Surface Microstructure Extraction from Multiple Aerial Images

Xiaoguang Wang[a], Allen R. Hanson[a], Robert T. Collins[b], and Jeff Dehart[c]

[a] Dept. of Computer Science, Univ. of Massachusetts, Amherst, MA 01003
Email: {xwang, hanson}@cs.umass.edu
[b] The Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA 15213
Email: rcollins@cs.cmu.edu
[c] Army Research Labs, 2800 Powder Mill Rd., Adelphi, MD 20783
Email: dehart@msc.arl.mil

ABSTRACT

In this paper we present a system that recovers building facet images from multiple source images and, as a first step towards detailed analysis of microstructures, extracts windows from walls. The system employs a sophisticated multi-image texture mapping technique to eliminate the corrupting effects of shadows and occlusions and to find a "best piece representation" of each facet. The system is model-driven, providing a context-based environment for microstructure analysis. The 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 technique. High-level knowledge is incorporated to simplify the computation of symbolic window extraction. The algorithms are typically useful in urban sites. Experiments show successful applications of this approach to site model refinement.

Keywords: microstructure extraction, model refinement, texture mapping, aerial imagery, image understanding

1 INTRODUCTION

In many aerial imagery applications, advanced optical and digital technologies are capable of revealing not only *large-scale structures* such as buildings, but also *microstructures* such as windows, doors and roof vents on buildings. Microstructures have many potential uses in military and civilian applications, because they often provide functional and cultural information, which is critical in applications such as landmark recognition, mission planning and assessment, and other applications requiring detailed models. Traditional building detection techniques[2] 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.

Aerial image understanding systems have traditionally focused on the detection and recognition of large-scale structures, leaving the microstructure untouched. This is partly due to the difficulties encountered when attempting to extract structures near the resolution of the image. 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.

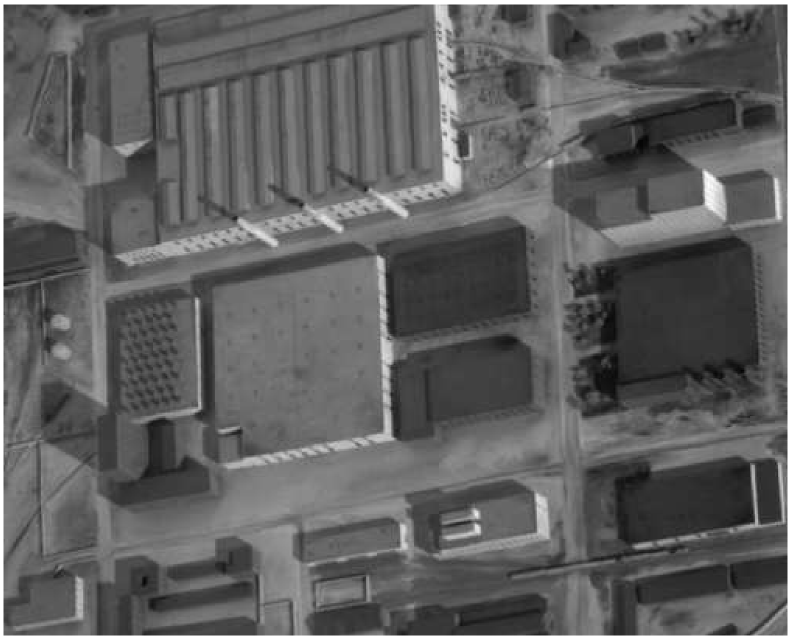


Figure 1: Part of site image J1 from Model Board 1

The above analysis suggests that a microstructure extraction 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. 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.

In this paper we present a system that extracts 2-D microstructures such as windows and doors from walls. The 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 goal of the window extraction module is to provide image analysts with an automatic environment within which the geometric site models acquired from the original *site images* can be refined. The test imagery is the Model Board 1 image set. There are altogether 8 images, J1-J8, taken independently at random times and positions under different lighting situations. The photometric and lighting parameters are given with each image. In order to automatically extract the windows from the building walls, 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 Ascender automated site modeling system[1,2] to the Model Board 1 imagery, resulting in an *initial site model* of 25 modeled buildings that represent most of the large-scale structures in the site. It gives a coarse geometric description of the structures in the site. Figure 2 is a CAD display of the initial model.

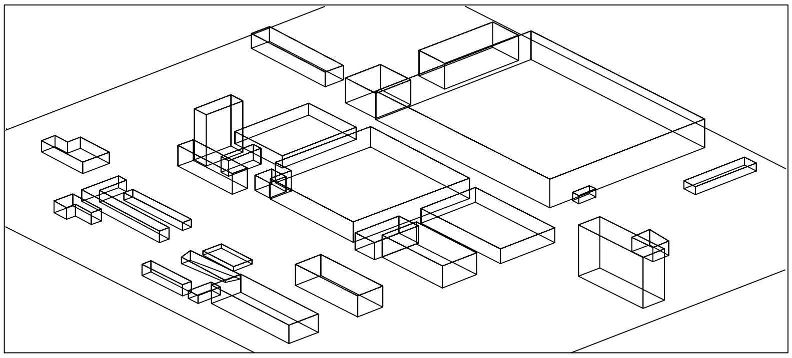


Figure 2: A CAD display of the initial site model

We employ the *Orthographic facet image library* (OFIL)[6,8] as the infrastructure for the window extraction task. The OFIL system systematically collects the building facet intensities from multiple aerial images into an organized orthographic library, eliminates the effects of shadows and occlusions, and combines the intensities from different sources to form a complete and consistent intensity representation for each facet. The organization of an OFIL is described in the next section. The window extraction algorithms are presented in Section 3, and the experimental results are given in Section 4.

2 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 our system through the Orthographic Facet Image Library. An OFIL for an initial site model is an image database whose records are orthographically projected *facet images* of all the modeled building faces, together with relevant information (in particular, *labeling images* that record the attributes of each pixel in facet images) to aid retrieval and analysis of these images. Details of the construction of such an OFIL can be found elsewhere.[6,8] We use an example facet to show the role of the OFIL in the microstructure extraction system.

2.1 Modeling of occlusions and shadows

Figure 3(a) shows a set of facet images, as well as the corresponding labeling images, for the right wall of the largest building shown at the top Figure 1. This wall appears only in site images J1, J2, J6, and J8. In 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 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.

One important feature of the OFIL architecture is its ability to handle occlusions and shadows that arise on the object surfaces. 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

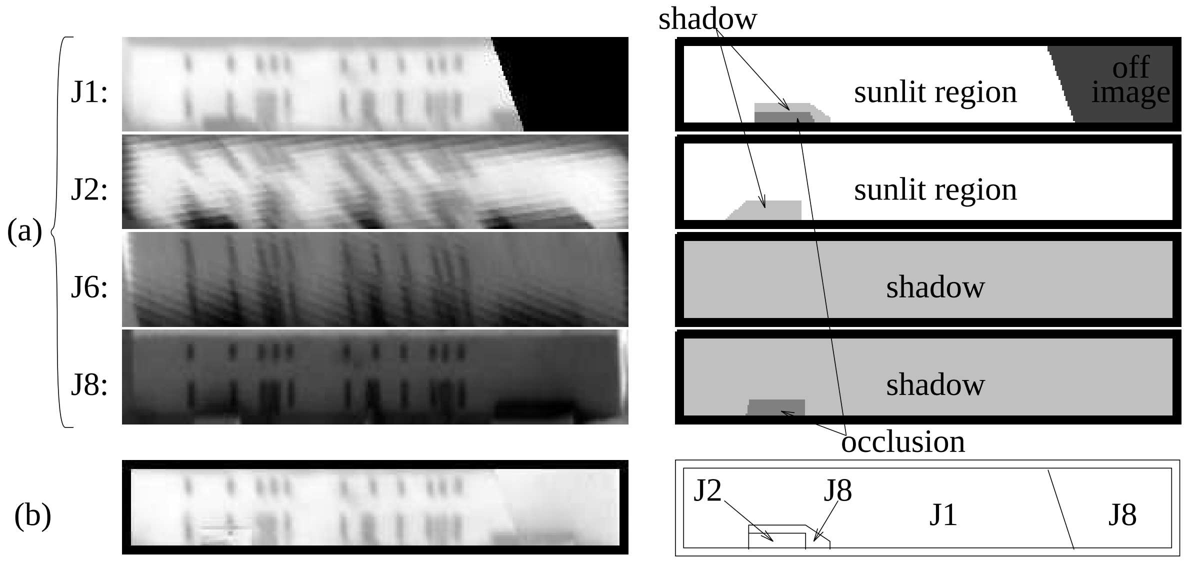


Figure 3: The application of OFIL architecture to a building facet
(a) orthographic facet images and labelings (b) the BPR image of the facet and regions of its intensity sources

graphics.[5] The result is saved in the pixels of the labeling images, which are composed of a number of attribute bits that record whether the corresponding pixel on the facet image is occluded or in shadow.

2.2 Best piece representation

For many tasks it is desirable to produce a single, unique intensity representation of a facet. The OFIL system uses 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. Each facet image version is generally partitioned into three pieces: *sunlit piece*, *shadow piece* and *useless piece* (i.e. occlusions or pixels falling outside the site image). 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, and finally picks as the representative intensity of the facet pixel the corresponding pixel value from the highest quality piece. To evaluate the quality of a piece, a heuristic function f is computed for each pixel i_v on piece p_v in facet image version v:

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

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 $\alpha(p_v)$ reflects the effect of the attributes that the piece possesses. In many applications, shadow pieces have a smaller range of intensity values, and are assumed to reveal less information than sunlit pieces. 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,

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

Intensities selected from different facet image versions cannot be juxtaposed directly in the BPR image because they often come from different pieces captured under different lighting conditions. A histogram adjustment algorithm is run, prior to the BPR algorithm, to make the intensities from different facet image versions consistent. The algorithm chooses a useful piece (a sunlit piece or a shadow piece) as an *exemplar piece*, and computes its intensity histogram distribution. The intensities of all the other pieces for that surface are adjusted to have the same histogram distribution as the exemplar piece. A 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 factor $S(p_v)$ is considered because a bigger piece usually contains richer texture, and is more qualified to be 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 3(b) shows the BPR image synthesized using the facet images in Figure 3(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 consistent.

3 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 4(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.

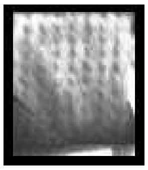
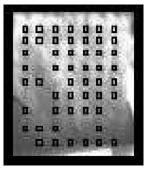

(a)

(b)

Figure 4: A noisy BPR map (a) and the result of window extraction (b)

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; and 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.

3.1 Oriented region growing

To avoid the difficulty of global thresholding algorithms for symbolic window extraction, we resort 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 3.2 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 5(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 5(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 6 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 5(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.

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

3.2 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

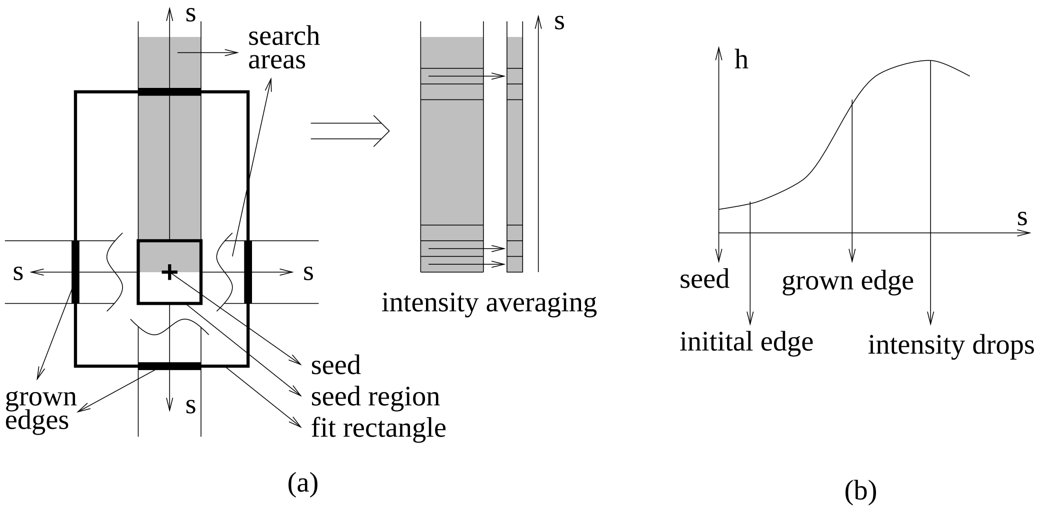


Figure 5: Illustration of oriented region growing
(a) one iteration of region growing (b) the criterion to determine the position of resulting edge

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 7 shows the horizontal and vertical intensity projection on the x and y axes of the BPR facet image generated in Figure 3(b). Since windows are 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 3(b), which makes the vertical projection (Figure 7(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 7(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 8(b).

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 8(a), 610 initial seeds have been found on the candidate lattice (Figure 8(b)). Only 79 of them have survived the seed merging at local minima (Figure 8(c)). The merged seeds are then used for region growing.

3.3 Window filtering and adjustment

It can be seen from Figure 8(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 arbitrary shapes or sizes. To remove the bad window

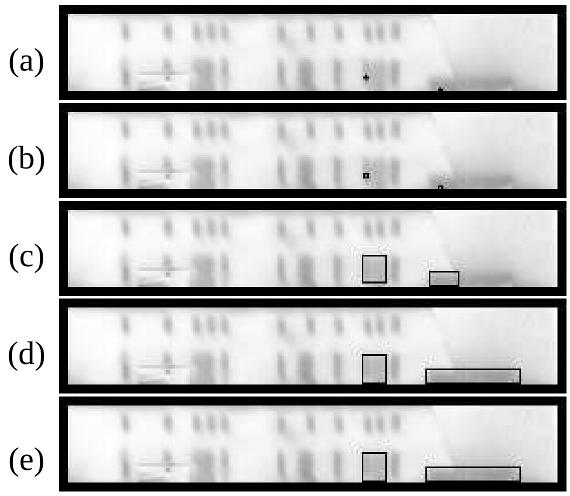


Figure 6: Two region growing examples
(a) the initial seeds (b) the initial seed regions (c) the first iteration of region growing (d) the second iteration of region growing (e) both regions stop growing at the third iteration

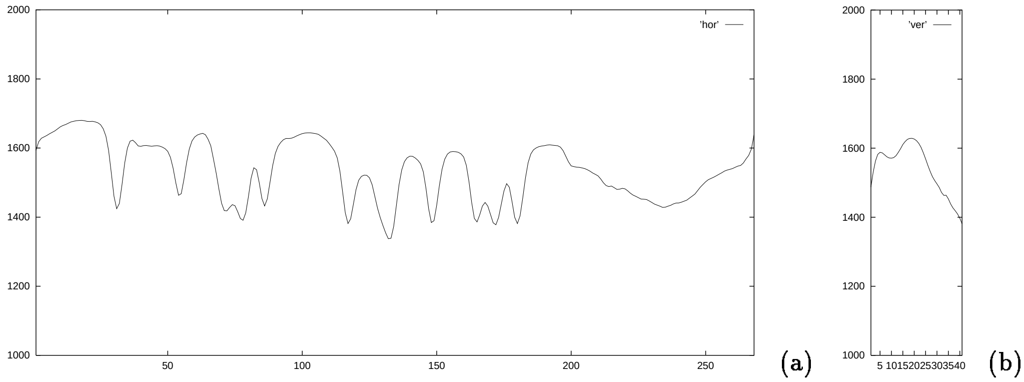


Figure 7: Intensity projections on the x and y axes of the BPR facet image in Figure 3(b)
(a) intensity projection on the x axis (b) intensity projection on the y axis

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 8(d)), 21 pass the qualification test and are believed to be true windows (Figure 8(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 9 shows the scenario of the adjustment of the 21 windows obtained from Figure 8(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 9(f).

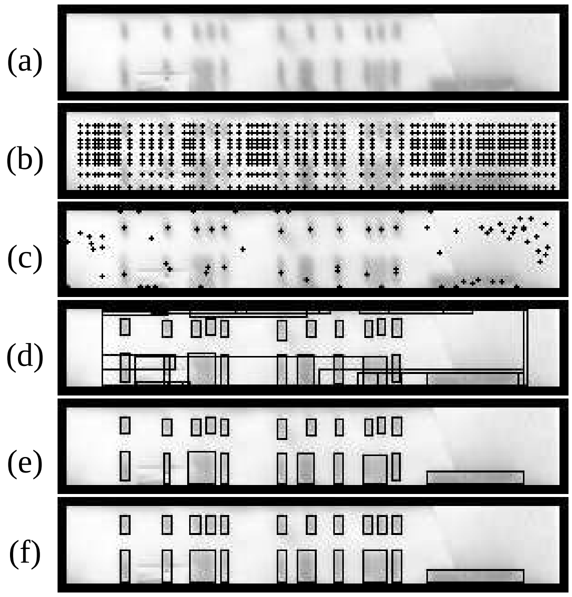


Figure 8: Symbolic window extraction
(a) the BPR facet image (b) initial seeding (c) seed merging (d) region growing (e) window filtering (f) window lattice adjustment

3.4 Algorithm outline

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 3.1.
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.

4 EXPERIMENTAL RESULT AND DISCUSSION

Figure 8(f) and Figure 4(b) show two examples of window extraction results using the symbolic window extraction algorithm. The algorithm has been tested on the RADIUS Model Board 1 data set. A complete orthographic facet image library has been built up from images J1-J8 for the 25 buildings provided by the initial

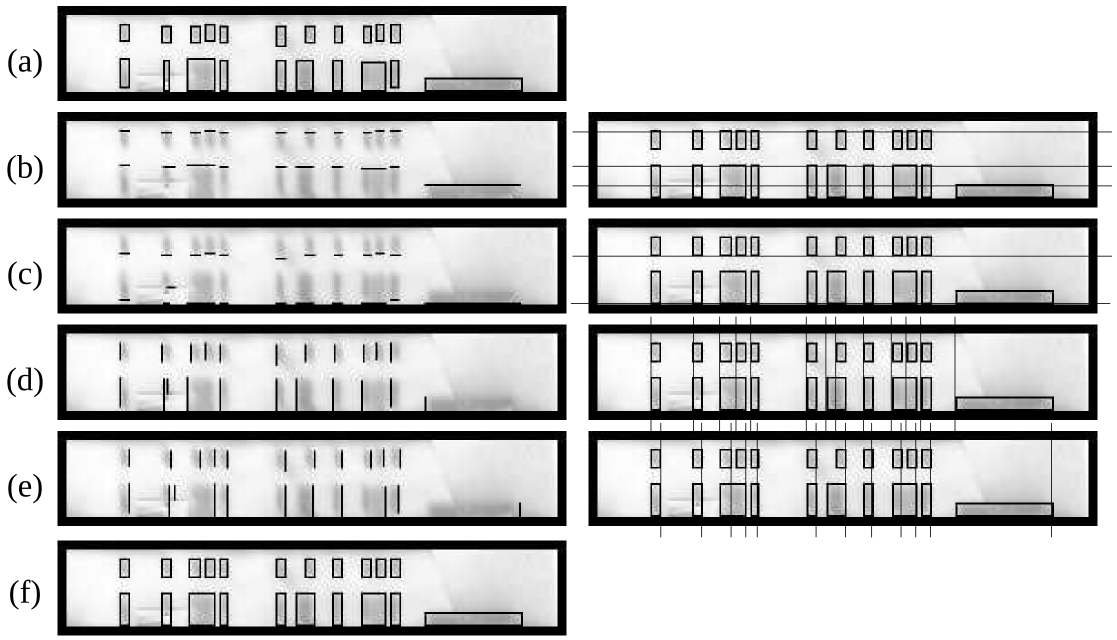


Figure 9: Window position adjustment by clustering
(a) windows to be adjusted (b) adjusting the top edges (c) adjusting the bottom edges (d) adjusting the left edges (e) adjusting the right edges (f) adjusted windows

site model. 133 facet images are created automatically using the BPR algorithm as surface texture maps. Among the 108 wall images, 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 10, which includes the walls that are illustrated in Figure 4 and Figure 8.

Figure 11 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 close distance. The refined model can also be used for special visual effects using graphics tools. Figure 11 (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.

The major advantage of the algorithms is the easy and natural way in which various types of knowledge are integrated into the data-driven processing. The ORG algorithm 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 8(a) is a fused image from more than one source. On the resulting Figure 8(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

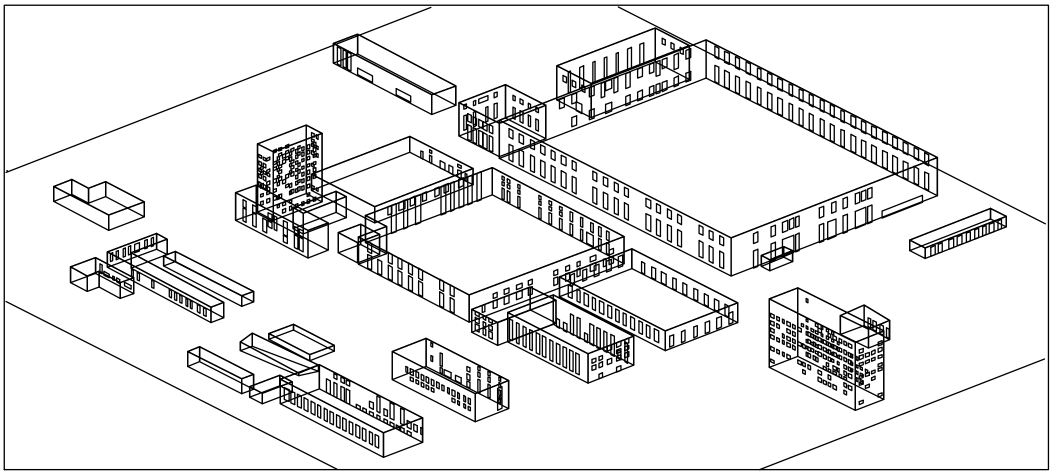


Figure 10: A CAD display of the refined site model

in Figure 3(a), suggesting that the BPR algorithm maintains good consistency of image intensities from different sources.

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 4(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.

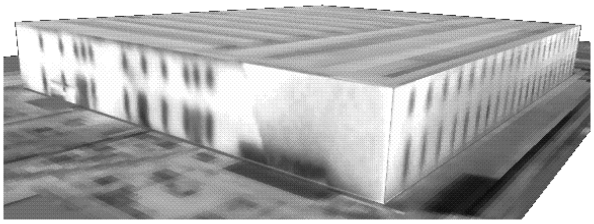
Currenly the system has been tested only on 2-D microstructures, i.e. windows and doors attached to building surfaces. However, the algorithms we have developed have the potential to deal with 3-D microstructures as well. Recent studies in terrain reconstruction and 3-D modeling[4,3] have shown that 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.

5 ACKNOWLEDGEMENTS

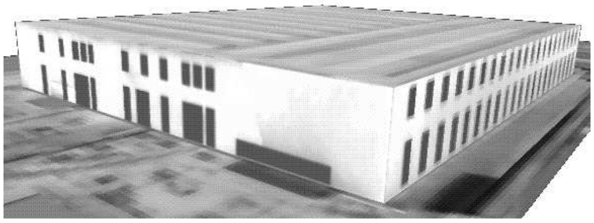
We would like to thank Professor Edward Riseman for his discussions and comments, and Jonathan Lim for his technical support. This work was funded by the RADIUS project under ARPA/Army TEC contract number DACA76-92-C-0041 and NSF grant CDA-8922572.

6 REFERENCES

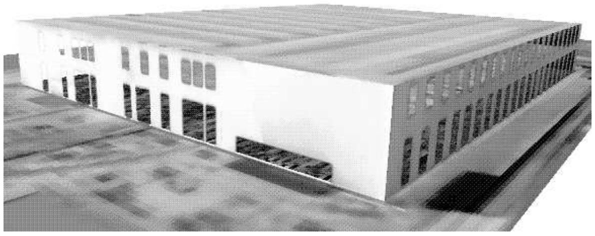
[1] R. Collins, A. Hanson, E. Riseman, C. Jaynes, F. Stolle, X. Wang, and Y. Cheng, "UMass Progress in 3D Building Model Acquisition", *Proc. Arpa Image Understanding Workshop*, pp. 305-315, Palm Springs, CA, 1996.



(a)



(b)



(c)

Figure 11: Visualization of the large building
(a) with BPR texture maps (b) with the refined model (c) with transparent windows and synthesized floors

[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] 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.

[4] 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.

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

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

[7] X. Wang and A. Hanson, "A System for Surface Texture and Microstructure Extraction from Mutiple Aerial Images," Technical Report #97-022, Department of Computer Science, University of Massachusetts at Amherst, 1997.

[8] 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.