MODELING FROM PHOTOGRAPHS

M. R. TAN[1] AND T. S. TAN[2]

School of Computing, National University of Singapore
3 Science Drive 2, Singapore 117543

ABSTRACT

In many areas, we need a virtual representation of our real world surroundings. However, reconstructing a model of an actual scene is usually a tedious process. In this report, we explore a novel workflow for architectural model reconstruction from photographs. We keep the interaction with our system at a 2D level to reduce the complexity of our user interaction using a mesh-grid that the user interactively builds from a supplied photograph. At every step, the user can adjust the grid appropriately to ensure the quality of the resulting model. The interaction is kept in 2D by employing the constraint that the shifting of the gird lines is parallel to one of the 3 orthogonal axis of the structure. To facilitate the building of the mesh-grid, we make use of a tile finding algorithm to find the period of a section of the image. We demonstrate our proposed workflow with a prototype implementation.

1 INTRODUCTION

One of the many common tasks faced by 3D artists is the reconstruction of actual scenes. Scene reconstruction has a wide variety of applications such as virtual reality or in urban town planning. Nowadays, buildings form an integral part of our modern landscape. Unlike trees or roads which is usually be approximated by a general procedural model, buildings are readily identifiable landmarks and therefore needs to be faithfully reconstructed for the virtual scene to be convincing. However, the traditional method of building reconstruction is extremely labor intensive, usually taking more than a day of work by an experienced artist.

In this project, we explore a novel work flow for the reconstruction of architectural models from photographs in order to reduce the amount of effort required. Our system focuses on user interaction in 2D, making use of well studied image synthesis techniques and basic geometrical properties to determine the geometry of the scene and also texture the recovered geometry at the same time. The eventual goal of this project is to have a complete system that can quickly and easily create an accurate model in 3D from a sparse set of photographs.

2 SYSTEM OVERVIEW

As our system is a semi-automatic system, we need to balance the amount of user interaction and the quality of the result. We choose to keep the interaction in 2D to avoid the complexity introduced with working in another dimension, especially since our display interfaces are all 2D. Architectural scenes provides a great deal of constraints that we work with, such as most buildings are repeated layer by layer vertically upwards, and building surfaces are generally planar and lie orthogonal to each other. Despite these constraints, our system handles a good variety of buildings which forms a large part of the urban landscape.

---

[1] Student

[2] Supervisor

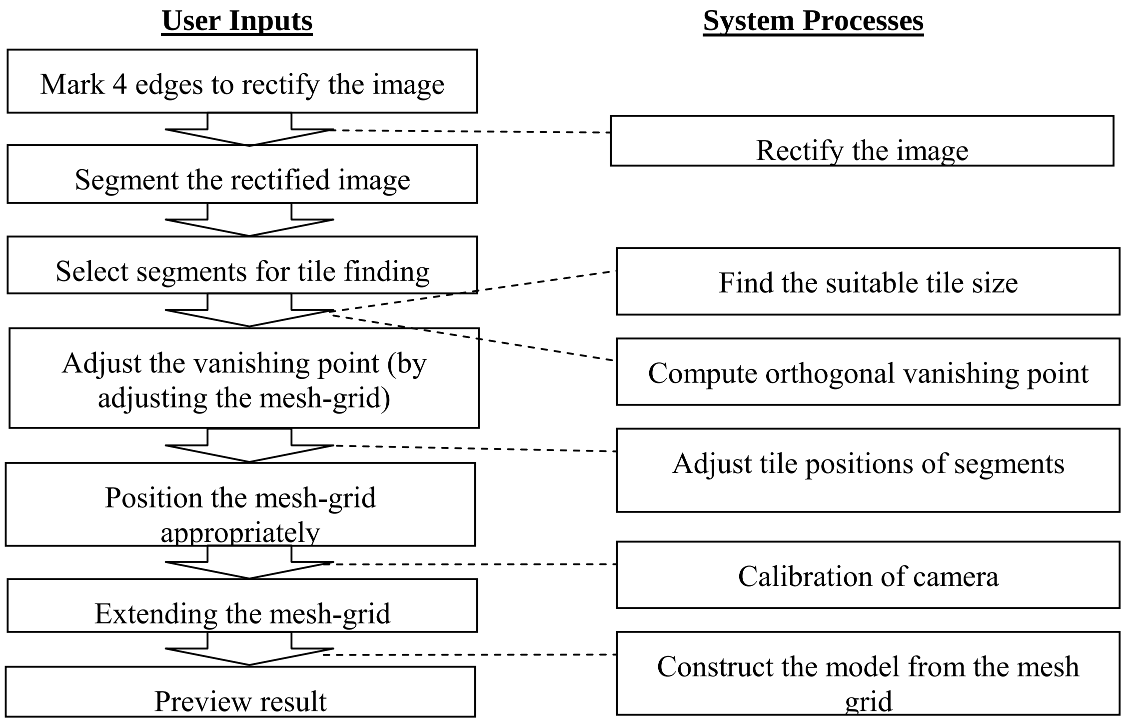

Figure 1. Workflow of our proposed system.

3 IMPLEMENTATION

We have implemented a prototype of our proposed system in C++, using the OpenCV and OpenGL libraries. All interactions and feedback from the system are near real time.

3.1 RECTIFICATION

The first step of the reconstruction process of our system is the rectification of the image. When a structure is projected onto an image plane, it exhibits perspective distortion, where angles between lines are no longer maintained and parallel lines begin to converge. The rectification process removes these perspective effects on the image. Rectification is a plane-to-plane transformation related by a planar homography[3]. The perspective image is related to its Euclidean equivalent by a homography matrix as follows:

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} \approx \begin{bmatrix} h_{00}\ h_{01}\ h_{02} \\ h_{10}\ h_{11}\ h_{12} \\ h_{20}\ h_{21}\ h_{22} \end{bmatrix} \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} \tag{1}$$

where h_{ij} represents the homography matrix, and x_i,y_i and x'_i,y'_i are homogenous coordinates of the perspective and rectified image respectively. The homography matrix has nine variables, and by ignoring the overall scale, it has an overall 8 degrees of freedom. A pair of point correspondence between the original and rectified image will provide two constraints on the equation, and thus, the homography matrix can be uniquely determined from four pairs of point correspondence.

In order to get the four pairs of point correspondences required to carry out rectification, we require the user to define four edges. These four edges should follow a rectangular surface parallel to the desired rectified plane. The four intersections of the edges give the four points on the original image. By ignoring the aspect ratio and scale of the photo, the corresponding points for these four points can simply be taken as the four corners of a square.

3.2 SEGMENTING THE IMAGE

In the next step, we mark out near uniform planes on the building using segment lines. Since the image is rectified, the segment lines correspond to vertical lines in the image. These segments form the basic unit of an image. The remaining parts of our system work on the relationship of these segments to recover the resulting textured partial model of the building. In our prototype system, the user defines segments on the rectified image by clicking a point which the vertical segment line will pass through.

3.3 TILE FINDING

We perform a tile finding algorithm on selected segments to the height of a tile for the segment. Our tile finding algorithm is based on the method proposed by Djado, Egli and Deschenes [2]. First, the method line up the image with another copy of the image. Then it performs a pixel-by-pixel displacement in the x direction for one of the image (Figure 2), computing the Sum of Squared Difference Normalized for the overlapped portion of the images at each step to form a curve. If the texture is a periodic texture, the curve will resemble a sinusoidal curve. The period of the curve, and thus the tile length, is given by the position of the last peak, divided by the number of peaks found.

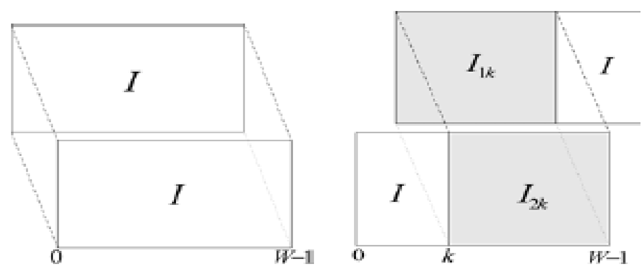

Figure 2. Pixel-by-pixel comparison of the image to determine the tile size

Since buildings are usually repeated vertically due to their layered structure, the segments marked out in the previous step correspond to a periodic texture in the y-direction. The texture height of a suitable tile is the period of the segment in the y-direction.

3.4 DEPTH FINDING

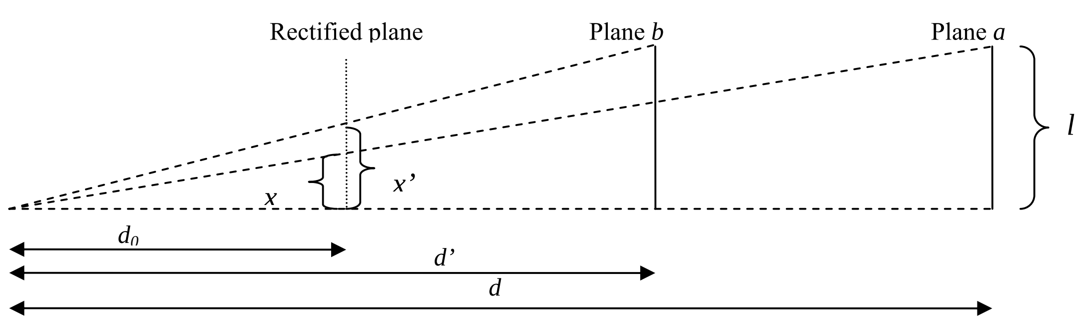

Figure 3. The relation between the depth and imaged length of a plane

Next, we make use of the recovered tile sizes to determine the depth of the tiled segment. The previous rectification process rectifies all planes parallel to the rectification plane. However, the planes that do not lie exactly on the rectified plane have a scale different from the other planes. The difference in scale is directly related to the distance of the plane from the camera center as shown in Figure 3. *a* and *b* are two planes with equal length *l*. *x'* and *x* are the lengths of the plane *a* and *b* on the rectified image respectively. *d'* and *d* are distance from the camera position of the original image, and d_0 is the distance of the rectified plane from the camera. By symmetry of triangles, we can equate the two triangles in the image and get the following relation on the relative orthogonal distance of the two planes from the camera center.

$$\frac{d}{d'} = \frac{x'}{x} \tag{2}$$

The tile heights of each segments provides the required equivalent parallel length property since they belong to a single layer of the building. Therefore, the relative depth of two segments is the ratio of the tile height of the segments.

3.5 ORTHOGONAL VANISHING POINT

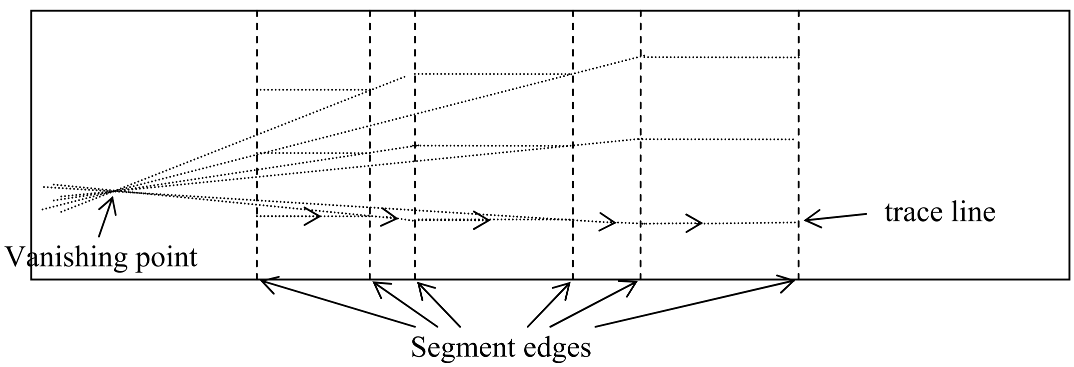

Figure 4. Location of the orthogonal vanishing point and the flow of the trace line

The segments of the rectified image are related by the orthogonal vanishing point. The orthogonal vanishing point is simply the point on the rectified image where lines orthogonal to the rectified plane converge. With the knowledge of the tile heights, the required orthogonal parallel lines are easily obtained as lines that join corresponding sections of two tiled segments. When multiple tiled segments are available, the orthogonal vanishing point for the image is estimated using the Gauss-Newton algorithm to minimize the back-projected error of the tile sizes.

Once the vanishing point has been located, it is possible to 'trace' every single layer. The trace line marks the points on the image which lie on the same horizontal plane parallel to the ground. The purpose of this trace line is to align the tile positions across the segments. The orthogonal vanishing point serves as the common point of reference for all the segments, in particular, the planes orthogonal to the rectified plane. Hence, to properly align the tiles belonging to the same layer, it is only necessary to adjust the vanishing point in the y-direction.

In our implementation, the user does not adjust the orthogonal vanishing point directly. Instead, the user shifts the mesh-grid, the combination of the traceline and segment edges.

The next parallel segment to the left is assumed to be the anchor, and the new trace line between the two parallel segments is used to determine the new y-position of the orthogonal vanishing point. The remaining segments can then be aligned to the new vanishing point accordingly.

3.6 CAMERA CALIBRATION

With the orthogonal vanishing point recovered, we have the necessary information to perform the calibration of the camera as described in [4]. The method uses three orthogonal vanishing points of the image to find the intrinsic parameters of the camera. From the initial rectification process, we have two vanishing points that are orthogonal to each other, and the third orthogonal vanishing point is the one orthogonal to the rectified plane.

After suitably adjusting the mesh-grid in the rectified image view, we project the mesh grid back onto the original image view (Fig. 5). The next step is to extend the existing mesh grid to include more of the scene. We allow either extending parallel or orthogonal to an existing plane. User can drag the edges of the mesh-grid to the new position, and with the computed camera matrix, we can find the ray from the camera and together with the knowledge of the plane from the mesh-grid, we can find the appropriate position for the newly created plane.

4 RESULTS

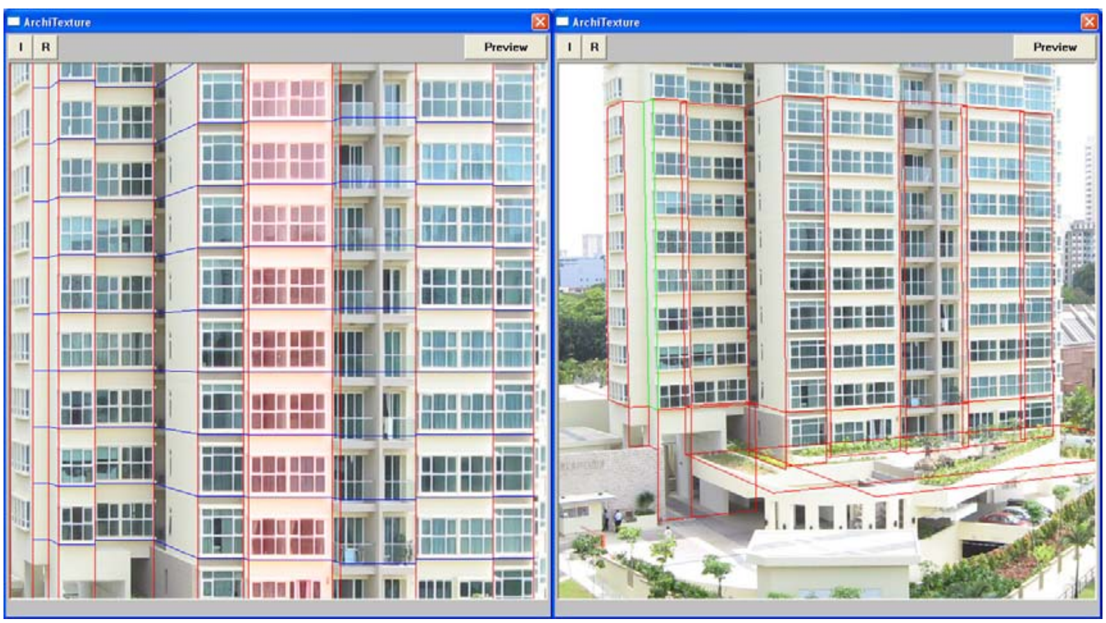

Figure 5.(left) View of Rectified Image with mesh grid adjusted into position. (right) View of original image with mesh grid projected on it. Additional planes has been defined

The interface of our system is extremely simple, with only three buttons and an image display canvas. The two top left buttons switches between original and rectified image view. The top right button displays the resulting recovered model in a new window. Most of the interactions on the drawing canvas are carried out with combination of mouse clicks and key presses.

Reconstructing the model from the mesh-grid starts from a base reference segment which the other segments will scale to the size of it. The depth of this base reference segment also needs to be provided, and the depth of the other segments can be computed from the known tile heights ratios of the segments. Each segment forms a plane which is either parallel or orthogonal to each other. The structure is built from the reference segment, where adjacent segment edges form the connecting points for the planes.

Figure 6. Two rendered views of the resulting model.

5 CONCLUSION AND FUTURE WORK

In this report, we have looked at a new workflow for reconstruction of an architectural scene from a given photograph. We simplify the user interface and reduce the amount of effort by keeping our user interactions solely at the 2D level. The system uses the natural constraints available in architectural scenes, and together with a combination of user inputs and automatic tile detection, recover an estimate of the geometry of the scene. At every step of our workflow, we provide immediate feedback to the user through our mesh-grid, allowing for adjustments to be made interactively to ensure the accuracy of the resulting model. Our system is by no means complete, but it demonstrated what may be an intuitive first step in building reconstruction. More work can be done to recover the complete model from multiple photographs, and removing the many visible artifacts on the resulting texture.

6 REFERENCE

[1] Debevec P., Taylor C., and Malik J. (1996) Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In Proceedings of ACM SIGGRAPH 1996, p11-20

[2] Djado K., Egli R. and Deschenes F. (2005). Extraction of a Representative Tile from a Near-Periodic Texture. In GRAPHITE 2005 p331-337

[3] Faugeras O. and Luong Q.T. (2001). The Geometry of Multiple Images. The MIT Press, Cambridge Massachusetts 2001

[4] Liebowitz D., Criminisi A. and Zisserman A.(1999). Creating Architectural Models from Images. In Eurographics 1999, 18-3, p39-50