SILHOUETTE BASED RECONSTRUCTION OF SYMMETRICAL BUILDINGS

SAMUELCIK Martin (SK)

Abstract. This method is designed for a 3D reconstruction of buildings from terrestrial images based on a visual hull method. Our approach is based on silhouette extracting from an image, extruding the silhouette in a given direction (based on the image orientation in the real world) so we get a silhouette prism and then we perform an intersection of these prisms. This intersection is called a visual hull of the object and with some error it approximates the original building. Images are then reused for texture mapping on a resulting solid. As a result we usually have a polygonal 3D model of the building with a low count of polygons. The accuracy of this model depends on the camera position for the given images and the shape of the building, but if no high accuracy is needed, this approach is useful, e.g. for creating building models as a part of the virtual city model.

Keywords**: silhouette, building reconstruction, visual hull, solids intersection, image unwarping, texture mapping**

1 Introduction

We present in this paper an algorithm for reconstructing building from terrestrial images. This approach can be used with a good precision on symmetrical buildings and if we do not need a model with a high count of polygons. This can be used for creating models that can be accessed via Internet.

As an input we need at least two terrestrial images of the building. For each terrestrial image we also need an angle of the directional vector. This directional vector describes direction of the camera in the given image related to the 3D world coordinate system. We call this angle azimuth (e.g. if we are looking in the image on the building from the north, the azimuth is equal to 0). If the footprint of the building is given, we can use it as another optional input parameter for a more accurate reconstruction.

Output of this method is a 3D model of the building. We can view and store this model in several ways. This approach can render the model in the window using OpenGL or save it into VRML file and so it can be viewed via Internet. It also has its inner format for storing geometry and topology of the model.

2 Related work

There are many papers related to building reconstruction from terrestrial images. Most of them are based on finding projection matrices for each image, detection of corresponding objects (points, lines, windows, …). As a result we can have point clouds [5] or polygonal model [3]. The visual hull method (intersecting of silhouette cones) is usually used for reconstruction of small objects from a set of images using a volumetric intersection [1], [2]. Another algorithm with a similar approach is used for generating models of trees from terrestrial images using B-rep [4]. The exact position of the camera must be given. Then the silhouette is extruded into the cone with its top at the camera location and the intersection is calculated.

3 Process description

The main idea in our approach is slightly different from presented works. In many cases the buildings have an important property: The opposite lines and faces are parallel. So we extrude a silhouette not into a cone, but into a prism and then the opposite lines will be parallel. If the footprint of the building is given, we get more accurate result by extruding that footprint in the direction of z axis. By intersecting of these prisms we get a visual hull of the building that can be treated as its 3D reconstruction. The advantage of this approach is that we don’t need the exact position of the camera, just the direction in which we are going to extrude the silhouette. The main disadvantage is that for more accurate model we need an ortho image of a building, so images need to be unwarped. This still leaves some scaling errors, but if images are taken from a longer distance from the building and the viewing direction is perpendicular to the facade, the result is more accurate. Also the algorithm is more successful in the reconstruction of symmetrical buildings like towers. The steps of the algorithm are as follows.

3.1 Image unwarping

We have to unwarp image to obtain the one that is similar to an orthophoto image. The idea is to find two lines that are parallel in the real world and then to unwarp the image so that these two lines become parallel. Figures 1 and 2 give an example of this process.

3.2 Silhouette detection and extruding

There exist algorithms for finding a silhouette in the image based on boundary detection [6],[7]. But these algorithms are unusable for complex image with more then one building. Then the silhouette can not be found automatically. Our idea is to find the corners in the image and then input the silhouette manually by using only these corner points. In the Figure 5 is drawn such silhouette of St. Michaels tower in Bratislava.

Given image azimuth is then used for direction calculation and then the silhouette is extruded in that direction into prism. If the footprint is given, it’s extruded too. The silhouette solids are then stored. Figure 3 shows example of two silhouettes extruded into prisms.

3.3 Solids scaling and intersection

All stored solids are scaled so that the height is equal for each silhouette solid. If the footprint of the building is given, solids are scaled so the width of solids from given azimuths is equal to the footprint width from given azimuth.

An intersection of all stored silhouette solids is computed and the resulting solid is stored, this is illustrated in the Figure 4. We are using our own function for solids intersection computation.

3.4 Texture mapping

Each input unwarped image is back projected on the resulting solid in the direction of the azimuth. Then for each face containing a part of actual texture the texture mapping coordinates are calculated. If the aerial image of a building is given, it can be used, too. If the textures for some faces are missing, the textures from an opposite face can be reused.

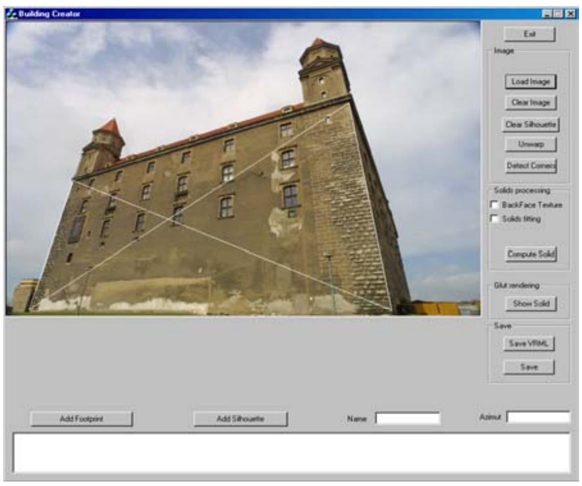

Figure 1: Image of Bratislava castle before unwarping process with auxiliary lines

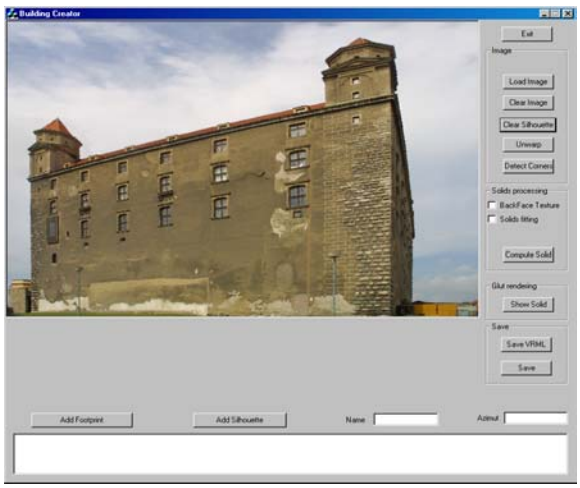

Figure 2: Image of Bratislava castle after one step of unwarping process

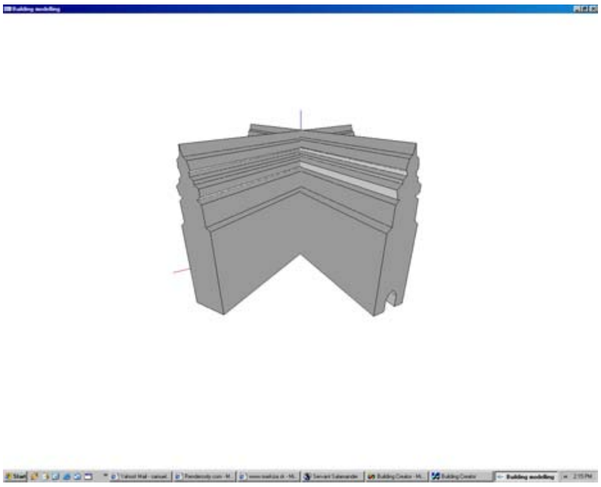

Figure 3: Example of silhouettes extruded into prisms

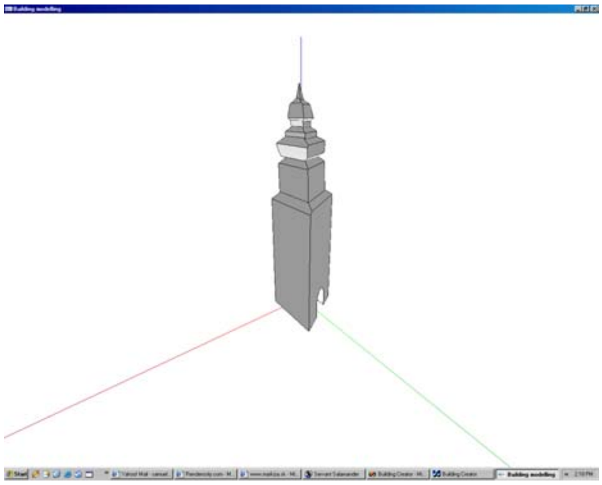

Figure 4: Intersection of prisms from Figure 3

4 Results

The presented algorithm has been implemented in Microsoft Visual C++ 6.0 environment. For 3D visualization we are using Open GL with Glut extension. In this stage the algorithm and the implementation give useful results. For storing solids we are using boundary representation. Our own classes handle the intersection of solids. The result can be stored in files using VRML file format or an inner file format. Figures 5 - 8 show images of resulting application called Building Creator and 3D models of some reconstructed buildings. In the Table 1 we present polygon count and input parameters for the presented models. As we can see, we get good textured models with low count of polygons in a few minutes and with a few input parameters. Reconstruction of these buildings with modelling software will take much more time.

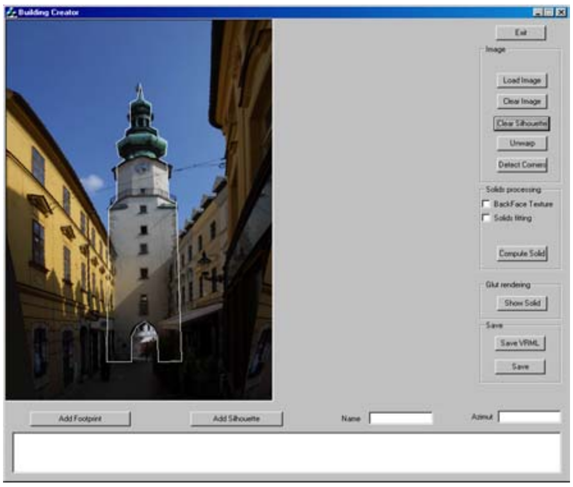

Figure 5: Main window of Building Creator with an image of St. Michael's Tower in Bratislava with silhouette

Figure 6: 3D model of St. Michael's Tower in Bratislava rendered in Glut window. For modeling and texture mapping was used image from Figure 5

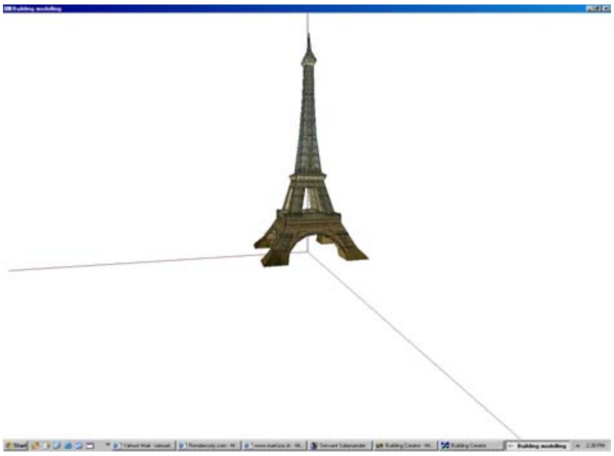

Figure 7: 3D model of the Eiffel Tower in Paris

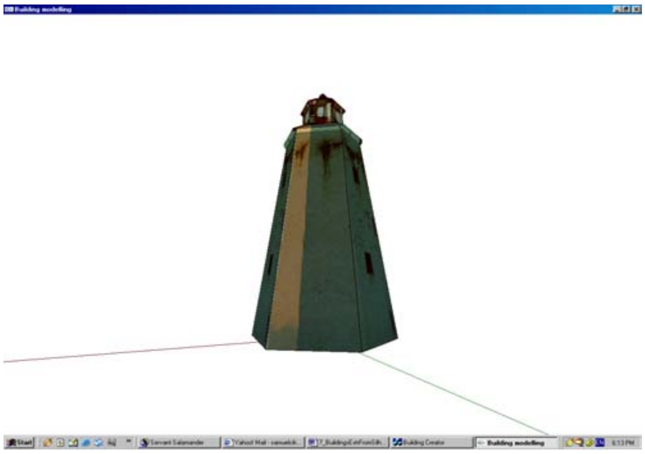

Figure 8: 3D model of lighthouse

Model	Input parameters	Polygons count	Time (minutes)
St. Michael's Tower	1 image, 2 azimuths	64	5
Eiffel Tower	1 image, 2 azimuths	120	4
Lighthouse	1 image, 3 azimuths	62	3,5

Table 1: Complexity of input and output of presented models

5 Acknowledgements

This work within the Virtual Environments for WWW project has been supported by VEGA Grant Agency under contract 1/0174/03.

Literature

[1] Dyer, Ch. R., *Volumetric Scene Reconstruction from Multiple Views*, in Foundations of Image Understanding, pp. 469-489, Kluwer Boston, 2001.

[2] Ramanathan, P., Steinbach, E., and Girod, B., *Silhouette-based Multiple-View Camera Calibration,* Proc. Vision, Modelling and Visualization 2000, pp. 3-10, Saarbruecken, Germany, November 2000.

[3] Debevec, P. E., Taylor, C. J. and Malik, J., *Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach*, In Computer Graphics (SIGGRAPH '96 Proceedings), volume 30, pp. 11-20, New Orleans, Louisiana, 1996.

[4] Shlyakter, I., Rozenoer, M., Dorsey, J. and Teller, S., *Reconstructing 3D Tree Models from Instrumented Photographs*, in IEEE CG&A 21(3), pp. 53-61, May/June 2001.

[5] Klaus, Bauer, J., Karner, K. and Schindler, K., *MetropoGIS: A Semi-Automatic Documentation System*, in PCV02, Graz, 2002.

[6] Ma, W. Y. and Manjunath, B. S., *Edge Flow: A Framework of Boundary Detection and Image Segmentation*, Proc. IEEE Conf on Computer Vision and Pattern Recognition, 1997.

[7] Desolneux A., Moisan L. and Morel, J.M. *Edge detection by the Helmholtz principle*, JMIV, 14(3): 271-284, 2001.

Contact address

Martin Samuelcik, Mgr., KPGSO, FMPI UK, Mlynska dolina, 842 48 Bratislava, samuelcik@fmph.uniba.sk