A System for Surface Texture and Microstructure Extraction from Multiple Aerial Images *

Xiaoguang Wang and Allen R. Hanson

Department of Computer Science
University of Massachusetts
Amherst, MA 01003
Email: {xwang, hanson}@cs.umass.edu

Abstract

Building surfaces with textures and microstructures provide important information for many military and civilian applications. The extraction of the information from aerial imagery is difficult due to problems such as perspective distortion, data deficiency, and corruption caused by shadows and occlusions. In this paper we present a Surface Texture and Microstructure Extraction (STME) system to deal with these problems, under the assumption that an initial site model is given and sufficient camera and light source information is known. The infrastructure of the system is an Orthographic Facet Image Library (OFIL) that systematically collects building facet intensities from multiple aerial images into a datebase, eliminates the effects of shadows and occlusions, and uses a Best Piece Representation (BPR) method to form a complete and consistent intensity represen-

*This work was funded by the RADIUS project under ARPA/Army TEC contract number DACA76-92-C-0041 and NSF grant CDA-8922572.

tation for each facet by combining intensities from different sources. A window extraction module focuses attention on wall facets, attempting to extract the 2-D window patterns attached to the walls using an Oriented Region Growing (ORG) technique. Combined with the UMass Ascender site modeling system and scene rendering algorithms, the system is typically useful for urban site model refinement and visualization.

Keywords: texture extraction, microstructure extraction, model refinement, texture mapping, scene rendering, aerial imagery, image sequence

1 Introduction

Recovery of 3-D geometric structures of ground objects from aerial imagery is an important issue in many military and civilian applications. Traditional machine vision systems (e.g. [2]) usually focus on *large-scale structures* such as buildings. *Microstructures* [9] such as windows, doors and roof vents on buildings are left untouched in these systems. As a matter of fact, microstructures have many potential uses in military and civilian applications, because they often provide functional and cultural information, which is critical in landmark recognition, mission planning and assessment, and other applications requiring detailed models. Traditional building detection techniques usually treat a building as a simple 3-D polygonal structure; the functionality of the building is not easily determined from such a coarse model. An analysis of the window lattices on the walls may provide much richer information: the number of windows in the vertical direction shows the number of levels (floors) inside the building; the distances between the windows may indicate the size of the rooms and the height of each floor; the sizes and alignment patterns of the windows may provide useful cues concerning the functionality of the building (a factory, a school, or a military base?). Generally speaking, microstructure often acts as a functional

and cultural signature of the large-scale structure it is attached to.

Advanced optical and digital technologies are now capable of revealing not only large-scale structures but also microstructures in aerial images. A large number of microstrucutres are associated with surfaces of large-scale structures and are contained in the textures of these surfaces. Extraction of surface textures of 3-D objects is itself an increasingly important aspect of both computer graphics and computer vision, because high-quality renderings and animations of textured geometric objects are widely demanded in visualization and virtual reality tasks in military and civilian applications. However, automatic surface texture extraction from aerial images of a scene is by no means an easy task, and has not been widely addressed.

Consider the problems involved in extracting the windows on the buildings in the site shown in Figure 1, which is a subimage from the RADIUS Model Board 1 image set. Some walls are occluded by other buildings, some are corrupted by shadows, and some are completely invisible due to the viewpoint. Even if some of the invisible or corrupted parts appear in other images, the resolution and brightness of the surfaces may vary considerably from image to image, because the images may be taken at various times of day, from different positions, and under varied weather conditions. Furthermore, most visible windows occupy no more than a few pixels.

This discussion suggests that a *surface texture and microstructure extraction* (STME) system must have the following properties to cope with all the difficulties. First, it must support multiple image input in order to recover data missing from individual images due to occlusions and shadows. Second, data fusion techniques must be incorporated to deal with data from multiple source images taken under very different situations. Third, model-driven processing using large-scale structures can focus attention on relevant areas of the image and provide the infrastructure for context sensitive microstructure extraction.

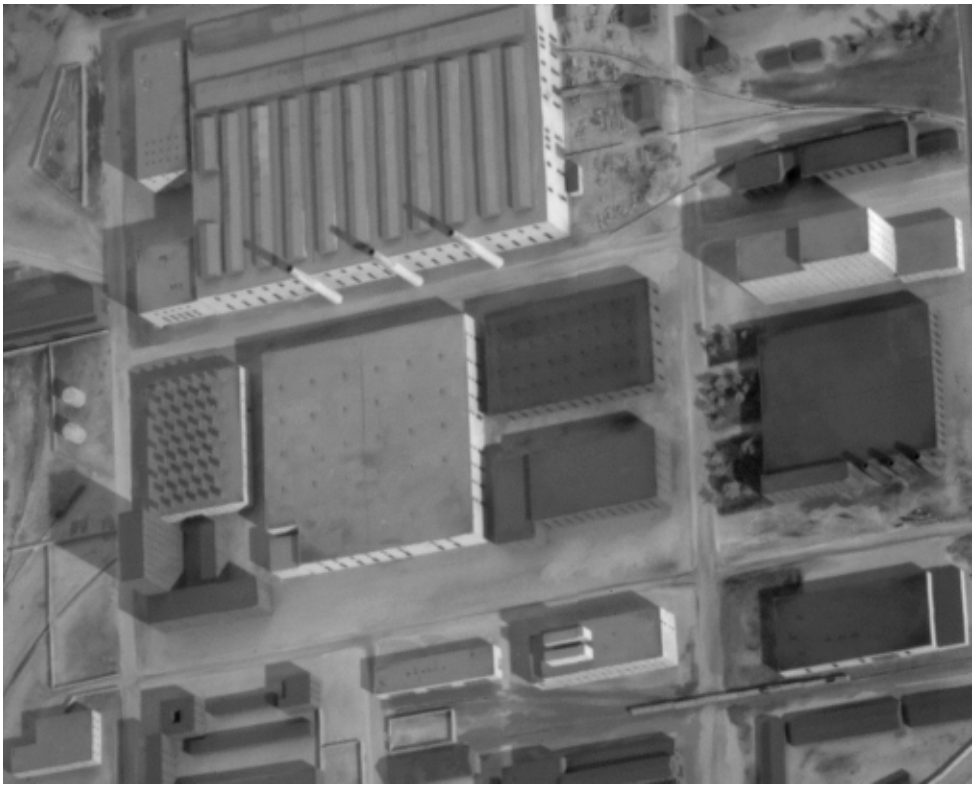


Figure 1: Part of site image J1 from Model Board 1

Finally, microstructure analysis is a typical domain where low-level data are insufficient and high-level knowledge must be used to guide the analysis. Some constraints in man-made structures may be made use of, such as that windows on a wall are usually aligned rectilinearly.

Recovery of a high quality image representation of building walls from multiple images is an essential issue in an STME system. Texture mapping [6, 5] is believed by many researchers to be a key technique for this purpose. Cohen-Or et al. [1] used a voxel-based model to generate visual fly-throughs over terrain. The purpose of their system, however, is to render distant views rather than reveal details. Their method for obtaining the voxels of an object from multiple images is not automatic, and is not sufficiently accurate for microstructure analysis. Forsyth and Rothwell [4] used texture mapping methods to

represent microstructures on building surfaces, but the issue of fusing multi-image data were not explored. None of the authors considered a systematic removal of shadows and occlusions in their texture mapping methods.

In this paper we present an STME system that recovers building facet images from multiple source images and, as a first step towards detailed analysis of microstructures, extracts windows from walls. We make the assumption that an initial site model is given and that photometric and lighting information are known. The system employs a sophisticated multi-image texture mapping technique to eliminate the corrupting effects of shadows and occlusions and to find the best resolution. The system is model-driven, providing a context-based environment for microstructure analysis. The system explicitly extracts the 2-D window patterns attached to building walls to construct a refined model. In Section 2 we first give an overview of the system. Section 3, 4 and 5 describes the method in detail and Section 6 gives experimental results and conclusion.

2 System Overview

The STME system was originally developed under the ORD/ARPA RADIUS project, whose goal was to develop soft-copy model-based aerial image exploitation tools and infrastructure for image analysts. The test imagery is the Model Board 1 image set. There are altogether 8 original *site images*, J1-J8, taken independently at random times and positions under different lighting situations. In order to extract the surface textures and windows, some initial knowledge of the site is necessary, such as the shapes and locations of the buildings in each site image. This is obtained by applying the UMass Ascender automated site modeling system [3, 2] to the Model Board 1 imagery, resulting in 25 modeled buildings that represent most of the large-scale structures in the site. We call this initial representation of the buildings the *initial site model*. It gives a coarse

geometric description of the structures in the site. Figure 2 is a CAD display of the initial model. The largest building at the top of Figure 1, and on the right side of Figure 2, will be used to illustrate the STME system.

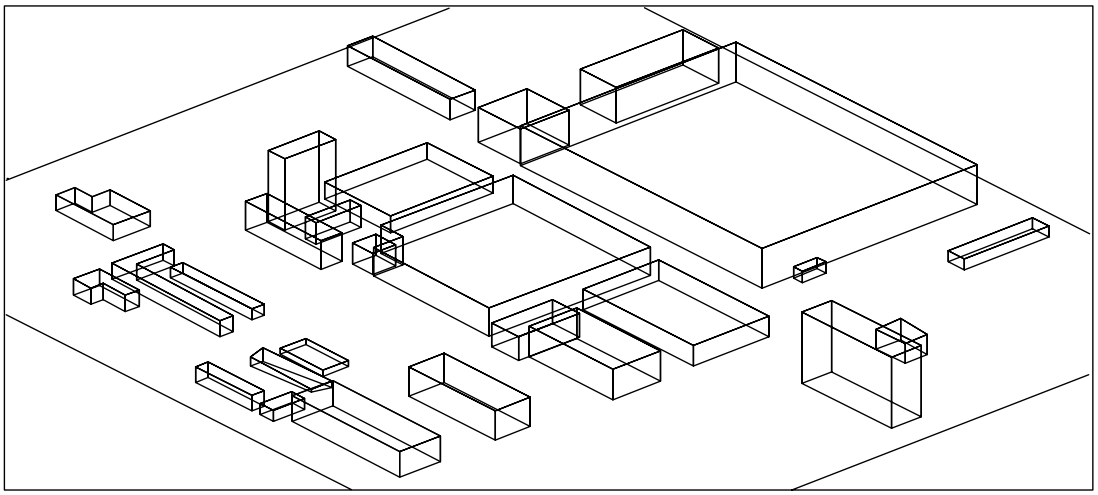


Figure 2: A CAD display of the initial site model

Using the initial model, each individual wall can be localized in each site image; however, due to perspective distortions and the small number of pixels on most walls, it is difficult to accurately map the windows from the original site images. What we need is a high quality texture map for each individual wall. This involves an evaluation of the "quality" of the visible walls in each site image. In addition, occlusion and shadowing may break up a texture map and it may be necessary to "piece together" the map from several images in order to remove the occluded or shadowed portions. All these are integrated in a texture extraction architecture, called the Orthographic Facet Image Library (Section 3). The final representation of a wall is called a Best Piece Representation (Section 4), whose pixels are composed of the "best pixels" taken from all the site images.

Although the texture mapping technique results in views without perspective distortion and free from occlusion and shadow corruption, microstructure analysis still suffers from the deficiency of data. How to integrate high-level knowledge into the analysis pro-

cess in an efficient way is a critical engineering issue. Section 5 focuses on this issue and shows how knowledge can be effectively used during window detection.

3 Orthographic Facet Image Library

The analysis of 2-D surface microstructures requires a high quality image representation of the surfaces. The goal of such a representation includes: 1. good resolution and brightness, 2. free from occlusions and shadows, and 3. free from perspective distortion. These goals are obtained in the STME system through the construction of an *orthographic facet image library* (OFIL). A *facet* is a modeled polygonal surface of a large-scale structure, such as a wall or a roof of a building. A *facet image* is a texture map of the facet as seen under orthographic projection. An OFIL for an initial site model stores indexed orthographic images of all the polygonal building facets that have been modeled in the site. The intensity values of each facet image are sampled from a site image using a texture mapping procedure. If the facet appears in more than one site image, the library will hold all the facet images (*versions*) for the facet. These multiple versions are individually indexed to facilitate library access. For example, a horizontal roof facet usually appears in all the aerial site images and thus has a complete set of orthographic versions in the library, whereas other facets like vertical walls only appear in some of the site images. Thus the availability of a facet version is an important piece of information to be indexed in the library. Other information such as the obliqueness and lighting conditions of the facet in the site image need to be recorded as well. In summary, an OFIL is an image database whose records are orthographically projected facet images together with relevant information to aid retrieval and analysis of these images.

Construction of the OFIL, which will be used for subsequent surface analysis, has many advantages. First, individual facets are stored separately, supporting context-based

processing. Specific surface structure extraction techniques can be applied only to relevant surfaces. Second, the orthographic facet images are free from perspective distortion, which is critical to the development of efficient techniques for extracting rectilinear, repetitive patterns many man-made structures possess. Third, the collection and alignment of all the visible versions of a building facet facilitates combining and fusing intensities from multiple views to produce a better, or clearer, view of each facet. Once the OFIL is built up, the original site images are no longer needed in subsequent surface analysis.

One important feature of the OFIL architecture is its ability to handle *occlusions* and *shadows* that arise on the object surfaces in a scene. For this purpose, an extra record, called the orthographic *labeling image*, is associated with each facet image. Each pixel of the labeling image is composed of a number of "attribute" bits that record whether the corresponding pixel on the facet image is occluded or in shadow. Figure 3 shows an example of this kind of labeling. The computation of occlusions and shadows in the current OFIL system is performed in a model-driven way, using the geometric data contained in the site model, the camera parameters of the site image, and light source parameters. This is a classic problem of hidden surface and shadow computation in computer graphics [12].

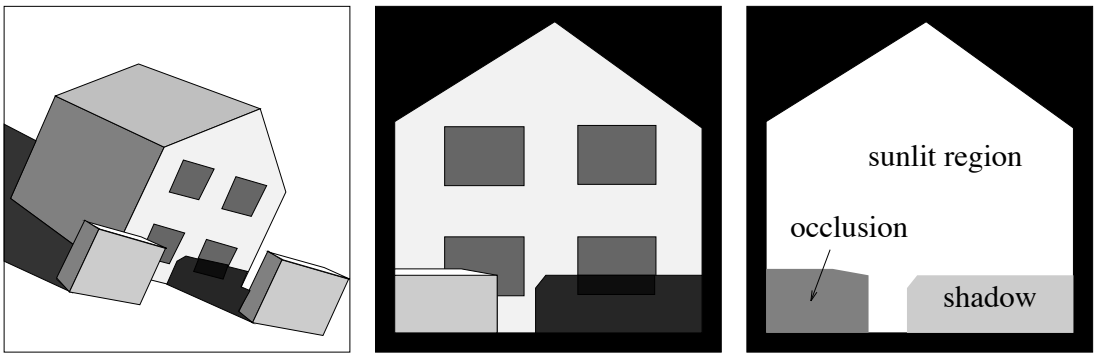


Figure 3: Occlusion and shadow labeling (left: site image, middle: facet image, right: facet labeling image)

Labeling images play an important role in indexing the pixel attributes of facet

images. In the current system, labeling images also provide other information besides occlusion and shadow. The complete set of attribute bits in a labeling image is:

1. *Facet Bit.* The system is able to handle arbitrary polygonal facets. This bit tells whether a pixel in the rectangular facet image is contained in the facet polygon.

2. *Presence Bit.* A building surface may lie partly outside of the boundaries of a site image. This bit labels whether a pixel's intensity is present in the site image.

3. *Occlusion Bit*, telling whether the pixel is occluded.

4. *Shadow Bit*, telling whether the pixel is in shadow.

To provide a glimpse of the OFIL, Figure 4(a) shows a set of facet images, with labeling, for a particular building facet in Model Board 1. This rectangular facet is the right wall of the largest building shown at the top of site image J1 in Figure 1. This wall appears only in site images J1, J2, J6, and J8, and thus only these four versions are available. In site image J1, part of the wall is cut by the image border, as is marked in the labeling image for the version from J1. Facet versions from J6 and J8 look darker because they are self-shadowed, i.e. oriented away from the light source. In site image J2 and J6, this wall is viewed from such an oblique angle that the textures mapped from these two images provide very little additional information over much of the wall surface. However, near the lower left of the wall there is another small building that occludes the wall in versions J1 and J8, but not in J2 and J6 due to the extreme obliqueness of the viewing angle. From this example we can see that multiple images are necessary to see all the portions of this particular building face, and that the OFIL has collected and organized the available information about this wall facet.

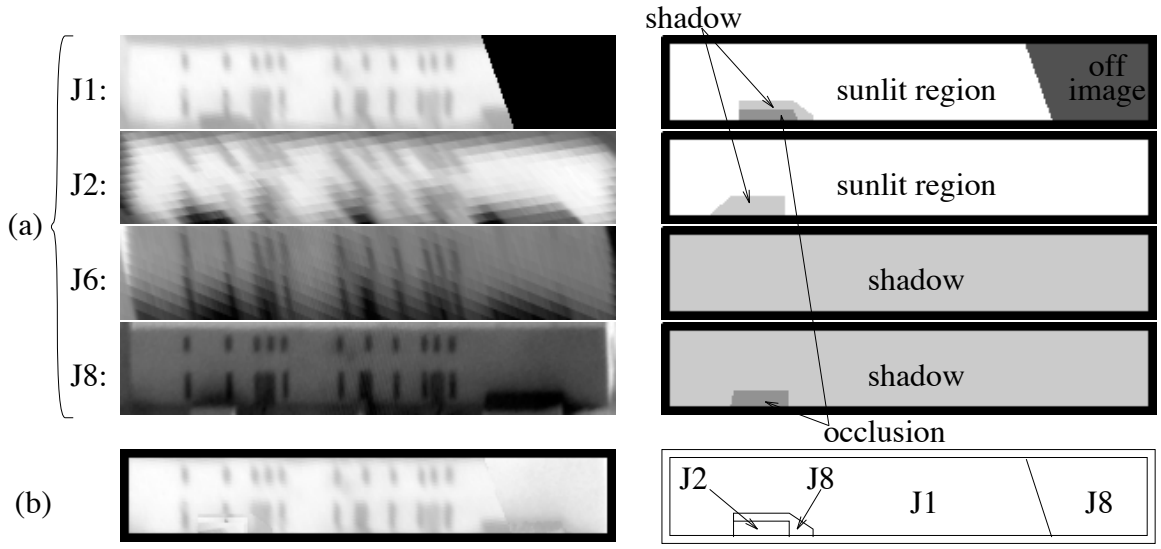


Figure 4: The application of OFIL architecture to a building facet

(a) left: orthographic facet image versions, right: their labelings

(b) left: the BPR image of the facet, right: regions of the intensity sources

4 Unique Intensity Representation

For many tasks it is desirable to produce a single, unique intensity representation of a facet. A simple approach is to select one "good" version from the facet images as the unique representation. The drawback of this approach is that any occlusions or shadows in that facet version will be included as artifacts in the resulting representative texture map. In addition, the version that is least corrupted by occlusions and shadows is not necessarily the clearest one. In Figure 4, the version that is least corrupted is the one from site image J2, but it is too blurred to be a good representation due to the obliqueness of the viewing angle.

In the STME system, we use a *best piece representation* (BPR) method for combining intensities into a unique representation of a facet. This method is based on the observation that different regions on a facet may only have good visibility in different image versions

due to the existence of occlusions and shadows. The final representation is a combined image whose pixel intensities are selected from among multiple image versions in the OFIL. A representative facet image synthesized in this way is called a *BPR image*.

The orthographic alignment of the facet image versions and the occlusion and shadow labeling make the BPR method very easy to compute under the OFIL architecture. We first define the term *piece*. Each facet image version is generally partitioned into three pieces: *sunlit piece*, *shadow piece* and *useless piece*. Sunlit pieces contain all the pixels that are labeled by Facet and Presence bits and not labeled by Occlusion and Shadow bits. They represent the sunlit portions of the surface. Shadow pieces, representing the shadowed part on the surface, contain all the pixels that are labeled by Facet, Presence and Shadow bits. All the other pixels are considered part of a useless piece (typically an occlusion) that provides no intensity information for the facet. Any particular pixel in the facet falls into a piece of one of these three kinds in each of its versions. To synthesize a representative facet image, the BPR algorithm runs through the pixels of the representative image, determines for each pixel which piece it falls into in each available facet version, evaluates the quality of each piece based on a criterion described below, and finally picks as the representative intensity of the facet pixel the corresponding pixel value from the highest quality piece.

Some issues arise in implementing the BPR method. One of them is how to evaluate the quality of a piece. Generally speaking, a good piece is a piece that reveals a clear, high resolution look at the surface detail. The detail-revealing ability of a piece is evaluated by a heuristic measure that takes into account such factors as the distance between the camera and the surface, the obliqueness of the viewing angle, and the lighting condition on the surface. In the BPR algorithm, for each pixel i_v on piece p_v in facet image version

v, we assign it a value given by the function

$$f(i_v) = \alpha(p_v)A(v),$$

in which $A(v)$ is the area that the facet occupies in the site image from which version v was obtained, and

$$\alpha(p_v) = \begin{cases} 1, & \text{if piece } p \text{ is a sunlit piece} \\ a, & \text{if piece } p \text{ is a shadow piece} \\ 0, & \text{if piece } p \text{ is a useless piece.} \end{cases}$$

The area factor $A(v)$ reflects the combined effects of surface-camera distance and viewing obliqueness. The weighting factor $\alpha(p_v)$ is set according to the attribute bits of the piece. In many applications, shadow pieces have a smaller range of intensity values, and are assumed to reveal less information than sunlit pieces. In our system we lower the heuristic value of shadow pieces by letting $a = 0.5$, an empirical constant. With the heuristic function defined in this way, every pixel i in the BPR image comes from the associated piece with the highest value of $f(i_v)$. That is,

$$i = i_u \text{ such that } f(i_u) = \max_v\{f(i_v)\}.$$

Another issue is the consistency of the intensity data. A BPR image is a synthesized image whose intensities are selected from different facet image versions. These intensities cannot be juxtaposed directly because they often come from different pieces captured under different lighting conditions. We solve this problem by making two assumptions. One is that every local piece on a surface has a similar intensity histogram distribution to the whole surface when seen under the same lighting conditions. This is true when the texture is fairly uniform on the surface, like on a wall where the windows are aligned evenly. The other assumption is that the intensities in a piece never reverse their order

under any lighting conditions. This assumption asserts that if windows are darker than the wall under sunlight, they will remain darker than the wall even when seen in shadows. Under these assumptions, we use a histogram adjustment algorithm, prior to running the BPR algorithm, to make the intensities from different facet image versions consistent. The algorithm has two steps. First, it chooses a useful piece (a sunlit piece or a shadow piece) as an *exemplar piece*, and computes its intensity histogram distribution. Second, the intensities of all the other pieces for that surface are adjusted to have the same histogram distribution as the exemplar piece. Another heuristic function, g, is used for selecting the exemplar piece. For any piece p_v on the facet image version v,

$$g(p_v) = \alpha(p_v)A(v)S(p_v),$$

where $\alpha(p_v)$ and $A(v)$ are as described above, and $S(p_v)$ is the ratio of the area of piece p_v to the area of the whole facet. The meaning of $A(v)S(p_v)$ is the area of piece p_v in the site image from which facet image version v is texture mapped. The bigger the area a piece occupies, the richer the texture it contains, and the more qualified it is to be chosen as the exemplar piece. The exemplar piece p is chosen by

$$p = p_u \text{ such that } g(p_u) = \max_{v,p_v}\{g(p_v)\}.$$

Figure 4(b) shows the BPR image synthesized using the facet images in Figure 4(a). The sunlit piece from the J1 version is chosen as the exemplar piece for histogram adjustment. We can see that some regions in the BPR image contain intensities from version J1, some others from J8 and J2. The intensities from different sources are more or less consistent.

5 Symbolic Window Extraction

The BPR texture maps provide a best image representation of the building walls without perspective distortion and, if there are enough views in the site images, free from occlusion and shadow corruption. However, symbolic extraction of windows is still difficult on BPR images, because building walls are usually viewed very obliquely in aerial images and contain very few pixels. In addition, the brightness of a wall image may vary considerably across its components even if they come from the same facet version, due to complicated sunlight reflections by the ground and other buildings in an urban area. A global intensity thresholding applied to a wall image would not give a satisfactory segmentation of windows. This is easily seen in Figure 5(a), which is a BPR image of one of the walls on the tall building on the left side in Figure 1. That the windows are darker than the wall is true only locally; in a global view, since the lower part of the wall is even darker than the windows on the upper part, global thresholding would clearly produce unacceptable results. Furthermore, since the image is noisy, it is difficult for a thresholding algorithm to maintain the rectangular shape of the windows.

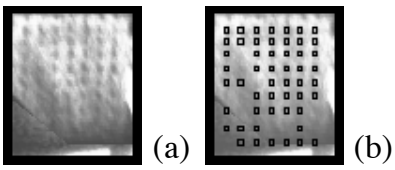


Figure 5: A noisy BPR map and the result of window extraction

(a) the noisy BPR map

(b) window extraction by using the ORG algorithm

A successful window extraction system must make use of high-level knowledge to deal with the deficiencies of the low-level data. In this section, we develop a generic algorithm to detect the windows. Three types of knowledge are incorporated into the algorithm:

1. the windows are oriented rectangles, whose edges are either vertical or horizontal;

2. the size and shape (ratio of height and width) of a window is constrained to a limited range;

3. a window lattice is a rectilinear, repetitive pattern. (That is, window edges are often regularly aligned.)

How to integrate this kind of high-level knowledge into the data-driven system is an important engineering issue. In practice, it is difficult to combine this kind of high-level knowledge with a primarily data-driven algorithm like global thresholding.

5.1 Oriented region growing

To avoid the difficulty of global thresholding algorithms for symbolic window extraction, the STME system resorts to an *oriented region growing* (ORG) algorithm, which handles the windows locally. Windows are defined as local "intensity dips" or "blobs" on a wall facet image. This is a reasonable assumption in aerial images where window structures are sufficiently distinguishable: although a window pixel might exceed the brightness of a wall pixel on the same wall, the pixels belonging to a window should on average be darker than the wall pixels nearby. With this assumption, once a window pixel (a *seed*) is found, we can grow locally from this pixel to a region that best symbolizes the window under certain criteria.

Section 5.2 will describe how to find the seeds for region growing. For now, we assume that a seed has been found and is to be grown into a window. Knowledge about window shapes is used here: regions are forced to grow as rectangles, and only in the four vertical and horizontal directions; that is, a window is grown from an initial, smaller rectangle to a larger one. The general idea is to search outward from the seed region until

a window edge is encountered. The search is done independently in the four directions, as shown in Figure 6(a). For each boundary of the seed region, the search is constrained to the rectangular area whose width is determined by the length of the corresponding boundary. The rectangular region is searched to determine the location of a boundary segment corresponding to a possible window edge. This is done by examining the intensity profile in the rectangular region as shown in Figure 6(a)(b). The criteria for determining the location of this edge is discussed below. A rectangular shape is fit to the four edges, which corresponds to a possible window hypothesis. In practice, this region growing step is applied iteratively, treating the newly found region as a new seed region. The iteration stops when the region ceases to grow; the resulting region is treated as the final window hypothesis. Figure 7 shows examples of two seeds growing to two windows. The initial regions are set to be 3×3 squares around the seeds. Note that the two windows are produced independently, although they are drawn in the same figure and they both terminate after three iterations.

The intensity profile in the rectangular search area is used to determine the position of the edge. In the examined area, the intensity values of the pixels are averaged to a 1-D signal, h, along the normal of the edge. Typically h is an increasing curve since the starting point (seed) is a local intensity minimum. A principal criterion for locating the edge is the position where the local second order derivative of h is sufficiently close to zero (Figure 6(b)). In real data, there may exist more than one zero second order derivative due to noise. Hence, among all the zero second order derivatives before the value of h starts to drop, we choose the one for which the slope of h is a maximum as the location of the edge (s_{edge}), i.e.

$$\left.\frac{\mathrm{d}h}{\mathrm{d}s}\right|_{s = s_{\text{edge}}} = \max\left\{\frac{\mathrm{d}h}{\mathrm{d}s}\,\middle|\,\frac{\mathrm{d}^2h}{(\mathrm{d}s)^2} = 0, \frac{\mathrm{d}h}{\mathrm{d}s} > 0\right\}.$$

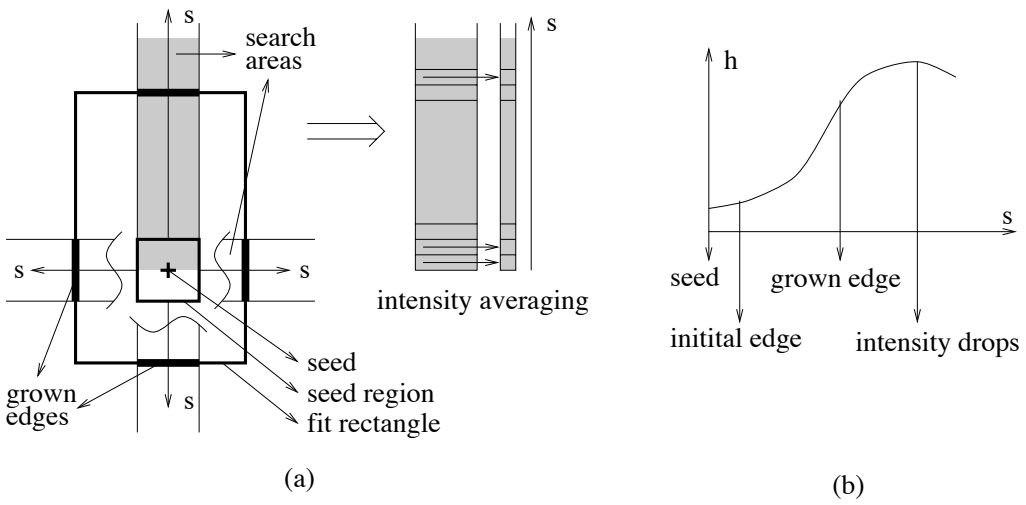


Figure 6: Illustration of oriented region growing

(a) one iteration of region growing

(b) the criterion to determine the position of resulting edge

5.2 Seeding

It is obvious that finding correct seeds is very important to the efficiency of the symbolic window extraction system. Incorrect seeds will not only lead to wrong window candidates, but also waste system resources in region growing. To reduce the number of invalid seeds, two filtering operations are applied prior to using the ORG algorithm. First, high-level knowledge about window lattices is used to restrict seeding positions. Second, a gradient descent algorithm is employed to merge repeated seeds.

Window lattices on building walls typically have a rectilinear, repetitive pattern, since windows are usually aligned vertically and horizontally. In the OFIL system, a window lattice on a facet image has an even simpler structure: windows are oriented parallel to the x and y axes in the facet images. With this knowledge, the initial seeding is done very easily. Figure 8 shows the horizontal and vertical intensity projection on the x and y axes of the BPR facet image generated in Figure 4(b). Since windows are

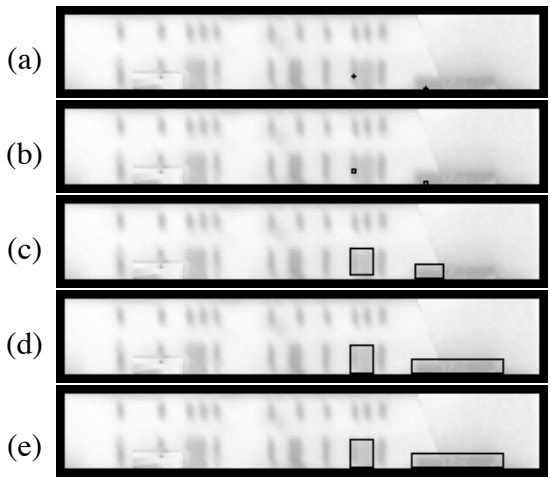


Figure 7: Two region growing examples

(a) the initial seeds

(b) the initial seed regions

(c) the first iteration of region growing

(d) the second iteration of region growing

(e) both regions stop growing at the third iteration

intensity dips, the 1-D projections of vertically or horizontally placed windows become 1-D dips, and thus the local minima (zero first order derivatives) are positions of good candidates. Sometimes, however, the 1-D dips in the projection are not complete due to the existence of irregular objects, such as the rightmost dark area on the bottom of the image in Figure 4(b), which makes the vertical projection (Figure 8(b)) a slope instead of a dip shape. To solve this problem, we use a zero third order derivative as the criterion to determine the seeding positions. In this case, the small "bumps" (not local minima) on the slope in Figure 8(b) are detected as well as true dips. A combination of the seed candidates from the two 1-D projections gives the 2-D positions (a "candidate lattice") of the initial seeds shown in Figure 9(b).

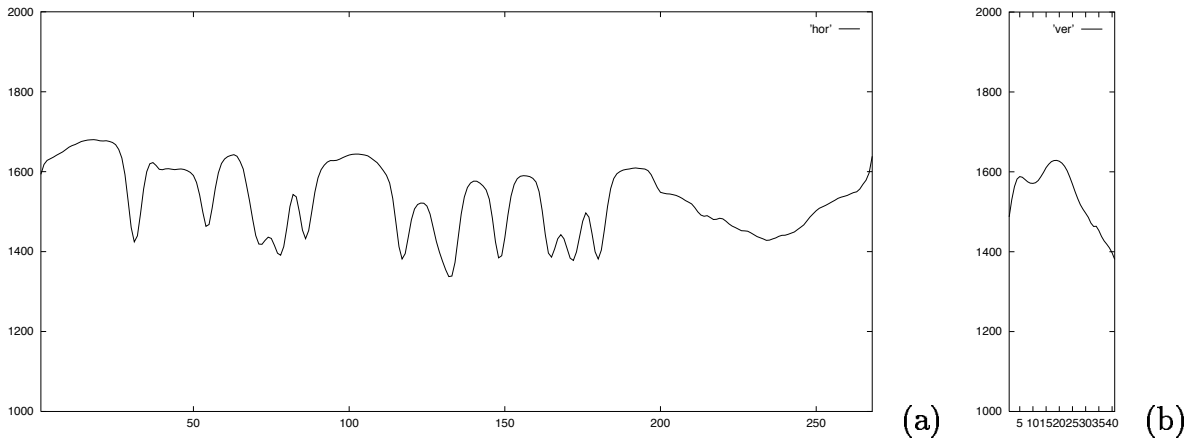


Figure 8: Intensity projections on the x and y axes of the BPR facet image in Figure 4(b)

(a) intensity projection on the x axis

(b) intensity projection on the y axis

Although application of high-level knowledge has reduced the initial seeds to a candidate lattice, in many cases the number of seeds is still too high because of the existence of the false positives on the lattice. The number of seeds can be reduced dramatically by noting that one characteristic of a window is that its intensity values correspond to a local minimum in the image. The facet image is used in conjunction with a gradient descent algorithm to move the initial seeds on the lattice to their nearest intensity minima. All seeds that arrive at the same minima are replaced by a single seed. In the experiment on the image of Figure 9(a), 610 initial seeds have been found on the candidate lattice (Figure 9(b)). Only 79 of them have survived the seed merging at local minima (Figure 9(c)). The merged seeds are then used for region growing.

5.3 Window filtering and adjustment

It can be seen from Figure 9(d) that those seeds that fall into the true windows have grown to be good window candidates, while bad seeds may grow into rectangles of ar-

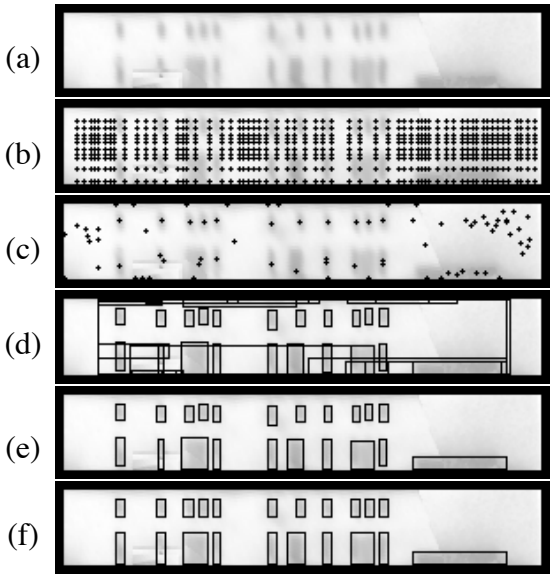


Figure 9: Symbolic window extraction

(a) the BPR facet image

(b) initial seeding

(c) seed merging

(d) region growing

(e) window filtering

(f) window lattice adjustment

bitrary shapes or sizes. To remove the bad window candidates, a set of "qualification tests" are applied to all the rectangles. These tests include: window size (area), window shape (ratio of height and width), and intensity variance within the window rectangle (we make the assumption that the intensities in a window region must be similar). The parameters in these tests could be set ahead of time or automtically determined as the result of preliminary experiments. Experimental results indicate that these parameters are usually very robust, because true windows often have very similar values in these physical measurements, while bad candidates are often far apart from these values. Of

the 79 rectangles grown from the seeds (Figure 9(d)), 21 pass the qualification test and are believed to be true windows (Figure 9(e)).

Due to local intensity fluctuations, the rectangles obtained thus far might not be aligned properly. The knowledge that windows are typically aligned in rows and columns is used to adjust the positions and sizes of these rectangles. Specifically, those windows whose top edge positions are vertically close enough to each other are forced to move their top edges to the same vertical position. The same technique is applied to the bottom, left and right edges. Figure 10 shows the scenario of the adjustment of the 21 windows obtained from Figure 9(e). A clustering algorithm is employed for this purpose. For example, the vertical positions of the bottom edges of all the windows are clustered into two classes; hence, the positions of the bottom edges are all unified to these two classes, resulting in the symbolic representation in Figure 10(f).

5.4 Algorithm Outline and conclusion

The following is an outline of the symbolic window extraction algorithm, with a BPR facet image as input and the symbolic windows as output.

1. Initial seeding: Project the intensity of the BPR image vertically and horizontally to get the 1-D projection signals along the x and y axes. Obtain the initial seed candidate lattice by computing the zero third order derivatives.

2. Seed merging: Apply the gradient descent algorithm to each seed to find its nearest 2-D local minimum. Merge the seeds that share the same local minimum.

3. Region growing: For each seed, set a 3×3 square as the initial region and grow the region iteratively until the region stops growing. The criterion for each iteration is described in Section 5.1.

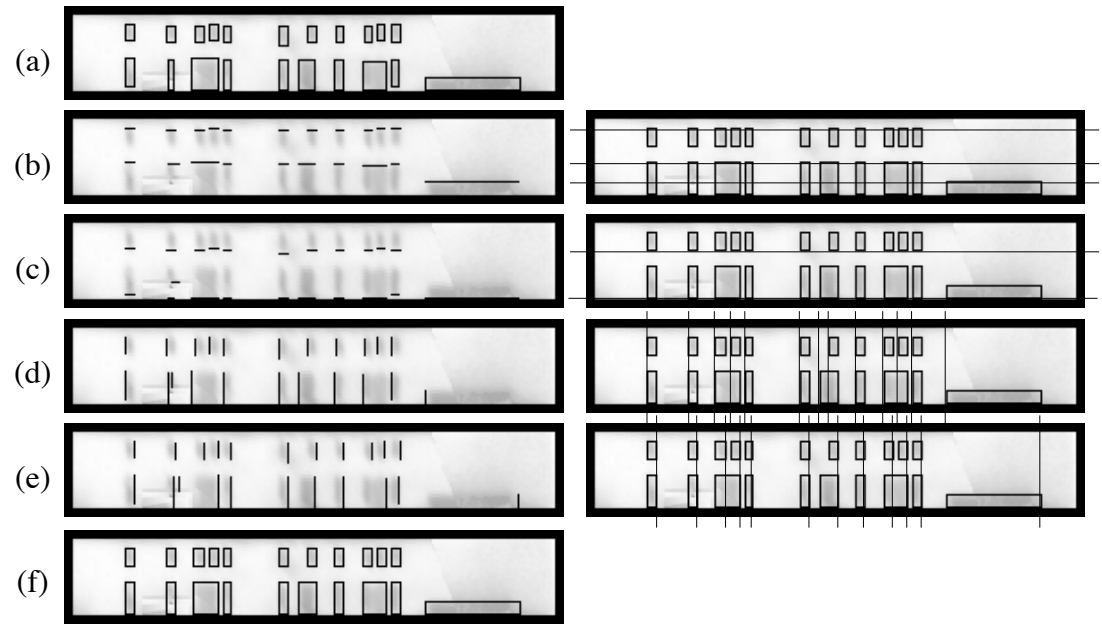


Figure 10: Window position adjustment by clustering

(a) windows to be adjusted

(b) adjusting the top edges

(c) adjusting the bottom edges

(d) adjusting the left edges

(e) adjusting the right edges

(f) adjusted windows

4. Window filtering: Filter out the unqualified rectangles using parameters based on window shapes, sizes, and intensity variances.

5. Window adjustment: Cluster the positions of window edges into several classes. Move window edges in the same class to the average position of them.

Figure 9(f) and Figure 5(b) show two examples of window extraction results using the algorithm. The major advantage of the algorithm is the easy and natural way in which various types of knowledge are integrated into the data-driven processing. The

oriented, local information based region growing technique avoids the problems that arise in the global thresholding method. The integration of knowledge does not complicate the processing, which is a critical issue in AI systems; rather, it simplifies the window detection process: regions are grown only in fixed directions and windows always keep their rectangular shapes. The other advantage of the system is that the processing can be easily parallelized: growing different regions are completely independent processes and can be done concurrently.

This window extraction strategy also takes advantage of the OFIL architecture. The use of the orthographic image from the OFIL simplifies the processing in that the effect of perspective has been removed. Consequently, processing in the vertical and horizontal directions is completely separable to procedures such as seeding and region growing. This simplifies the computation significantly. Moreover, recall that Figure 9(a) is a fused image from more than one source. On the resulting Figure 9(f), the rightmost flat rectangle and the second-left rectangle on the bottom of the image cross over two or more image pieces whose intensities come from different image versions in Figure 4(a), suggesting that the BPR algorithm maintains good consistency of image intensities from different sources.

6 Experimental Result and Discussion

Surface texture and microstructure extraction from aerial imagery is an important issue in many civilian and military applications. The problem is difficult due to perspective distortion, deficiency of supporting pixels, and corruption from shadows and occlusions. In this paper we have proposed an STME system, which contains a set of algorithms for dealing with these problems, under the assumption that an initial site model is given and sufficient camera and light source information is known. The first part of the algorithms systematically collects the facet intensity information from multiple site images

into an organized orthographic library, eliminates the effects of shadows and occlusions, and combines the intensities from different sources into a complete and consistent intensity representation for each facet. The second part focuses attention on wall facets and attempts to extract the 2-D window patterns attached to the walls. The algorithms are typically useful in urban sites where buildings are thickly settled.

The algorithms have been tested on the RADIUS Model Board 1 data set. A complete OFIL has been built up from images J1-J8 for the 25 buildings provided by the initial site model. 133 facet images are created automatically using the BPR algorithm as surface texture maps. Among them, there are 108 rectangular walls, 21 rectangular roofs and 4 L-shaped roofs. Among the 108 walls, windows appear on 42 of them, including all the walls of the major buildings. There are altogether 719 windows detected using the symbolic window extraction algorithm. The symbolically extracted windows are then incorporated into the initial site model to form an extended, refined model. A CAD display of the refined model is shown in Figure 11, which includes the walls that are illustrated in Figure 5 and Figure 9.

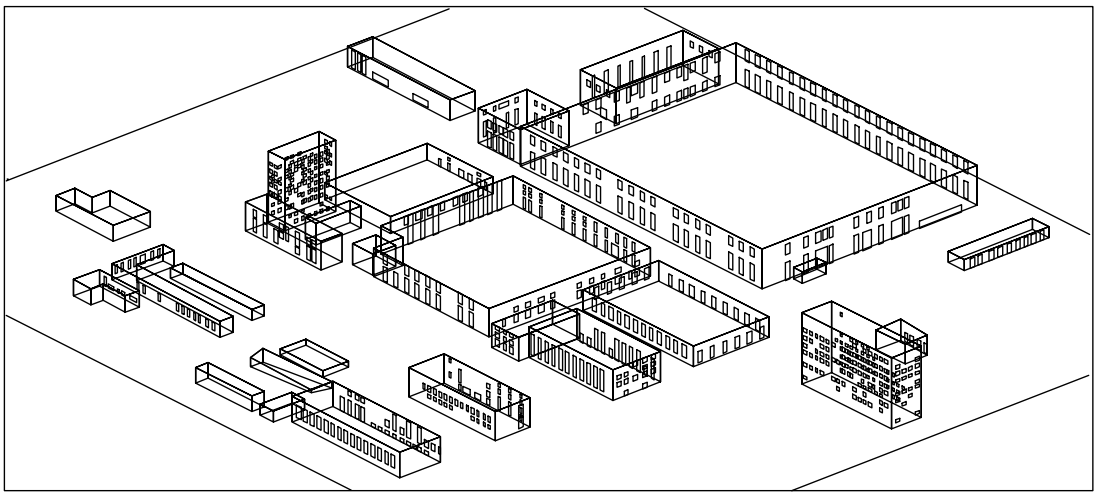


Figure 11: A CAD display of the refined site model

Site model visualization is an important area in which the STME system finds an

immediate application [10, 11]. While Cohen-Or et al.'s system [1] requires that the user stay reasonably distant from the terrain to maintain visual realism (in a fly-through, for example), the STME system allows closeup inspection of the model. Figure 12 shows a refined site model visualization of the large building in Figure 1. The windows in (a) are blurred and disappear when the viewpoint approaches the building because of the low intensity resolution on the wall. After an extracted window model is inserted in (b), the surface structure can be seen clearly at any reasonable close distance. The refined model can also be used for special visual effects using graphics tools. Figure 12 (c) shows a resulting view after a transparent attribute is assigned to the window model and synthesized textures pasted to the inserted floors inside the building. Rendering of the whole Model Board 1 site with surface texture mapping is shown in Figure 13. Using the fly-through generation software developed in UMass Computer Vision Lab, we have generated video sequences of renderings that simulate fly-throughs over the site.

The drawback of the OFIL architecture is that it is almost purely top-down, or model-driven. Inaccuracy and incompleteness in the initial model can cause problems. For example, the absence of undetected objects (such as smokestacks) in the site may lead to mis-identification of the occlusions and shadows caused by them. Inaccuracy of the camera model and/or camera parameters can cause mis-alignment of the different versions of a facet. Inaccuracy of sun angle parameters can cause erroneous shadow labeling and bad data fusion. One possible way to address these deficiencies, which is currently under study, is to introduce bottom-up, or data-driven, processing modules into the library. This includes shadow detection on the facet image and model/camera/sun parameter adjustment as directed by the comparison between the model-driven predicted and data-driven extracted shadows.

We emphasize the use of knowledge in the noisy environment in window detection.

Additional knowledge may also be integrated into the current system. For example, in the window lattice in Figure 5(b), some windows do not appear, because the support from the pixels (intensity dips) are not sufficient, i.e., seeding fails to find likely candidates. If we included a stronger constraint on the repetitive window patterns, these windows could be added back into the lattice by reasoning over the detected windows.

Currenly the system has been tested only on 2-D microstructures, i.e. windows and doors attached to building surfaces. However, the algorithms we have developed have the potential to deal with 3-D microstructures as well. Recent studies in terrain reconstruction and 3-D modeling [8, 7] have shown that elevation maps of building roofs can be obtained by a stereo matching algorithm from multiple aerial images. 3-D microstructures, such as air conditioners on top of building roofs, appear as small bumpy regions in the elevation maps. This is very similar to what windows present – dips – in the intensity maps of wall facets. Hence, as a topic of current research, the symbolic window extraction algorithm will be directly applied to the elevation maps to search for "elevation bumps" as 3-D rectangular microstructures.

Acknowledgements

We would like to thank Edward Riseman and Robert Collins for their discussions and comments, and Jonathan Lim for his technical support.

References

[1] D. Cohen-Or, E. Rich, U. Lerner, and V. Shenkar, "A Real-Time Photo-Realistic Visual Flythrough," *IEEE Trans. on Visualization and Computer Graphics*, vol. 2, no. 3, pp. 255-265, 1996.

[2] R. Collins, Y. Cheng, C. Jaynes, F. Stolle, X. Wang, A. Hanson, and E. Riseman, "Site Model Acquisition and Extension from Aerial Images," *Fifth International Conference on Computer Vision*, pp. 888-893, Cambridge, MA, 1995.

[3] R. Collins, A. Hanson, E. Riseman, C. Jaynes, F. Stolle, X. Wang, and Y. Cheng, "UMass Progress in 3D Building Model Acquisition," *Proc. Arpa Image Understanding Workshop*, pp. 305-315, Palm Springs, CA, 1996.

[4] D. Forsyth and C. Rothwell, "Representations of 3D Objects that Incorporate Surface Markings," *Applications of Invariance in Computer Vision*, Springer-Verlag Lecture Notes in Computer Science no. 825, pp. 341-357.

[5] P. Haeberli and M. Segal, "Texture Mapping as a Fundamental Drawing Primitive," in *Fourth Eurographics Workshop on Rendering*, Paris, France, 1993.

[6] P. Heckbert, "Survey of Texture Mapping," *IEEE Computer Graphics and Applications*, vol. 6, no. 11, pp. 56-67, 1986.

[7] C. Jaynes, F. Stolle, H. Schultz, R. Collins, A. Hanson, and E. Riseman, "UMass Progress in 3D Building Model Acquisition," *Proc. Arpa Image Understanding Workshop*, pp. 479-490, Palm Springs, CA, 1996.

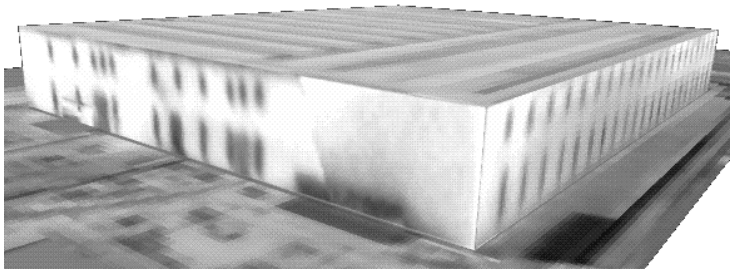
[8] H. Schultz, "Terrain reconstruction from widely separated images," *Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision II*, SPIE Proceedings Vol. 2486, pp. 113-123, Orlando, FL, 1995.

[9] X. Wang, A. Hanson, R. Collins, and J. Dehart, "Surface Microstructure Extraction from Multiple Aerial Images," in *Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision III*, SPIE Proceedings Vol. 3072, Orlando, FL, 1997.

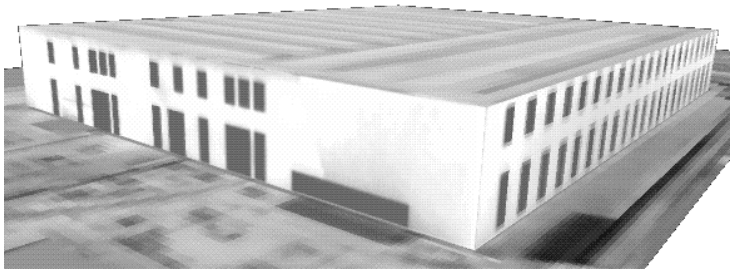
[10] X. Wang, R. Collins, and A. Hanson, "An Orthographic Facet Image Library for Supporting Site Model Refinement and Visualization," Technical Report #95-100, Dept. of Computer Science, Univ. of Massachusetts at Amherst, November 1995.

[11] X. Wang, J. Lim, R. Collins, and A. Hanson, "Automated Texture Extraction from Multiple Images to Support Site Model Refinement and Visualization," *The Fourth Int. Conf. in Central Europe on Computer Graphics and Visualization 96*, pp. 399-408, Plzen, Czech Republic, 1996.

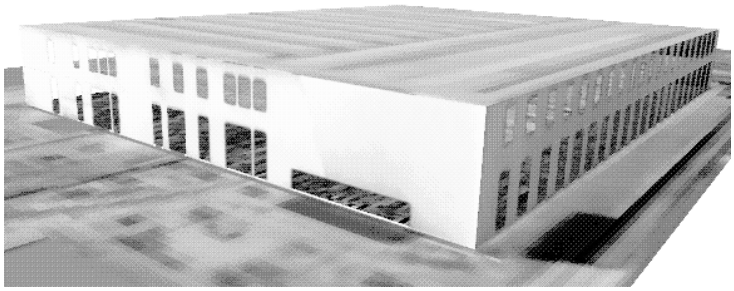
[12] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques: Theory and Practice*, ACM Press, New York, NY, 1992.



(a)



(b)



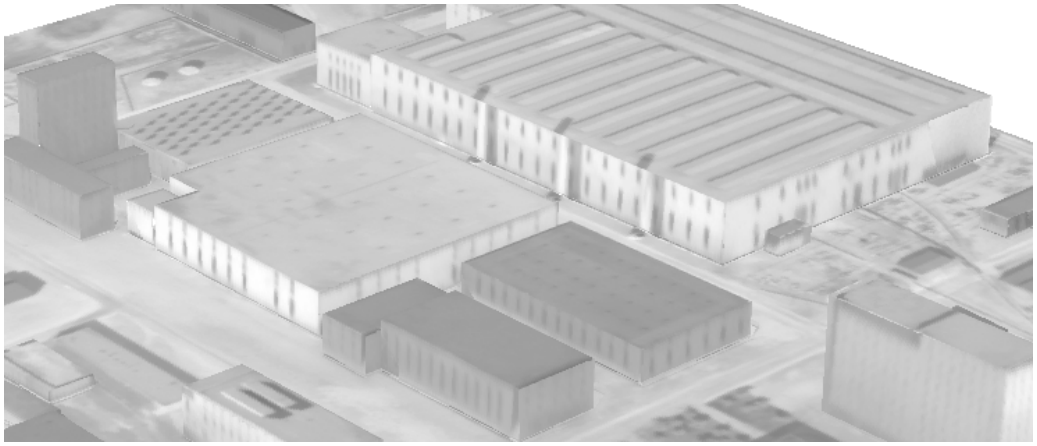
(c)

Figure 12: Visualization of the large building

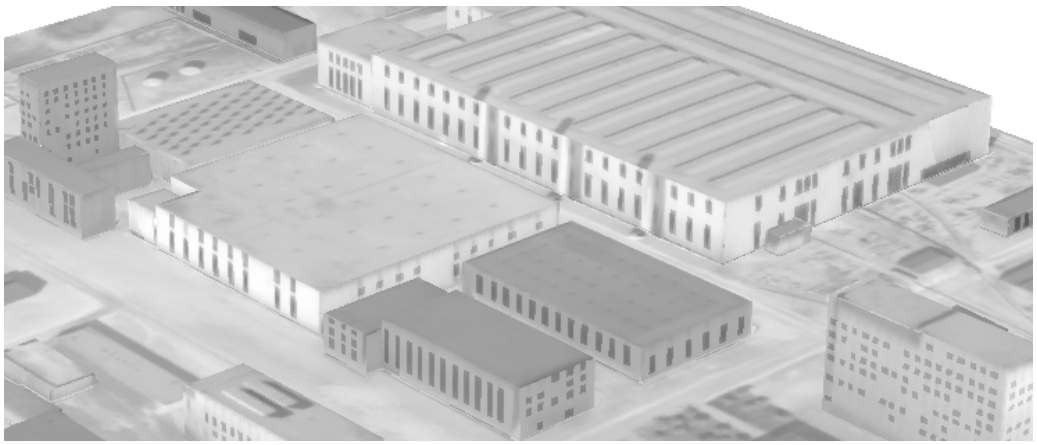
(a) with BPR texture maps

(b) with the refined model inserted

(c) with transparent windows and synthesized floors



(a)



(b)

Figure 13: Visualization of the Model Board 1 site

(a) with BPR texture maps

(b) with the window enhanced by the refined model