413867.4190	4542608.6250	35.9019	2
22	413672.2180	4542632.9608	48.6434	2
24	413500.3035	4543080.9564	12.2296	2
25	413585.4508	4543248.2851	5.4952	2
26	413783.3033	4543193.6664	4.1000	2
27	414742.6071	4543133.7844	6.3043	2
28	414832.1193	4542985.0706	4.7801	2
29	414836.0213	4542751.8481	4.5628	2
30	414764.3866	4542559.1931	17.1103	2
31	414486.6179	4542545.6448	52.3738	2
32	414364.5803	4542592.2437	36.3629	2
33	414416.3870	4542998.3994	8.6836	2
34	414411.5711	4543092.0811	5.1891	2

The total object points = 29

The accuracy of object points

Point ID	mX	mY	mZ	mP	Overlap
1	0.5431	0.5800	0.7840	1.1162	2
2	0.5015	0.5156	0.7611	1.0472	2
3	0.4850	0.4982	0.7631	1.0324	2
4	0.5627	0.5998	0.7905	1.1407	2
6	0.4609	0.4875	0.7381	0.9975	2
7	0.5805	0.6211	0.7985	1.1663	2
9	0.4717	0.5020	0.7453	1.0149	2
10	0.4253	0.4389	0.7358	0.9565	2
11	0.4991	0.5101	0.7579	1.0410	2
12	0.4762	0.4827	0.7451	1.0075	2
13	0.5661	0.5624	0.7879	1.1214	2
14	0.6193	0.6513	0.8356	1.2272	2
17	0.6047	0.6592	0.8228	1.2153	2
18	0.5555	0.5629	0.7890	1.1171	2
19	0.4729	0.4992	0.7583	1.0236	2
20	0.5056	0.5591	0.7800	1.0848	2
21	0.5164	0.5198	0.7616	1.0568	2
22	0.5126	0.5242	0.7511	1.0497	2
24	0.4833	0.5341	0.7665	1.0519	2
25	0.5119	0.5204	0.7855	1.0723	2
26	0.4658	0.4492	0.7678	1.0042	2
27	0.4907	0.4724	0.7740	1.0310	2
28	0.4648	0.4829	0.7542	1.0090	2
29	0.4562	0.4948	0.7449	1.0039	2
30	0.4918	0.5212	0.7575	1.0427	2
31	0.4731	0.5066	0.7325	1.0085	2
32	0.4842	0.5470	0.7464	1.0444	2

33	0.4344	0.4547	0.7541	0.9819	2
34	0.4721	0.4812	0.7726	1.0254	2

amX	amY	amZ
0.5030	0.5255	0.7676

The residuals of image points

Point	Image	Vx	Vy
1	1	0.0435	0.0151
1	2	-0.0442	-0.0125

Point	Image	Vx	Vy
2	1	-0.0243	0.0365
2	2	0.0216	-0.0366

Point	Image	Vx	Vy
3	1	0.0172	-0.0074
3	2	-0.0170	0.0068

Point	Image	Vx	Vy
4	1	-0.0100	-0.0128
4	2	0.0106	0.0146

Point	Image	Vx	Vy
6	1	-0.0158	0.0052
6	2	0.0182	-0.0070

Point	Image	Vx	Vy
7	1	-0.0068	-0.0344
7	2	0.0070	0.0346

Point	Image	Vx	Vy
9	1	-0.0031	0.0018
9	2	0.0052	-0.0002

Point	Image	Vx	Vy
10	1	-0.0008	-0.0045
10	2	0.0005	0.0033

Point	Image	Vx	Vy
11	3	-0.0279	0.0615
11	4	0.0307	-0.0590

Point	Image	Vx	Vy
12	3	0.0037	0.0413
12	4	-0.0036	-0.0435

Point	Image	Vx	Vy
13	3	-0.0432	0.0013

13	4	0.0440	-0.0014

Point	Image	Vx	Vy
14	3	0.0067	-0.0463
14	4	-0.0041	0.0496

Point	Image	Vx	Vy
17	3	0.0154	-0.0222
17	4	-0.0182	0.0272

Point	Image	Vx	Vy
18	3	0.0446	-0.0282
18	4	-0.0495	0.0245

Point	Image	Vx	Vy
19	6	0.0177	0.0369
19	5	-0.0205	-0.0373

Point	Image	Vx	Vy
20	6	-0.0125	0.0107
20	5	0.0148	-0.0131

Point	Image	Vx	Vy
21	6	0.0218	-0.0380
21	5	-0.0203	0.0389

Point	Image	Vx	Vy
22	6	-0.0131	0.0144
22	5	0.0149	-0.0125

Point	Image	Vx	Vy
24	6	-0.0216	0.0576
24	5	0.0189	-0.0573

Point	Image	Vx	Vy
25	6	0.0352	-0.0674
25	5	-0.0342	0.0680

Point	Image	Vx	Vy
26	6	-0.0284	-0.0101
26	5	0.0273	0.0098

Point	Image	Vx	Vy
27	8	0.0120	0.0293
27	7	-0.0126	-0.0291

Point	Image	Vx	Vy
28	8	-0.0077	-0.0025
28	7	0.0079	0.0022

Point	Image	Vx	Vy
29	8	-0.0156	-0.0185
29	7	0.0173	0.0180

Point	Image	Vx	Vy
30	8	0.0171	0.0027
30	7	-0.0173	-0.0031

Point	Image	Vx	Vy
31	8	0.0194	0.0097
31	7	-0.0203	-0.0076

Point	Image	Vx	Vy
32	8	-0.0309	0.0110
32	7	0.0307	-0.0121

Point	Image	Vx	Vy
33	8	0.0024	-0.0214
33	7	-0.0020	0.0226

Point	Image	Vx	Vy
34	8	0.0028	-0.0111
34	7	-0.0028	0.0101

The image residuals of the control points

The image ID = 1

Point ID	Vx	Vy
1	0.0435	0.0151
2	-0.0243	0.0365
3	0.0172	-0.0074
4	-0.0100	-0.0128
6	-0.0158	0.0052
7	-0.0068	-0.0344
9	-0.0031	0.0018
10	-0.0008	-0.0045

RMSE of 8 points: mx=0.0199, my=0.0194

The image ID = 8

Point ID	Vx	Vy
27	0.0120	0.0293
28	-0.0077	-0.0025
29	-0.0156	-0.0185
30	0.0171	0.0027
31	0.0194	0.0097
32	-0.0309	0.0110
33	0.0024	-0.0214
34	0.0028	-0.0111

RMSE of 8 points: mx=0.0161, my=0.0159

The image ID = 2

Point ID	Vx	Vy
1	-0.0442	-0.0125
2	0.0216	-0.0366
3	-0.0170	0.0068
4	0.0106	0.0146
6	0.0182	-0.0070
7	0.0070	0.0346
9	0.0052	-0.0002
10	0.0005	0.0033

RMSE of 8 points: mx=0.0201, my=0.0194

The image ID = 7

Point ID	Vx	Vy
27	-0.0126	-0.0291
28	0.0079	0.0022
29	0.0173	0.0180
30	-0.0173	-0.0031
31	-0.0203	-0.0076
32	0.0307	-0.0121
33	-0.0020	0.0226
34	-0.0028	0.0101

RMSE of 8 points: mx=0.0165, my=0.0158

The image ID = 3

Point ID	Vx	Vy
11	-0.0279	0.0615
12	0.0037	0.0413
13	-0.0432	0.0013
14	0.0067	-0.0463
17	0.0154	-0.0222
18	0.0446	-0.0282

RMSE of 6 points: mx=0.0286, my=0.0386

The image ID = 6

Point ID	Vx	Vy
19	0.0177	0.0369
20	-0.0125	0.0107
21	0.0218	-0.0380
22	-0.0131	0.0144
24	-0.0216	0.0576
25	0.0352	-0.0674
26	-0.0284	-0.0101

RMSE of 7 points: mx=0.0228, my=0.0398

The image ID = 4

Point ID	Vx	Vy
11	0.0307	-0.0590
12	-0.0036	-0.0435

13	0.0440	-0.0014
14	-0.0041	0.0496
17	-0.0182	0.0272
18	-0.0495	0.0245

RMSE of 6 points: mx=0.0308, my=0.0391

The image ID = 5

Point ID	Vx	Vy
19	-0.0205	-0.0373
20	0.0148	-0.0131
21	-0.0203	0.0389
22	0.0149	-0.0125
24	0.0189	-0.0573
25	-0.0342	0.0680
26	0.0273	0.0098

RMSE of 7 points: mx=0.0225, my=0.0401

APPENDIX A.2: ArcScene TIN to CityGML Exporter Written by Thomas H. Kolbe, Technical University of Berlin

```
Private Sub CommandButton1_Click()
    ExportTINasCityGML
End Sub

' Converts Double numbers to Strings using "." as the decimal point.
' This function was implemented, because I haven't found an alternative
' way to avoid localized formatting of double numbers (in a German
' Windows installation the decimal separator is printed as ",")

Private Function DoubleToStringUS(dnum As Double) As String
    DoubleToStringUS = Replace(Format(dnum), ",", ".")
End Function

Private Sub ExportTINasCityGML()
    Dim pSxdoc As ISxDocument
    Set pSxdoc = ThisDocument
    Dim pScene As IScene
    Set pScene = pSxdoc.Scene

    Dim pLayer As ILayer
    Dim pDataLayer As IDataLayer
    Dim pDatasetName As IDatasetName
    Dim pWSName As IWorkspaceName
    Dim sFDS As String
    Dim pFCName As IFeatureClassName
    Dim i, j As Long
    Dim pTinLayer As ITinLayer
    Dim pTriangle As ITinTriangle
    Dim pTinNode As ITinNode
    Dim pTin As ITinAdvanced
    Dim bSuperNode As Boolean
    Dim lInvalidTriangles, lNumTriangles As Long
    Dim pEnumTinTriangle As IEnumTinTriangle

    Dim lFileID As Long
    Dim sTextFileName As String
    sTextFileName = "D:\Temp\TINWriter.xml"

    Set pLayer = pSxdoc.SelectedLayer
```

```
    If Not pLayer Is Nothing Then
        ' Debug.Print pLayer.Name
    Else
        MsgBox "Please select a TIN Layer before starting this tool!"
        Exit Sub
    End If

    If TypeOf pLayer Is ITinLayer Then
        Set pTinLayer = pLayer
        ' If pTinLayer.Dataset.IsDelaunay Then Debug.Print "TIN is Delaunay
triangulated"
        Set pTin = pTinLayer.Dataset

        ' open file:
        sTextFileName = TextBox1.Text
        lFileID = FreeFile()
        If Len(Dir(sTextFileName)) > 0 Then Kill sTextFileName
        Open sTextFileName For Append As lFileID

        Print #lFileID, "<?xml version=" + Chr(34) + "1.0" + Chr(34) + " encoding=" +
Chr(34) + "UTF-8" + Chr(34) + "?>"
        Print #lFileID, "<CityModel xmlns:xsi=" + Chr(34) +
"http://www.w3.org/2001/XMLSchema-instance" + Chr(34);
        Print #lFileID, " xsi:schemaLocation=" + Chr(34) +
"http://www.citygml.org/citygml/1/0/0
http://www.citygml.org/citygml/1/0/0/CityGML.xsd" + Chr(34);
        Print #lFileID, " xmlns:gml=" + Chr(34) + "http://www.opengis.net/gml" +
Chr(34);
        Print #lFileID, " xmlns:xlink=" + Chr(34) + "http://www.w3.org/1999/xlink" +
Chr(34);
        Print #lFileID, " xmlns:xAL=" + Chr(34) +
"urn:oasis:names:tc:ciq:xsdschema:xAL:2.0" + Chr(34);
        Print #lFileID, " xmlns=" + Chr(34) + "http://www.citygml.org/citygml/1/0/0" +
Chr(34) + ">"
        Print #lFileID, "<cityObjectMember>"
        Print #lFileID, "<ReliefFeature>"
        Print #lFileID, "<gml:name>" + TextBox3.Text + "</gml:name>"
        Print #lFileID, "<lod>" + TextBox4.Text + "</lod>"
        Print #lFileID, "<reliefComponent>"
        Print #lFileID, "<TINRelief>"
        Print #lFileID, "<gml:name>" + TextBox5.Text + "</gml:name>"
        Print #lFileID, "<lod>" + TextBox6.Text + "</lod>"
        Print #lFileID, "<tin><gml:TriangulatedSurface srsName=" + Chr(34) +
TextBox2.Text + Chr(34) + ">"
        Print #lFileID, "<gml:trianglePatches>"

        lInvalidTriangles = 0
        lNumTriangles = 0
        Set pEnumTinTriangle = pTin.MakeTriangleEnumerator(Nothing, esriTinAll,
Nothing)
```

```
        Do
            Set pTriangle = pEnumTinTriangle.Next
            If Not (pTriangle Is Nothing) Then
                bSuperNode = False
                For i = 0 To 2
                    If pTin.IsVoidZ(pTriangle.Node(i).Z) Then bSuperNode = True
                Next i
                If Not bSuperNode Then
                    lNumTriangles = lNumTriangles + 1
                    Print #lFileID, "<gml:Triangle><gml:exterior><gml:LinearRing>";
                    Print #lFileID, "<posList srsDimension=" + Chr(34) + "3" + Chr(34) +
">";
                    For i = 2 To 0 Step -1
                        Set pTinNode = pTriangle.Node(i)
                        Print #lFileID, DoubleToStringUS(pTinNode.x) + " " +
DoubleToStringUS(pTinNode.y) + " " + DoubleToStringUS(pTinNode.Z) + " ";
                    Next i
                    Set pTinNode = pTriangle.Node(2)
                    Print #lFileID, DoubleToStringUS(pTinNode.x) + " " +
DoubleToStringUS(pTinNode.y) + " " + DoubleToStringUS(pTinNode.Z);
                    Print #lFileID, "</posList>";
                    Print #lFileID, "</gml:LinearRing></gml:exterior></gml:Triangle>"
                Else
                    lInvalidTriangles = lInvalidTriangles + 1
                End If
            End If
        Loop Until pTriangle Is Nothing

        Print #lFileID, "</gml:trianglePatches></gml:TriangulatedSurface></tin>"
        Print #lFileID, "</TINRelief>"
        Print #lFileID, "</reliefComponent>"
        Print #lFileID, "</ReliefFeature>"
        Print #lFileID, "</cityObjectMember>"
        Print #lFileID, "</CityModel>"

        Close lFileID
        MsgBox Format(lNumTriangles) + " triangles exported!"
        Unload Me
    Else
        MsgBox "Please select a TIN Layer before starting this tool!"
        Exit Sub
    End If
End Sub


Private Sub CommandButton2_Click()
    Unload Me
End Sub
```

```
Private Sub CommandButton3_Click()
    MsgBox "File selection is not working yet; please enter the filename (incl. path)
into the text field to the left. (I'll have to find out how to open a file dialog in
ArcScene...)"
End Sub

Private Sub TextBox4_AfterUpdate()
    Dim iLOD As Integer
    iLOD = Val(TextBox4.Text)
    If (iLOD < 0 Or iLOD > 4) Then iLOD = 0
    TextBox4.Text = Format(iLOD)
End Sub

Private Sub TextBox6_AfterUpdate()
    Dim iLOD As Integer
    iLOD = Val(TextBox6.Text)
    If (iLOD < 0 Or iLOD > 4) Then iLOD = 0
    TextBox6.Text = Format(iLOD)
End Sub
```

CURRICULUM VITA

Candidate's full name: **Ahmet ŞENGÜL**

Place and date of birth: **Erzincan - 07.09.1981**

E-mail Address: ahmetsengul@gmail.com

Universities and Colleges attended:

- Geodesy and Geoinformation Science, ERASMUS MSc Exchange Student at TU Berlin, Germany (2008-2009)
- Geomatics Engineering - MSc Student at Istanbul Technical University (2006-2010)
- Geodesy and Photogrammetry Eng. BSc at Yıldız Technical University, Istanbul (2000-2005)

Publications:

▪ **Sengul A.,** 2008, 3D GIS Example in the Historical Peninsula of Istanbul, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing 2008, p 659-664.

Work Experiences

- TERRA IMAGING Company – Berlin, Germany (2009 August -)
  - Classification of LIDAR data, Freelancer – Student Job

- EMIN Limited A.S. & Vadi Yazilim (2006 September – 2008 October)
  - Urban Information Department, GIS Expert, Provide GIS reports and analyses for the management authorities, Municipality of Eminönü and Uskudar, Istanbul.
  - Worked as the general coordinator for the house number data collection project of the district Eminönü- Quality, Management and data procedure producer.

- INFOTECH A.Ş - TELE ATLAS (2005 October – 2006 August)
  - Worked in the Navigation project of Turkey as a team member, worked at the big cities of Turkey-Istanbul, Ankara and Bursa, for implementing the navigation database, as a Local Data Collector (LDC),

- IAESTE PRACTICAL TRAINING Agriculture and Food of Ss Cyril and Methodius University, Skopje / Macedonia (2005 July- September)
  - Projects carried on for the application of GIS in land management with special emphasis on soil, erosion, DTM, slope and their interactions
  - Digitalization, Geo-referencing & development of thematic maps for different areas of R. Macedonia
  - Analyses of agricultural areas based on CORINE and DTM data

- NETSOFT Computer Systems (2004 November – 2005 June)
  - Team member, Urban transformation focused on Earthquake Master Plan for Istanbul Metropolitan Municipality by means of Earthquake Information System Project
  - Worked as one of the Coordinators of the GIS and supported architectural project for Metropolitan Municipal Office attended the International Project Financing Investment Fair (MIPIM) for Istanbul Settlements & Urban Transformation Department

- TUBITAK – Turkish Scientific Technical Research Organization, MIT and CNRS-National Center for Scientific Research Corporation (2004 September)
  - Team member, Surveying deformations of Earthquake faults with GPS

- ERDEM EMİ Mapping LTD.ŞTİ (2003 July- December)
  - Part-time student job- Erdem EMİ Mapping Information Company August-December 2003