SIGRADI2018
TECHNOPOLITICAS
xxii congresso da sociedade
iberoamericana de gráfica digital
*22th conference of the
iberoamerican society
of digital graphics*
07|08|09|novembro|2018
iau usp | são carlos | sp br

Geometry from 3D Photogrammetry for Building Energy Modeling

Mohammad Alawadhi
Texas A&M University | United States | mawadhi@tamu.edu

Wei Yan
Texas A&M University | United States | wyan@tamu.edu

Abstract

Building energy modeling requires skilled labor, and there is a need to make environmental assessments of buildings more efficient and accessible for architects. A building energy model is based on collecting data from the real, physical world and representing them as a digital model. Recent digital photogrammetry tools can reconstruct real-world geometry by transforming photographs into 3D models automatically. However, there is a lack of accessible workflows that utilize this technology for building energy modeling and simulations. This paper presents a novel methodology to generate a building energy model from a photogrammetry-based 3D model using available tools and computer algorithms.

Keywords 3D scanning; Building energy modeling; Building energy simulation; Digital photogrammetry; Photo-to-BEM

INTRODUCTION

Current digital photogrammetry software can automatically match photographs taken with a camera and generate a 3D model. Also, there is existing interest in using 3D scanning for building energy modeling (BEM) in the architecture, engineering and construction (AEC) industry (Autodesk, 2011; Deodhar, 2015). While previous studies touched upon the topic of automatically converting 3D scanned data to a building energy model (Turner, 2015; Wang, 2014), they are based on customized laser scanners and did not address available photogrammetry-based methods.

A 3D mesh automatically generated from current photogrammetry software lacks building semantics. A mesh is a 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 generated mesh data can contradict other sources of data such as existing architectural drawings, empirical site measurements, or other photogrammetry methods. This is due to the limitations from the software or human errors. These issues pose challenges to translate photogrammetry-based mesh data into inputs for BEM. The 3D geometry for building performance models such as energy, daylighting, and Computational Fluid Dynamics (CFD) 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). BEM and simulation engines have limitations in handling 3D geometry. For example, using EnergyPlus, any 3D geometry that represents a thermal zone must contain a closed volume or shell (Kota et al., 2016). Photogrammetry 3D models have open mesh surface geometry as opposed to closed geometry such as boundary representation (BREP) models. While previous methods were developed to simplify 3D geometry of buildings from point cloud data (3DReshaper, 2015; Arikan et al., 2013; Reisner-Kollmann, 2013) and to extract building semantics from 3D scanned buildings (Frommholz et al., 2015; Haala & Kada, 2010; Ochmann et al., 2016; Ochmann et al., 2014; Pu et al., 2006; Tamke et al., 2014; Zwierzycki et al., 2016), these methods did not address converting photogrammetry-based mesh data to geometry data suitable for BEM.

We present a tested methodology that can be used to generate a building energy model from a digital photogrammetry-based 3D model. This methodology can be applied to create a semi-automated workflow to generate a building energy model from site photographs by automating 3D modeling and inputting procedures through available tools and computer algorithms. This contributes towards the creation of an accessible platform for BEM and energy simulations, with limited user inputs that can provide assessments for architects and designers interested in understanding the environmental performance of existing buildings.

METHODOLOGY

The outline of the presented methodology can be seen in Figure 1. There are two main inputs within this framework; a 3D mesh generated from a photogrammetry software, and an annotated texture map. The output of this framework is a building energy model that can be verified through running a successful energy simulation.

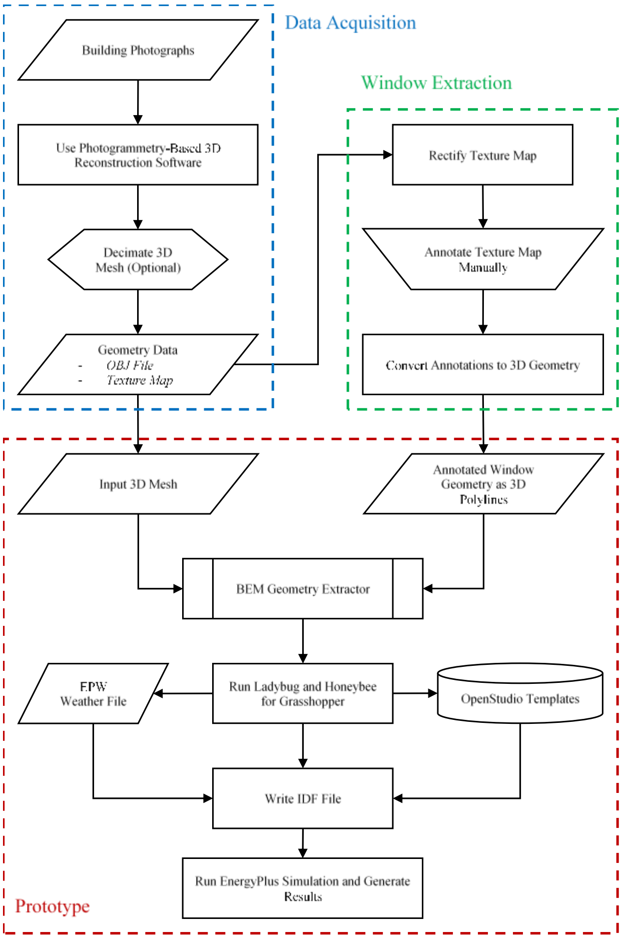

Data Acquisition
Building Photographs
Use Photogrammetry-Based 3D Reconstruction Software
Decimate 3D Mesh (Optional)
Geometry Data
- OBJ File
- Texture Map
Window Extraction
Rectify Texture Map
Annotate Texture Map Manually
Convert Annotations to 3D Geometry
Input 3D Mesh
Annotated Window Geometry as 3D Polylines
BEM Geometry Extractor
EPW Weather File
Run Ladybug and Honeybee for Grasshopper
OpenStudio Templates
Write IDF File
Run EnergyPlus Simulation and Generate Results
Prototype

Figure 1: Methodology framework.

DATA ACQUISTION USING DIGITAL PHOTOGRAMMETRY

Data acquisition involved conducting a field survey where photographic data was acquired using a consumer UAV drone fitted with a HD camera. Photographs of a case-study building were taken by the UAV drone while following a circular flight path. The photographs from the field survey were processed using photogrammetry software. A 3D model of the building was automatically generated using the photogrammetry software. The resulting textured 3D mesh model was edited and exported to OBJ - an open file format.

ANNOTATION OF TEXTURE DATA

3D scanned image data was contained in a texture map, which is a single raster image containing pixels that wrap around all parts of the 3D model. Building semantics were annotated on the texture map directly based on an approach similar to the one proposed by Frommholz et al., (2015). The annotations were applied manually on the texture map at this stage (but can be automated through machine learning in the future). For window extraction, the annotated 2D pixels in the texture map were automatically converted to semantic 3D geometry. The window annotations were used as inputs for the BEM process.

PROTOTYPE DEVELOPMENT AND OUTPUT SIMULATION

A prototype program was developed using Grasshopper that utilizes the 3D scanned data as geometric modeling inputs for BEM. The inputs were a photogrammetry-based mesh and an annotated texture map. A number of algorithms and tools were applied to automatically convert both the 3D mesh and the annotated texture map to geometry compatible with EnergyPlus. Through the prototype program, Ladybug and Honeybee (Sadeghipour Roudsari & Pak, 2013) were used with the geometric inputs to generate a building energy model. The feasibility of the output was verified by running a successful energy simulation. The outline of the prototype's overall framework can be seen in Figure 2.

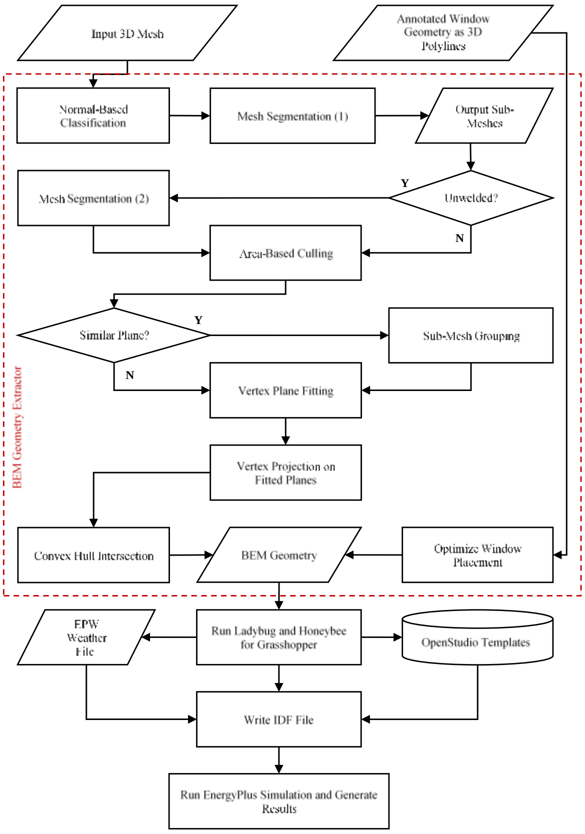


Figure 2: Prototype framework with the BEM geometry extractor shown in Figure 1 detailed here.

EXPERIMENTS AND RESULTS

FIELD SURVEY

The case-study building was a single-story farmhouse. The UAV was set on an automatic point-of-interest flight path, which enabled the drone to fly continuously in a fixed circular path while facing the building. A total of 111 photographs were taken at arbitrary intervals. The photographs were uploaded to ReCap 360 – an image-based 3D reconstruction tool.

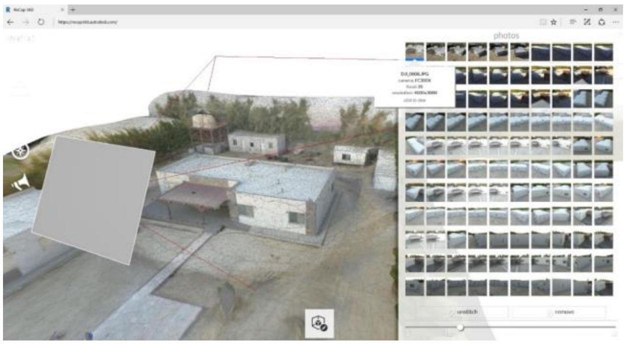


Figure 3: The reconstructed 3D mesh.

Figure 3 shows the automatically generated 3D mesh. The photographs were taken at an oblique angle; there were no photographs that clearly showed the area beneath the porch. Since this part was obstructed in view, the software was not able to reconstruct this geometry properly. Also, holes were identified on the output mesh that do not exist in the actual building. The walls in the 3D mesh measured around 4 meters high (around 13 feet) in software, 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; therefore, the walls were not manually scaled. Otherwise, they would be manually scaled. The 3D mesh was saved in the OBJ format which includes a text list of the vertices as Cartesian coordinate values, texture UV values for each vertex, and the mesh face definitions. A texture map was saved with the file.

TEXTURE MAP ANALYSIS

Building semantics cannot be recognized from the disorganized texture map that was generated using the software (Figure 4). The texture map must be rectified.

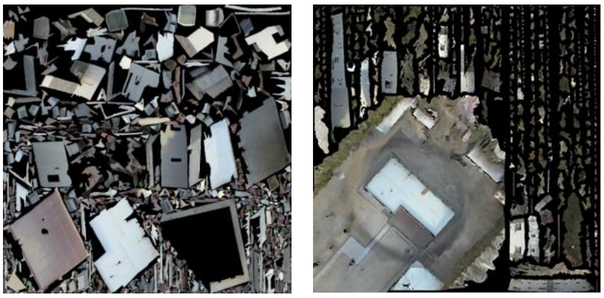

(a) (b)

Figure 4: Unrectified texture map images generated with the OBJ mesh using two different photogrammetry software.

The 3D mesh of the building was unwrapped to match the texture map. This is a UV map which results from converting the mesh vertices’ 3D coordinates to their respective UV coordinates. The texture map was rectified automatically by recalculating the UV coordinates of the mesh using the least squares conformal maps (LSCM) unwrapping algorithm (Lévy et al., 2002). This resulted in a new UV map for the 3D mesh. The mesh with the new UV mapping was saved as a separate file. The texture map was reorganized in the same layout as the rectified UV map through texture baking by aligning the two meshes then casting rays from the first mesh to the second to record the texture detail on the second mesh, while using the first mesh’s UV coordinates. The resulting OBJ mesh was unwrapped and overlaid over the rectified texture map image.

By comparing the rectified texture map with the actual building, the wall, window, and roof pixels were annotated manually. Figure 5 shows polylines drawn manually over the texture map. For the prototype, only the window polylines were relevant, since other methods were developed to extract wall and roof elements. The arbitrary (or manually drawn) 2D polylines can be converted to 3D geometry corresponding to their locations on the mesh by calculating the barycentric coordinates explained in the following.

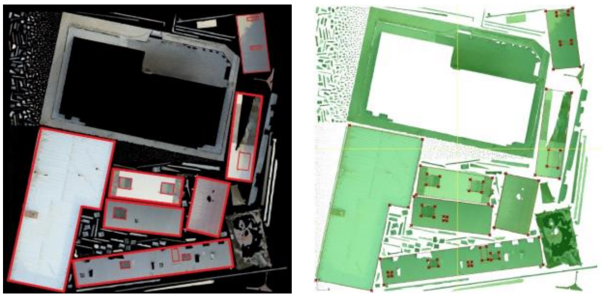

(a) (b)

Figure 5: Manually annotated texture maps. (a) Hypothesis of extracted contours manually drawn on the texture map. (b) Implementation of polyline contours for barycentric interpolation.

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* (λ_a, λ_b, λ_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 illustrated in Figure 5 (b), each corner point of the annotations is inscribed inside a triangular face within the unwrapped 2D mesh. 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 6). The point *p* in Figure 6 was created by manual annotation. Once the barycentric coordinates of the arbitrary 2D texture coordinates relative to the corresponding 3D faces are calculated, the annotated 2D polylines can be redrawn on their exact 3D locations on the mesh. The polylines shown in Figure 5 (b) were used to parse the mesh and generate the corresponding 3D points for the 2D polylines' corner points drawn on the texture map.

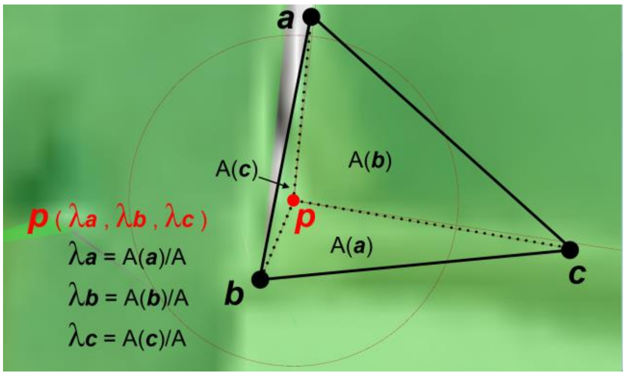


Figure 6: Barycentric coordinate calculation. The point *p* represents a polyline corner point, while the triangle *abc* represents a mesh face.

The polylines have no semantic labeling; they are boundary contours with no classification on which contour is a wall, window, or roof. However, if a photograph or a 3D model of a building can be recognized through human knowledge, then we can program a computer to do the same by programming geometry constraints and rules to extract wall, window, and roof features (Pu et al., 2006). 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 interpolated 3D polylines were 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 window areas were labeled automatically (Figure 7).

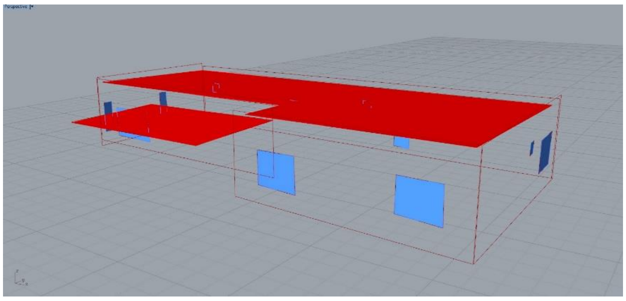


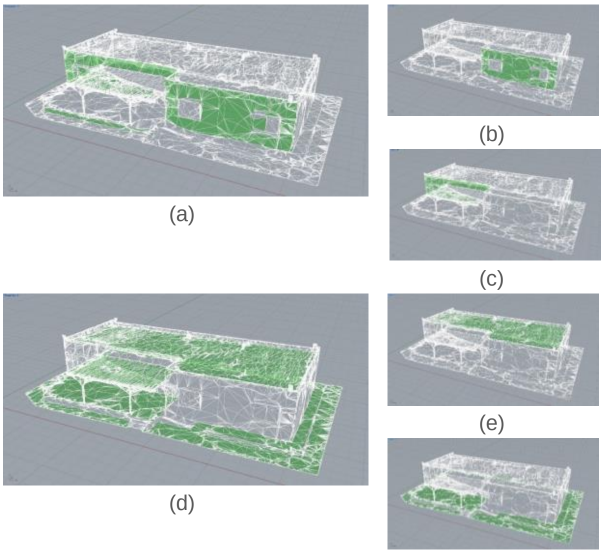
Figure 7: Automatically labeled window surfaces.

EnergyPlus can accept rectangular and triangular geometry for window surfaces. Since the window polylines were not true rectangles, a bounding box algorithm was applied for each window, and a genetic algorithm was used to rotate each bounding box for achieving the minimum area for each bounding box, and then rectangular windows – the final bounding boxes – were generated.

MESH GEOMETRY PROCCESSING

Each face of the mesh has a normal vector with three component values between -1 and 1 in the Cartesian coordinate system. Based on this data, to simplify the mesh for BEM, surfaces of the building that share a similar orientation were categorized. For example, mesh faces that represent walls that face a similar direction are in the same 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 introduced 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 produced more possible categories that represented more orientations since the Cartesian values that were categorized are -1, -0.5, 0, 0.5, and 1 for each of *x*, *y*, and *z* component in the normal vector values. After this process was applied to the normal vector values of all the faces in the input mesh, a mathematical finite set function was applied to group all the unique normal vector values into a category. The maximum number of possible orientation categories based on normal vector direction was 98. 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 was 74.

Segmenting the mesh into sub-meshes based on the normal vector direction was not sufficient to categorize building wall and roof areas. The resulting categorization was that any surfaces facing a similar direction were categorized as one sub-mesh group. Figure 8 (a) illustrates two wall surfaces in one category. The prototype introduced an algorithm by Mans (n.d.) to separate unwelded meshes in each category into new categories. Applying the algorithm to the mesh further segmented the wall meshes showed previously in Figure 8 (a) into separate categories as shown in Figure 8 (b) and (c).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 8: 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).

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

ENERGYPLUS MODEL GENERATION

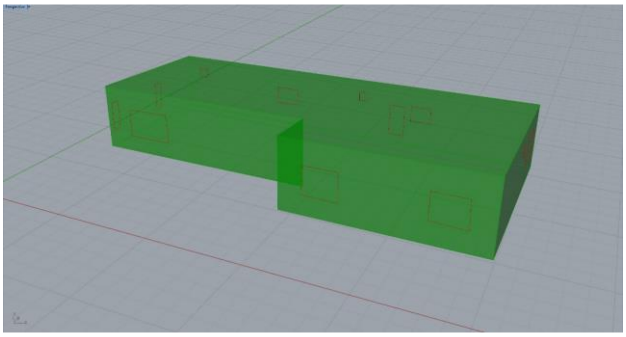


Figure 9: Closed planar BREP with projected window polylines.

After separating the mesh segments into groups, a least square plane fitting algorithm was applied on the vertices in each group. The vertices are then projected on the fitted planes. Subsequently, a convex hull algorithm was applied for each vertex group. The resulting convex surfaces were scaled by a factor of 2. The surfaces were intersected using a boundary volume algorithm and a closed BREP geometry was computed. The BREP was oriented by using the largest generated convex hull surface and assuming that it represents the ground surface. This surface's plane was then re-mapped on the world coordinate system's XY

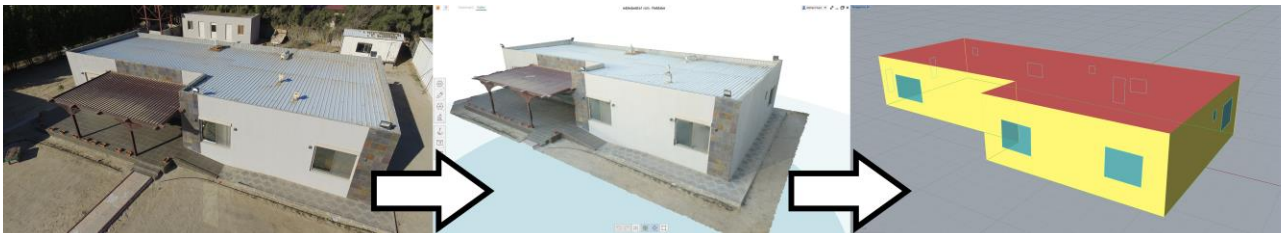


Figure 10: Left: UAV photograph of the farmhouse. Middle: 3D photogrammetry mesh. Right: Visualized EnergyPlus model.

plane, and the BREP was subsequently oriented. A variable was used to adjust the rotation of the building model relative to the cardinal directions. The window corner points were projected to the BREP as shown in Figure 9.

The BREP model with co-planar windows was converted automatically into an EnergyPlus thermal zone including the walls, windows, floor, ceiling, and roof by using Ladybug and Honeybee (Sadeghipour Roudsari & Pak, 2013). A number of assumptions were made for describing the building energy model. Since the building model in the prototype was a single closed BREP model, only 1 zone was specified; the resulting energy model was considered as a single-zone building energy model that did not consider the internal walls and rooms within the building. This zone was specified to be air-conditioned. As for the zone program, which includes materials and constructions, Honeybee was used to access the OpenStudio library which provides a list of pre-defined building programs that can be used as templates for EnergyPlus inputs. The usual energy modeling process requires strict geometry rules, manual work, and debugging, and since the input closed BREP model was compatible for BEM, it was successfully converted into a defined thermal zone without issues. Honeybee was used to input the weather file, simulation analysis period, simulation outputs, and shadow parameters. The simulation analysis period was defined from June until September. The selected EnergyPlus outputs for the simulation were solving for heating, cooling, and electricity loads, in addition to calculating the mean air temperature, operative temperature, and relative humidity. The resulting output was written as an EnergyPlus model in IDF format (Figure 10).

VERIFICATION

The objective of the prototype was not to conduct a calibrated energy simulation, but to produce a verified one. In the context of modeling and simulation, while validation is making sure that the model represents the real-world conditions, verification is making sure that a model is debugged and works as intended (Chung, 2003). The IDF file was 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. The cooling, lighting, and equipment loads from June to September were simulated. The cooling set point was 23.9 C° (75.02 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 were not verified or calibrated against actual load measurements. Table 1 contains EnergyPlus inputs that were automatically generated from the OpenStudio library.

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 9/30
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	23.9 C° (75.02 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.

An alternative simulation run was conducted by specifying the building zone as unconditioned. The appropriate graphical charts were generated automatically using Ladybug. Figure 11 shows the plotted psychrometric chart for the unconditioned zone. The plotted hours for dry-bulb temperatures in degrees Celsius and humidity ratios appeared to be reasonable for the simulated location in Kuwait from June to September.

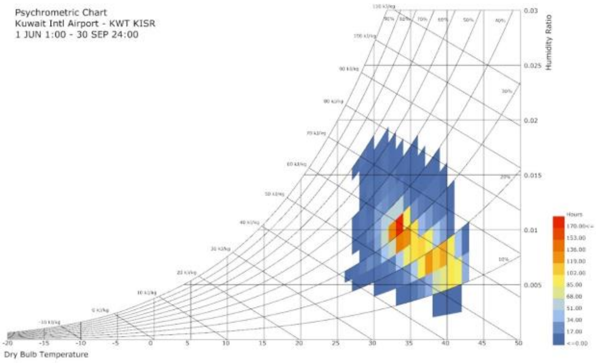


Figure 11: Dry-bulb temperatures and humidity ratios for the unconditioned zone from June to September (Chart was generated using Ladybug).

DISCUSSION AND CONCLUSIONS

For the purposes of environmental assessments and the sustainable management of existing buildings, the methodology presented in this paper can be applied to create a semi-automated workflow to generate a building energy model from site photographs of a building. The workflow consists of algorithms and tools accessible to architects and designers. This could reduce time and labor costs by automating many of the processes involved in creating 3D geometry for BEM, and by taking the 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 by requesting minimal user input and providing quick, visual, and graphical assessments for a stakeholder interested in understanding the environmental performance of an existing building.

LIMITATIONS

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

FUTURE WORK

Further technological improvements and developments in computer processing power, photogrammetric 3D reconstruction, image recognition, and machine learning (ML) can be applied within the framework of the presented methodology, where building constructions and materials are automatically labeled from photogrammetry data, and environmental assessments for large scale urban models can be realized. Time and labor costs which are inherent with the BEM process can be further reduced by capitalizing on new developments in the field of 3D reconstruction and image analysis by adding information (i.e., semantics) from the physical and visual world and applying it to BEM. Bridging different processes together can potentially enable any stakeholder, either an owner, architect, or engineer to measure the environmental performance of an existing building.

REFERENCES

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

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. doi:10.1145/2421636.2421642

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

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

Deodhar, A. (2015). Autodesk Presents: Rapid Energy Modeling for Existing Buildings. Presented at the Greenbuild International Conference and Expo 2014, New Orleans, LA. Retrieved from https://www.youtube.com/watch?v=0Bm-GvQ7Ww4&t=9s

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

Haala, N., & Kada, M. (2010). An Update on Automatic 3D Building Reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing, 65(6), 570–580. Retrieved from https://doi.org/10.1016/j.isprsjprs.2010.09.006

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.

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. (n.d.). Mesh+ (Version 2.1) [Computer software]. Retrieved from http://www.food4rhino.com/app/mesh

Ochmann, S., Vock, R., Wessel, R., & Klein, R. (2016). Automatic Reconstruction of Parametric Building Models from Indoor Point Clouds. Computers & Graphics, 54, 94–103. Retrieved from http://dx.doi.org/10.1016/j.cag.2015.07.008

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

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

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

Sadeghipour Roudsari, Mostapha; Pak, Michelle, 2013. Ladybug: a parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design. In: Proceedings of the 13th International IBPSA Conference Held in Lyon, France Aug 25–30th. Retrieved from http://www.ibpsa.org/proceedings/BS2013/p_2499.pdf

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

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

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