Integrating Sensors on a Smartphone to Generate Texture Images of 3D Photo-realistic Building Models

Sendo Wang

Assistant Professor, E-Mail: sendo@ntnu.edu.tw
Department of Geography, National Taiwan Normal University
162 HePing East Rd. Sec. 1, Taipei 10610, Taiwan
Tel: +886-2-77341683; FAX: +886-2-23691770

Abstract

A 3D photo-realistic building model which can represent the real appearance of the building should be composed by precise geometric model and realistic façade images. There are a number of approaches to reconstructing the geometric model from photogrammetric images and LiDAR points cloud. However, the façade texture generation still relies on massive labor works, therefore, are still the bottle neck in the photo-realistic building modeling process. The emphasis of this paper will be put on the integration of sensors on a smartphone. While the camera is collecting the realistic photo of the building façade, the built-in GPS receiver and G-sensors are recording the approximate 3D coordinates and 3 rotation angles of the exposure station. However, these data are not accurate enough for the texture image generation. A semi-automated approach is proposed to improve the accuracy of the position and orientation data and to generate the precise façade texture images. The reconstruction of 3D realistic building models consists of three major issues: (1) geometrically modelling the object; (2) determining the image orientation; (3) generating the realistic façade texture from photographs. By introducing the "Floating Model" concept, the object modelling and image orientation problems can be solved efficiently through the semi-automated procedures based on the Least-squares Model-image fitting (LSMIF). A friendly human-machine inter-acting interface is designed for an operator to choose suitable model. Then the operator can move, rotate, or resize the model to approximately fit among all of the images. An ad-hoc LSMIF algorithm is developed to solve the optimal fitting between projected model line segments and extracted edge pixels. Since the object model can be extracted and the photo orientation can be determined, the creation of realistic texture image, which is also called inverse mapping, can be automated by coordinate transforming and image resampling. For better understanding the camera characteristics on a smartphone, a series of camera calibration process has been executed to derive interior orientation parameters before taking façade pictures. Three representative buildings in NTNU campus are selected for the experimental tests. The geometric models are reconstructed by fitting floating models to aerial photogrammetric images. The façade photos are taken by smart phones on the ground. The result shows that the proposed approach is practicable, but the lens distortion must be corrected before creating texture image. Since the iterative LSMIF algorithm requires initial parameters, the position and pose derived from built-in sensors must fall in the pull-in range.

Keywords: Three-dimensional Building Modelling, Photo-realism City Model, Texture Rendering, Direct Georeferencing, Automation, Digital Close-range Photogrammetry, Virtual Reality

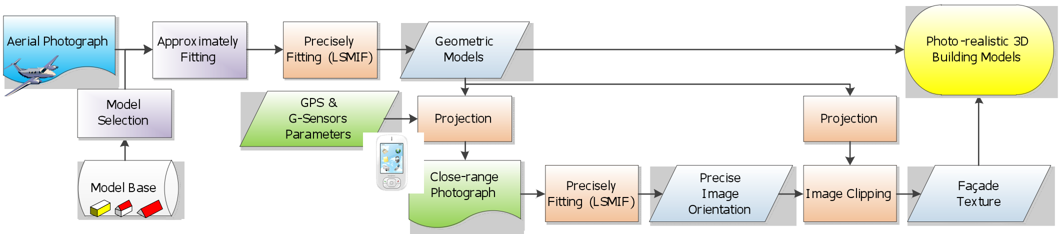
1 Introduction

The photo-realistic 3D building model does not only present the building in the three-dimension perspective but also reflects its real appearance. There are a number of approaches for reconstructing the geometric model from photogrammetric images, from LiDAR point cloud, or from both of them (Braun, et al. 1995; Chapman, et al. 1992; Förstner, 1999; Grün, 2000, Lang and Förstner, 1996; Lowe, 1991; Tseng and Wang, 2003; Veldhuis, 1998; Wang and Tseng, 2009). However, the façade mapping relies on the manual operations to create texture images is still the bottle neck of the photo-realistic building modeling. The recent mobile computing devices, such as smartphones and tablet PCs, usually equip with not only high-resolution camera but also built-in GPS receiver and G-sensors. These sensors can be used for the direct geo-referencing while taking pictures of buildings. This paper proposes the concept of a semi-automated approach to modeling photo-realistic 3D buildings using a smartphone or a tablet PC. When the picture is taken, the device's 3D coordinates are recorded from the built-in GPS receiver and its three rotation angles are also recorded from the G-sensors. However, these parameters are too rough to reconstruct the object space stereo model by Photogrammetric means. Therefore, a model-image fitting algorithm is necessary to determine precise image orientation.

The reconstruction of photo-realistic 3D building models consists of three major issues: (1) modeling the object; (2) determining the image orientation; (3) creating the realistic texture image from photos. In this paper, the aerial photographs are used to reconstruct the geometric models of buildings, while the pictures taken by the personal computing device is used as the façade texture. By introducing the "Floating Model" concept, the object modeling and image orientation problem can be solved efficiently through the semi-automated procedures based on the Least-squares Model-image fitting (LSMIF). A friendly human-machine inter-acting interface program is designed for an operator to choose suitable model, and to move, to rotate, or to resize the model so it can approximately fit to all of the images. An ad-hoc Least-squares Model-image Fitting algorithm is developed to solve the optimal fitting between projected model line segments and extracted edge pixels. Since the object model can be extracted and the photo orientation can be determined, the creation of realistic texture image, which is also called inverse mapping, can be automated by coordinate transforming and image resampling. Figure 1 shows the workflow of the proposed semi-automated photo-realistic 3D building modeling procedures.

In the proposed workflow, there are two procedures - “model selection” and “approximately fitting” requires human interactions. This is because manual image interpretation is more robust and more efficient than computer algorithms. While the other computational work, such as “model projection”, “precisely fitting”, and “Image Clipping”, are carried out by computer algorithms. Therefore, the proposed semi-automated procedures shall improve the efficiency from the full-manual methods, while remain robust than full-automated approaches.

Figure 1: The proposed semi-automated procedure

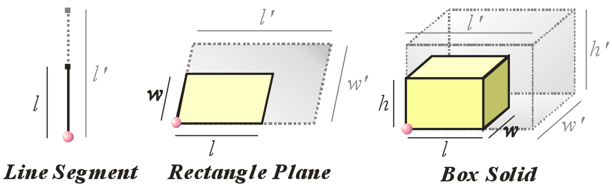


2 Model-image Correspondance

To deal with the modeling problem, this paper introduced the concept of floating models. The floating models can be categorized into four types: point, linear feature, plane, or volumetric solid. Each type contains various primitive models for the practical needs. For example, the linear feature includes the line segment and the arc. The plane includes the rectangle, the circle, the ellipse, the triangle, the pentagon, etc. The volumetric solid includes the box, the gable-roof house, the cylinder, the cone, etc. Despite the variety in their shape, each primitive model commonly has a datum point, and is associated with a set of pose parameters and a set of shape parameters. The datum point and the pose parameter determine the position of the floating model in object space. It is adequate to use 3 translation parameters (dX, dY, dZ) to represent the position and 3 rotation parameters, tilt (t) around Y-axis, swing (s) around X-axis, and azimuth (α) around Z-axis to represent the rotation of a primitive model. Figure 2 shows four examples from each type of models with the change of the pose parameters. X'-Y'-Z' coordinate system defines the model space and X-Y-Z coordinate system defines the object space. The little pink sphere indicates the datum point of the model. The yellow primitive model is in the original position and pose, while the grey model depicts the position and pose after changing pose parameters (dX, dY, dZ, t, s, α). The model is “floating” in the space by controlling these pose parameters. The volume and shape of the model remain the same while the pose parameters change. The shape parameters describe the shape and size of the primitive model, e.g., a box has three shape parameters: width (w), length (l), and height (h). Changing the values of shape parameters elongates the primitive in the three dimensions, but still keeps its shape as a rectangular box. Various primitive may be associated with different shape parameters, e.g., a gable-roof house primitive has an additional shape parameter – roof’s height (rh). Figure 3 shows three examples from each type of models with the change of shape parameters. The point is an exceptional case that does not have any shape parameters. The yellow one is the original model, while the grey one is the model after changing the shape parameters. The figure points out the other important characteristic of the floating model – the flexible shape with certain constraints. Changing the shape parameters does not affect the position or the pose of the model.

Figure 2: Pose parameters adjustment of floating models

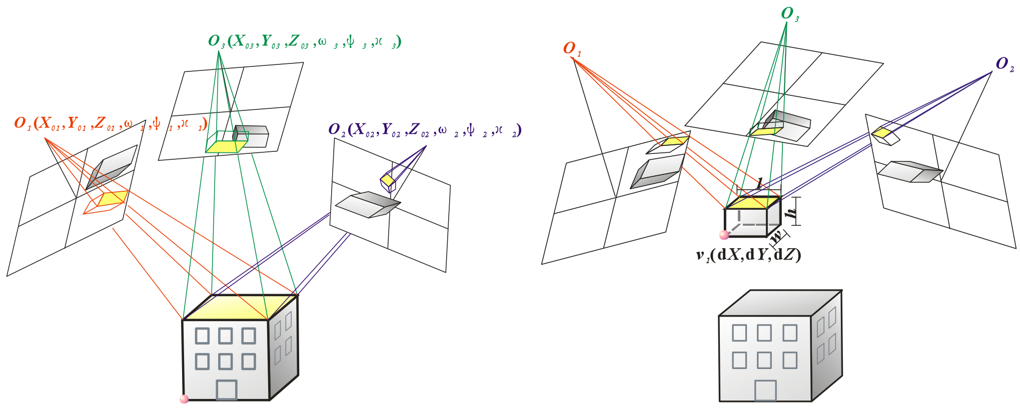
Figure 3: Shape parameters adjustment of floating models



When the pictures are taken, buildings are projected on the photo based on the central projection. The object point, exposing center, and the image point should lie on the same straight line. This is the foundation of the collinearity equations. If the object point is expanded to a volumetric solid, there would be unlimited rays along the boundary. When re-project a floating model onto the taken photo, all of the model parameters and image orientation parameters must be accurate so the wireframe model can perfectly superimpose on the building’s image. Figure 4 shows the examples either the model parameters or the image orientation parameters were wrong.

Figure 4: Precise image orientation and model parameters are both required for correct model projection

(a) Incorrect image orientation parameters **(b) incorrect model parameters**



In this paper, the image orientation parameters of the aerial photographs are known and fixed, while floating models are used for reconstructing geometric 3D building models. These reconstructed geometric models are later used for refining the image orientation parameters of the close-range photographs taken by personal mobile computing devices. So the reconstructed model parameters remain fixed while fitting floating model to the close-range photographs.

3 Least-squares Model-image Fitting

The principle of model-image fitting algorithm is to adjust either model parameters or the image orientation parameters, so the model projection fit the building images. Since the floating model can be taken as a wire-frame model, the edge pixels are selected as fitting targets. The optimal fit is achieved by minimizing the sum of the perpendicular distances from the edge pixels to the corresponding projected line of the wire-frame model. Figure 5 depicts the optimal fitting procedure. The selected primitive model is projected onto the image and fit the extracted edge pixels.

Figure 5: Optimal model-image fitting

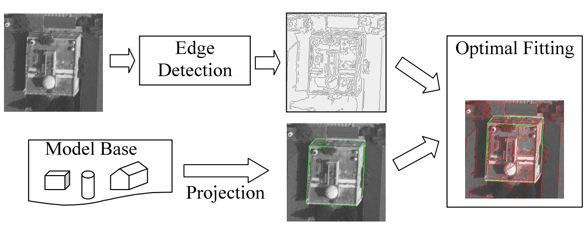
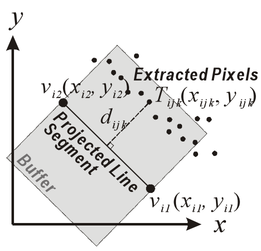


Figure 6: Extracted edge pixels and buffer



Either for geometric modeling or image orientation, an approximate fitting is required before applying the LSMIF algorithm. An interactive program is developed for model selection, approximate fitting, and visualization. To obtain as close as to the right fitting, this program provides a user interface that allows the operator to resize, rotate, and move a model to fit the corresponding building images approximately. Benefited from the approximate fitting, the LSMIF iteratively pulls the model to the optimal fit instead of blindly searching for the solution. To avoid the disturbance of irrelevant edge pixels, only those edge pixels distributed within the specified buffer zones will be used in the calculation of the fitting algorithm. Figure 6 depicts the extracted edge pixels T_{ijk} and the buffer determined by a projected edge $\overline{v_{i1}v_{i2}}$ of the model. The suffix *i* represents the index of edge line, *j* represents the index of overlapped image, and *k* represents the index of the edge pixel. Filtering edge pixels with buffer is reasonable, because the discrepancies between the projected edges and the corresponding edge pixels should be small, as either the model parameters or the image orientation parameters are approximately known.

The optimal fitting condition we are looking for is the projected model edge line exactly falls on the building edges in the images. In Eq.(1), the distance d_{ijk} represents a discrepancy between an edge pixel T_{ijk} and its corresponding edge line $\overline{v_{i1}v_{i2}}$, which is expected to be zero. Therefore, the objective of the fitting function is to minimize the squares sum of d_{ijk}. Suppose a projected edge line is composed of the projected vertices $v_{i1}(x_{i1}, y_{i1})$ and $v_{i2}(x_{i2}, y_{i2})$, and there is an edge pixel $T_{ijk}(x_{ijk}, y_{ijk})$ located inside the buffer. The distance d_{ijk} from the point T_{ijk} to the edge $\overline{v_{i1}v_{i2}}$ can be formulated as the following equation:

$$d_{ijk} = \frac{\left|(y_{ij1} - y_{ij2})x_{jk} + (x_{ij2} - x_{ij1})y_{jk} + (y_{ij2}x_{ij1} - y_{ij1}x_{ij2})\right|}{\sqrt{(x_{ij1} - x_{ij2})^2 + (y_{ij1} - y_{ij2})^2}} \tag{1}$$

where i = the index of the edge line
j = the index of the overlapped image
k = the index of the edge pixel

The photo coordinates $v_{i1}(x_{i1}, y_{i1})$ and $v_{i2}(x_{i2}, y_{i2})$ are functions of the unknown model parameters, comparatively the exterior-orientation parameters of photos are known. Therefore, d_{ijk} will be a function of the model parameters. Taking a box model for instance, d_{ijk} will be a function of *w*, *l*, *h*, *α*, d*X*, d*Y*, and d*Z*, with the hypothesis that a normal building rarely has a tilt angle (*t*) or swing angle (*s*). The least-squares solution for the unknown parameters can be expressed as:

$$\Sigma d_{ijk}^2 = \Sigma[F_{ijk}(w, l, h, \alpha, dX, dY, dZ)]^2 \rightarrow min. \tag{2}$$

Eq.(2) is a nonlinear function with regard to the unknowns, so that the Newton's method is applied to solve for the unknowns. The nonlinear function is differentiated with respect to the unknowns and becomes a linear function with regard to the increments of the unknowns as follows:

$$d_{ijk} - F_{ijk0} = \left(\frac{\partial F_{ijk}}{\partial w}\right)_0 \Delta w + \left(\frac{\partial F_{ijk}}{\partial l}\right)_0 \Delta l + \left(\frac{\partial F_{ijk}}{\partial h}\right)_0 \Delta h + \left(\frac{\partial F_{ijk}}{\partial \alpha}\right)_0 \Delta \alpha + \left(\frac{\partial F_{ijk}}{\partial dX}\right)_0 \Delta dX + \left(\frac{\partial F_{ijk}}{\partial dY}\right)_0 \Delta dY + \left(\frac{\partial F_{ijk}}{\partial dZ}\right)_0 \Delta dZ \tag{3}$$

in which, F_{ijk0} is the approximation of the function F_{ijk}, calculated with given approximations of the unknown parameters. Given a set of unknown approximations, the least-squares solution for the unknown increments can be obtained, and the approximations are updated by the increments. Repeating this calculation, the unknown parameters can be solved iteratively. Eq.(2) and Eq.(3) are used for geometric model reconstruction. As for image orientation determination, they are modified as Eq.(4) and Eq.(5). The unknowns becomes the increments of the image orientation parameters.

$$\Sigma d_{ijk}^{\ 2} = \Sigma[F_{ijk}(\omega, \varphi, \kappa, X_0, Y_0, Z_0)]^2 \ \rightarrow min. \tag{4}$$

$$d_{ijk} - F_{ijk0} = \left(\frac{\partial F_{ijk}}{\partial \omega}\right)_0 \Delta \omega + \left(\frac{\partial F_{ijk}}{\partial \varphi}\right)_0 \Delta \varphi + \left(\frac{\partial F_{ijk}}{\partial \kappa}\right)_0 \Delta \kappa + \left(\frac{\partial F_{ijk}}{\partial X_0}\right)_0 \Delta X_0 + \left(\frac{\partial F_{ijk}}{\partial Y_0}\right)_0 \Delta Y_0 + \left(\frac{\partial F_{ijk}}{\partial Z_0}\right)_0 \Delta Z_0 \tag{5}$$

The linearized equations can be expressed as a matrix form: $\boldsymbol{V}=\boldsymbol{AX}-\boldsymbol{L}$, where $\boldsymbol{A}$ is the matrix of partial derivatives; $\boldsymbol{X}$ is the vector of the increments; $\boldsymbol{L}$ is the vector of approximations; and $\boldsymbol{V}$ is the vector of residuals. The objective function actually can be expressed as $q=\boldsymbol{VTV}\rightarrow min$. For each iteration, $\boldsymbol{X}$ can be solved by the matrix operation: $\boldsymbol{X}=(\boldsymbol{A}^{T}\boldsymbol{A})^{-1}\boldsymbol{A}^{T}\boldsymbol{L}$. The iteration normally will converge to the correct answer. However, inadequate relevant image features, affected by irrelevant features or noise, or given bad initial approximations may lead the computation to a wrong answer.

4. Conclusions

Photo-realistic 3D building models are the basic geospatial information infrastructure for many applications. This paper proposed a semi-automated reconstruction approach to overcome the bottleneck. Instead of using the precise and expensive mobile mapping instruments, the personal mobile computing devices are used to collect façade images of the buildings. Benefit from the built-in GPS receiver and G-sensors, the approximate image orientation parameters are directly recorded as the picture was taken. Then the orientation is refined by fitting model to image, which is automated computed by an *ad hoc* Least-squares Model-image Fitting algorithm. Some experiments are still undergoing, so the results will be presented in the conference in an interactive way.

ACKNOWLEDGEMENT

This research is sponsored under the project (NSC 100-2221-E-003 -025) from the National Science Council of Republic of China. The author would also like to show his gratitude to the National Taiwan Normal University for supporting his travel grant.

REFERENCE

Braun, C., Kolbe, T. H., Lang, F., Schickler, W., Steinhage, V., Cremers, A. B., Förstner, W., and Plümer, L., 1995. "Models for Photogrammetric Building Reconstruction." Computers & Graphics, 19(1), pp. 109-118.

Chapman, D. P., Deacon, A. T. D., Hamid, A. A., and Kotowski, R., 1992. "CAD Modeling of Radioactive Plant - The Role of Digital Photogrammetry in Hazardous Nuclear Environments." Proceedings of the XIXth Congress of the International Society for Photogrammetry and Remote Sensing, Washington D.C., USA, Vol. XXIX, Part B5, pp. 741-753.

Förstner, W., 1999. "3D-City Models: Automatic and Semiautomatic Acquisition Methods." Photogrammetric Week '99, Stuttgart, Germany, pp. 291-304.

Grün, A., 2000. "Semi-automated Approaches to Site Recording and Modeling." Proceedings of the XIXth Congress of the International Society for Photogrammetry and Remote Sensing, Amsterdam, The Netherlands, Vol. XXXIII, Part B5, pp. 309-318.

Lang, F., and Förstner, W., 1996. "3D-City Modeling with a Digital One-eye Stereo System." Proceedings of the XVIIIth Congress of the International Society for Photogrammetry and Remote Sensing, Vienna, Austria, Vol. XXXI, Part B, pp. 415-420.

Lowe, D. G., 1991. "Fitting Parameterized Three-Dimensional Models to Images." IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(5), pp. 441-450.

Tseng, Y.-H., and Wang, S., 2003. "Semi-automated Building Extraction Based on CSG Model-Image Fitting." Photogrammetric Engineering & Remote Sensing, 69(2), pp. 171-180.

Veldhuis, H., 1998. "Performance Analysis of Two Fitting Algorithms for the Measurement of Parameterised Objects." International Archives of Photogrammetry and Remote Sensing, Columbus, OH, USA, Vol. XXXII, pp. 400-406.

Wang, S. and Tseng, Y.-H., 2009. “Least-Squares Model-Image Fitting of Floating Models for Building Extraction from Images”, Journal of the Chinese Institute of Engineers, 32(5), pp.667-677.