Peek-in-the-Pic: Architectural Scene Navigation from a Single Picture Using Line Drawing Cues

Amit Shesh Baoquan Chen

University of Minnesota at Twin Cities
Email:{ashesh, baoquan}@cs.umn.edu

Abstract

The ability to navigate through the scene captured in a photograph generates much more excitement than merely looking at one. Through our system, SMARTPhoto, we accomplish navigation by utilizing work done in reconstruction of 3D geometry from 2D line drawings. We regard objects traced from a photograph as line drawings, formulate the reconstruction problem as an optimization problem and extend current work done in reconstruction of orthographic line drawings to perspective images to build navigable models. We also provide a basic sketching tool to progressively complete invisible parts of the reconstructed buildings. We use texture synthesis to interactively fill holes created in the original image when reconstructed buildings are removed from it. Minimal manual work, ease of use and interactivity are the salient features of our approach.

1. Introduction and Motivation

Photographs have traditionally been used as visual records of people met and places visited. Though photographs create a sense of “being there”, they do not provide as immersive an experience as say, navigation of the 3D scene that they represent.

Extracting 3D geometry from photographs is a well-known and studied problem both in computer graphics and vision. There exists rich literature on using photogrammetry and computational stereopsis [7, 11] to extract 3D geometry from multiple images. However these methods are not applicable in cases where only a single photograph is available. We propose a solution in this area. The traditional vision approach has been to determine camera parameters and then construct homographies. The camera parameters are determined by using line cues [1, 5], and then reconstruction is performed plane-wise by ortho-rectification. This process critically requires information about relationships between all planes and inherently requires a lot of human intervention. Methods suggested by Horry *et al.* [3] and Oh *et al.* [8] are examples of approaches proposed by the computer graphics community. The spidery mesh in [3] generates very crude approximation of the geometry depicted in the scene. The results by Oh *et al* [8] produced by *depth painting* are very compelling, but involve hours of tedious manual work.

A parallel and loosely related area of research is sketch reconstruction, which aims to reconstruct geometry from single freehand 2D line drawings. But as most of the work done in this area [6, 10, 12] applies only to orthographic drawings, they cannot be used to reconstruct perspective images.

We propose a simple “trace-to-reconstruct” method to construct 3D geometry from a single uncalibrated photograph. Our system, SMARTPhoto, applies the concepts of sketch reconstruction by generalizing them to reconstruct perspective images. Our main goal is to reconstruct geometry for navigational purposes to get a feel of 3D, instead of emphasizing on acquiring accurate geometry. Minimal manual work, ease of use and interactivity are the salient features of our approach.

2. From Image to Navigable 3D Geometry

The general theme of SMARTPhoto is “trace and reconstruct”. Our pipeline is summarized in Figure 1. The input is a single, casually taken photograph like Figure 1(a). To determine the camera parameters for this input image, the user specifies any three pairs of parallel lines that are mutually perpendicular. We use the method in [1] to infer the focal length and center of projection of the camera.

Now the user can begin reconstructing geometry from the image. The user places strokes along the visible edges of a building in the image. Figure 1(b) shows all the line drawings sketched by the user (in practice, the user traces out one building at a time). The line drawing of an object is converted into a 2D graph by treating each edge as a graph edge and clustering their end points to form vertices. Unlike SMARTPAPER [10], the graph does not represent a closed object as no object is fully visible in the image. Some natural constraints like attachment to the ground, walls being

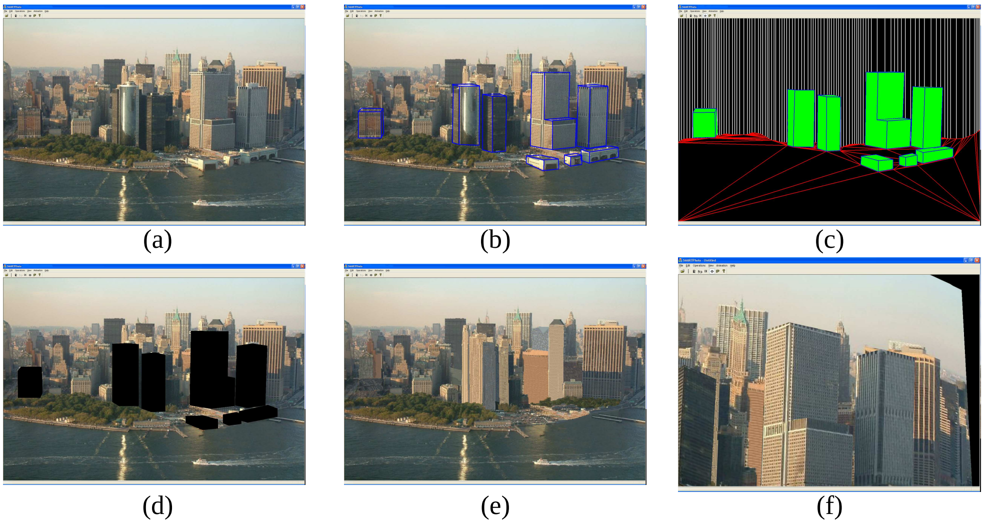


Figure 1. ***Partial geometry reconstruction of lower Manhattan, New York City from a single image. (a) the original image. (b) all line drawings made by the user (figure shows all traced lines; actually, one building is traced and reconstructed at a time). (c) eight reconstructed buildings, with the ground relief and the background. (d) the original image with holes to be synthesized. (e) the synthesized image for background and ground geometry. (f) an alternate view of the city.***

vertical, etc. are assumed, while other constraints like parallelism and perpendicularity of edges and planes can be explicitly specified by the user by simple gestures. The accuracy of the reconstructed geometry increases as more constraints are imposed.

The graph is now "inflated" by assigning a suitable Z-coordinate to each vertex through an optimization process, thereby resulting in the 3D geometry of the object it represents. Unlike inflation of orthographic sketches [6, 10], fewer constraints can be assumed about perspective line drawings. Also, in order to place constraints on the "correct" geometry, a candidate graph must be undistorted *before* evaluating them. The compliance function that the optimization attempts to minimize is of the form $f = w_i * t_i$ where $w = [w_i]$ is a weight vector. The terms t_i encapsulate all constraints determined earlier as penalties, resulting in an unconstrained, non-linear, multi-dimensional optimization problem. We use Brent's minimization [9] to solve this problem. We provide a basic sketching tool to sketch the missing parts of the object to complete it. Figure 1(c) shows all the buildings reconstructed from the line drawings in Figure 1(b) in green.

After all buildings are reconstructed, ground and background geometry must be constructed. The user sketches a horizon line which is used to reconstruct the ground and background geometry. Figure 1(c) shows the ground and background geometry in red and white respectively.

As a building is reconstructed, it leaves a hole in the image. Figure 1(d) shows all such holes created when all buildings traced in Figure 1(b) are reconstructed. We use interactive texture synthesis to fill these holes. The user specifies a source region and a target region using lassoing, and the system synthesizes the target region using the methods in [2, 4]. As this process is progressive, the user specifies small source and target regions, and hence this operation is interactive. Figure 1(e) shows the final synthesized image after all holes in Figure 1(d).

The geometry is then texture-mapped using the synthesized image to form a navigable 3D environment. A unique view of such an environment generated from the photograph in Figure 1(a) can be seen in Figure 1(f).

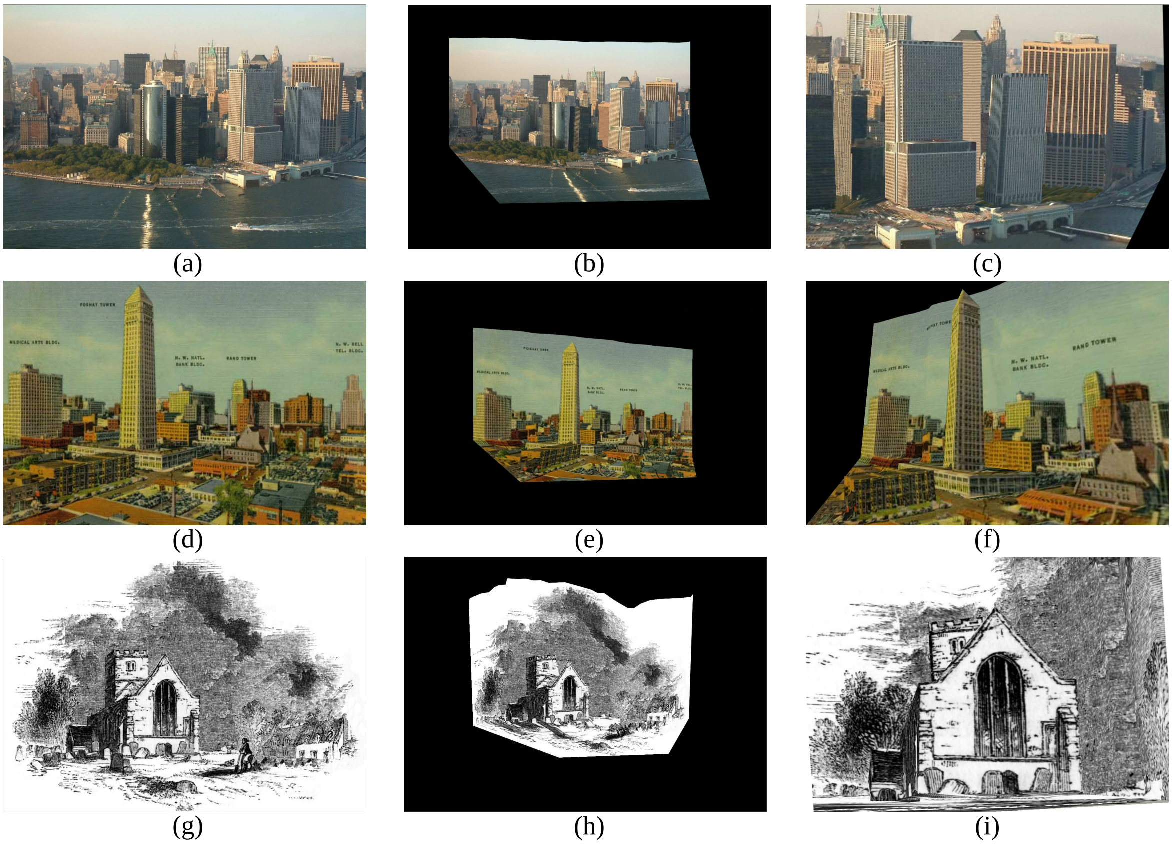
3. Implementation and Results

Figure 2(c) shows a part of Manhattan, NY[1] reconstructed using Figure 2(a) as input. Figure 2(b) shows how the constructed 3D environment looks from a distance. Eight buildings have been reconstructed in this example. On an average, it took 2-3 seconds to construct every building, and 5-7 seconds per texture synthesis operation.

Figure 2(d) shows a drawing of Foshay Tower in Minneapolis, MN[2] from the 1930s. This is a case where multiple images are not available as this is a drawing. Figure 2(f) shows an alternate view obtained by reconstruction (note the correctly reconstructed pyramidal top of the tower). Three buildings were reconstructed in this view.

[1] Source:http://www.pilotlist.org/balades/manhattan/manhattan.html

[2] Source:http://www.minneapolishistory.com/marriott3.htm


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 2. ***Image navigation. (a) input photograph of Manhattan, NY (b) view from a distance. (c) a unique view of the reconstructed scene. (d) input painting of 1930s downtown Minneapolis, MN. (e) view from a distance. (f) a unique view of the reconstructed scene. (g) input painting of 1305 church (h) view from a distance. (i) a unique view of the reconstructed scene.***

Figure 2(g) shows a drawing of the 1305 Church of Aston-Cantlow, Warwickshire, England[3]. Figure 2(i) shows a zoom and change of angle towards the church. Four buildings were reconstructed in this view. In each case, the optimization took about 5 seconds and the texture synthesis took about 4 seconds per source-target-pair.

References

[1] R. Cipolla, T. Drummond, and D. P. Robertson. Camera calibration from vanishing points in image of architectural scenes. In *Proc. British Machine Vision Conference*, 1999.

[2] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. *IEEE Intl. Conf. Computer Vision*, pages 1033–1038, September 1999.

[3] Y. Horry, K. ichi Anjyo, and K. Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. *Proc. SIGGRAPH*, pages 225–232, 1997.

[4] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum. Real-time texture synthesis by patch-based sampling. *ACM Trans. Graphics.*, 20(3):127–150, 2001.

[5] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. *Proc. Eurographics*, 18:39–50, 1999.

[6] H. Lipson and M. Shpitalni. Optimization-based reconstruction of a 3d object from a single freehand line drawing. *J. Computer Aided Design*, 28(8):651–663, 1996.

[7] L. McMillan and G. Bishop. Plenoptic modeling: an image-based rendering system. *Proc. SIGGRAPH*, pages 39–46, 1995.

[8] B. M. Oh, M. Chen, J. Dorsey, and F. Durand. Image-based modeling and photo editing. *Proc. SIGGRAPH*, pages 433–442, 2001.

[9] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, New York, NY, 2002.

[10] A. Shesh and B. Chen. Smartpaper–an interactive and user-friendly sketching system. *Proc. Eurographics*, 23:301–310, 2004.

[11] R. Szeliski. Image mosaicing for tele-reality applications. *WACV94*, pages 44–53, 1994.

[12] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. Sketch: an interface for sketching 3d scenes. *Proc. SIGGRAPH*, pages 163–170, 1996.

[3] Source:http://www.holoweb.net/ liam/pictures/oldbooks/OldEngland/ pages/1305-Church-of-Aston-Cantlow/