BUILDING ENERGY MODEL GENERATION USING A DIGITAL PHOTOGRAMMETRY-BASED 3D MODEL

A Thesis

by

MOHAMMAD S A A M ALAWADHI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Wei Yan
Committee Members,	Jeff Haberl
	Julian Kang
Head of Department,	Robert Warden

May 2017

Major Subject: Architecture

ABSTRACT

Buildings consume a large amount of energy and environmental resources. At the same time, current practices for whole-building energy simulation are costly and require skilled labor. As Building Energy Modeling (BEM) and simulations are becoming increasingly important, there is a growing need to make environmental assessments of buildings more efficient and accessible. A building energy model is based on collecting input data from the real, physical world and representing them as a digital energy model. Real-world data is also collected in the field of 3D reconstruction and image analysis, where major developments have been happening in recent years. Current digital photogrammetry software can automatically match photographs taken with a simple smartphone camera and generate a 3D model.

This thesis presents methods and techniques that can be used to generate a building energy model from a digital photogrammetry-based 3D model. To accomplish this, a prototype program was developed that uses 3D reconstructed data as geometric modeling inputs for BEM.

To validate the prototype, an experiment was conducted where a case-study building was selected. Photographs of the building were taken using a small remotely-controlled Unmanned Aerial Vehicle (UAV) drone. Then, using photogrammetry software, the photographs were used to automatically generate a textured 3D model. The texture map, which is an image that represents the color information in the 3D model, was semantically annotated to extract building elements. The window annotations were

used as inputs for the BEM process. In addition, a number of algorithms were applied to automatically convert both the 3D model and the annotated texture map into geometry that is compatible for a building energy model. Through the prototype, pre-defined templates were used with the geometric inputs to generate an EnergyPlus model (as an example building energy model). The feasibility of this experiment was verified by running a successful energy simulation. The results of this thesis contribute towards creating an automated and user-friendly photo-to-BEM method.

DEDICATION

To my beloved parents.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my committee chair, Dr. Wei Yan, for his guidance, encouragement, and support. His knowledge, overwhelming dedication towards his students, and being always available to help them grow academically, has been invaluable for completing my work. I would also like to thank my committee members, Dr. Jeff Haberl and Dr. Julian Kang, for their time and advice throughout this experience.

I would also like to extend my thanks to the various people who contributed to the realization of this work, including Dr. Charles Culp, who helped me gain a better understanding of EnergyPlus, and to my fellow graduate students from the BIMSIM Group, who showed me their support by being such a pleasant group. I really enjoyed their acquaintance.

Finally, I wish to thank my wonderful family and friends back home, especially the two dearest persons to me, my parents, for their love, and for their encouragement throughout my study.

CONTRIBUTORS AND FUNDING SOURCES

1) Contributors

This work was supported by a thesis committee consisting of Professor Wei Yan and Jeff Haberl of the Department of Architecture and Professor Julian Kang of the Department of Construction Science.

All work conducted for the thesis was completed independently by the student, under the advisement of Wei Yan of the Department of Architecture.

2) Funding Sources

Graduate study was supported by a scholarship from Kuwait University under the supervision of The Cultural Office of the Embassy of the State of Kuwait.

NOMENCLATURE

2D	Two-Dimensional
2.5D	Two-and-a-half-Dimensional
3D	Three-Dimensional
AEC	Architecture, Engineering, and Construction
BEM	Building Energy Modeling
BIM	Building Information Modeling
BREP	Boundary Representation
CAD	Computer-Aided Design
CFD	Computational Fluid Dynamics
CityGML	*Open standard file format for GIS applications*
ECM	Energy Conservation Measures
EPW	*EnergyPlus file format for weather data*
f	*Face of a 3D mesh in an OBJ text file*
gbXML	*Open standard file format for engineering analysis applications*
GIS	Geographical Information Systems
HD	High Definition
ID	Identification
IDF	*EnergyPlus file format for building description data*
INP	*DOE-2.2 file format for building description data*
IR	Infra-Red

LIDAR	Light Detection and Ranging
LOD	Level Of Detail
MTL	*OBJ companion file format for texture definitions*
MVS	Multi-View Stereo
OBIA	Object-Based Image Analysis
OBJ	*Open standard file format for 3D geometry data*
RANSAC	Random Sample Consensus
REM	Rapid Energy Modeling
RGB	*Red, Green, and Blue color model*
SfM	Structure from Motion
TMY	Typical Meteorological Year
ToF	Time-of-Flight
UAV	Unmanned Aerial Vehicle
v	*Vertex of a 3D mesh in an OBJ text file*
vn	*Normal of a vertex in an OBJ text file*
vt	*Texture coordinate of a vertex in an OBJ text file*
X, Y, Z	Axes in the Cartesian coordinate system
x	*X axis value in the Cartesian coordinate system*
y	*Y axis value in the Cartesian coordinate system*
z	*Z axis value in the Cartesian coordinate system*
λ	*Lambda, a mathematical symbol representing a ratio of an area in the context of Barycentric coordinates*

TABLE OF CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION

In the Architecture, Engineering and Construction (AEC) field, using automatic 3D reconstruction methods such as digital photogrammetry in conjunction with image analysis could streamline the labor-intensive process of Building Energy Modeling (BEM) and Building Information Modeling (BIM) of existing buildings.

The proliferation of 3D reconstruction methods is becoming more relevant in recent years, with end-to-end solutions, tools, and software being used in the AEC, GIS, archeology, and virtual reality fields. Current photogrammetry-based methods in particular enable the automated reconstruction of detailed and high-fidelity 3D built environments from a set of photographs. Developments in this area are also evident in the public realm with the accessibility of solutions ranging from Google Earth (Google Earth, 2017), which provides viewable 3D cities reconstructed from aerial imagery, to Autodesk 123D Catch (123D, 2017), a free and easy-to-use consumer software that automatically generates 3D geometry from uploaded pictures.

In the context of BEM, modeling and simulation for building performance analysis are useful for sustainable building design. In the case of existing buildings, calibrated building energy models are constantly being used for evaluating ECMs (Energy Conservation Measures), building diagnostics, and retrofit analysis simulations. The workflow of building performance analysis for existing buildings is a multidisciplinary approach that includes architectural measurement from existing

construction drawings or site surveys, a definition of building components and systems, and in many cases, 3D modeling using BEM and/or BIM tools. This process is often labor-intensive (O'Donnell et al., 2014) and is usually performed manually by knowledgeable experts. Any human errors would decrease the efficiency and accuracy of the modeling process. An example of BEM inputs for EnergyPlus (DOE, 2015) - a well-established whole-building energy simulation program, is shown in Appendix A.

By combining both building 3D reconstruction and image processing methods, creating building energy models of existing buildings could be automated or semi-automated to enable an easier and more time-efficient workflow. This thesis presents a literature review of previous 3D reconstruction methods that were utilized for BEM, BIM, and relevant fields in recent research, and presents methods to generate a building energy model from a photogrammetry-based 3D mesh. The methods are presented using an experimental prototype program that applies geometric algorithms and image analysis concepts on a 3D reconstructed case-study building to produce a semantic 3D building model; where the building geometry is semantically classified further into walls, windows, and roofs. This model is then converted into an EnergyPlus model as an example building energy model.

1.1 Research Problem

There has been a lack of progress toward the development of an automated process that converts photographs of an existing building into a building energy model. Furthermore, exporting models from CAD (Computer-Aided Design) and/or BIM

software to BEM tools is a potentially costly and labor-intensive process that requires editing and robust software interoperability, while the costs of interoperability (including software interoperability) issues alone in the United States AEC industry amounted to $15.8 billion in 2003 (Gallaher et al., 2004). On the other hand, recent attempts have looked into utilizing various 3D reconstruction methods for creating models for BEM.

The use of photogrammetry-based 3D reconstruction methods for BEM was proposed in the AEC software industry by researchers. Autodesk proposed Rapid Energy Modeling (REM) workflows for existing buildings using their available software solutions such as ReCap 360 (ReCap 360, 2016), ImageModeler (ImageModeler, 2009) and Revit (Revit, 2016) to streamline the process between vision-based measurements, such as photographs, and BEM (Autodesk, 2011, 2016a, 2016b). One approach by Autodesk was Project Rosenfeld - a prototypical mobile device software that attempted to combine photogrammetry, BEM, and energy simulation in a simple user interface with a semi-automatic workflow (Autodesk, 2016). However, Autodesk REM workflows do not include automatic recognition of building components and they have to be manually defined by the user.

The automatic recognition of building components and automated 3D reconstruction of built environments from laser-based point cloud data have been discussed in previous research. For Example, C. Wang (2014) proposed a methodology to generate a building energy model using the gbXML open standard schema by using computer algorithms that automatically recognize wall, window, and roof components of a building envelope from LIDAR-based 3D point cloud data. The point cloud data was

acquired from a data collection system that uses a laser scanner and an Infra-Red (IR) camera (C. Wang, 2014), which is different from photogrammetry-based 3D reconstruction methods. A literature review of 3D reconstruction methods and the work of C. Wang among other research is further explained in Chapter II.

The scope of this research addresses the limitations of using digital photogrammetry geometry for BEM, namely:

- Information output from digital photogrammetry, such as 3D geometry and texture, lacks the building information semantics that are relevant to BEM and BIM, such as wall, window, and roof elements. A necessary aspect of BEM or BIM is the definition and classification of building components. In BEM, semantics in the 3D geometric components such as walls, windows, roofs and floors with their respective material properties are usually manually defined by the user before being saved in file formats such as DOE-2.2 (DOE & Hirsch, 2016) INP files, EnergyPlus IDF files, or gbXML open standard schema. However, the output of digital photogrammetry is a 3D textured mesh that lacks the additional semantic information that is relevant to either BEM or BIM. This raises a challenge to convert the 3D mesh to a semantically enriched building model.
- Geometry data from digital photogrammetry cannot be readily used as BEM geometry. Not all 3D geometry can be translated into a building energy model. For example, a 3D model that does not define a closed volume cannot be modeled as a volume input for thermal calculations, and boundary conditions of

surfaces (inside/outside) cannot be geometrically defined. As for surface geometry that represent building surfaces, both EnergyPlus and DOE-2.2 do not directly support curved geometry. Also, OpenStudio (OpenStudio, 2016), which is a software suite that allows 3D modeling for EnergyPlus, requires window surfaces to be coplanar to wall surfaces. As for window shapes, EnergyPlus only supports rectangular and triangular windows. DOE-2.2 only supports rectangular windows (DOE & Hirsch, 1998). Additionally, there might be some limitations in handling concave surfaces.

1.2 Research Objectives

This thesis aimed to develop and test a methodology through a prototype program that semi-automatically converts digital photogrammetry data of an existing building into geometry that is compatible with BEM tools. Additionally, it aimed to resolve the following technical challenges relevant to this objective:

- To present a methodology for semantically enriching digital photogrammetry data of existing buildings.
- To develop toolsets that translates digital photogrammetry geometry into BEM geometry.
- To verify that the resulting BEM geometry is working as intended by running a successful energy simulation.

1.3 Background

In addition to the literature review presented in Chapter II, the following is a brief overview of the relevant topics to the scope of the research:

1.3.1 Building Energy Modeling (BEM)

"Simulation is the process of creating and experimenting with a computerized mathematical model of a physical system" (Chung, 2003). In the context of buildings, BEM is the virtual representation of a building system to simulate energy use with computer software (Jefferis et al., 2012). A building energy model is defined through various inputs that attempt to accurately simulate a building in real-world conditions, such as surface areas, space volumes, material thermal properties, layering of wall components, and boundary conditions. BEM workflows for energy-use simulations (and daylighting analysis) commonly start with manually modeling buildings in a simple 3D geometrical representation (although heat transfer calculations do not rely on 3D modeling) using whole-building simulation software, or by editing and exporting available digital models from CAD or BIM software to BEM tools such as EnergyPlus or DOE-2.2.

1.3.2 Building Information Modeling (BIM)

BIM is an activity (Eastman et al., 2011) that involves the digital creation, maintenance, and distribution of a building's physical and functional characteristics (Briscoe, 2015). The concept of BIM evolved from object-based parametric modeling

into a number of widely-used BIM software tools in the AEC industry and in academia (Kensek & Noble, 2014). When compared to CAD, a BIM model would contain much more semantic information of geometric objects such walls, windows, roofs, materials, and costs, whereas a CAD model would often only represent the graphical information of the same objects. Since BEM requires definitions of building objects, BIM is considered a good platform for BEM, as BIM model geometry is semantically enriched and can be exported to different file formats. Furthermore, previous research looked into integrating BIM into parametric workflows for building performance simulation (Asl et al., 2014; Asl et al., 2013; Kim et al., 2015; Yan et al., 2013). In this study, BIM is not part of the scope, but is identified as a relevant field.

1.3.3 Image Analysis

Image analysis is a computer vision-based field that applies image processing algorithms to extract information from digital images and photographs, such as shapes, contours, and colors. It can be used for image segmentation and object recognition to extract semantic information from building photographs, such as annotating the windows on a facade. Object-based image analysis (OBIA) is a computer vision-based method for image recognition and an improvement to previous pixel-based image analysis methods (Burnett & Blaschke, 2003). It is used to classify objects in digital photographs. Frommholz et al., (2015) applied OBIA by using Trimble eCognition (eCognition, 2016) software to extract window areas from 3D building model textures for virtual reality applications. In another example that applied image analysis, Cao et al., 2015 presented

a method to extract window geometry from low-resolution single-view oblique aerial imagery and proposed complementing their method with current 3D reconstruction methods for BIM and BEM applications. Other similar methods for extracting windows are explained in Chapter II and Chapter III.

1.4 Methodology

The thesis contains a literature review of previous research that utilized 3D reconstruction methods such as digital photogrammetry for BEM and BIM. The literature's significance, limitations, and its relationship to the research problem was studied, and is followed by an experiment that addresses a novel approach to the problem. The experiment was conducted through the following steps:

1.4.1 Data Collection of a Building Case Study in a Field Survey Using Aerial Digital Photogrammetry

An existing case-study building was selected for 3D reconstruction using digital photogrammetry software. Data was collected in a field survey using a remotely-controlled UAV drone fitted with a HD camera. Photographs of the building were taken by the UAV drone while following a circular flight path. The photographs from the field survey were then processed in Autodesk's cloud-based photo-to-3D service, which is available for both Autodesk ReCap 360 and Autodesk Remake (Remake, 2016). The resulting textured 3D mesh was then edited and exported to OBJ; an open file format.

1.4.2 Processing Image-based Photogrammetry Data

The case-study building's image data was contained in a texture map. A texture map could be a single (or multiple) raster image(s) containing pixels that wrap around parts of a 3D model. This model would contain the texture coordinates (also known as UV coordinates) of each vertex and a reference to the texture map. Using Grasshopper (Rutten, 2014) - a visual programming tool, a toolset was developed that automatically converts semantically annotated 2D pixels in the texture map to semantic 3D geometry. A proposed methodology to annotate the texture map is presented and intended to be integrated with the outputs of available computer vision and image processing tools such as OpenCV (OpenCV, 2016) and MATLAB (MATLAB, 2017) as future work. Image processing and analysis tools were not used in this study and are not part of the scope.

1.4.3 Prototyping a Building Geometry Extractor and Validation of the Prototype Through Building Energy Simulation

A prototype was developed using Rhinoceros (Rhinoceros, 2016) (a 3D modeling tool) and Grasshopper that inputs a digital photogrammetry geometry and an annotated texture map, and then automatically converts the photogrammetry 3D mesh into a semantic building model. The model was then translated into an EnergyPlus IDF file using the Ladybug (Roudsari, 2016a) and Honeybee (Roudsari, 2016b) plugins for Grasshopper. The prototype's model output was verified by running a building energy simulation using the EnergyPlus engine. Integration with BIM is suggested as future work. The research methodology framework is shown in Figure 1.

The first and second parts of the framework; data acquisition and window extraction respectively, are presented in Chapter III. Data acquisition involved conducting a field survey where photographic data was collected using a UAV drone. A 3D model of the building was then automatically generated using a digital photogrammetry software tool. For window extraction, the texture map of the 3D model was analyzed and a methodology was presented to semantically annotate the 3D model with building element labels, such as wall, window, and roof elements. The window annotations were used for the third part of the framework, which involved the development of the prototype. The prototype program is presented in Chapter IV, and a more detailed framework is expanded in the same chapter.

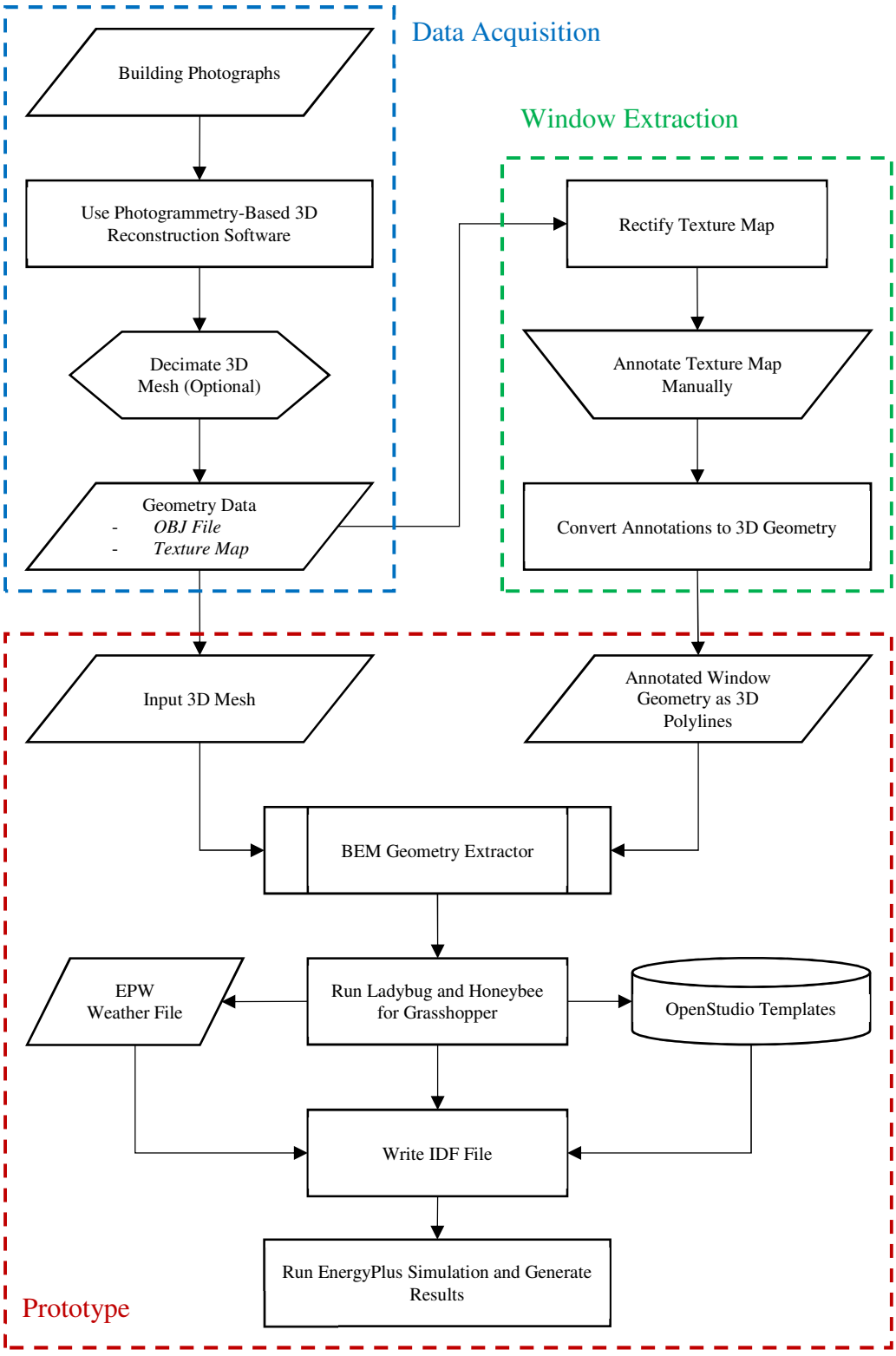


Figure 1. Research methodology framework.

1.5 Significance

This study is expected to provide a simple and improved method for the simulation of existing buildings using photogrammetry-based 3D reconstruction, more specifically:

- Provide an architect or an engineer with a rapid workflow from on-site photography to a building energy model.
- Allow a user-friendly method for a building stakeholder to evaluate the performance of an existing building.

1.6 Outcomes

A comprehensive literature review of recent methods on converting 3D reconstructed geometry to BEM and BIM is presented in this thesis. The relevance of these methods to the research problem were addressed. A prototype program was developed to demonstrate a novel process of bridging digital photogrammetry with BEM tools through the use of geometry processing algorithms and image analysis concepts. Finally, in order to validate the methodology, the output of the prototype, which is a building energy model, was verified by running a debugged and successful energy simulation.

CHAPTER II

LITERATURE

The literature review is divided into three parts. The first part will present various data acquisition methods for use in 3D reconstruction. The second part covers the data structure of the geometry resulting from digital photogrammetry software. The third part will review previous work that are relevant to the scope of this study.

2.1 Data Acquisition Methods

3D reconstruction can be defined as recording the measurements and/or appearance of physical objects as three-dimensional data. It can be alternatively known as 3D scanning. The data can be collected through a variety of techniques, and has applications in multiple research fields such as computer vision, remote sensing, and virtual reality. More recent applications include reverse engineering of mechanical parts (Górski et al., 2010). In this section, two methods for collecting data have been identified: laser scanning, and digital photogrammetry.

2.1.1 Laser Scanning

LIDAR (Light Detection and Ranging) or laser scanning, is a common technique for 3D reconstruction which produces point cloud data. A laser scanner is a sensor that emits a beam of light (i.e., laser) to record the distance and/or appearance (such as color information) of an object based on the reflected beam measured by the sensor.

One method to measure the reflected beam is through Time-of-Flight (ToF), where the sensor, which is a laser range finder, records the distance of a point by measuring the time it takes for the beam to travel to the scanned object and then reflected back. This method often employs a system of rotating mirrors to rapidly scan a real-world object by registering a large number of points in a short period of time.

Another method of laser scanning is based on the concept of triangulation. In this method, the sensor used is a camera that is configured beside the beam emitter within a certain distance. Based on where the beam's visible appearance is captured on the camera's sensor, the distance between the object and the scanner is recorded. Color information can also be recorded through the camera's sensor. The triangulation method is often employed for hand-held laser scanners (Salomon, 2011).

2.1.2 Digital Photogrammetry

Photogrammetric theory has existed since the 15th century (Burtch, 2008). It is being employed today to obtain measurements of physical objects from digital photographs, and the same principles are used in photogrammetry-based 3D reconstruction software. In contrast to the laser scanning method, digital photogrammetric reconstruction is based on matching points between two or more photographs. By matching points, a photogrammetry software can construct a sparse point cloud and determines the positions of where the photographs were taken. A dense point cloud can be constructed through interpolation by referencing the pixels in the images. The output of some photogrammetry software also includes a 3D mesh, which is

generated by triangulating the points of the point cloud. Textures can be added on the mesh by wrapping and blending the photographs on the mesh.

Photogrammetry as a discipline is constantly changing and influenced by the developments in computer science (Schenk, 2005). Applied photogrammetry methods relevant to the AEC field (including the methods used in software packages) vary and implement different algorithms based on previous research and proprietary processes.

One of the algorithms used for matching points between photographs is known as feature matching, which was developed in the fields of computer vision and image processing. Feature matching is performed by analyzing the pixels between multiple images and finding the same feature such as edges, corners, and blobs. A blob refers to a region in an image that differs from surrounding regions, such as in color, texture, or brightness.

Photographs can be aligned by finding the position and orientation of the camera in 3D space. Structure-from-motion (SfM) algorithms are used to reconstruct the camera's parameters such as its position in the world coordinate system (x, y, z) and the focal length.

In order to produce more dense point clouds, multi-view stereo (MVS) algorithms are used. MVS method assumes known camera parameters and attempts to interpolate the 3D positions of all pixels in a series of photographs, and is often a subsequent process following SfM to produce a more detailed 3D model.

In the AEC, heritage conservation, and archeology fields, examples of commercial software that use automatic photogrammetry-based methods for 3D

reconstruction of built environments include ReCap 360 (online-only), Remake, and Agisoft Photoscan (Photoscan, 2016). Other software such as ImageModeler and PhotoModeler (PhotoModeler, 2016) use semi-automatic methods which provide manual user inputs for generating the output 3D data.

While laser scanning generally provides higher resolutions and more details than digital photogrammetry, the latter provides a better representation of surface textures. Also, digital photogrammetry equipment, which could be a simple mobile phone camera, costs significantly less than laser scanners.

This study utilizes digital photogrammetry as a research tool in the development of the experimental prototype. Photogrammetry software was used to acquire and reconstruct geometric data of an existing building. The geometric data was processed as input data for the prototype, and this study focused on methods to process 3D reconstructed geometry into a building energy model. The scope of this study did not cover the details and technicality of the photogrammetry methods used for the prototype, due to the difficulty of identifying the exact implementation of the methods and algorithms used in proprietary software, while the mathematics of the methods and algorithms is discussed in computer visions textbooks (Förstner & Wrobel, 2016; Szeliski, 2010).

2.1.3 Unmanned Aerial Vehicles (UAVs)

While the vision-based input data required for photogrammetry (e.g., photographs) can be collected using a simple camera, the quality of the photographs,

such as image resolution and camera positioning, play an important role in the quality of the output data.

In terms of positioning, there are two major types of photogrammetry: close-range photogrammetry using ground-based cameras, and aerial photogrammetry which involves taking photographs from an aircraft. Vision-based data of existing built environments is acquired using ground or aerial equipment depending on the applications. One issue when using a handheld (ground-based) camera to take photographs of a building, is that the roof area is often not visible in the camera's field of view. This can be resolved if aerial photogrammetry is used to capture photographs of the roof in addition to the walls. Small consumer and recreational UAV drones equipped with HD digital cameras can be used as a cost-effective solution for aerial photogrammetry. During the field survey, photographic data was acquired using this approach in this study.

2.2 3D Geometry Data Structure

The output of the photogrammetry software used in this study is a 3D textured mesh. This mesh will be used as an input for the methodology presented in Chapter III and the experimental prototype in Chapter IV.

2.2.1 3D Meshing

In the context of computer graphics, a mesh is a data structure that represents 3D geometry and is composed of vertices, edges, and faces. A vertex is a data structure that

represents a point in 3D space defined in the Cartesian coordinate system (x, y, z), while an edge is a line joining two vertices. Faces are the elements that represent the surface of a mesh, and in many computer software, each single face is represented as a 3-sided polygon (triangle).

2.2.1.1 Vectors

Similar to a vertex, a vector is defined by the Cartesian values of $<x, y, z>$, however, these values represent a direction and magnitude as opposed to a point (Issa, 2013). The length of the vector $<x, y, z> = \sqrt{(x^2 + y^2 + z^2)}$. A vector with a length of 1 is known as a unit vector. Given any arbitrary vector, a unit vector is calculated by dividing each vector's x, y, and z components by the vector's length. This is useful for normalizing vector magnitudes.

A normal vector is a unit vector that is perpendicular to a surface at a given point. In 3D meshes, the direction of each face can be known by computing its normal vector. For a face that is defined by three corners a, b, and c in the counter-clockwise direction, the normal vector = $(b - a)$ x $(c - a)$, or equals to the unitized cross product of the vectors.

2.2.2 Texture Mapping

A texture map is an image that is wrapped on a 3D model. It is used to add colors, textures, and material appearance to a model. Texture mapping is one of the essential methods in the field of 3D rendering. The resolution of a texture map depends

on the capability of the hardware and the rendering software, but it is not uncommon to find texture maps with very high resolutions used in real-time 3D applications.

Most software process texture maps as a square-shaped image. In order to project the map on a 3D mesh, the square image is treated as a 2D space with a Cartesian system, and the designations of the X and Y axes are replaced with U and V respectively. These coordinates are known as UV coordinates (also referred to as texture coordinates). UV values range from 0 to 1. Each vertex in a 3D mesh is assigned a UV coordinate that corresponds to a point on the texture map, and this process is known as UV mapping. Subsequently, each texture region between the assigned UV coordinates will be mapped on the faces of the mesh. UV mapping was used as a technique in this study.

2.2.2.1 OBJ

Wavefront OBJ is a common text-based file format that is used in many 3D software tools to represent 3D geometry. In order to represent texture, an external text-based MTL file is used, which references an image file. The MTL file should be in the same file directory as the OBJ.

Within the OBJ file, each vertex is represented as *v* followed by its three Cartesian values. If a vertex has a texture coordinate, it is specified following each vertex as *vt* with its UV values. Optionally, the text file can include *vn* which is the normal vector values. Faces are identified as *f* followed by 9 values: each value

represents the vertex index, vertex texture index, or vertex normal index. Below is a text example of an OBJ file that represents a textured 3D mesh:

```
# 26387 vertices, 37890 faces
mtllib TextureMap.mtl
v -6.769291 -5.476615 1.808448
vn -0.974540 0.170803 0.145252
vt 0.987427 0.170564
…
f 1/1/1 2/2/2 3/3/3
…
f 19001/25439/19001 9582/26118/9582 9520/26344/9520
…
```

The texture mapping process has been automated in most available photogrammetry software and is part of an end-to-end solution to generate realistic, textured, 3D reconstructed objects. Figure 2 shows a texture map that is generated automatically from a photogrammetry software (Photoscan) following the 3D reconstruction process. After the generated mesh was saved as an OBJ file, an MTL file is created to reference the texture map. All files were saved in the same directory.

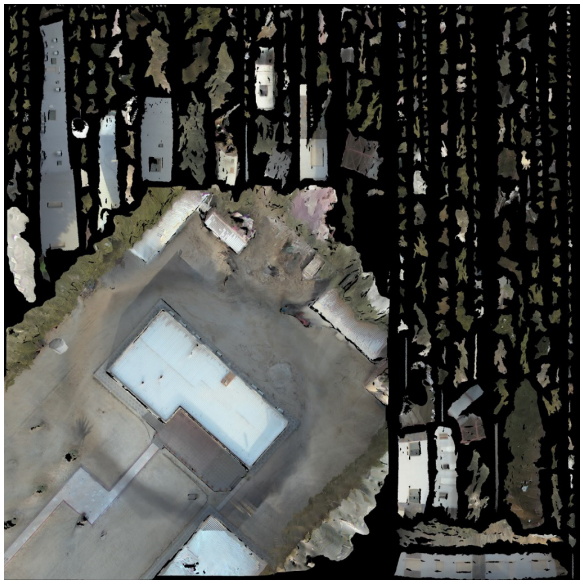


Figure 2. A texture map for a photogrammetry-based 3D mesh.

2.3 Using 3D Reconstructed Data

One of the challenges that is inherent in 3D data acquired from vision-based hardware such as photographs and laser scanners is dense information that must be processed and remodeled before the use for BEM or BIM. The 3D geometry for building performance models such as energy, daylighting, and CFD (Computational Fluid Dynamics) models should be simple and not contain redundant spatial information. Building geometry for BEM applications must be simplified, abstracted, and approximated (Kota et al., 2016).

Methods and algorithms are constantly being developed to simplify 3D geometry and identify building components for the 3D reconstruction of built environments from dense point clouds and meshes. Reisner-Kollmann (2013) presented 3D reconstruction algorithms that process photographs or point cloud data of built environments into 3D

geometric primitives, which leads to a simple 3D model. Simon (2011) applied procedural modeling for automated reconstruction of buildings components from single-view images. Additionally, Arikan et al., (2013) introduced novel 3D reconstruction pipelines for modeling architecture from point cloud data, which is based on RANSAC plane fitting method, and compared their findings with previous related research. A similar method can be found in 3DReshaper (3DReshaper, 2016a) software, which introduced the Building Extractor functionality in 2015 (3DReshaper, 2016b). Building Extractor enables automatic extraction of 2D planes from dense point cloud data to produce simple 3D geometric models of buildings.

2.3.1 3D Reconstructed Data to Semantic Building Models

Previous work has been conducted to extract semantic information from 3D data of existing buildings, such as floor plans, walls, openings, and furniture. This literature review section will outline two main areas of focus based on the type of 3D data collected; indoor laser scans (interior data), and outdoor or aerial data (exterior).

2.3.1.1 Interior Data

In his dissertation, Turner (2015) presented techniques to extract useful information from interior laser scans of existing buildings. Interior point cloud data was collected using a custom backpack scanning system developed by the Video and Image Processing Lab at UC Berkeley (Turner, 2015). The system employs vertical and horizontal laser scanners, cameras (including IR), and Wi-Fi antennas to conduct a

comprehensive walk-through survey of a building (Turner, 2015). Using the data produced by the system, multiple methods and algorithms were developed to produce 2D floor plans, simple 2.5D extruded models, and detailed 3D mesh models (Turner, 2015). Floor plans were generated by using the data from the horizontal laser scanner which represents a top-down cross section of the building similar to a floor plan (Turner, 2015). Dense areas of the point cloud are assumed to be walls (Turner, 2015). By assuming that the building is 2.5D in nature; meaning that all walls are vertical, a simple 3D model was generated (Turner, 2015). Detailed 3D models were generated by triangulating surfaces from the point cloud and using voxel carving and octree carving methods (Turner, 2015). Identifying interior objects such as furniture was based on intersecting the detailed 3D models with the simple 2.5D models (Turner, 2015). Turner suggested that presented techniques can be applied for building energy simulation. Turner elaborated that instead of relying on manual user input to model geometry for simulation tools such as EnergyPlus, these techniques can be used to automate this process. An example of a building energy model generated using the backpack system was shown in the study (Turner, 2015). To model the windows in the building energy model, window regions were automatically detected using the same backpack system (Turner, 2015), which was based on methods developed in another study (Zhang & Zakhor, 2014). To detect plug loads and occupancy, Turner (2015) proposed classifying objects such as computers with depth map algorithms.

Related research produced a parametric building model from indoor point cloud data (Ochmann et al., 2016) building upon similar concepts and methods discussed in

(Ochmann et al., 2014; Tamke et al., 2014). Ochmann et al. (2016) identified room regions by assuming that each room was scanned separately. The point cloud was segmented and labeled based on that assumption, and vertical planes are detected using a RANSAC plane fitting method to identify the wall candidates (Ochmann et al., 2016). While viewing the wall candidates on a horizontal plane, any edges that separate different room labels are identified as the walls (Ochmann et al., 2016). Openings such as doors and windows were identified by intersecting the reconstructed planar walls with simulated rays from the scan positions to the point cloud (Ochmann et al., 2016). The intersection produces point cloud segments which are further identified as doors or windows based on each segment's features (Ochmann et al., 2016). For example, a segment's feature includes its bounding box width and height, distance from the floor, and proximity to the outside area (Ochmann et al., 2016). Ochmann et al., also presented techniques that allow parametric editing functionality for the walls while maintaining room topology, which is relevant for parametric BIM applications.

This literature review did not cover similar research on 3D reconstructed building interiors from photogrammetry-based methods. According to Furukawa et al. (2009) and Reisner-Kollmann (2013), photographs of architectural interiors contain many texture-poor areas such as walls painted with a single color, causing feature matching algorithms which are used in photogrammetry and discussed in Section 2.1.1 to be not reliable for 3D reconstruction. Feature matching and consequently SfM and MVS methods are often employed in digital photogrammetry software. Additionally, the prototype in this study used data acquired from exterior scenes.

2.3.1.2 Exterior Data

Modeling detailed and realistic building models from aerial data is a relevant area of research in the fields of photogrammetry, remote sensing, and GIS. Kolbe et al., (2005) defined 5 Levels Of Details (LODs) for reconstructed building models. LOD0 is a simple elevation model, LOD1 is an extruded plan model, and LOD2 is an extruded plan model with simplified roof geometry. As for LOD3, it represents a detailed exterior geometry of a building and includes semantic information such as wall, roof, door, and window surfaces. LOD4 is when the model contains interior walls and furniture. Automated building modeling with varying levels of detail is an active research area, and tools and methods used for building 3D reconstruction from aerial data were reviewed in previous literature (Haala & Kada, 2010).

A semi-automatic technique to generate LOD3 models from exterior point clouds was presented by Tutzauer & Haala, (2015) which was built upon previous work (Becker, 2009; Becker & Haala, 2009). This approach assumes the availability of a LOD2 model in addition to a point cloud dataset (Tutzauer & Haala, 2015). Another assumption is that window areas in a laser scan appear as holes (Tutzauer & Haala, 2015). This often due to glass causing laser beams to penetrate through or reflect away from the sensor (Tutzauer & Haala, 2015). Based on this information, a grid of horizontal and vertical edges were placed on the borders of windows where the point cloud terminates (i.e., no further points can be found) (Tutzauer & Haalal, 2015). This cell structure is then aligned with the façade surfaces of a LOD2 model (Tutzauer & Haala, 2015). To identify the cells that represent façade walls or windows, a black-and-

white reference image is generated from the point cloud and mapped on the cells, and cells were classified based on pixel density (Tutzauer & Haala, 2015). Classified window cells were added as geometry to the LOD2 model to generate a CityGML LOD3 model (Tutzauer & Haala, 2015). Tutzauer and Haala proposed an alternative method when using a photogrammetry-based point cloud. This method assumes that façade and window areas each have unique and homogenous color values (Tutzauer & Haala, 2015). Dual-color reference images were presented to follow a similar method to the one detailed above (Tutzauer & Haala, 2015).

A method to model street façade geometry from SfM-based photogrammetry point clouds was presented by Xiao et al. (2008) and Xiao et al. (2009). The approach is based on using street-view orthographic projections of a point cloud to generate regularized depth maps (Xiao et al., 2009). The depth maps, including color information, are used to extrude rectangular geometry to add more details (Xiao et al., 2009). This method can be compared to a previous procedural method of modeling facades from a single orthographic image presented by Müller et al. (2007). Recent work improved upon this procedural approach by generating semantic 2D and 3D building façade elements from single-view images (Simon, 2011).

A recent study proposed utilizing the texture map that is generated with a 3D photogrammetry-based model to extract semantic information of buildings, such as roof, wall, and window surfaces (Frommholz et al., 2015). The study suggested the use of OBIA software to apply image segmentation algorithms on a 3D model's texture map to classify windows (Frommholz et al. 2015). The contours of labeled window pixels are

then converted into polygons that are applied to the 3D model (Frommholz et al. 2015). The texture patches on the texture map were assumed to be unrectified, with their positions calculated by the software (Frommholz et al. 2015).

In Chapter III, this study will present a similar approach to Frommholz et al. by using an annotated texture map to extract wall, roof, and opening surfaces from a texture map, then re-apply the annotations on the 3D building model.

2.3.2 3D Reconstructed Data to BIM

Due to the developments in the field of 3D building reconstruction as discussed previously, there is an avenue to utilize methods of extracting semantic information for BIM. In addition to proprietary point cloud-to-BIM software, there is recent research interest to provide openly accessible links between 3D reconstruction methods and architectural design tools such as BIM (Zwierzycki et al., 2016). Volvox (Evers & Zwierzycki, 2016) is a free plugin developed for Grasshopper which enables users to extend its functionality to reconstruct point cloud datasets. The EU funded DURAARK project (DURAARK, 2017) implemented a parametric association between point clouds and BIM elements using Volvox (CITA, 2017a, 2017b). This enables an automatic workflow to generate architectural geometries (Ochmann et al., 2016).

BIM is beyond the scope of this study, as the implemented prototype focuses on establishing a link between 3D reconstructed geometry and BEM. Converting 3D reconstructed geometry into BIM will be future work.

2.3.3 3D Reconstructed Data to BEM

A methodology to automatically create a building energy model using a custom laser scanning system and computer algorithms was proposed by C. Wang (2014). The scanning system is composed of a laser scanner in addition to an IR camera in order to generate point clouds infused with thermal data (C. Wang, 2014). For example, in addition to point coordinates (x, y, z) and color values (RGB), a generated dataset would include temperature values collected from the IR camera (C. Wang, 2014). The methodology presented by C. Wang to generate a building energy model started with applying a region growing plane segmentation algorithm to segment the dataset into groups of points that lie on the same plane. A boundary detection algorithm is then applied to generate the outer and inner boundary edges of each segmented point group (C. Wang, 2014). The outer boundary edges represent the wall and roof elements (C. Wang, 2014). Since window areas appear as holes in the point cloud, the inner boundary edges represent window elements (C. Wang, 2014). Surfaces were generated from the boundaries using a concave hull algorithm (C. Wang, 2014). Gaps between the surfaces are closed by a geometry size fitting algorithm that extends surfaces and replaces the edges with the intersections (C. Wang, 2014). C. Wang proposed a number of classification rules to identify different building elements from the surfaces. Based on the proposed rules, the ID and geometry data of the components were saved as a text file for inputs into a gbXML file format (C. Wang, 2014). The methodology was validated by successfully opening the file in Autodesk Ecotect (C. Wang, 2014). Energy

simulations were not conducted using the software and were not part of the scope (C. Wang, 2014).

The literature review identified a limited number of research projects in the area of photo-to-BEM or 3D reconstructed data to BEM. Only two previous studies touched upon the specific topic of automatically converting 3D reconstructed data to BEM (Turner, 2015; C. Wang, 2014). While C. Wang (2014) proposed a methodology to convert a point cloud to a building energy model, the resulting gbXML model was not verified by running a successful energy simulation. Also, Turner (2015) only discussed the application of his methodology in the area of BEM. Furthermore, both approaches by C. Wang (2014) and Turner (2015) are based on highly customized laser scanning hardware and did not address available photogrammetry-based methods. Thus, an area of interest has been identified for further research.

CHAPTER III

BUILDING INFORMATION EXTRACTION FROM TEXTURED MESHES

A 3D mesh that is generated from currently available photogrammetry software does not contain semantics that are relevant to buildings. A mesh is a simple representation of 3D geometry with limited information; there are no boundary conditions, building element descriptions, or material information. Furthermore, 3D reconstruction as a data acquisition phase of an existing building would contain ambiguous data (Prabhu, 2010). While photogrammetry is a potentially highly accurate measurement technique, the 3D geometry represented in the generated mesh data often contradicts other sources of data such as existing architectural drawings or empirical site measurements, or even other photogrammetry methods. This is due to limitations from the software or mistakes by the user. These issues pose challenges to translate photogrammetry-based mesh data as inputs for BEM.

This chapter covers the development of input data that was used for the prototype presented in Chapter IV. An example building was selected as a case study for 3D reconstruction. Afterwards, the 3D mesh data was generated using a photogrammetry software. A methodology was proposed to extract building elements (semantic data). Finally, the proposed methodology was tested using visual programming as a proof-of-concept.

3.1 Field Survey

The case study that was selected for 3D reconstruction is a 1-storey farmhouse building located in Al Wafra farm area in Kuwait, which is shown in Figure 3. The farmhouse was selected for two main reasons: The first is that there were no obstructions blocking the view towards the building in all directions. This is important so that the software would be able to reconstruct the 3D mesh that will represent the building as accurately as possible with no missing features due to camera obstructions. The second reason is that building form is relatively simple and box-shaped, and lacks any large curved surfaces. For this study, the prototype program will focus on this typology of buildings.

Figure 3. Photograph of the farmhouse taken with a Phantom 3 UAV drone.

3.1.1 Data Acquisition Using UAV

The photo data collection method used is a remotely-controlled Phantom 3 UAV drone fitted with an HD camera. Aerial photographs were taken at three time periods. The first batch was photographed at 2:15 PM totaling to 6 photos. On 4:15 PM, 23 photographs were taken at a similar oblique angle to the first batch. The final batch was 82 photos taken on 5:00 PM with similar angles to the previous batches in addition to both higher and lower oblique angles. The UAV was set on an automatic point-of-interest flight path, which enables the drone to fly continuously in a fixed circular path while facing the building. A total of 111 photographs were taken manually by the author at arbitrary intervals.

3.1.2 3D Reconstruction from Photo Data

The photographs were uploaded to ReCap 360's photo-to-3D service, which is a cloud-based photogrammetric 3D reconstruction solution provided by Autodesk. Cloud computing is a convenient way to take advantage of photogrammetry software since they require high-end computers and processing power for high-fidelity results. Figure 4 shows the automatically generated 3D mesh in ReCap 360 after uploading the photographs.

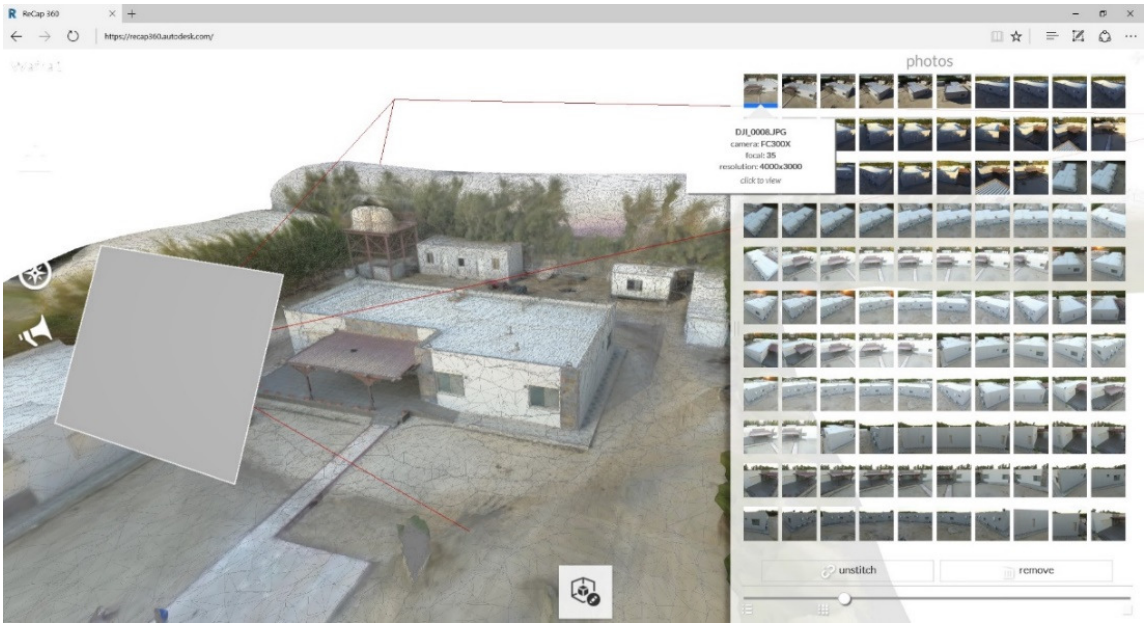


Figure 4. The reconstructed 3D mesh in ReCap 360.

Further tests have been conducted using other photogrammetry software, namely, Photoscan and Remake. The results from Photoscan yielded similar albeit lower quality results. Results from Remake were identical to ReCap 360, which suggests that the same algorithms were being used. One advantage Photoscan has over the other tested software is that it can export colored point cloud files in addition to textured 3D meshes. Figure 5 shows the uploaded photographs and generated mesh in Photoscan. The blue patches represent the reconstructed camera parameters. The 3D reconstruction process in Photoscan was run locally on a laptop computer as opposed to cloud computing.

Figure 5. The reconstructed 3D mesh in Photoscan.

The 3D model that was selected as an input for the prototype was the 3D mesh generated from ReCap 360 from Figure 4 because it has better quality geometry and texture. Furthermore, Remake was used to view, edit, and export the mesh. Figure 6 shows the selected 3D mesh in Remake.

Figure 6. The 3D mesh in Remake.

3.1.2.1 Photo-to-3D Inaccuracies

Accurate 3D reconstruction using digital photogrammetry software relies heavily on the quantity and quality of the input data. Furthermore, the photogrammetry method is not a completely accurate empirical measurement method, so any limitations in the photogrammetry software or user errors would be reflected in the mesh.

Figure 7 shows a distorted part of the mesh that does not represent the actual building. During the data acquisition phase, the photographs were taken at an oblique aerial view angle, there were no photographs that clearly show the area beneath the overhang. Since this part is obstructed in view by the building's overhang, consequently, the software was not able to reconstruct this geometry properly.

Figure 7. Distorted mesh beneath the overhang.

Furthermore, some holes can be identified on the output mesh that do not exist in the actual building. One example of these holes can be shown in Figure 8. These inaccuracies were taken into consideration during the development of the prototype.

Figure 8. Mesh holes in the 3D reconstructed geometry.

Scaling the 3D geometry properly in the world coordinate system is another issue. The mesh must be scaled to match actual measurements. The walls of the 3D mesh measured around 4 meters high (around 13 feet) in Remake and Rhinoceros, which is relatively accurate when compared to the actual building. Manual scaling and georeferencing tools are often provided in photogrammetry software. 4-meter-high walls are reasonable for this study; therefore, they were not manually scaled. Otherwise, they would be manually scaled.

3.1.2.2 Mesh Decimation

Mesh decimation is a process in which an algorithm is applied to reduce the number of faces in a mesh while preserving the overall shape of the mesh. An

application of this procedure is to simplify a mesh for faster computation performance, or to overcome the limitations of computer hardware when handling 3D geometry.

Figure 9. Undecimated mesh.

The output mesh was decimated using the decimation tool provided in Remake to reduce the mesh face-count from 1,536,739 faces to 37,919 faces. This is not considered a necessary step for using the mesh in the prototype, but is recommended for faster computing speed.

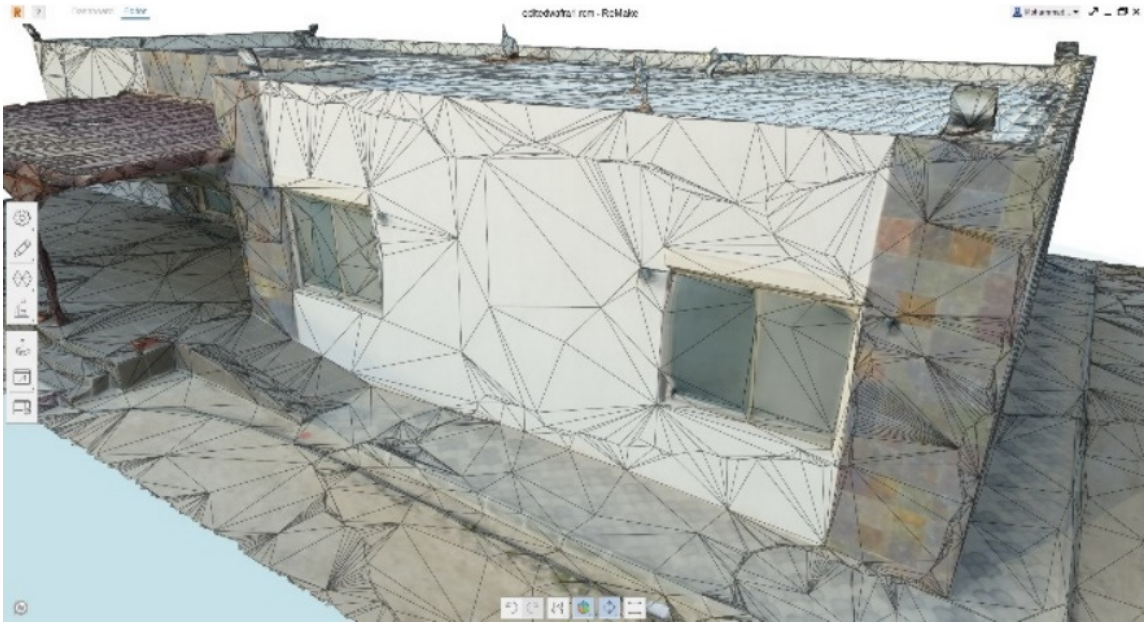
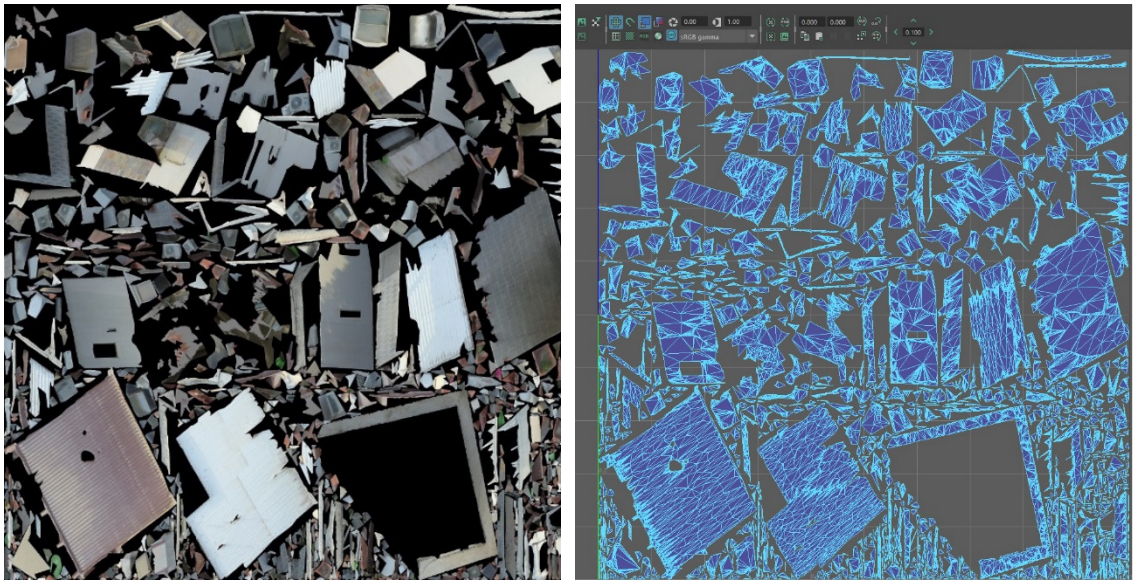


Figure 10. Mesh after decimation.

3.2 Texture Map Analysis

The selected mesh was saved in OBJ format. Figure 11 (a) shows the texture map that was saved with the OBJ file. The 3D mesh of the building was unwrapped to match the texture map image as shown in Figure 11 (b). This is a UV map which results from converting the mesh vertices' 3D coordinates to their respective UV coordinates. At first glance, a human viewer cannot easily distinguish the different elements of the building in a disorganized texture map that was generated using software. This is further exacerbated by the fact that the computer only identifies the map as a raster image with no semantic labeling or classification.



(a) (b)

Figure 11. Unrectified texture maps. (a) The texture map image generated with the 3D mesh. (b) The UV mapping of the unwrapped mesh.

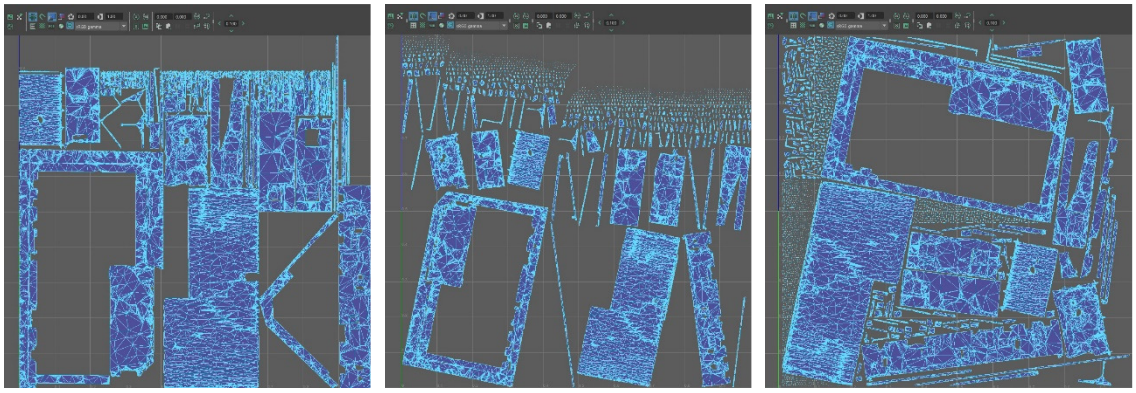
If a photograph or 3D model of a building can be recognized through human knowledge, then we can teach a computer to do the same (Pu et al., 2006). Pu et al., (2006) argued that since humans recognize a wall as both vertical and planar, a window as a smaller plane within a wall, and a roof as a horizontal element, then this human knowledge can be modeled in a machine. Pu et al. presented a number of geometry constraints and rules to extract wall, window, and roof features from point clouds.

Similarly, this study proposes a methodology to extract semantic building information; namely wall, window, and roof elements, from a rectified texture map based on human knowledge and assumptions.

3.2.1 Texture Map Rectification

A texture map can be rectified automatically with adequate results using an appropriate 3D modeling software tool. This process starts by recalculating the UV coordinates of a mesh based on unwrapping algorithms such as least-squares conformal maps (LSCM) (Lévy et al., 2002), or angle-based methods. This results in a new UV map for a 3D mesh. The texture map can then be reorganized in the same layout as the rectified UV maps in a subsequent process known as texture baking. This is a method that is often employed in video game development to transfer textures from a detailed 3D model to a simpler one. Texture baking is based on aligning two models then casting rays from the first model to the second to record the texture detail on the second model, while using the first model's UV coordinates.

In this study, UV map rectification was tested on the 3D mesh using Blender (Blender Foundation, 2016), Autodesk Maya (Maya, 2016), and Modo (Modo, 2016). All software generated similar results that only differ in the organization of the unwrapped mesh as shown in Figure 12. The mesh with the new UV mapping was saved as a separate file. A rectified texture map was created by aligning the unrectified 3D mesh with the rectified mesh and baking the texture from the original mesh to the new mesh.



(a) (b) (c)

Figure 12. Results of UV map rectification using various software. (a) Blender. (b) Maya. (c) Modo (selected result).

Figure 13 shows a script developed by the author in Grasshopper that unwraps the resulting OBJ mesh (green color) and overlays it over the rectified texture map image.

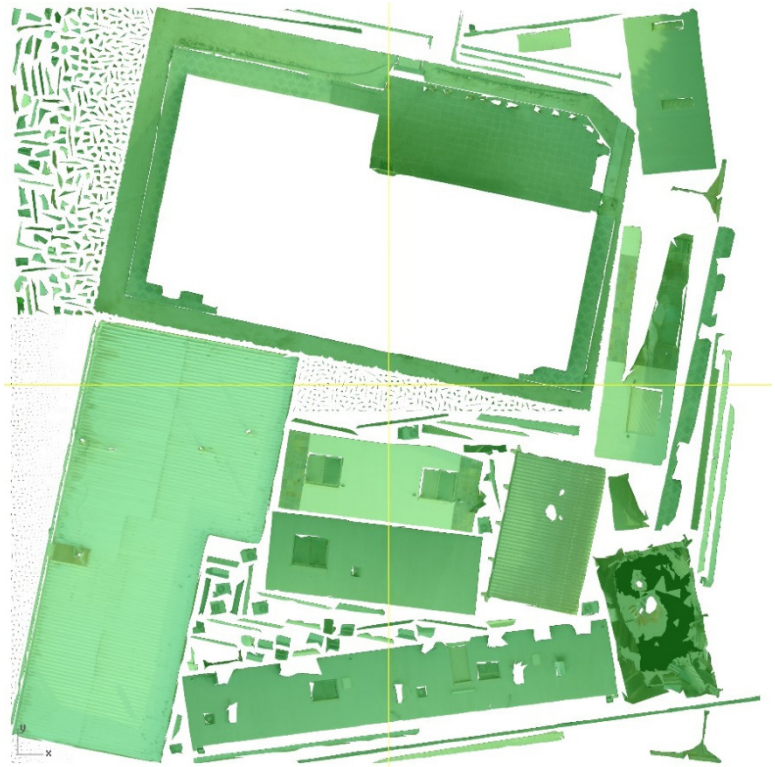


Figure 13. 3D mesh with rectified UV mapping unwrapped over the new texture map.

3.2.2 Semantic Annotation of Images

A rectified texture map such as the one shown in Figure 13 does provide a better foundation to start extracting the components of the building. By comparing the map with the original existing building, a human viewer can identify most of the walls, the windows, and the roof.

The proposed approach to extract building semantics in this study is to annotate texture map directly based on an approach similar to the one proposed by Frommholz et al., (2015), which this study identifies as a unique method. Additionally, the following is an evaluation of different methods used to extract building semantics:

3.2.2.1 Histogram Analysis

Lee and Nevatia (2004) presented a technique to identify windows from a rectified ground-view façade texture by computing edge histograms. The technique exploits the regularity of most building façade patterns and projects two profiles (horizontal, and vertical) on façade images, and the peaks in the histogram from edge pattern repetitions are intersected for window candidate locations (Lee & Nevatia, 2004). More recent work followed a similar approach (Haugeard et al., 2009; Vračar et al., 2016). Building upon the work done by Becker and Haala (2009) referenced in Chapter III and Lee and Nevatia (2004), R. Wang et al., (2012) identified window candidates from LIDAR point clouds by the histogram analysis of window points. As far as the current knowledge suggests, histogram analysis is limited to planar facades with rectilinear windows. Also, histogram analysis methods assume the availability of a ground-view projection (2D elevation) of a building façade. A projection is either manually provided or automatically extracted with software. For example, R. Wang et al., (2012) used surface normal computation to identify potential street-facing facades from 3D point clouds. In comparison to ground-view orthogonal projections, texture maps represent 2D map projections of a 3D object. Due to this difference, histogram analysis might not be useful on the rectified texture map.

3.2.2.2 Template Matching

Nguatem et al., (2014) proposed a way to search for windows and doors from a point cloud dataset based on a Monte Carlo simulation search routine. Templates of

control points and curves representing windows and doors can be pre-defined (Nguatem et al., 2014). The hypothesis was based on a plane sweep through the point cloud that reveals window and door outlines, where probabilistic template matching can be used to place the templates on the façade (Nguatem et al.). The proposed method enables the extraction of arc-shaped windows and doors in addition to rectilinear ones (Nguatem et al.). This method for window and door extraction is solely based on the 3D geometry and does not use image data, and thus not applicable on a texture map.

3.2.2.3 Pixel Analysis

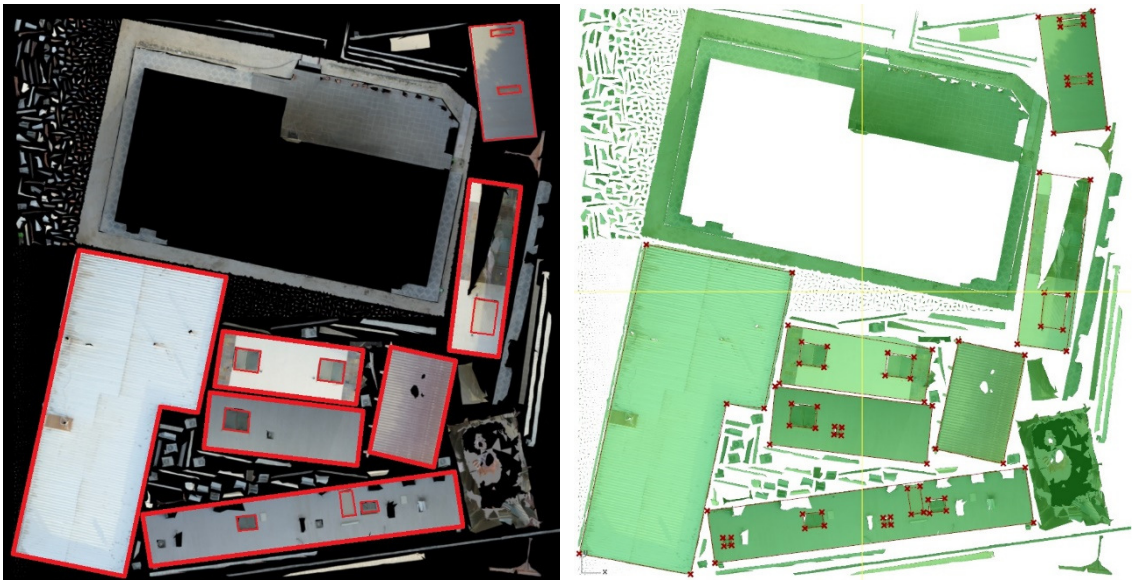
In the field of computer vision, labeling pictures of building facades is research problem that received active research attention in recent years (Sprengel, 2014). There has been a number of attempts to apply pixel-based image analysis methods to extract building semantics from photographs. Jahangiri and Petrou (2008) used blob detection algorithms to identify parts of building facades other than the walls. Furthermore, canny edge detection and watershed segmentation algorithms were used to label a building facade image into groups that represent walls and/or windows (girish_m & user667804, 2017; Sprengel, 2014). These labels can be used for further classification. In another approach, Larsen and Marsh (2010) identified the three largest pixel clusters on the texture of a simple 3D building model. Based on the assumption that walls are the largest surfaces in a building, the largest pixel cluster was classified as walls, and the other pixel clusters as openings (i.e., windows and doors) (Larsen, 2010). For more specific labeling and classification, Rafeek (2008) studied the advantages and

disadvantages of OBIA to segment a building image into wall, window, door, and roof elements. Using OBIA to semantically annotate texture maps was similarly proposed by Frommholz et al. (2015) as referenced in Chapter II.

3.2.2.4 Suggested Method

This study suggested a pixel-based image analysis approach to annotate building features on the texture map. Assuming the wall, roof, and more specifically the window pixel areas are classified, then these annotations can be reapplied on the 3D geometry using barycentric interpolation. A contour extraction algorithm can be applied on the texture map to find potential object boundaries (David, 2017). The contours can then be converted to polyline boundaries.

In this study, the annotations were applied manually by the author. Figure 14 shows polylines drawn manually over the texture map to represent this hypothesis. For the prototype in Chapter IV, only the window polylines are relevant, since other methods were developed to extract wall and roof elements.



(a) (b)

Figure 14. Semantically annotated texture maps. (a) Hypothesis of extracted contours manually drawn on the texture map. (b) Implementation of polyline contours for barycentric interpolation.

3.2.3 UV to XYZ Coordinates Using Barycentric Interpolation

Once the general polyline definitions of building components are annotated on the 2D texture map, theoretically, based on the concept of texture UV mapping, the corresponding locations of the annotations can be found in 3D space on the mesh. The locations of the corner points of the annotated polylines are assumed to be defined on the texture map image, so the corner points are defined as pixel coordinates which can be easily converted to a UV coordinate format. Since the mesh is in OBJ file format, it should contain a list of (1) the vertices as Cartesian values, (2) texture UV values for each vertex, and the (3) mesh face definitions all in text format. But generally, an arbitrary point on the texture map usually does not translate directly to a vertex on the

mesh; instead an annotated corner point could be somewhere inside a face. This raises a challenge to convert the arbitrary (or manually drawn) 2D polylines to 3D geometry corresponding to their locations on the mesh. In order to solve this problem, the texture map must contain no overlaps, and overlaps are common in many textured 3D models. For example, a single pixel location on the texture map does not necessarily wrap on a single point on the mesh, but the pixel's location might be used for multiple points on the mesh based on multiple UV coordinates for a single vertex defined in an OBJ file. Photogrammetry meshes have no overlaps, so this problem can be solved through calculating barycentric coordinates.

Barycentric coordinates in the context of triangles can be defined as a location of a point relative to a triangle. The coordinates of a point *p* can be defined relative to the triangle *abc* as: $p\ (\lambda_a, \lambda_b, \lambda_c)$. The variable λ is a ratio of the area of the triangle relative to the point *p*. For a point that lies inside a triangle, the following equation is applied: $\Sigma\lambda = 1$. Consequently, any barycentric coordinate or λ value that is negative means that the point *p* is located outside of the triangle.

In the context of the unwrapped mesh and the annotated texture map which was illustrated in Figure 14 (b), each corner point of the annotations is inscribed inside a triangular face within the unwrapped 2D mesh as shown in Figure 15. To convert an arbitrary 2D texture coordinate to its 3D position on the 3D mesh, the mesh face that contains that 2D coordinate must be identified, then the barycentric coordinates of that point relative to the mesh face can be calculated (Figure 16). The point *p* in the Figure 16 was created by user annotation.

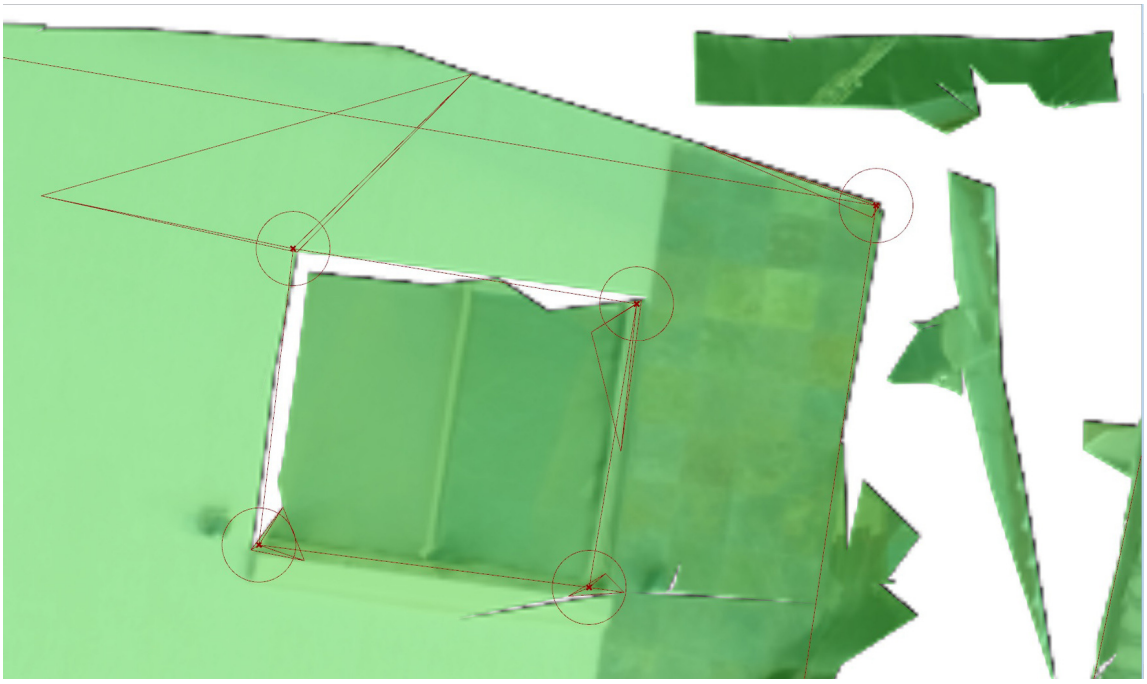


Figure 15. An example of a window contour polyline. The polyline edge points are highlighted by circles. The triangles represent the mesh faces.

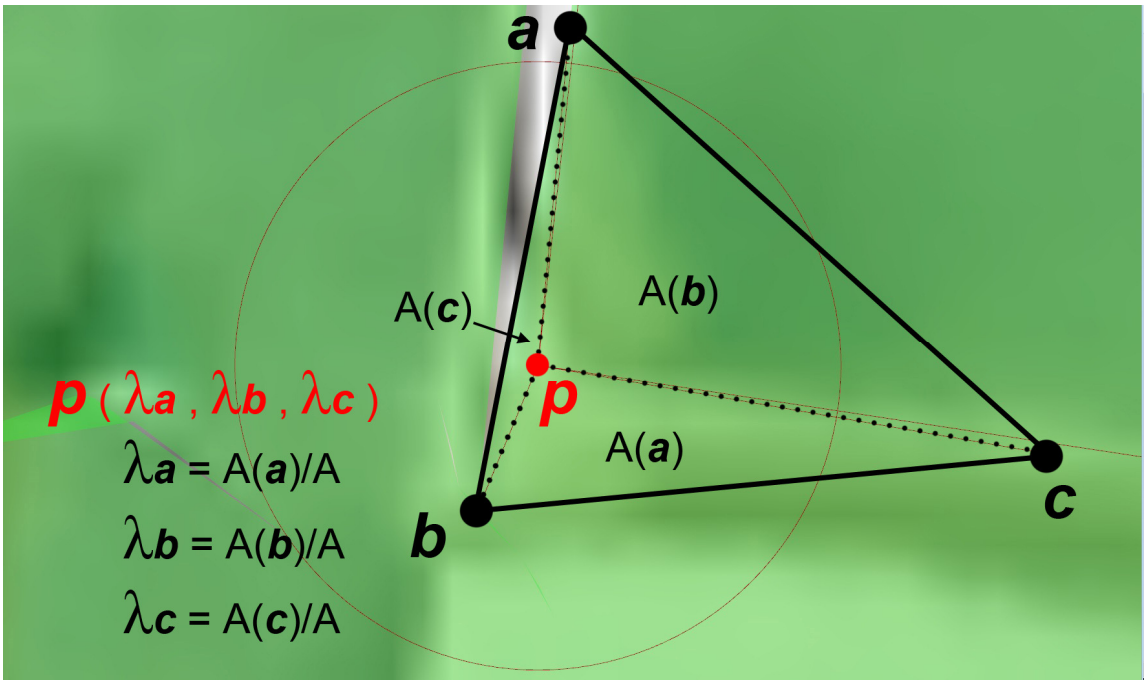


Figure 16. Barycentric coordinate calculation. The point *p* represents a polyline corner point, while the triangle *abc* represents a mesh face.

Once the barycentric coordinates of the arbitrary 2D texture coordinates relative to the corresponding 3D faces are calculated, the annotated 2D polylines can be drawn on their exact 3D locations on the mesh. A proof-of-concept program was built that takes the polylines shown in Figure 14 (b) and parses the mesh to find the corresponding 3D points for the 2D polylines' corner points, and then create a 3D model as shown in Figure 17.

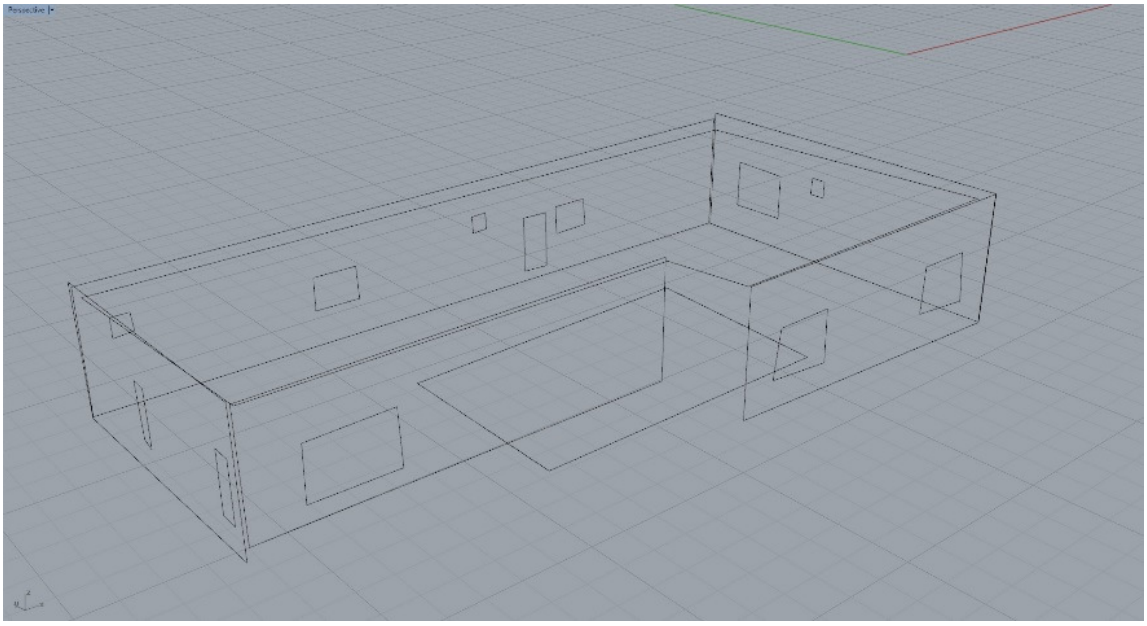


Figure 17. Proof-of-concept program showing the interpolated 3D polylines.

3.3 Building Information Extraction

This study assumes that the polylines have no semantic labeling. Meaning they are simply boundary contours with no classification on which contour is a wall, window, or roof. However, through human knowledge, the building's geometry semantics can be extracted. By assuming that all building surfaces are planar, and walls are vertical, windows are vertical but smaller in area, and that roofs are horizontal, these assumptions can be programmed in a computer.

The model shown in Figure 17 was converted to a surface model. The surfaces were made planar by using a least-squares plane fitting algorithm. Based on the areas and normal vectors of each building surface, and by assigning threshold values to them, the different components of the building can be labeled automatically. A proof-of-

concept for of this methodology was developed using Grasshopper, and is shown in Figure 18 and Figure 19.

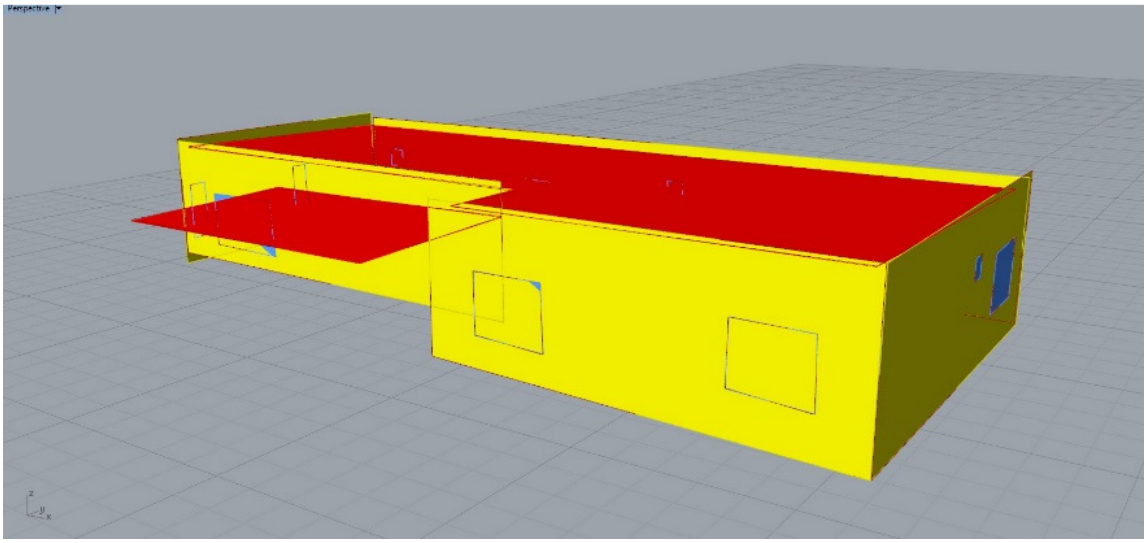


Figure 18. Proof-of-concept program showing a 3D building model with automatically labeled wall (yellow), roof (red), and window (blue) surfaces.

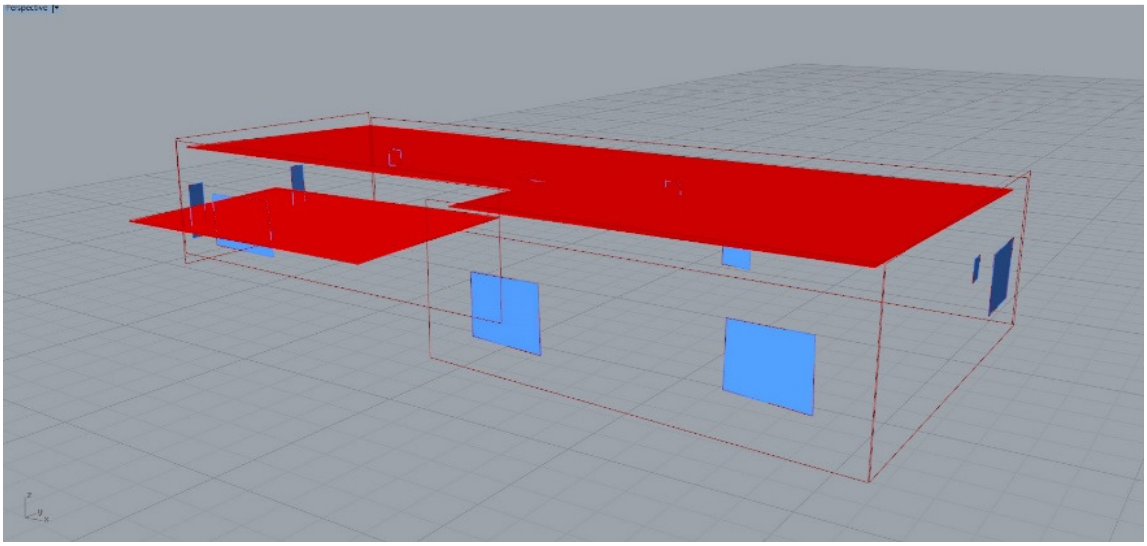


Figure 19. Hidden wall surfaces revealing the window surfaces.

CHAPTER IV BUILDING ENERGY MODEL CONSTRUCTION

This chapter will introduce the methodology of the developed prototype that generates a building energy model from a photogrammetry-based mesh of an existing building. The outline of the prototype's overall framework can be seen in Figure 20. There are two main inputs within this framework; a 3D mesh generated from a photogrammetry software, and an annotated texture map such as the one discussed in Chapter III. The output of this framework is a building energy model that can be verified through running a successful energy simulation.

The prototype proposes a number of integrated processes and algorithms that reconstructs the input mesh automatically into a geometric model that is compatible with building energy analysis tools. Grasshopper, which is a visual programming language developed for Rhinoceros, was used to develop the prototype.

In the case of this study, the output energy model is an EnergyPlus IDF file, including a 3D representation of the energy model. EnergyPlus was chosen due to the available integration between Grasshopper and EnergyPlus via the Ladybug and Honeybee plugins developed by Mostapha Sadeghipour Roudsari. Ladybug and Honeybee were developed as open-source plugins for Grasshopper to provide an environmental design toolset for architects and engineers. The plugins provide integration with simulation tools such as EnergyPlus for energy analysis and Radiance (LBNL & Ward, 2014) for daylighting analysis.

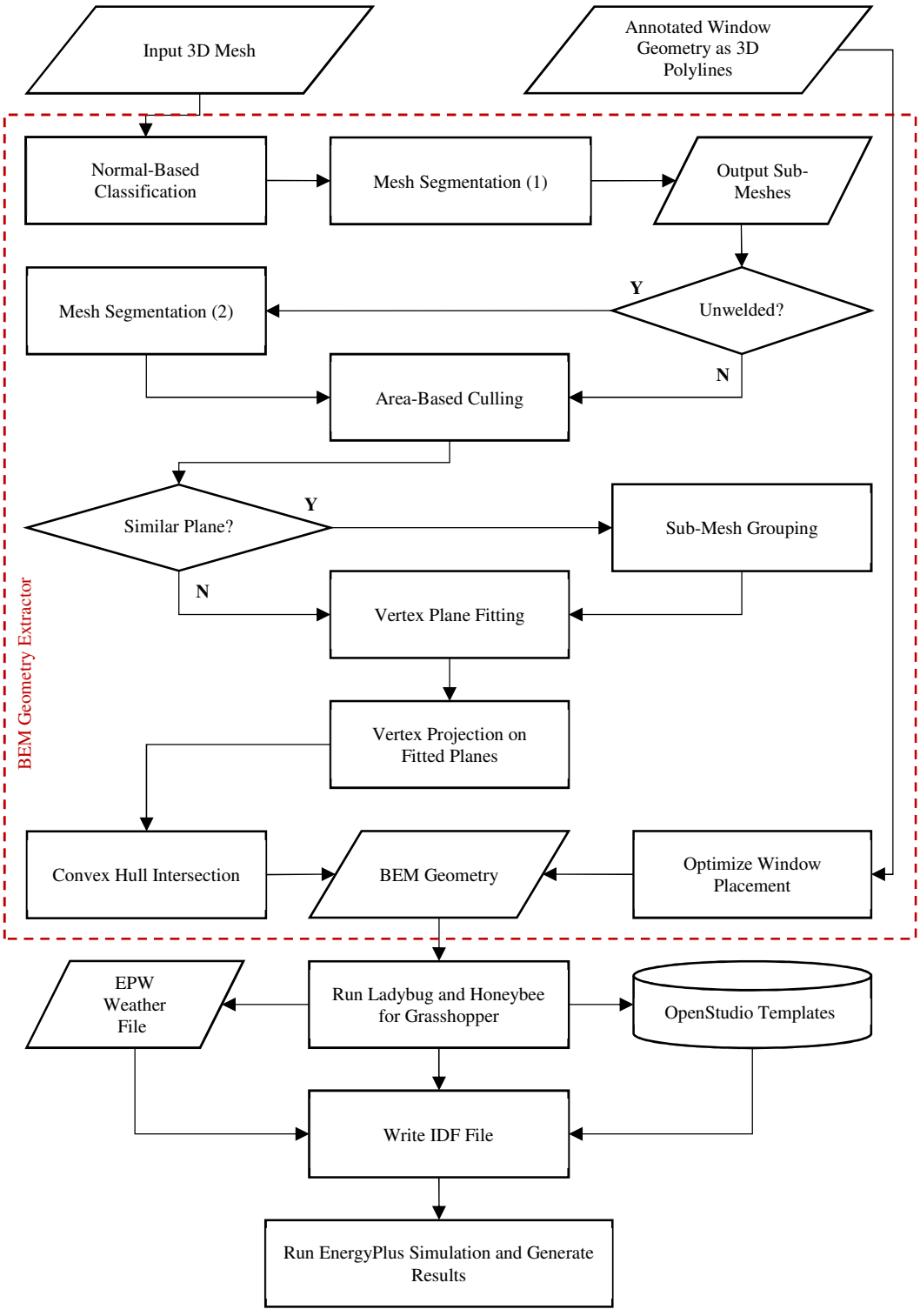


Figure 20. Prototype framework.

4.1 BEM Geometry Limitations

BEM and simulation engines such as EnergyPlus have certain limitations when handling geometry. For EnergyPlus, any 3D geometry that represents a thermal zone must contain a closed volume or shell (Kota et al., 2016). This poses a challenge when working with an output mesh from a photogrammetry software as these types of geometry are considered an 'open' mesh surface as opposed to 'closed' geometry such as a boundary representation (BREP) model, and inaccuracies such the mesh holes discussed in Chapter III further compounds this issue. Also, all of the surfaces in the geometry of an EnergyPlus model must be planar surfaces; curved geometry is not supported. While the individual faces of a triangulated mesh are planar, too many planar surfaces in EnergyPlus would cause slowdown issues and drastically increase the processing time. For faster simulation performance, it would be preferable to have the minimum number of planar surfaces that most accurately represent building surfaces. For example, a curved wall could be represented as a series of planar segments in order to be compatible with EnergyPlus, however, it is still possible to represent the same wall as a single planar surface without impacting the simulation results. Additionally, when using OpenStudio to model 3D EnergyPlus Geometry, any 'child' surface, such as a window contained within a certain wall element, must be coplanar to the 'parent' wall. Further limitations of EnergyPlus include the geometry of the windows, as only triangular and rectangular windows are supported; four-sided non-rectangular windows would cause the simulation to return an error.

When taking these BEM geometry limitations into consideration, it becomes clear that unprocessed geometry from digital photogrammetry would introduce many issues. The processes and algorithms of the prototype as illustrated in the overall framework were designed to overcome these issues, and the experimentation results will be introduced in the following sections.

4.2 Segmenting Mesh Based on Normal Vectors

Since the 3D mesh is based on a real-world physical object, it would not be robust enough to be used as an energy model. For example, any irregularities and imperfections in the actual building surfaces would be represented in the triangulated mesh. While each individual polygon in the mesh would be planar, the group of polygons representing the wall would be a group of polygonal faces that do not share the same normal vector orientation. Therefore, a solution must be found to extract the building's walls and roofs as individual surfaces for each relative direction.

Each face of the mesh has a normal vector with three component values in the Cartesian coordinate system between -1 and 1. Based on this data, to simplify the mesh for BEM, surfaces of the building that share a similar orientation in the Cartesian coordinate system can be categorized. For example, mesh faces that represent walls and windows that face a similar direction can be a category. The same can be said about roof surfaces that share a similar normal vector orientation.

In order to identify each category based on similarity, the prototype introduces a rounding function to round the normal vector values of the faces to the closest vector of

integers, which would result in the Cartesian values of -1, 0, and 1. Furthermore, the normal vector values were divided by a variable of 0.5 before applying the rounding function and multiplied by the same value after applying the function. This would produce more possible categories that represent more orientations since the Cartesian values that will be categorized are -1, -0.5, 0, 0.5, and 1 for each of *x*, *y*, and *z* component in the normal vector values. To further illustrate, Figure 21 shows the possible normal vector directions in the positive Cartesian direction after applying a rounding function to the closest positive integer, and comparing it to the possible directions after applying the variable of 0.5.

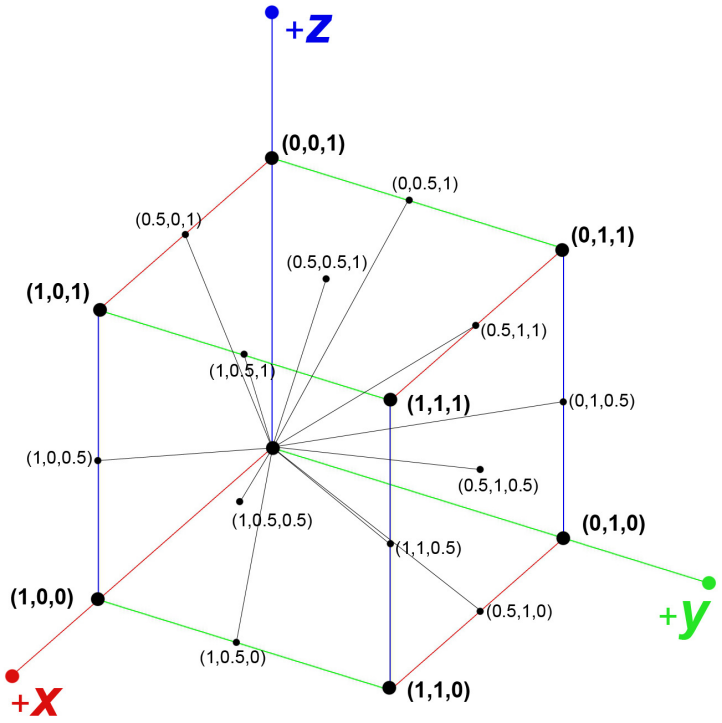


Figure 21. Possible vector directions in the positive Cartesian direction.

After this process was applied to the normal vector values of all the faces in the input mesh, a mathematical finite set function is applied to group all the unique normal vector values into a category. This categorization was used to segment the mesh into sub-meshes based on these normal vector categories. The total number of the sub-meshes produced from the segmentation algorithm is 74. The normal-based segmentation approach was designed to take into account buildings with gable roofs that have varying slopes.

4.3 Segmenting Unwelded Sub-Meshes and Culling

Segmenting the mesh into sub-meshes based on the normal vector direction is not sufficient to categorize building elements. The resulting categorization is that any surfaces facing a similar direction would be categorized as one sub-mesh group. Figure 22 (a) illustrates two wall surfaces in one category.

The program introduces an algorithm to separate unwelded meshes in each category into new categories (Mans, 2016). An unwelded mesh is a mesh that is not connected in any vertex with another mesh. Applying the algorithm to the mesh would further segment the wall meshes showed previously in Figure 22 (a) into separate categories as shown in Figure 22 (b) and (c).

Additionally, a culling function is applied to cull (remove) any sub-mesh categories that are smaller than a specified area threshold. This threshold can be adjusted. This will remove any small meshes that can be neglected or might cause slowdowns in processing time.

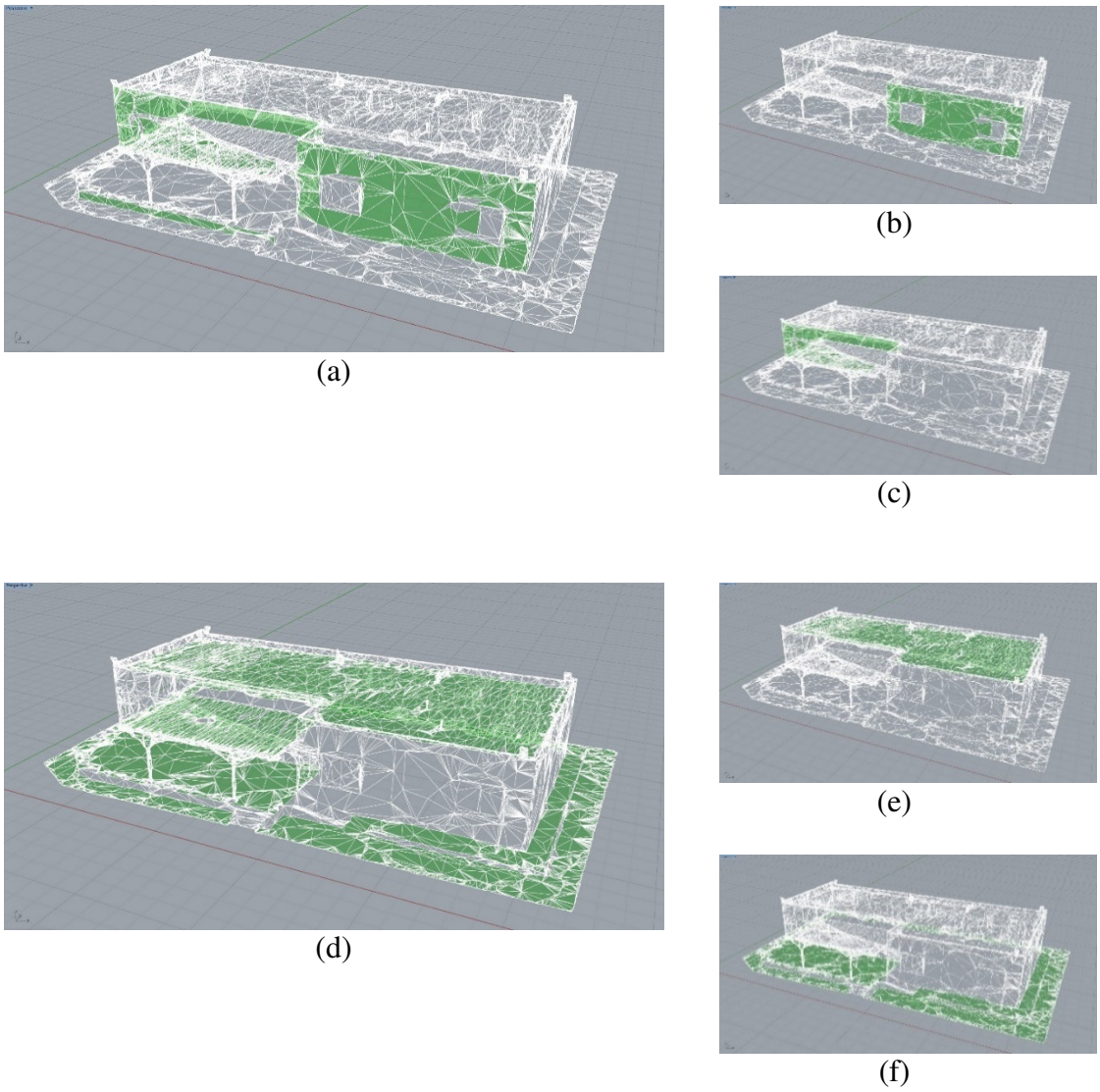


Figure 22. Results of mesh segmentation. Normal-based segmentation results in classifications such as (a) and (d). By segmenting unwelded meshes, (a) is segmented to (b) and (c), and (d) to (e) and (f).

4.4 Grouping Sub-Meshes on a Similar Plane

The mesh segmentation processes do not resolve the issue of unwelded meshes that actually lie on a similar plane. Figure 23 shows of an example of two wall segments that would not share the same category after applying segmentation based on normal

vector direction and segmenting unwelded meshes. In order to categorize the two segments as one sub-mesh that represents the same wall, an algorithm is introduced to group sub-meshes that exist on a similar plane. This is done by calculating the least-square plane for each vertex group in each sub-mesh, then by introducing the variable *d*, whereas planes with vertical distances between each other equal to or smaller than *d* would be grouped together. Since each plane represents a sub-mesh, similarly, the sub-meshes that exist on a similar plane will be grouped together.

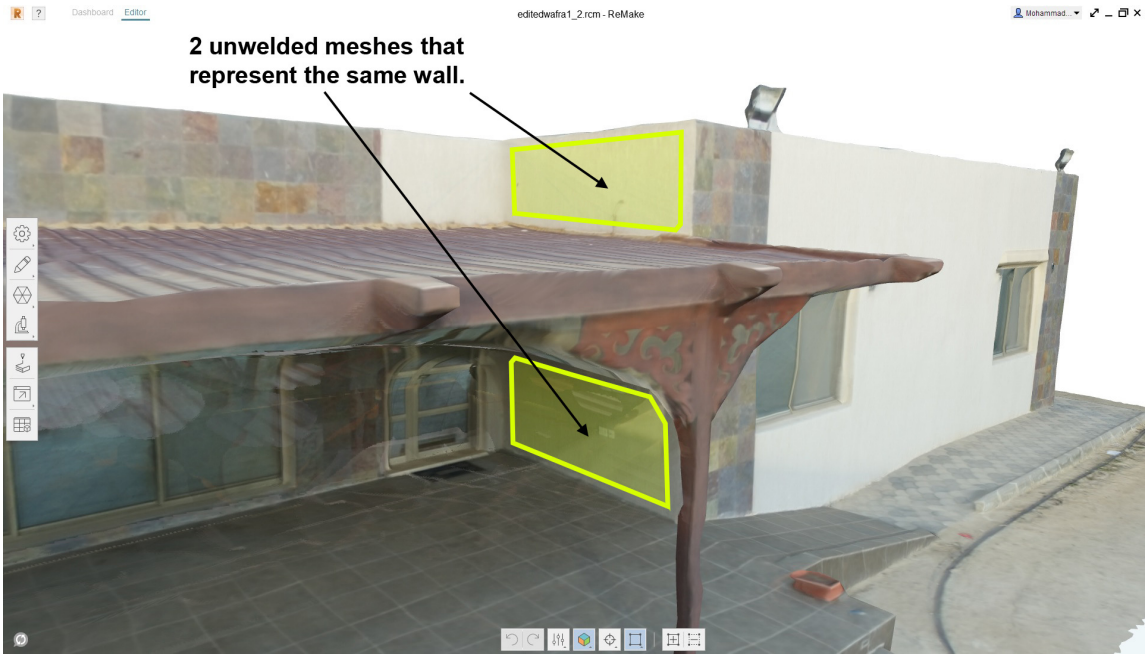


Figure 23. Previous segmentation algorithms do not resolve the case of two unwelded meshes that might represent the same wall.

4.5 Converting Mesh to BEM-Compatible Geometry

A technique is presented based on the constraints explained in Section 4.1. After separating the mesh segments into groups, a least-square plane fitting algorithm is

applied on the vertices in each group. The vertices are then projected on the fitted planes. This will result in ‘flattening’ the vertex groups so that each group of vertex points lies on the same plane. Subsequently, a convex hull algorithm is applied for each group. This will result in a group of planar surfaces that can be intersected using a boundary volume algorithm. The resulting convex surfaces were scaled by a factor of 2. This is to make sure that these surfaces contain a volume within a closed boundary, and avoid gaps. The closed BREP geometry of that volume was computed as shown in Figure 24.

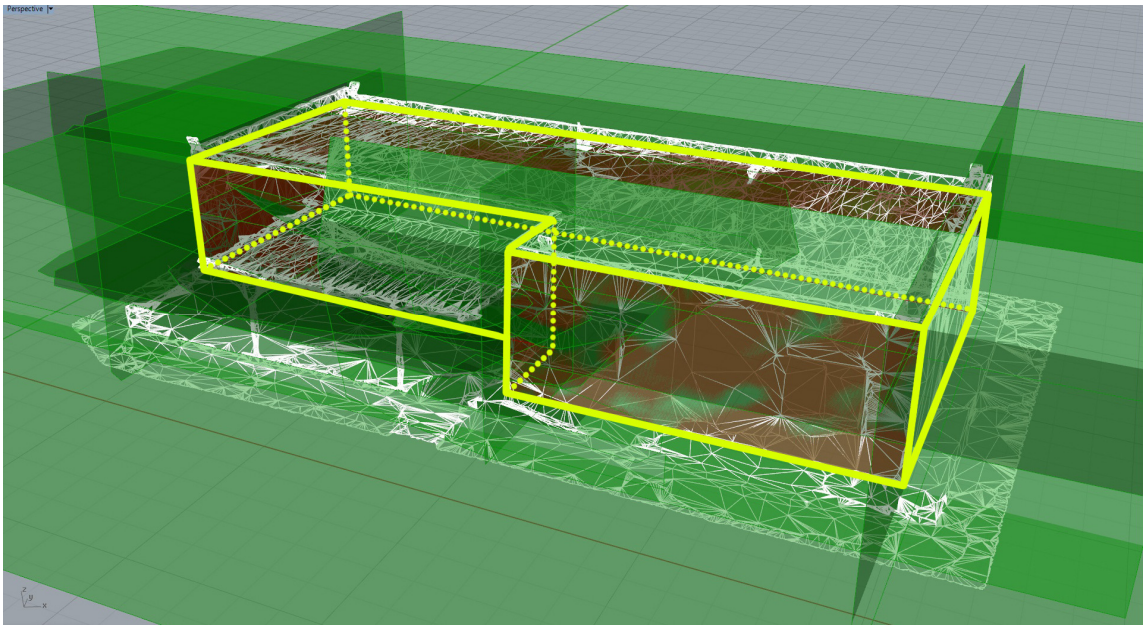


Figure 24. Convex hull plane intersection to generate closed BREP geometry.

Currently, this method only works adequately on buildings with simple geometry such as the example used in this experiment, as using it on complex geometry might produce geometry results that do not match the building's form.

In order to orient the building properly relative to the world coordinate system's 2D XY plane, a simple script was developed by the author that automatically orients the building by using the largest generated convex hull surface and assuming that it represents the ground surface. This surface's plane is then re-mapped on the world coordinate XY plane, and the building geometry is subsequently oriented. A variable is used to adjust the rotation of the building relative to the cardinal directions. This enables the prototype to accept an input mesh that may not be properly oriented for the BEM process, then automatically orient the building mesh by using a single variable.

4.6 Placing Openings Using Annotated Texture Map

Windows are an important factor in BEM as they contribute largely towards the heat gain of buildings. The annotated window polylines from Chapter III were integrated into the prototype as shown in Figure 25. Since the window surfaces must be co-planar to the wall surfaces, a script was developed to automatically wrap the 3D window polylines on the closed BREP generated from the intersection process (Figure 24). This is done by projecting the corner points from the 3D polylines to the BREP. The door polyline was considered as a window in the prototype.

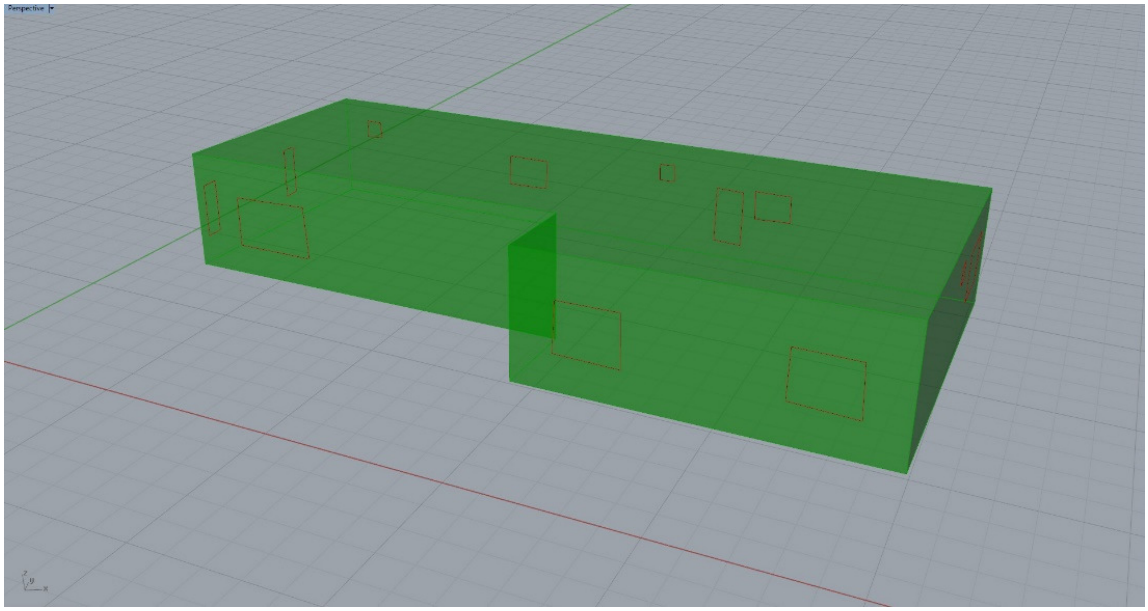


Figure 25. Closed planar BREP with projected window polylines.

However, these window polylines do not represent perfect rectangles, as the textured mesh represents a real building with all imperfections in construction and photogrammetric 3D reconstruction. This is an issue for EnergyPlus as the simulation engine only accepts rectangular and triangular geometry for window surfaces as mentioned in Section 4.1. To resolve these issues, the minimum bounding rectangle area was calculated for each opening (Figure 26). This method was performed by using a genetic algorithm process to minimize the bounding rectangle areas and make them rectangular for EnergyPlus. A bounding box algorithm was applied for each window polyline, and Galapagos - a genetic algorithm tool in Grasshopper, was used to rotate each bounding box for achieving the minimum area for each bounding box.

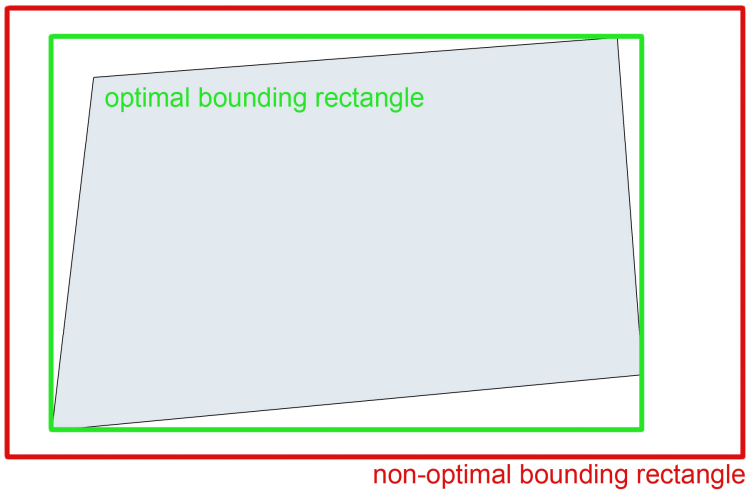


Figure 26. Calculating the minimum bounding rectangle area for a four-sided non-rectangular window.

4.7 Converting BEM-Compatible Geometry to Energy Model

This section covers the processes of how the resulting building model, which is defined by the generated closed BREP model with co-planar window surfaces, is converted automatically into an EnergyPlus model by using the toolsets provided by Ladybug and Honeybee. During these processes, a number of assumptions were made in terms of describing the building energy model. The first process is to convert the closed BREP model into an EnergyPlus thermal zone by using Honeybee's 'Masses2Zone' component, which assigns to each input closed BREP geometry a thermal zone name, a zone program, and specifies if the zone is conditioned or not. The component then converts the input geometry into an EnergyPlus thermal zone including the building surfaces such as the walls, floors, ceilings, and roof. Since the building model in the prototype program is a single closed BREP model, only 1 zone was specified; the

resulting energy model will be considered as a 1-zone building energy model that will not consider the internal walls or rooms within the building. This zone is specified to be air-conditioned, which will assign an 'Ideal Air Loads System' input object from the EnergyPlus library. As for the zone program, which includes materials and constructions, Honeybee accesses the OpenStudio library which provides a list of pre-defined building programs that can be used as templates for EnergyPlus inputs. The zone in the prototype was set to 'apartment', since this was the closest building template to a small residential farmhouse. The usual energy modeling process requires strict geometry rules, manual work, and debugging, and since the input closed BREP model was already made compatible during the previous processes, it was successfully converted into a defined thermal zone without issues.

The window surfaces can be converted into EnergyPlus glazing inputs by using Honeybee's 'addHBGlz' component, however, the component requires any surface to be converted to be co-planar and connected to a thermal zone. This is also a non-issue since the prototype applied the window surfaces on the closed BREP model as explained in Section 4.6.

The resulting EnergyPlus zones and surfaces can be visualized within the Rhinoceros interface as seen in Figure 27.

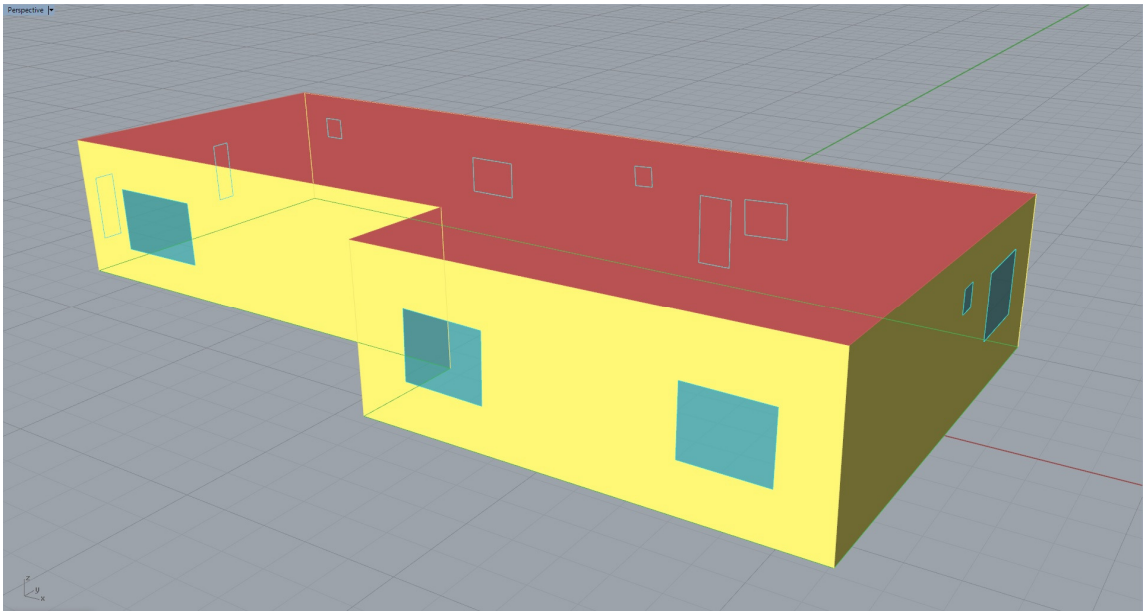


Figure 27. Energy model visualized in Rhinoceros.

In order to write the currently defined model into an EnergyPlus IDF file, Honeybee's 'runEnergySimulation' was used in addition to further inputs. These inputs are the weather file, simulation analysis period, simulation outputs, and EnergyPlus shadow parameters. The selected weather file is TMY (Typical Meteorological Year) weather data measured at Kuwait International Airport and provided by a Kuwait Institute for Scientific Research (KISR) report in 2000 (Shaban, 2000). The weather data was downloaded from the EnergyPlus website in EPW file format (EnergyPlus, 2016). The simulation analysis period was defined from June until September. The selected EnergyPlus outputs for the simulation are solving for heating, cooling, and electricity loads, in addition to calculating the mean air temperature, operative temperature, and relative humidity. As for setting the shadow parameters, the solar distribution calculation is set to 'Full Exterior', which accounts for direct solar radiation and shading on exterior

geometry. However, for the interior of the building, all solar radiation entering the zones is assumed to fall on the floor, and simple window view factor calculations are used to calculate diffuse radiation on interior surfaces. This is contrast to the 'Full Interior And Exterior' setting, which will calculate the solar radiation falling on the interior surfaces in a more accurate manner by ray tracing the solar rays entering the building. To account for the geometric model used in the prototype, selecting 'Full Exterior' is important since the building's thermal zone consists of concave surfaces; more specifically the L-shaped floor area and ceiling/roof. As of this study, EnergyPlus cannot solve solar distribution for concave surfaces using the 'Full Interior And Exterior' setting and doing so will result in a simulation run error. Any concave surfaces must be converted to convex surfaces for this calculation method. Since the prototype did not address this issue, the 'Full Exterior' calculation setting was chosen to account for the limitations inherent in the model and the simulation engine used for the prototype.

The resulting output after these processes was writing an EnergyPlus model in IDF format which can be opened in IDF Editor and EP-Launch.

4.8 Verifying Energy Model

The objective of this experiment is not to conduct a calibrated energy simulation, but to produce a verified one. In the context of modeling and simulation, verification is making sure that a model is debugged and works as intended, while validation is making sure that the model represents the real-world conditions (Chung, 2003). The IDF file is

not considered verified until a successful energy simulation is conducted without any errors.

The prototype managed to produce a building energy model that was successfully simulated using the EnergyPlus engine. Figure 28 shows graphical charts of the simulated cooling, lighting, and equipment loads from June to September. The cooling set point is 24 C° (74 F°), and the total simulated cooling load for the analysis period amounted to 18,065 kWh. The peak cooling hour was found to be 7:00 PM, August 6th. The simulation results are not verified or calibrated against actual load measurements.

Appendix A contains the inputs for EnergyPlus that were automatically generated from the OpenStudio library.

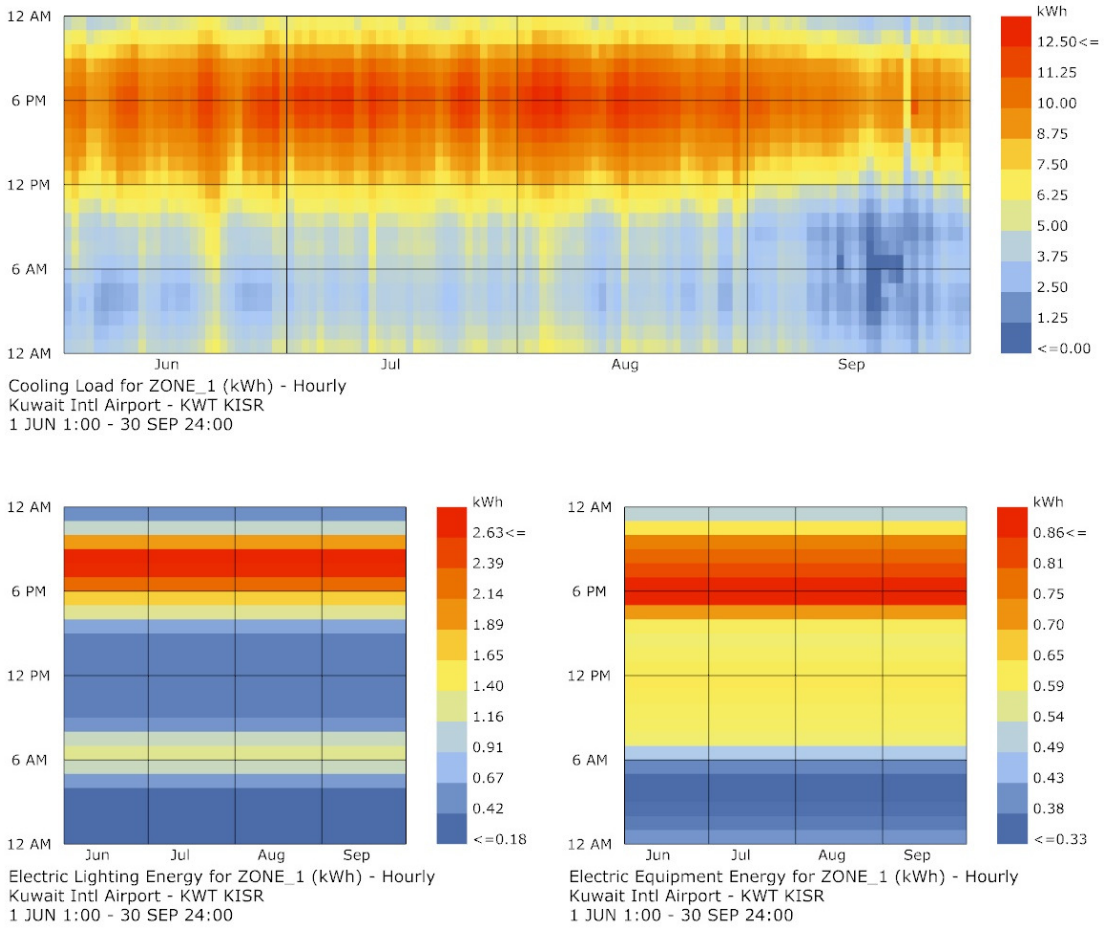


Figure 28. Simulated energy loads of the farmhouse from June to September (Charts were generated using Ladybug).

An alternative simulation run was conducted by specifying that the building zone is unconditioned. This is done by switching a simple Boolean toggle (true/false) within the prototype to switch between simulating an unconditioned building, or a conditioned building using the 'Ideal Loads Air System' input. The appropriate graphical charts are generated automatically using Ladybug. Figure 29 shows the plotted psychrometric chart for the unconditioned zone.

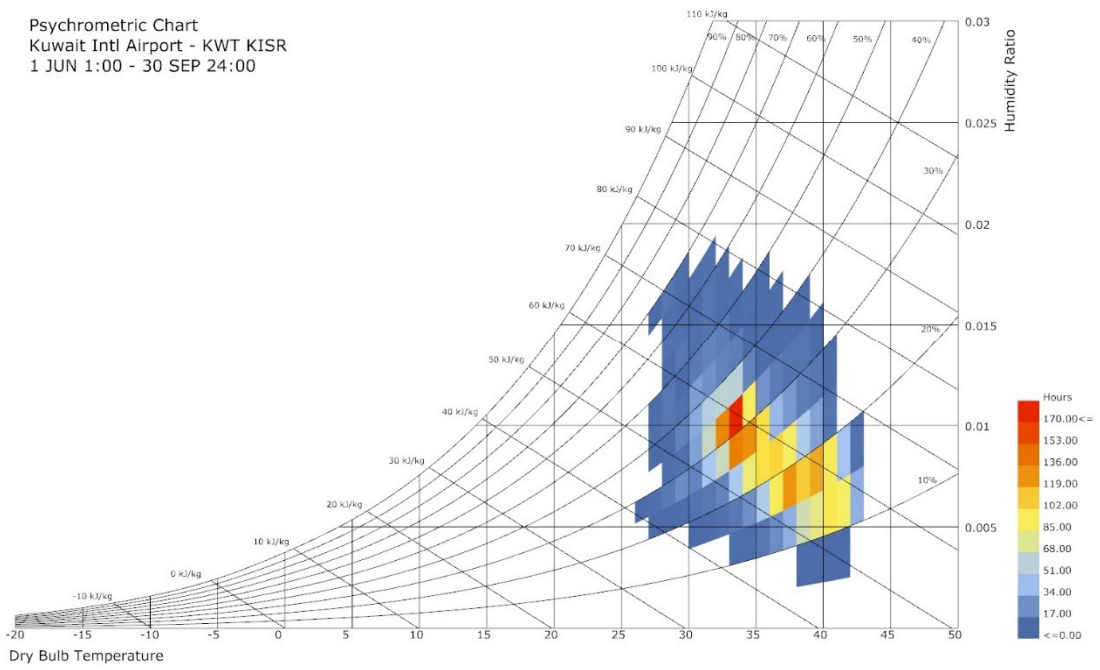


Figure 29. Dry-bulb temperatures and humidity ratios for the unconditioned zone from June to September (Chart was generated using Ladybug).

CHAPTER V

CONCLUSION AND DISCUSSION

In this study, a number of methods and techniques were presented to generate a verified building energy model from a digital photogrammetry-based 3D model. A field survey was conducted to collect photographs of a selected case-study building. The photographs were processed in a 3D reconstruction software to automatically generate a textured 3D mesh. A methodology was presented to semantically annotate the mesh's texture map. By using the 3D mesh and annotated texture map as inputs, a prototype Grasshopper program was developed to automatically output an EnergyPlus model. Finally, the EnergyPlus model was simulated successfully to validate the prototype.

5.1 Significance and Applications

For the purposes of environmental assessments and the sustainable management of existing buildings, the methods and techniques presented in this thesis can be applied to create a semi-automated workflow to generate a building energy model from site photographs of a building. This could potentially reduce time and labor costs by automating many of the processes involved in creating 3D geometry for BEM, and by taking advantage of pre-defined templates as BEM inputs. Also, this workflow would reduce the human error factor by reducing the manual user inputs required, and by semi-automating 3D modeling and inputting procedures through computer algorithms.

Another contribution of this work is towards the creation of an easy-to-use platform for BEM and energy simulations, with limited user inputs that can provide quick, visual, and graphical assessments for a stakeholder interested in understanding the environmental performance (including energy use) of an existing building.

Further technological improvements and developments in computer processing power, photogrammetric 3D reconstruction, and image analysis can be applied within the framework of the presented prototype, where environmental assessments for large scale urban models can be realized.

For applications beyond the AEC field such as virtual reality, this study describes methods to simplify a 3D geometry model of a building. Another method is described for adding building semantics from the texture map of a 3D mesh model.

5.2 Limitations and Future Work

While the prototype program operated successfully on the selected case study, there are some limitations in handling other 3D geometry. For example, the prototype assumes that the building model does not contain two or more protruding walls that lie on the same plane (a limitation with the mesh distance-based grouping algorithm in Chapter IV Section 4.4). Additionally, initial tests using buildings with gabled roofs proved to be unsuccessful. With further improvements to the prototype, these issues can be overcome.

As for the proposed method to semantically annotate the texture map, an automatic process has not yet been developed and was not in the scope of this study. In

order to realize a fully automated process from photo-to-BEM, more research and testing is required in this area.

In terms of BEM inputs, the 3D mesh that was used did not contain any control points and was not georeferenced, and adjustments had to be made to orient the building. Furthermore, there are limitations with the EnergyPlus engine, such as limited solar radiation calculation on concave surfaces, and only accepting rectangular or triangular windows. Also, the BEM process in general relied on many assumptions and the use of pre-defined templates. This study suggests that with future research in the field of computer vision, building constructions and materials could be automatically labeled on the texture map as additional inputs for BEM. Labeling methods can be improved further with developments in online cloud-based image recognition, adding additional layers of semantics.

5.3 Closing Remarks

This study attempted to contribute towards automating the process of BEM in a visual, accessible, and user-friendly manner. The goal is to reduce time and labor costs which are inherent with this process. By capitalizing on new developments in the field of 3D reconstruction and image analysis, there is a potential to add information (i.e., semantics) from the physical and visual world and apply it to BEM. By bridging these processes together, this can potentially enable any stakeholder in the area of existing buildings, either an owner, architect, or engineer to measure the environmental performance of a building.

REFERENCES

3DReshaper [Computer software]. (2016a). (Version 13.0.23910.0). Technodigit. Retrieved from http://www.3dreshaper.com/en/

3DReshaper. (2016b). Building Extractor - 3DReshaper 2015. Retrieved from https://www.youtube.com/watch?v=AH73YqT4yJc

123D [Computer software]. (2017). Autodesk. Retrieved from http://www.123dapp.com/catch

Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M., & Maierhofer, S. (2013). O-Snap: Optimization-Based Snapping for Modeling Architecture. *ACM Transactions on Graphics*, *32*(1), 6.

Asl, M. R., Bergin, M., Menter, A., & Yan, W. (2014). BIM-Based Parametric Building Energy Performance Multi-Objective Optimization. In *Education and Research in Computer Aided Architectural Design in Europe (eCAADe): Fusion*. Newcastle upon Tyne, England. Retrieved from http://papers.cumincad.org/data/works/att/ecaade2014_224.content.pdf

Asl, M. R., Zarrinmehr, S., & Yan, W. (2013). Towards BIM-Based Parametric Building Energy Performance Optimization. In *Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA): ACADIA 13: Adaptive Architecture*. Cambridge, Massachusetts. Retrieved from http://papers.cumincad.org/data/works/att/acadia13_101.content.pdf

Autodesk. (2011). Streamlining Energy Analysis of Existing Buildings with Rapid Energy Modeling. Autodesk White Paper. Retrieved from http://images.autodesk.com/adsk/files/rem_white_paper_2011.pdf

Autodesk. (2016). Autodesk Labs: Project Rosenfeld (v1.1). Retrieved from https://www.youtube.com/watch?v=psPIUqPW4Ic

Autodesk. (2016). Autodesk ReCap 360: Rapid Energy Models from Photos into Autodesk Revit. Retrieved from https://www.youtube.com/watch?v=8gBApip-qqM

Autodesk. (2016). Rapid Energy Modeling for Existing Buildings (REM). Retrieved from http://sustainability.autodesk.com/available-solutions/rapid-energy-modeling/

Becker, S. (2009). Generation and Application of Rules for Quality Dependent Façade Reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, *64*(6), 640–653.

Becker, S., & Haala, N. (2009). Grammar Supported Facade Reconstruction from Mobile LIDAR Mapping. In *ISPRS Workshop: CMRT09 - City Models, Roads and Traffic*. Paris, France. Retrieved from http://www.ifp.uni-stuttgart.de/publications/2009/CMRT_Becker_Haala_final.pdf

Blender Foundation. (2016). Blender (Version 2.78). Retrieved from https://www.blender.org/

Briscoe, D. (2015). *Beyond BIM: Architecture Information Modeling*. Abingdon, UK: Routledge.

Burnett, C., & Blaschke, T. (2003). A Multi-Scale Segmentation/Object Relationship Modelling Methodology for Landscape Analysis. *Ecological Modelling*, *168*(3), 233–249.

Burtch, R. (2008). History of Photogrammetry. Retrieved from https://assets.documentcloud.org/documents/3235835/History-of-Photogrammetry.pdf

Cao, J., Metzmacher, H., O'Donnell, J., Kobbelt, L., & van Treeck, C. (2015). BIM Geometry Generation from Low-Resolution Aerial Photographs for Building Energy Modeling. In *International Conference of International Building Performance Simulation Association (IBPSA)*. Hyderabad, India. Retrieved from http://www.ibpsa.org/proceedings/BS2015/p2451.pdf

Chung, C. A. (2003). *Simulation Modeling Handbook: A Practical Approach*. Boca Raton, FL, USA: CRC Press.

CITA. (2017a). DURAARK Integration of DURAARK Workbench in Rhino - Automated Reconstruction of BIM Model from Point Clouds. Retrieved from https://vimeo.com/181613946

CITA. (2017b). DURAARK Reconstruction Component. Retrieved from https://vimeo.com/123437666

David, R. (2017). Chapter 5: Line, Edge and Contours Detection. Retrieved from http://www.robindavid.fr/opencv-tutorial/chapter5-line-edge-and-contours-detection.html

DOE. (2015). EnergyPlus (Version 8.4.0) [Computer software]. Retrieved from https://energyplus.net/

DOE, & Hirsch, J. J. (2016). DOE-2.2 (Version 2.2) [Computer software]. James J. Hirsch & Associates. Retrieved from http://doe2.com/

DURAARK. (2017). Durable Architectural Knowledge (DURAARK). Retrieved from http://duraark.eu/

Eastman, C., Eastman, C. M., Teicholz, P., Sacks, R., & Liston, K. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors* (2nd ed.). Hoboken, NJ, USA: John Wiley & Sons.

eCognition [Computer software]. (2016). Trimble. Retrieved from http://www.ecognition.com/

EnergyPlus. (2016). Weather Data by Location. Retrieved from https://energyplus.net/weather-location/asia_wmo_region_2/KWT//KWT_Kuwait.Intl.AP.405820_KISR

Evers, H. L., & Zwierzycki, M. (2016). Volvox (Version 0.3.0.0). Retrieved from http://www.food4rhino.com/app/volvox

Förstner, W., & Wrobel, B. P. (2016). *Photogrammetric Computer Vision: Statistics, Geometry, Orientation and Reconstruction*. Cham, Switzerland: Springer International Publishing.

Frommholz, D., Linkiewicz, M., Meissner, H., Dahlke, D., & Poznanska, A. (2015). Extracting Semantically Annotated 3D Building Models with Textures from

Oblique Aerial Imagery. In *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS Archives)*. Munich, Germany. Retrieved from http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-3-W2/53/2015/isprsarchives-XL-3-W2-53-2015.pdf

Furukawa, Y., Curless, B., Seitz, S. M., & Szeliski, R. (2009). Manhattan-World Stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Miami, Florida. Retrieved from http://grail.cs.washington.edu/projects/manhattan/manhattan.pdf

Gallaher, M. P., O'Connor, A. C., Dettbarn, Jr., J. L., & Gilday, L. T. (2004). *Cost Analysis of Inadequate Interoperability in the US Capital Facilities Industry* (No. NIST GCR 04-867). Gaithersburg, Maryland: National Institute of Standards and Technology (NIST). Retrieved from http://fire.nist.gov/bfrlpubs/build04/PDF/b04022.pdf

girish_m, & user667804. (2017). OpenCV Canny + Watershed. Retrieved from http://stackoverflow.com/questions/21060804/opencv-canny-watershed

Google Earth [Computer software]. (2017). Google. Retrieved from https://www.google.com/earth/

Górski, F., Kuczko, W., Wichniarek, R., & Zawadzki, P. (2010). Application of Close Range Photogrammetry in Reverse Engineering. In *International DAAAM Baltic Conference: Industrial Engineering*. Tallinn, Estonia. Retrieved from http://innomet.ttu.ee/daaam10/proceedings/PDF/gorski.pdf

Haala, N., & Kada, M. (2010). An Update on Automatic 3D Building Reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, *65*(6), 570–580.

Haugeard, J.-E., Philipp-Foliguet, S., Precioso, F., & Lebrun, J. (2009). Extraction of Windows in Facade Using Kernel on Graph of Contours. In *Scandinavian Conference on Image Analysis (SCIA)*. Oslo, Norway. Retrieved from https://pdfs.semanticscholar.org/616e/73effbc1b2e94962d516e960326809ed00c5.pdf

ImageModeler [Computer software]. (2009). Autodesk. Retrieved from http://usa.autodesk.com/adsk/servlet/pc/index?id=11390028&siteID=123112

Issa, R. (2013). *Essential Mathematics for Computational Design* (3rd ed.). Seattle, WA, USA: Robert McNeel & Associates.

Jahangiri, M., & Petrou, M. (2008). Fully Bottom-Up Blob Extraction in Building Facades. In *International Conference of Pattern Recognition and Image Analysis (PRIA): New Information Technologies*. Nizhny Novgorod, Russian Federation. Retrieved from http://www.ipb.uni-bonn.de/projects/etrims/publications/jahangiri-petrou-08-Fully%20Bottom-Up%20Blob%20Extraction%20in%20Building%20Facades.pdf

Jefferis, A., Madsen, D. A., & Madsen, D. P. (2012). *Architectural Drafting and Design* (6th ed.). Boston, MA, USA: Cengage Learning.

Kensek, K., & Noble, D. (2014). *Building Information Modeling: BIM in Current and Future Practice*. Hoboken, NJ, USA: John Wiley & Sons.

Kim, H., Asl, M. R., & Yan, W. (2015). Parametric BIM-Based Energy Simulation for Buildings with Complex Kinetic Façades. In *Education and Research in Computer Aided Architectural Design in Europe (eCAADe): Real Time*. Vienna, Austria. Retrieved from http://papers.cumincad.org/data/works/att/ecaade2015_202.content.pdf

Kolbe, T. H., Gröger, G., & Plümer, L. (2005). CityGML – Interoperable Access to 3D City Models. In *Geo-information for Disaster Management (Gi4DM)*. Delft, Netherlands. Retrieved from http://www.sig3d.org/files/media/downloads/Papers/Gi4Dm_2005_Kolbe_Groeger.pdf

Kota, S., Stipo, F. J. F., Jeong, W., Kim, J. B., Alcocer, J. L. B., Clayton, M. J., Yan, W., Haberl, J. S. (2016). Development of a Reference Building Information Model for Thermal Model Compliance Testing-Part I: Guidelines for Generating Thermal Model Input Files. *ASHRAE Transactions*, *122*, 256–266.

Larsen, C. L. (2010). *3D Reconstruction of Buildings From Images with Automatic Façade Refinement* (Master's Thesis). Aalborg University, Denmark. Retrieved from http://projekter.aau.dk/projekter/files/32366963/report.pdf

LBNL, & Ward, G. (2014). Radiance [Computer software]. Retrieved from http://www.radiance-online.org/

Lee, S. C., & Nevatia, R. (2004). Extraction and Integration of Window in a 3D Building Model from Ground View Images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Washington, D.C. Retrieved from

https://pdfs.semanticscholar.org/d04e/24da69cdcd36cf42477d3b26cca75b296db8.pdf

Lévy, B., Petitjean, S., Ray, N., & Maillot, J. (2002). Least Squares Conformal Maps For Automatic Texture Atlas Generation. In *Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH)*. Retrieved from https://members.loria.fr/Bruno.Levy/papers/LSCM_SIGGRAPH_2002.pdf

Mans, D. (2016). Mesh+ (Version 2.1) [Computer software]. Retrieved from http://www.food4rhino.com/app/mesh

MATLAB [Computer software]. (2017). MathWorks. Retrieved from https://www.mathworks.com/products/matlab.html

Maya [Computer software]. (2016). Autodesk. Retrieved from http://www.autodesk.com/products/maya/overview

Modo [Computer software]. (2016). The Foundry. Retrieved from https://www.foundry.com/products/modo

Müller, P., Zeng, G., Wonka, P., & Van Gool, L. (2007). Image-Based Procedural Modeling of Facades. *ACM Transactions on Graphics*, *26*(3), 85.

Nguatem, W., Drauschke, M., & Mayer, H. (2014). Localization of Windows and Doors in 3D Point Clouds of Facades. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS Annals)*. Zurich, Switzerland. Retrieved from http://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/II-3/87/2014/isprsannals-II-3-87-2014.pdf

Ochmann, S., Vock, R., Wessel, R., & Klein, R. (2016). Automatic Reconstruction of Parametric Building Models from Indoor Point Clouds. *Computers & Graphics*, *54*, 94–103.

Ochmann, S., Vock, R., Wessel, R., Tamke, M., & Klein, R. (2014). Automatic Generation of Structural Building Descriptions from 3D Point Cloud Scans. In *International Conference on Computer Graphics Theory and Applications (GRAPP)*. Lisbon, Portugal. Retrieved from http://cg.cs.uni-bonn.de/aigaion2root/attachments/GRAPP_2014_54_CR.pdf

O'Donnell, J. T., Maile, T., Rose, C., Mrazović, N., Morrissey, E., Regnier, C., … Bazjanac, V. (2014). *Transforming BIM to BEM: Generation of Building Geometry for the NASA Ames Sustainability Base BIM* (No. LBNL-6033E). Berkely, CA, USA: Lawrence Berkeley National Labs (LBNL). Retrieved from https://simulationresearch.lbl.gov/sites/all/files/lbnl-6033e.pdf

OpenCV [Computer software]. (2016). (Version 3.2). Retrieved from http://opencv.org/

OpenStudio [Computer software]. (2016). (Version 1.13.0). Retrieved from https://www.openstudio.net/

PhotoModeler [Computer software]. (2016). Eos Systems Inc. Retrieved from http://www.photomodeler.com/

Photoscan [Computer software]. (2016). (Version 1.2.6). Agisoft. Retrieved from http://www.agisoft.com/

Prabhu, V. (2010). *Techniques for the Mediation of Heritage Preservation vs. Building Performance* (Master's Thesis). Carleton University, Ottawa, Ontario. Retrieved

from https://curve.carleton.ca/system/files/etd/baafbffa-0cd2-4371-8787-890bca4f710c/etd_pdf/ec2561df8f62ae1a0305429daea663ea/prabhu-techniquesforthemediationofheritagepreservation.pdf

Pu, S., & Vosselman, G. (2006). Automatic Extraction of Building Features from Terrestrial Laser Scanning. In *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS Archives)*. Dresden, Germany. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.2486&rep=rep1&type=pdf

Rafeek, M. T. M. (2008). Extraction of Building Facade Information from Terrestrial Image Using Segmentation Based Bbject-Oriented Image Analysis. In *Asian Conference on Remote Sensing (ACRS)*. Colombo, Sri Lanka. Retrieved from http://a-a-r-s.org/aars/proceeding/ACRS2008/Papers/TS%2016.3.pdf

ReCap 360 [Computer software]. (2016). (Version 3.0.2.12). Autodesk. Retrieved from https://recap360.autodesk.com/

Reisner-Kollmann, I. (2013). *Reconstruction of 3D Models from Images and Point Clouds with Shape Primitives* (Dissertation). Technische Universität Wien, Vienna, Austria. Retrieved from https://publik.tuwien.ac.at/files/PubDat_220568.pdf

Remake [Computer software]. (2016). (Version 17.23.1.69). Autodesk. Retrieved from https://remake.autodesk.com/

Revit [Computer software]. (2016). Autodesk. Retrieved from http://www.autodesk.com/products/revit-family/overview

Rhinoceros [Computer software]. (2016). (Version 5.13.60523.20140). Robert McNeel & Associates. Retrieved from https://www.rhino3d.com/

Roudsari, M. S. (2016a). Ladybug (Version 0.0.63). Retrieved from http://www.food4rhino.com/app/ladybug-tools

Roudsari, M. S. (2016b). Honeybee (Version 0.0.60). Retrieved from http://www.food4rhino.com/app/ladybug-tools

Rutten, D. (2014). Grasshopper (Version 0.9.0076). Robert McNeel & Associates. Retrieved from http://www.grasshopper3d.com/

Salomon, D. (2011). *The Computer Graphics Manual* (Vol. 2). London, UK: Springer-Verlag London Ltd.

Schenk, T. (2005). Introduction to Photogrammetry. The Ohio State University. Retrieved from http://www.mat.uc.pt/~gil/downloads/IntroPhoto.pdf

Shaban, N. (2000). *Development of Typical Meteorological Year for Kuwait* (No. KISR 5857). Kuwait: Kuwait Institute for Scientific Research (KISR).

Simon, L. (2011). *Procedural Reconstruction of Buildings: Towards Large Scale Automatic 3D Modeling of Urban Environments* (PhD Thesis). École Centrale Paris, Châtenay-Malabry, France. Retrieved from https://pdfs.semanticscholar.org/e2f2/72c0f2f9c837498ffd0719f2ba9d3fc1315a.pdf?_ga=1.45911012.1982736775.1490232190

Sprengel, E. (2014). *A Report on Facade Labeling*. Zurich, Switzerland: ETH Zurich. Retrieved from http://n.ethz.ch/~eliass/files/Facade-Labeling-Elias-Sprengel.pdf

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer-Verlag London Ltd.

Tamke, M., Blümel, I., Ochmann, S., Vock, R., & Wessel, R. (2014). From Point Clouds to Definitions of Architectural Space. In *Education and Research in Computer Aided Architectural Design in Europe (eCAADe): Fusion*. Newcastle upon Tyne, England. Retrieved from http://papers.cumincad.org/data/works/att/ecaade2014_138.content.pdf

Turner, E. L. (2015). *3D Modeling of Interior Building Environments and Objects from Noisy Sensor Suites* (Dissertation). University of California, Berkeley, Berkely, California. Retrieved from https://www2.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-105.pdf

Tutzauer, P., & Haala, N. (2015). Facade Reconstruction Using Geometric and Radiometric Point Cloud Information. In *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS Archives)*. Munich, Germany. Retrieved from http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-3-W2/247/2015/isprsarchives-XL-3-W2-247-2015.pdf

UC, & Hirsch, J. J. (1998). Overview of DOE-2.2. Camarillo, CA. Retrieved from http://www.doe2.com/Download/Docs/22_oview.pdf

Vračar, P., Kononenko, I., & Robnik-Šikonja, M. (2016). Obtaining Structural Descriptions of Building Façades. *Computer Science and Information Systems*, *13*(1), 23–43.

Wang, C. (2014). *Point Clouds and Thermal Data Fusion for Automated gbXML-Based Building Geometry Model Generation* (Dissertation). Georgia Institute of Technology, Atlanta, Georgia. Retrieved from https://smartech.gatech.edu/bitstream/handle/1853/54008/WANG-DISSERTATION-2014.pdf

Wang, R., Ferrie, F. P., & Macfarlane, J. (2012). A Method for Detecting Windows from Mobile LIDAR Data. *Photogrammetric Engineering & Remote Sensing*, *78*(11), 1129–1140.

Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., & Quan, L. (2008). Image-Based Façade Modeling. In *SIGGRAPH Asia*. Retrieved from http://vision.princeton.edu/projects/2008/TOG_facade/paper_low-res.pdf

Xiao, J., Fang, T., Zhao, P., Lhuillier, M., & Quan, L. (2009). Image-Based Street-Side City Modeling. In *SIGGRAPH Asia*. Yokohama, Japan. Retrieved from http://vision.princeton.edu/projects/2009/TOG/paper_high-res.pdf

Yan, W., Clayton, M., Haberl, J., Jeong, W., Kim, J. B., Kota, S., Alcocer, J. L. B., Dixit, M. (2013). Interfacing BIM with building thermal and daylighting modeling. In *International Conference of the International Building Performance Simulation Association (IBPSA)*. Chambery, France. Retrieved from http://www.ibpsa.org/proceedings/BS2013/p_1374.pdf

Zhang, R., & Zakhor, A. (2014). Automatic Identification of Window Regions on Indoor Point Clouds Using LIDAR and Cameras. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*. Steamboat Springs, Colorado. Retrieved from http://www-video.eecs.berkeley.edu/papers/rzhang/rzhang_wacv14_draft_v5.pdf

Zwierzycki, M., Evers, H. L., & Tamke, M. (2016). Parametric Architectural Design with Point-Clouds - Volvox. In *Education and Research in Computer Aided Architectural Design in Europe (eCAADe): Complexity & Simplicity*. Oulu, Finland. Retrieved from http://papers.cumincad.org/data/works/att/ecaade2016_171.pdf

APPENDIX A

ENERGYPLUS INPUTS

Table 1. Description of EnergyPlus template inputs for the prototype.

General	EnergyPlus Version	8.4.0
	Weather File	Kuwait Intl AP 405820
	North Axis	0° from true North
	Run Period	6/1 to 9/1
	Solar Distribution	FullExterior*
Location	Latitude	29.22
	Longitude	47.98
	Time Zone	3
	Elevation	55 m
Internal Loads	People	0.028 person/m^2
	Lights	11.840 W/m^2
	Equipment	3.875 W/m^2
HVAC Design	System	IdealLoadsAirSystem*
	Infiltration	0.0002 m^3/s per m^2
	Outdoor Air Method	Sum*
	Cooling Set Point	24 C° (74 F°)
	Cooling Sizing Factor	1.25

Construction	Exterior Wall	100 mm brick, 200 mm heavyweight concrete, 50 mm insulation board, 10 mm gypsum board
	Exterior Window	Double glazing (clear 3mm, air gap 13 mm, clear 3mm)
	Roof	100 mm lightweight concrete, acoustic tile

*EnergyPlus object name.

APPENDIX B

METHODS, TOOLS, AND TECHNIQUES

Table 2. List of used and developed software methods, tools, and techniques.

Method Name	**Software/Platform**	**Developed by Author?**	
		Yes	**No**
3D reconstruction	ReCap 360		✓
Barycentric interpolation tool	Grasshopper*	✓	
Boundary volume	Grasshopper*		✓
Bounding box	Grasshopper*		✓
BREP for BEM generation	Grasshopper*	✓	
BREP to IDF conversion	Ladybug and Honeybee**		✓
Convex hull	Grasshopper*		✓
Building energy simulation	EnergyPlus		✓
Genetic algorithm	Galapagos**		✓
Mesh area-based culling	Grasshopper*	✓	
Mesh decimation	Remake		✓
Mesh distance-based grouping	Grasshopper*	✓	
Mesh normal-based segmentation	Grasshopper*	✓	
Mesh unwelded-based segmentation	Grasshopper*, Mesh+**		✓
Plane fitting	Grasshopper*		✓
Texture baking	Modo		✓

Method Name	Software/Platform	Developed by Author?	
		Yes	No
Texture map annotation tool	Grasshopper*	✓	
UV mapping	Modo		✓
UV unwrapping tool	Grasshopper*	✓	
Window placement tool	Grasshopper*, Galapagos**	✓	

*Visual programming language and tool for Rhinoceros.

**Tools for Grasshopper.

APPENDIX C

GLOSSARY

The following (Table 3) is a list of technical terms and terms of art defined by the author based on the understanding of the terms. The terms are used in numerous publications accessible as textbooks, research papers, online documentations, etc.

Table 3. Terms and definitions.

Term	**Definition**
3D Reconstruction	Recording the measurements and/or appearance of physical objects as 3-dimentional data. Alternatively known as 3D scanning.
Aerial photogrammetry	Photogrammetry methods that involve taking photographs from an aircraft.
Ambiguous data	Data that contradicts another source of data that should represent the same information.
Barycentric coordinates	Coordinate system that defines a location of a point relative to a simplex (triangle or tetrahedron).
Blob	Pixel region in a digital image that differs from surrounding regions, such as in color, texture, or brightness.

Table 3. “Continued”.

Term	Definition
Bounding box	Algorithm for computing the minimum bounding box that encloses a geometry.
Camera parameters	Parameters that are used to define the position of a camera and/or photograph in an appropriate coordinate system.
Canny edge detection	Well-established image processing algorithm for extracting edges from photographs.
Cartesian coordinates	Coordinate system with three perpendicular axes (X, Y, and Z) that define the 2-dimentional position (x, y) or the 3-dimentional position (x, y, z) of a point.
Close-range photogrammetry	Photogrammetry methods that use ground-based cameras that are relatively close to the photographed subject.
Cloud computing	Internet-based computing for sharing and increasing overall processing power.
Computer vision	The science of artificial systems that understand imagery. It is related to various fields such as image processing and analysis, object recognition from imagery, and Artificial Intelligence (AI).

Table 3. “Continued”.

Term	Definition
Concave hull	Algorithm that computes the concave shape that contains a group of 2D points.
Convex hull	Algorithm that computes the minimum area convex shape that encloses a group of 2D points.
Co-planar	Two surfaces that lie on the same plane in the Cartesian coordinate system.
Data structure	The way computer data is defined and organized.
Dataset	Collection of computer data.
Depth map	Generated 2D image of a 3D scene that represents distance information, where a depth value (such as a specified color range) is assigned to each pixel.
Digital photogrammetry	Techniques within the field of photogrammetry that apply digital methods such as feature matching, SfM, and MVS for 3D reconstruction. Sometimes referred to as dense image matching.
End-to-end solution	Technological solution that provides all the user’s requirements without the need to use another solution.
Feature matching	Image processing method for analyzing the pixels between multiple images and finding the same feature such as edges, corners, and blobs

Table 3. “Continued”.

Term	Definition
Genetic algorithm	Well-established method for solving optimization problems.
Geometrical primitives	The simplest form of 3D geometry that can be described and standardized in a computer graphics system. Primitives include cubes, cylinders, spheres, cones, and pyramids.
Histogram	Graph that represents the distribution of data. For example, an image histogram graph distributes tonal values from pixels in a digital image.
Image analysis	Field that applies image processing algorithms to extract information from digital images and photographs, such as shapes, contours, and colors.
Image processing	Processing images using computer algorithms.
Image segmentation	Dividing an image’s pixels into separate parts.
Interoperability	Ability of different software to exchange information.
Laser scanner	Sensor that emits a beam of light (i.e., laser) to record the distance and/or appearance (such as color information) of an object based on the reflected beam measured by the sensor.

Table 3. “Continued”.

Term	Definition
Laser scanning	Technique for 3D reconstruction which produces point cloud data by using a laser scanner. It is also known as LIDAR.
Mesh	Data structure that represents 3D geometry and is composed of vertices, edges, and faces.
Mesh decimation	Process in which an algorithm is applied to reduce the number of faces in a mesh while preserving the overall shape of the mesh.
Mesh segmentation	Dividing a mesh’s data structure into separate datasets.
Monte Carlo simulation	Well-established computer algorithms for probability simulations.
Multidisciplinary	Involving multiple different academic and/or professional fields.
Normal (or normal vector)	Unit vector that is perpendicular to a surface at a given point.
Octree	Type of tree data structure that is used for real-time 3D graphics.
Parametric modeling	Modeling by defining and modifying object parameters and their relationships.

Table 3. “Continued”.

Term	Definition
Photogrammetry	The science of obtaining measurements of physical objects from photographs.
Pixel	Smallest element in a digital image. Pixel color can be represented by RGB components.
Pixel coordinate	Location of a pixel within an array of pixels in a digital image. It is defined by a horizontal and a vertical positive integer value starting from (1, 1).
Plane fitting	Fitting a plane through a group of points using a least-squares or RANSAC method.
Plane sweep	Algorithm that is based on the concept of sweeping a line across a plane to detect geometric intersections.
Point cloud	Data structure of points in 3D space.
Polyline	Connected line segments forming a single object.
Procedural modeling	Process of using computer algorithms to automatically generate computer geometry.
Rays (in computer graphics)	Computed lines intersected with virtual objects. Applied in techniques such as ray casting and ray tracing.
Recognition	Computer processes for identifying objects from imagery or 3D geometry.

Table 3. “Continued”.

Term	Definition
Remote sensing	The science of data acquisition of an object or phenomenon from a distance and without direct physical contact.
Rendering	Processes within the field of computer graphics that display and/or generate visualizations of 3D models (which can include textures and simulated lightning).
Rounding function	Mathematical function for reducing a number’s decimal digits.
Semantic 3D model	3D model where different parts of the geometry are classified or labeled based on meaningful knowledge.
Semantics	Meaningful definitions of information.
Set function	Function for listing computer data values (such as numbers) without repetitions.
Template matching	Algorithm for finding parts of a dataset (such as an image or a point cloud) that matches a pre-defined template.
Texture	Color information in a 3D model.
Texture baking	Method to transfer textures from one 3D model to another.
Texture coordinates	UV coordinates of a texture map.

Table 3. “Continued”.

Term	**Definition**
Texture map	Single (or multiple) raster image(s) containing pixels that wrap around parts of a 3D model.
Texture rectification	Improving the appearance of a 3D model’s texture map by changing the UV mapping.
Triangulation	Formation of triangles from a given set of points.
Unit vector	Vector with a length of 1.
Unrectified texture	3D model texture that is not rectified after an automatically computed texture mapping process.
Unwelded mesh	Mesh that is not connected in any vertex with another mesh.
UV coordinates	2D Cartesian coordinate system that defines the position of a point on a texture map with two values (u, v) ranging from 0 to 1. Alternatively known as texture coordinates.
UV map	2D map that assigns to each vertex in a 3D mesh a UV coordinate (u, v).
UV unwrapping	Unwrapping a mesh by replacing its vertex coordinates (x, y, z) with its UV coordinates (u, v).
Virtual reality	The science of digitally replicating real environments.

Table 3. “Continued”.

Term	Definition
Vision-based	Based on visual and optical systems, such as photography, thermography, multispectral imaging, and laser systems.
Voxels	Three-dimensional cubic grid of points in 3D space with color values (sometimes described as 3D pixels).
Voxel carving	Computer graphics technique that can be used to construct volumetric geometry using voxels.
Watershed segmentation	Image segmentation algorithm that can be used to label groups of pixels in order to identify potential objects.
World coordinate system	Cartesian coordinate system of a 3D scene.