

# RECOVERING FACADE TEXTURE AND MICROSTRUCTURE FROM REAL-WORLD IMAGES

Xiaoguang Wang<sup>a</sup>, Stefano Totaro<sup>b</sup>, Franck Taillandier<sup>c</sup>, Allen R. Hanson<sup>d</sup>, and Seth Teller<sup>e</sup>

<sup>a</sup> Cognex Corporation, Natick, MA 01760, USA - xwang@cs.umass.edu

<sup>b</sup> Dipartimento di Elettronica ed Informatica, Univ. of Padua, Italy - tost@dei.unipd.it

<sup>c</sup> Institut Géographique National, Saint-Mandé Cédex, France - franck.taillandier@ign.fr

<sup>d</sup> Dept. of Computer Science, Univ. of Massachusetts, Amherst, MA 01003, USA - hanson@cs.umass.edu

<sup>e</sup> Laboratory for Computer Science, MIT, Cambridge, MA 02139, USA - seth@graphics.lcs.mit.edu

## Commisson III, WG 7

**KEY WORDS:** Texture, Surface Geometry, Texture Fusion, Microstructure, Urban Site Reconstruction

### ABSTRACT:

We present a set of algorithms that recovers detailed building surface structures from multiple images taken under normal urban conditions, where severe occlusions and lighting variations occur and are difficult to be modeled effectively. An iterative weighted-average algorithm is designed to recover high-quality consensus texture of the wall facades. A generic algorithm is developed to extract the 2D microstructures. Depth is estimated and combined with 2D structures to obtain 3D structures, facilitating urban site model refinement and visualization.

## 1 INTRODUCTION

Extracting and rendering detailed 3D urban environments is an important problem in computer vision and computer graphics because of its numerous applications. The main bottleneck of this problem lies in the need for human intervention in current systems, preventing them from being scalable to large datasets (Teller, 1998). A large body of research has been made for automating some of the processes, including reconstruction of coarse 3D geometric models, mainly at the level of buildings (Collins et al., 1998; Coorg and Teller, 1999; Firschein and Strat, 1996). Detailed analysis of facade texture and substructures has been very limited (Mayer, 1999).

Real-world texture and detailed structure are important because they provide high visual realism as well as cultural and functional information of the urban site. However, it becomes increasingly difficult when the information of concern is detailed to the degree of *microstructure* (surface structures such as windows that possess few supporting pixels due to insufficient image resolution). Figure 1(a2, b2, c2, d) shows some sample (rectified) images of a real-world building facade captured from different viewpoints. Large portions of useful texture are either occluded by other objects or degraded due to significant lighting variations. Some occlusions are caused by regular structures, such as other buildings, which could be modeled using existing techniques; some others are caused by irregular objects, such as trees, which are very difficult to be modeled effectively.

In summary, a detailed analysis of such images poses a difficult problem due to various factors that affect image quality, including (1) varying resolution due to perspective effects, (2) noise introduction during acquisition, (3) non-uniform illumination caused by lighting variations and complex environmental settings, (4) occlusions caused by *modeled* objects, such as other buildings, (5) occlusions caused by *unmodeled* objects, such as trees, utility poles, and cars. A system must be capable of dealing with all these coexisting factors in order to facilitate a detailed analysis. In addition, interactive methods are not preferable because of the large number of pixels and structures present in many situations (e.g. more than a thousand windows for four or five buildings).

We develop an automated method for effectively recovering high-quality facade texture and its microstructure pattern from multiple images. Input to our method is a set of images annotated with camera intrinsic parameters and reasonably accurate (but not exact) camera pose, as well as a coarse geometric model, mainly the facade planes of buildings in the urban site.

The information required as input to our method is available using existing techniques developed in the City Scanning Project (Teller, 1998). Image acquisition is performed by a semi-autonomous robot (Bosse et al, 2000), which is a movable platform mounted with a differential GPS sensor, a relative motion sensor, and a rotative CCD camera. The robot acquires thousands of images from different locations (called *nodes*) on the ground surface, annotating the images with metadata such as time and pose estimation. Spatial positions of the nodes are refined using feature correspondences across images (Antone and Teller, 2000). A set of *facades*, each corresponding to a wall surface of a building, is then extracted from the original images using a priori constraints (such as vertical surfaces and horizontal lines) to establish the geometric model of the urban site (Collins et al., 1998; Coorg and Teller 1999).

Section 2 describes an iterative, weighted-average approach to high-quality facade texture recovery. Sections 3 and 4 introduce 2-D and 3-D methods for microstructure extraction. Section 5 concludes the paper with discussions.

## 2 TEXTURE RECOVERY

Facade texture recovery is itself an important task for computer graphics applications; it is even more important when microstructure is of concern, because a high-fidelity texture representation is key to the success of detailed image analysis. Multi-view methods have been proposed for texture fusion/recovery, such as interpolation methods (Debevec et al., 1996), reflectance models (Sato et al., 1997), and inpainting techniques (Bertalmio et al., 2000). The main drawback of these methods is that they do not handle occlusions automatically. A system introduced by (Wang and Hanson, 2001) is capable of determining occlusions caused by regular, modeled struc-

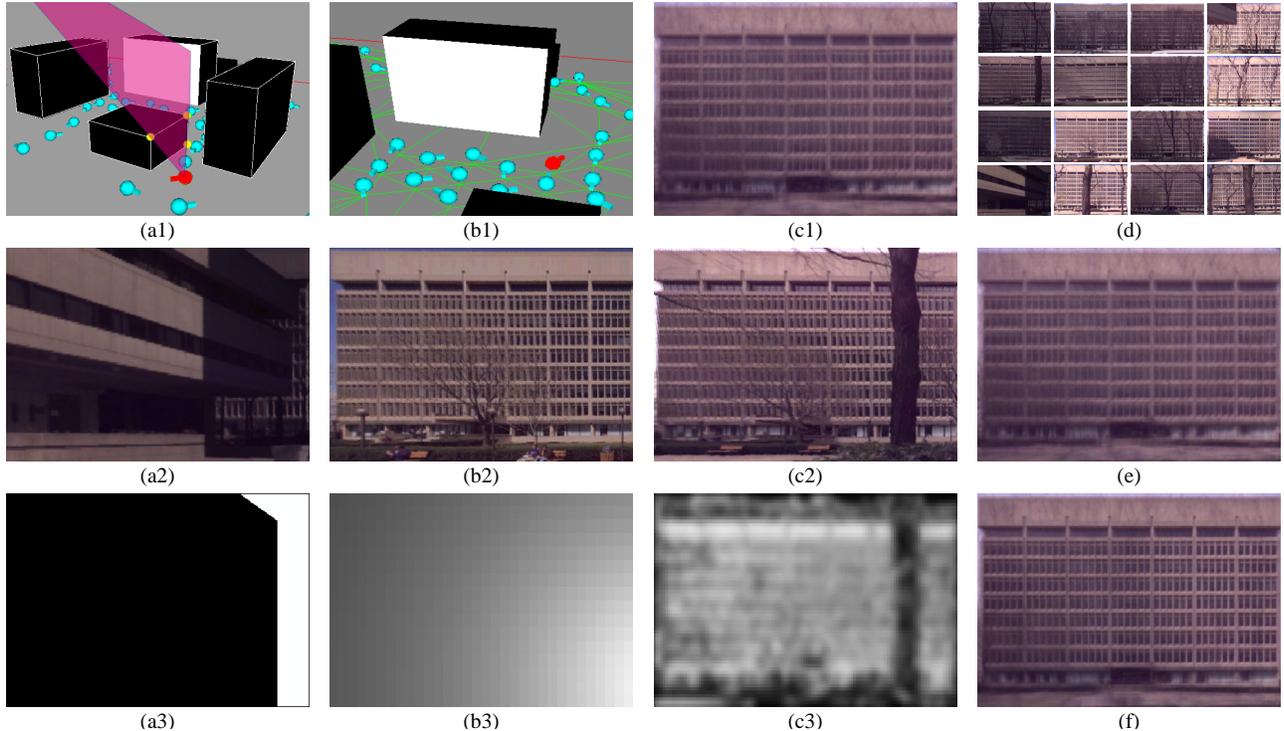


Figure 1: Texture recovery. (a) environment mask [a1: camera position, a2: LNF image, a3: mask]; (b) obliqueness mask [b1: camera position, b2: LNF image, b3: mask]; (c) correlation mask [c1: a version of CTF image, c2: LNF image, c3: mask]; (d) sample original facade images of this wall; (e) initial CTF image without deblurring; (f) CTF image after iterative deblurring.

tures, but not unmodeled ones. A median-based technique developed by (Coorg and Teller, 1999) repairs unmodeled occlusions; however, the method may cause blurred or disrupted boundaries of structures.

We describe a new method to obtain a realistic facade texture map, removing occlusions and effects of illumination variations. This method takes as input a coarse geometric model of buildings and a set of images taken from nodes at different locations and associated with reasonably accurate, but not exact, camera pose information. We assume that the light source for the urban images are normal sunlight (i.e. nearly white) and thus only the luminance in the color space is considered in this paper. We also assume that the building facades are close to the Lambertian surface model (Foley et al., 1990).

As a preprocessing step, the input images are rectified into *facade images*, i.e. images under orthographic projection of a facade. This happens only to a subset of the input images in which the facade is (at least partially) visible. For each image, the facade visibility and rectification is calculated based on the camera geometry at the node where the image is taken. Figure 1(d) shows some sample facade images in our experiments. To facilitate texture fusion for removing degradation effects (e.g. occlusions), the facade images are normalized by linear gray-level stretching; the resulting *luminance-normalized facade images* (or *LNF images*) have the same average luminance and thus are comparable to one another.

## 2.1 Weighted Averaging

The core of our method is a weighted-average algorithm that generates a *consensus texture facade image* (or *CTF image*) for each facade. The luminance value of pixel  $[i, j]$  in the CTF image of a facade is a weighted average of all LNF images of that facade:

$$Y_{\text{CTF}}[i, j] = \sum_{\tau} Y_{\text{LNF}}^{\tau}[i, j] * w^{\tau}[i, j], \quad (1)$$

$$\sum_{\tau} w^{\tau}[i, j] = 1, \quad (2)$$

in which  $Y_{\text{LNF}}^{\tau}$  is LNF image  $\tau$ ,  $Y_{\text{CTF}}$  is the fused CTF image, and  $w^{\tau}$  is the weight factor determined by three *masks* described below. A mask is an image whose pixel value indicates the relative importance of the corresponding pixel in the LNF image. The three masks measure three different physical attributes.

*Environment Mask* is a binary mask that specifies whether a pixel is occluded by a modeled object (Figure 1(a)). It is computed using the camera geometry and the 3D coarse model:  $M_{\text{E}}^{\tau}[i, j]$  is set to 0 if pixel  $[i, j]$  is occluded; otherwise, it is set to 1.

*Obliqueness Mask* is a grey-scale mask that represents the obliqueness of a facade as seen from the camera (Figure 1(b)), also computed from the geometry:

$$M_{\text{O}}^{\tau}[i, j] = \cos \theta^{\tau}(i, j), \quad (3)$$

in which  $\theta^{\tau}(i, j)$  is the camera viewing angle at  $[i, j]$  on the facade measured from the normal of the facade.

*Correlation Mask* is a grey-scale mask intended to deweight the effects of unmodeled occlusions and local illumination variations. To compute this mask, an initial CTF image is needed, and the mask is calculated using a standard linear correlation between the LNF image and the initial CTF image (Figure 1(c)):

$$M_{\text{C}}^{\tau}[i, j] = \frac{\text{Cov}_{i,j}[Y_{\text{LNF}}^{\tau}, Y_{\text{CTF}}]}{\text{Var}_{i,j}[Y_{\text{LNF}}^{\tau}] \text{Var}_{i,j}[Y_{\text{CTF}}]}, \quad (4)$$

in which  $\text{Cov}_{i,j}$  and  $\text{Var}_{i,j}$  are based in an image window, centered at  $[i, j]$ , of a predetermined size ( $8 \times 8$  in our experiments). In practice, the weighted-average algorithm is carried out iteratively (Section 2.2), and in each iteration a new CTF image is used to calculate  $M_{\text{C}}^{\tau}$ . The initial CTF image is obtained by the first iteration, in which only  $M_{\text{E}}^{\tau}$  and  $M_{\text{O}}^{\tau}$  are used.

Once the three masks are determined for each LNF image of a facade, the weight  $w^\tau$  at pixel  $[i, j]$  of LNF image  $\tau$  is computed by:

$$W^\tau[i, j] = M_E^\tau[i, j]M_O^\tau[i, j]M_C^\tau[i, j], \quad (5)$$

$$w^\tau[i, j] = \frac{W^\tau[i, j]}{\sum_\tau W^\tau[i, j]}. \quad (6)$$

## 2.2 Iterative Deblurring

The CTF image thus obtained may look blurred (Figure 1(e)) because the LNF images may not be perfectly registered due to errors in camera parameters. (Note that our algorithm do not require precise input camera parameters; that is, any two versions of LNF images may not align accurately to each other.) A deblurring process is used that warps the source LNF images to align with the CTF image, similar to that of (Szeliski, 1996):

$$[u, v, 1]^T \cong P[u', v', 1]^T, \quad (7)$$

which warps pixel  $[u', v']$  to  $[u, v]$  using  $P$ . Our goal is to find a warp  $P$  that best registers the two images. We use the following constraint functions in our method:

$$E_{CTF} = \sum_{u,v} [e(u, v)]^2, \quad (8)$$

$$[e(u, v)]^2 = W^\tau[u', v'](Y_{CTF}[u, v] - Y_{LNF}^\tau[u', v'])^2. \quad (9)$$

Note that the overall weight mask  $W^\tau$  is used, reflecting the degree of confidence we have for each pixel of  $Y_{LNF}^\tau$ . The Levenberg-Marquardt algorithm (Press et al., 1992) is employed to solve the constrained minimization problem. It is an iterative process (starting from the identity matrix); in each iteration,  $P$  is incremented by

$$\Delta p = -(H + \lambda I)^{-1}g, \quad (10)$$

where

$$g = \sum_{u,v} e(u, v) [\partial e(u, v) / \partial p], \quad (11)$$

$$H = \sum_{u,v} [\partial e(u, v) / \partial p][\partial e(u, v) / \partial p]^T, \quad (12)$$

in which  $p$  is a  $8 \times 1$  vector representation of  $P$  (note only 8 parameters are needed to describe  $P$ ), and  $\lambda$  is a parameter reduced to 0 as the procedure converges. After a new  $P$  is calculated, the LNF images are warped and the weighted-average algorithm (Section 2.1) is rerun using the warped LNF images to compute for a new CTF image. Figure 1(f) shows such a CTF image with deblurring.

The deblurring process is also executed in an recursive manner. Recall that the correlation mask  $M_C$  is dependent on an initial CTF image. After deblurring, the new CTF image is used to compute a more accurate  $M_C$ , which then again updates the CTF image and triggers another round of deblurring. The convergence of the recursion is ensured by stopping when the difference between two successive CTF images is sufficiently small.

## 2.3 Experiments

Experiments were carried out to test the consensus texture generation algorithm against an image dataset acquired at Technology Square, an office park of four buildings located on the MIT campus. About 4,000 images were captured using the movable platform (Section 1)

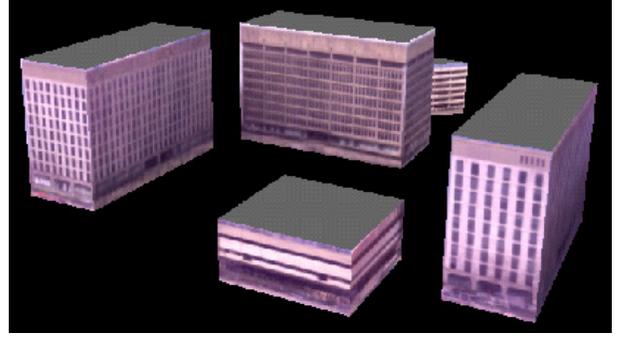


Figure 2: CTF textured model.

at 81 nodes in this site. At each node, 47 images were acquired with distinct rotations. LNF images were extracted for each facade.

Figure 1(f) shows the CTF result of the iterative weighted-average algorithm on a facade, for which 28 LNF images were extracted from the database and used to generate the CTF image. Most occlusions caused by modeled/unmodeled objects were satisfactorily removed; the luminance is also reasonably consistent across the entire CTF image. Figure 2 shows a perspective view of the resulting textured model of this site.

Our experiments also show that only about a dozen original facade images, with quality shown in Figure 1(d), are needed for texture recovery with a satisfactory result. In addition, the iterative deblurring is a very stable process; only a couple of iterations are necessary to reach the image quality as shown in Figure 1(f), under the condition of up to 5-pixel mis-alignment of LNF images due to input camera pose error. Therefore, the halting of the iterations can be simplified as to a certain number of iterations, instead of complex criteria.

## 3 MICROSTRUCTURE DETECTION

In the area of urban site reconstruction, a large body of research has been focusing on methods of establishing geometric models for large-scale structures, especially buildings, whose structural features (corners, edges, etc.) typically possess sufficient image cues to support direct and reliable 3D reconstruction from the images (Firschein and Strat, 1996; Mayer, 1999). In this paper, we emphasize the importance of microstructures because they provide rich information of the buildings and result in added realism for visualization.

Two pieces of evidence are used for microstructure extraction: the relative 3D depth of the structures and their 2D appearances. The relative depth of a surface microstructure is typically very small (see Section 4). Thus directly extracting these structures from noisy 3D depth data may be beyond the state-of-the-art of current computer vision algorithms without a priori knowledge. In this section, we use a 2D-based strategy to detect the locations of microstructures in the CTF images generated in Section 2.

The CTF images provide a good texture representation of the facade, free from effects of occlusions and local illumination variations if enough views are provided. However, symbolic extraction of windows is still difficult due to the existence of noise. One type of noise is the global illumination variation on the facade. It happens in Figure 2 that the lower part of the walls is universally darker than their upper part (sometimes even darker than upper windows). This is because lower parts of buildings typically receive less sunlight than upper parts in a densely urbanized area. The pixel-based weighted-average algorithm is unable to remove global illumination variations, because the lower part is darker on the majority of LNF images. A

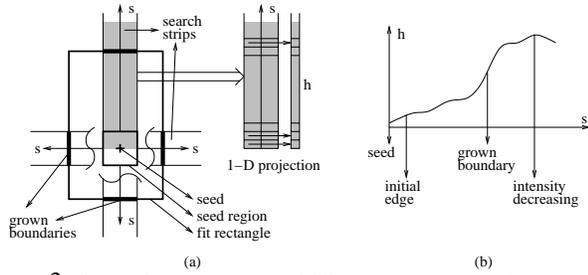


Figure 3: Oriented region growing (ORG). (a) one iteration of region growing; (b) determining the location of the grown boundary.

global thresholding algorithm is not useful for detecting microstructures in such images.

### 3.1 Oriented Region Growing (ORG)

We apply an *oriented region growing (ORG)* algorithm to the CTF images for window structure extraction (Wang and Hanson, 2001). This algorithm detects a generic class of objects that exhibit a regular size, pattern, and orientation. One of its major advantages is that it deals with global illumination variations and other types of noise. It requires that a window be on average darker than the wall locally but not necessarily globally.

In the current system, the symbolic microstructures are represented as a set of disjoint 2-D rectangles, each having two vertical and two horizontal edges. A large number of windows in urban areas fit well into this representation. Window extraction is performed on the CTF images, on which windows appear as dark, rectangular blobs on the brighter wall surface.

Details of the ORG algorithm are shown in Figure 3(a) in the facade image space. It runs iteratively, starting from a smaller rectangle (called a *seed*) and growing outward into a larger one that best fits the window blob. The growing processes are performed only in the two vertical and horizontal directions. In each direction, a search strip, e.g.  $s$  in 3(a), is established based on the seed. A zero second-order derivative criterion, shown in 3(b), is applied to the intensity profile,  $h(s)$ , of the strip for determining the grown boundary. A new rectangle is fit to the four boundaries, found in the four strips, to form a larger rectangle, which is then treated as a new seed to initiate another iteration of oriented growing. The iteration halts when the region ceases to grow; the resulting rectangular region is taken as a window candidate.

The ORG module requires only two user-provided parameters, the lower and upper bounds of window size. It attempts to find all windows of any size between the two bounds.

### 3.2 Periodic Pattern Fixing (PPF)

The ORG algorithm is a purely bottom-up process and may result in missing candidates due to image noise. A top-down module (Figure 4) is designed to fix the missing candidates by applying a high-level constraint about the microstructure pattern. This constraint states that microstructures of similar size have a periodic pattern in horizontal and vertical directions on the facade. Based on this constraint, a *periodic pattern fixing (PPF)* module is designed for repairing periodic microstructure patterns. Structures of similar size are grouped together; the horizontal and vertical periods of a microstructure group are then found using clustering algorithms based on their neighboring distances. Missing candidates are then hypothesized using interpolation or extrapolation.

In reality, the periodic pattern constraint may not always be strictly

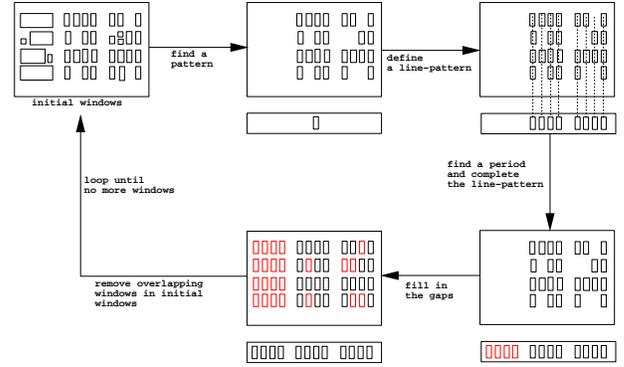


Figure 4: Periodic pattern fixing (PPF).

satisfied on all buildings. To ensure that missing candidate hypotheses are only filled in for windows that exist, a “bottom-up verification” test is used to verify their existence in the LNF images before interpolation/extrapolation. On each LNF image, a vertical and horizontal edge detection algorithm is performed at locations of missing candidates (if they are visible). A missing candidate is accepted for filling in only if there are sufficiently many LNF images that support the hypothesis.

### 3.3 Experiments

Ten facade images, representing the major buildings in Technology Square, were used to test the symbolic window extraction algorithm. Figure 5(a) shows a CTF image ( $512 \times 256$ ), where there are two types of windows on the facade: twenty small ones aligned on the top floor and 192 windows in a matrix pattern. The image is noisy: the lower part of the facade has significantly less luminance and contrast than the upper part; windows have different local background conditions and different degrees of blurriness on their edges. In addition, there are structures on the bottom floor with irregular reflectance.

In the experiments, the lower/upper bounds of window size are set to 3 and 100 pixels, respectively, for both height and width. Figure 5(b) shows the results of the ORG algorithm. The majority of the windows are extracted correctly; some windows are missing due mainly to extremely low contrast in the CTF image; there are some false extractions caused by the irregular reflectance on the bottom floor. Figure 5(c) shows the PPF results, in which missing windows are correctly filled in and irregular structures are properly removed.

Table 1 lists the extraction results in the experiments. Among the 1146 windows on the ten facades, 1119 of them have been extracted correctly. Only 27 are missing, accounting for 2.4% of the actual windows. Among the 1133 extracted structures, about 98.8% are correct windows and only 1.2% (14 extracted structures) are false positives. An examination of the images shows that the missing windows are mainly caused by low contrast of the windows and their blurred edges. The false positives are mostly due to the complex intensity patterns caused by rectangular structures on the wall surfaces that look like windows but are not.

The PPF module is based on the assumption that the windows possess a periodic pattern. This strong assumption did not cause significant false positives because of the bottom-up verification process. However, it is worth noting that PPF is an optional module highly dependent on a priori knowledge of the domain, scene, and/or specific object. For urban sites where windows do not show periodic patterns on buildings, this option may not be used.

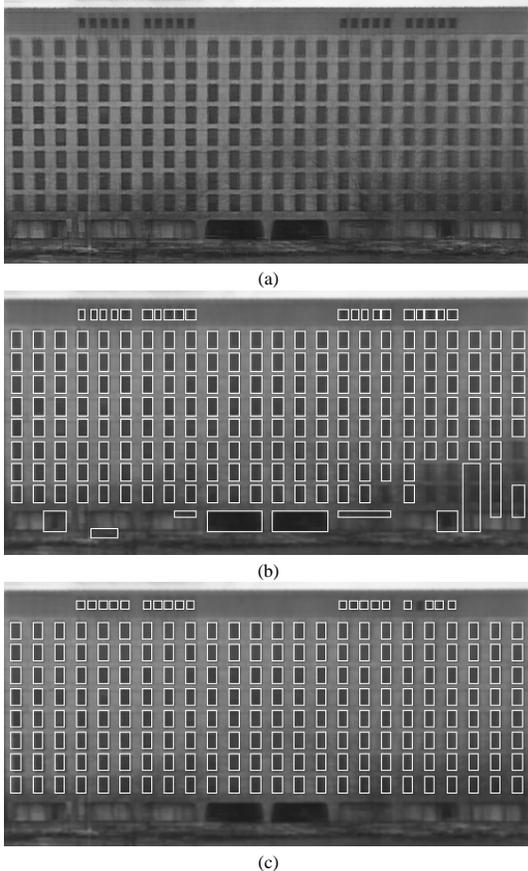


Figure 5: Symbolic window extraction. (a) CTF image; (b) results of ORG; (c) results of PPF.

#### 4 DEPTH ESTIMATION

The goal of depth estimation is to recover the relative depth of microstructures on the facade surfaces. Accurate depth recovery from images is very difficult in the context of the City Scanning Project, mainly due to the small ratio of the depth of microstructures (on the order of centimeters) to the camera-to-wall distance (on the order of tens of meters).

Stereo analysis (Faugeras, 1993) is widely used to recover depth from multiple images. There are some shortcomings in the pure form of stereo analysis when applied to the problem of urban site microstructure extraction. First, the analysis (e.g. epipolar matching) typically takes place on two images whereas many more facade images are available in our application. Second, geometric constraints (such as that microstructures are shallow structures on a largely planar surface) is not easily incorporated into stereo processes. Third, other knowledge, such as that of occlusions (both modeled and unmodeled) is not readily applicable in stereo analysis.

Fua and Leclerc developed a method that generates a 3D mesh to represent the 3D structures on a surface (Fua and Leclerc, 1994). This approach has a number of advantages over the pure form of stereo analysis. It is correlation-based and makes use of any number of images. It uses a minimization function in which geometric constraints can be added. In particular, the minimization process starts from a plane, addressing the planar constraint of the facades.

In this section, surface microstructures are recovered using a hybrid method that combines the 2D information obtained in Section 3 and the 3D information obtained using a revised version of Fua and Leclerc’s method.

| Facade | Actual | Extracted | Correct | False neg. | False pos. |
|--------|--------|-----------|---------|------------|------------|
| 7      | 301    | 303       | 300     | 1          | 3          |
| 17     | 18     | 17        | 17      | 1          |            |
| 18     | 54     | 54        | 54      |            |            |
| 19     | 18     | 11        | 6       | 12         | 5          |
| 24     | 192    | 192       | 192     |            |            |
| 25     | 82     | 72        | 72      | 10         |            |
| 26     | 212    | 211       | 211     | 1          |            |
| 27     | 72     | 72        | 72      |            |            |
| 44     | 144    | 150       | 144     |            | 6          |
| 45     | 53     | 51        | 51      | 2          |            |
| Total  | 1146   | 1133      | 1119    | 27         | 14         |

Table 1: Window extraction results

#### 4.1 3D Mesh Generation

The facade surface  $S$  is represented by a mesh, which is a hexagonally connected set of vertices organized into triangular elements called *facets*. The algorithm starts with a planar surface and deforms it by iteratively minimizing objective function  $E(S)$ :

$$E(S) = \lambda_D E_D(S) + \lambda_S E_S(S) + \lambda_G E_G(S), \quad (13)$$

$$\lambda_D + \lambda_S + \lambda_G = 1, \quad (14)$$

in which  $E_D$  is a planar surface constraint that controls the amount the surface is allowed to deviate from a plane,  $E_S(S)$  is a correlation-based stereo constraint attempting to minimize the appearance differences of each facet of the mesh across all the images, and  $E_G(S)$  is a geometric constraint. Details for these components and the minimization scheme can be found elsewhere (Fua and Leclerc, 1994).

In order to take advantage of knowledge obtained in Section 2, we modified Fua and Leclerc’s algorithm by excluding occlusions from stereo computation in  $E_S$ . We define an *occlusion-removed facade image* (or *ORF image*) by

$$Y_{\text{ORF}}^T[i, j] = Y_{\text{LNF}}^T[i, j] M_E^T M_C^T,$$

where  $M_E^T$  is the environment mask that represents the modeled occlusion, and  $M_C^T$  is a binary version of the correlation mask  $M_C^T$  that represents the unmodeled occlusion. Thus we use  $Y_{\text{ORF}}^T$  rather than  $Y_{\text{LNF}}^T$  to calculate  $E_S$ , using only unoccluded pixels and focusing on the visible parts of each facade.

We applied the 3D mesh generation algorithm to all the major facades in our dataset. In the experiments, we set  $\lambda_D = 0.1$ ,  $\lambda_S = 0.9$ , and  $\lambda_G = 0$ ; that is, the geometric constraint  $E_G(S)$  was not implemented. Figure 6 shows the depth estimation on one of the facades. These results are noisy, but the general pattern of windows is evident.

#### 4.2 2D/3D Combination

Figure 6 shows that an accurate shape recovery of 3D microstructures is nearly impossible from depth estimates alone, because the depth results are very noisy. For a higher-quality reconstruction, we assume that the facade surface structures can be approximated by two depth layers only: the wall layer and the window layer. This assumption is reasonable for normal walls. Non-flat portions on a wall are normally connections between windows and the wall; these detailed structures are beyond the scope of our current discussion.

With this assumption, we use the 2D rectangles extracted in Section 3 to represent the shape of the 3D microstructures, and use the average depth inside the rectangles to represent the depth of the structures. Figure 7 demonstrates an example of 2D/3D combination results. It shows that the 3D structures are well represented by combining the 2D symbolic representation and the 3D depth information.

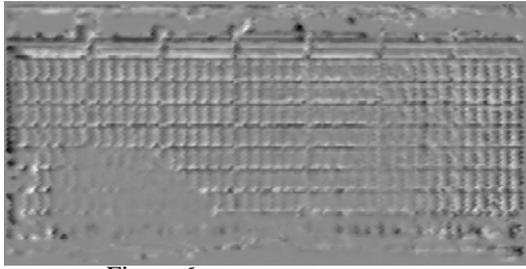


Figure 6: Facade depth estimation.

## 5 DISCUSSION

We described a suite of algorithms for detailed urban environment extraction, including texture recovery, microstructure detection, and depth estimation. An iterative, weighted-average algorithm is capable of recovering a consensus texture map, nearly free from occlusions (modeled and unmodeled) and local illumination variations. 2D microstructures are extracted using an algorithm that combines bottom-up and top-down strategies. Finally, the relative depth of 3D microstructures are estimated, facilitating a combination of 2D and 3D information for a complete representation of surface microstructures. Combining with previous techniques, these algorithms are capable of producing realistic, detailed, and texture-mapped 3D models of urban environments from large sets of real-world images.

The proposed algorithms are effective for solving a generic set of urban environment extraction and refinement problems, in which the wall surfaces are largely planar and the microstructures are mainly rectangular. Many buildings in urban environments satisfy these constraints. In addition, practicality is one of the design emphases of the algorithms. For example, significant efforts have been invested in the algorithms to deal with inaccuracy and uncertainty of the input data. The texture deblurring process allows the algorithm to tolerate camera pose error that often arises in real applications. The 2D microstructure module is adapted to extract structures of any size, requiring from the user only the upper/lower bounds of the structure size and needing no interactive parameter adjustment.

There are several directions in which the algorithms can be extended to solve more general problems. First, the extracted 2D microstructures can provide partial geometric constraints in  $E_G(S)$  for depth estimation. How to improve the depth estimation by incorporating the partial constraints is a topic for future study.

Second, the architecture of iterative texture recovery invites more information to be utilized for better results. For example, once the depth of the 3D microstructures are determined, occlusions caused by these structures on the facade can be computed for each LNF image. Therefore, the texture recovery algorithm can be rerun to take into account this additional information (excluding these occlusions from the CTF computation).

Third, the ORG algorithm is designed to extract a generic class of objects. Although a large variety of surface microstructures fit into this class, it has two major limitations: the shape of each microstructure is approximated by a rectangle, and the luminance of the microstructure must be relatively uniform. For more special problems, special object detection modules should be used as a successor of ORG/PPF.

Fourth, the global illumination variation problem has not been solved in the CTF algorithm. For rendering purposes, a better texture representation may be demanded. This problem could be solved using the heuristics given by the periodic pattern of microstructures. As the microstructures share the common shape and common period, they should also share the illumination in normal cases. An illumination adjustment algorithm could thus be designed for this end.

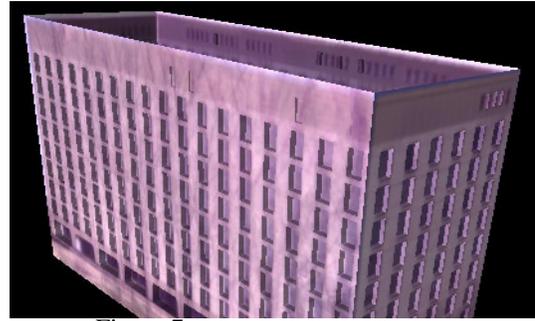


Figure 7: Microstructure visualization.

## ACKNOWLEDGMENTS

This work was funded in part by DARPA DACA76-97-K-0005, ARPA/ARL DAAL02-91-K-0047, ARPA/ATEC DACA76-92-C-0041, and ARO/ARL DAAG55-97-1-0026. The authors would like to thank Eric Amram and Neel Master for their technical support. Xi-aoguang Wang's contributions to the paper reflect his work at UMass Amherst and MIT.

## References

- Antone, M., and S. Teller, 2000. "Automatic Recovery of Relative Camera Rotations for Urban Scenes," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 282-289.
- Bertalmio, M., G. Sapiro, V. Caselles, and C. Ballester, 2000. "Image Inpainting," *SIGGRAPH'00*.
- Bosse, M., D. de Couto, and S. Teller, 2000. "Eyes of Argus: Georeferenced Imagery in Urban Environments," *GPS World*, pp. 20-30.
- Collins, R., C. Jaynes, Y. Cheng, X. Wang, F. Stolle, A. Hanson, and E. Riseman, 1998. "The Ascender System for Automated Site Modeling from Multiple Aerial Images," *Computer Vision and Image Understanding*, vol. 72, no. 2, pp. 143-162.
- Coorg, S., and S. Teller, 1999. "Extracting Textured Vertical Facades from Controlled Close-Range Imagery," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 625-632.
- Debevec, P., C. Taylor, and J. Malik, 1996. "Modeling and Rendering Architecture from Photographs: a Hybrid Geometry and Image-based Approach," *SIGGRAPH'96*, pp. 11-20.
- Faugeras, O., 1993. *Three-Dimensional Computer Vision, a Geometric Viewpoint*, The MIT Press.
- Firschein, O., and T. Strat (Ed.), 1996. *RADIUS: Image Understanding for Imagery Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA.
- Foley, J., A. van Dam, S. Feiner, and J. Hughes, 1990. *Computer Graphics, Principles and Practice*, Second Edition, Addison Wesley, Reading, MA.
- Fua, P., and Y. Leclerc, 1994. "Using 3-Dimensional Meshes to Combine Image-Based and Geometry-Based Constraints," *European Conference on Computer Vision*, pp. B:281-291.
- Mayer, H., 1999. "Automatic Object Extraction from Aerial Imagery - A Survey Focusing on Buildings," *Computer Vision and Image Understanding*, vol. 74, no. 2, pp. 138-149.
- Press, W., S. Teukolsky, W. Vetterling, and B. Flannery, 1992. *Numerical Recipes in C, The Art of Scientific Computing*, Cambridge University Press.
- Sato, Y., M. Wheeler, and K. Ikeuchi, 1997. "Object shape and Reflectance Modeling from Observation," *SIGGRAPH'97*, pp. 379-387.
- Szeliski, R., 1996. "Video Mosaics for Virtual Environments," *IEEE Computer Graphics and Applications*, vol. 16, no. 2, pp. 22-30.
- Teller, S., 1998. "Automated Urban Model Acquisition: Project Rationale and Status," *Image Understanding Workshop*, pp. 455-462, Monterey, CA.
- Wang, X., and A. Hanson, 2001. "Surface Texture and Microstructure Extraction from Multiple Aerial Images," *Computer Vision and Image Understanding*, vol. 83, no. 1, pp. 1-37.