An Orthographic Facet Image Library for Supporting Site Model Refinement and Visualization *

Xiaoguang Wang, Robert T. Collins, and Allen R. Hanson

Department of Computer Science
Box 34610, University of Massachusetts
Amherst, MA. 01003-4610

Abstract

This paper addresses the issue of combining intensity information from multiple images for analysis of object surface intensities to support model refinement and scene rendering. Given camera and light source parameters for each image, and a 3D CAD model of objects in the scene, the textures of object surfaces are systematically collected into an organized orthographic library. Occlusions and shadows caused by objects in the scene are predicted from the model and associated with each retrieved surface. The problem of combining intensities from multiple images is explored, and the result is an algorithm that produces a unique surface intensity representation that is as complete and consistent as possible. A successful application of this approach to model refinement and scene visualization is shown, using results from the RADIUS aerial image understanding project.

Keywords: texture mapping, model refinement, scene rendering

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

1 Introduction

A goal of the ORD/ARPA RADIUS project is to develop soft-copy model-based image exploitation tools and infrastructure for image analysts. An important function of the evolving system is the development of a *site model* of an area from multiple images (*site images*) taken over different days, from a variety of positions, and under variable weather and lighting conditions. Uses of the site model include 3D site visualization and familiarization, mission planning and assessment, and change detection [3, 9]. Consequently, an important component of the site model is an accurate geometric representation of significant objects in the site, such as buildings, as polygonal models with associated surface texture maps derived from the site images.

Collins, et al. [5, 4] describe recent progress in RADIUS site model acquisition. A set of image understanding algorithms have been developed for automated building model acquisition; the algorithms currently assume a generic class of flat roofed, rectlinear buildings. Line segments are extracted from each image of the site, and a building detection algorithm employs both bottom-up (data-driven) and top-down (model driven) perceptual grouping techniques to detect significant building rooftops from a single image [8]. Supporting geometric evidence is then located in other images via epipolar line segment matching, and the precise 3D shape and location of each building is determined by constrained multi-image line triangulation. New methods for performing camera resection and correspondence matching have also been investigated [2]. An initial model-building experiment was reported based on the RADIUS Model Board 1 images J1–J8 (a portion of image J1 is shown in Figure 1). The result was a 3D polygonal site model containing 25 buildings that represent most of the central structures in the Model Board 1 site. Figure 2 is a CAD display of the site model.

This paper focusses on automated acquisition of complete and consistent texture maps from multiple images in order to support subsequent detailed surface analysis, scene rendering, and other goals. Given a polygonal model, a camera viewpoint, and a texture map, rendering the texture onto the model surfaces is a well-understood technique in computer graphics [10, 7]. However, extraction of a complete and consistent texture map for all surfaces of a 3D building model is a challenging task. Consider the problems involved in generating texture maps for the largest building in the site, which appears at the top of Figure 1. Only two of the four building walls are visible in this image, so at least one other image must be used. There are only a few pixels in the vertical direction of the building wall; other, more oblique views, may have higher resolution. Buildings are often built fairly close to one another and it may not be possible to get a good view - 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. Since the images may be taken at various times of day, from different positions, and under varied weather conditions, the brightness of surface intensity maps may vary considerably from image to image. All of these factors must be considered when generating a consistent texture map from the site images. Traditional graphics algorithms provide no direct way for collecting and combining intensities under various conditions for texture consistency and completeness.

Once a consistent texture map is obtained, it can be applied to the building surfaces to yield more realistic graphical displays (e.g. see Figure 7) suitable for site visualization and familiarization. In addition, the texture maps can play an important role in detailed geometric analysis of the site and can aid in change detection. In some cases, it is desirable to model small building details such as windows, doors, and roof vents. This process, called *model refinement*, is facilitated by the texture maps, as they provide an intensity image for each building face that is free from perspective distortion, and together with the building model provide important contextual cues for focussing the analysis, i.e. door and window extraction procedures are applied only to wall texture maps, while roof vent extraction is applied only to roof maps.

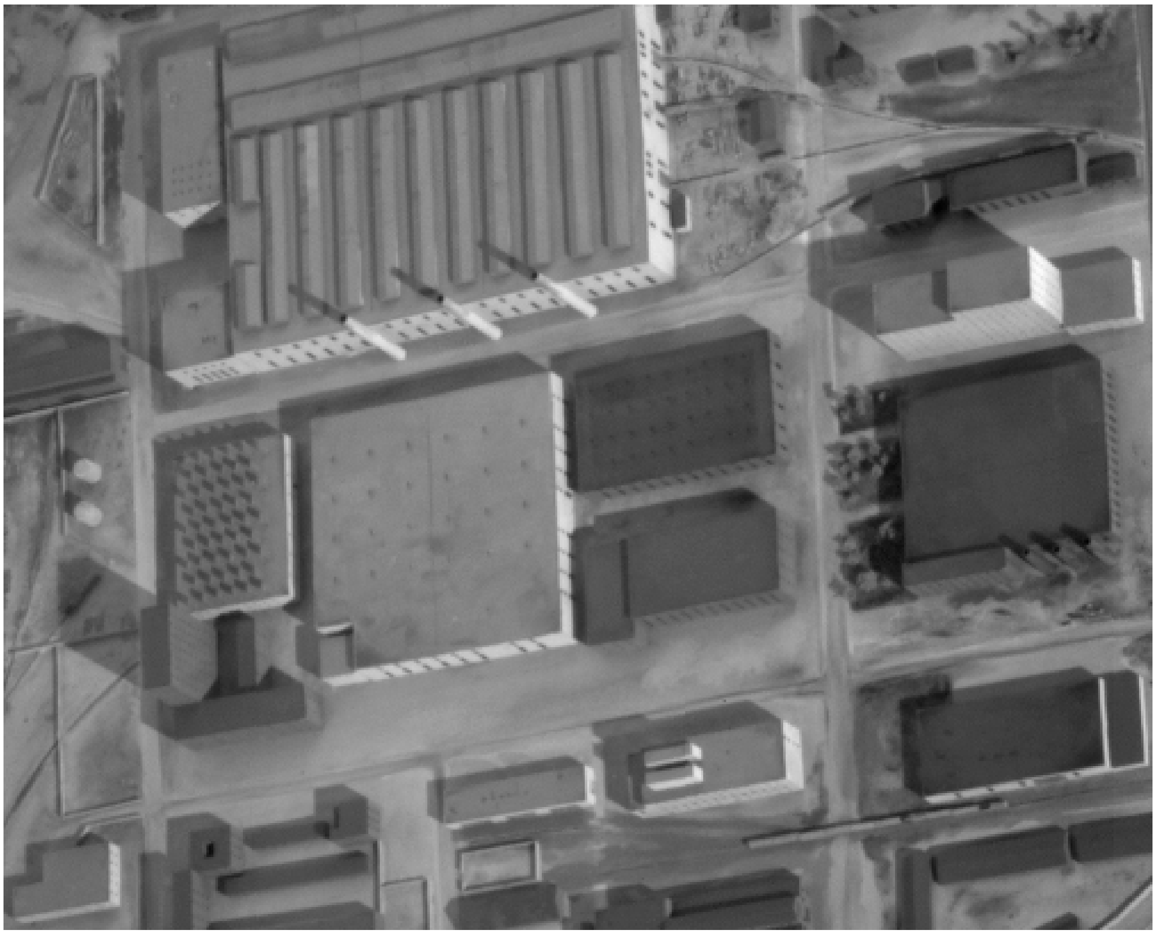

Figure 1: Part of site image J1 from Model Board 1

In the following sections, we describe a multi-image texture mapping architecture, which supports automatic extraction of high-quality texture maps for all modeled polygonal surfaces by combining intensity information from multiple images. The approach is based on an orthographic facet image library, which is described in the next section. Section 3 discusses modeling of occlusions and shadows and how this information is used

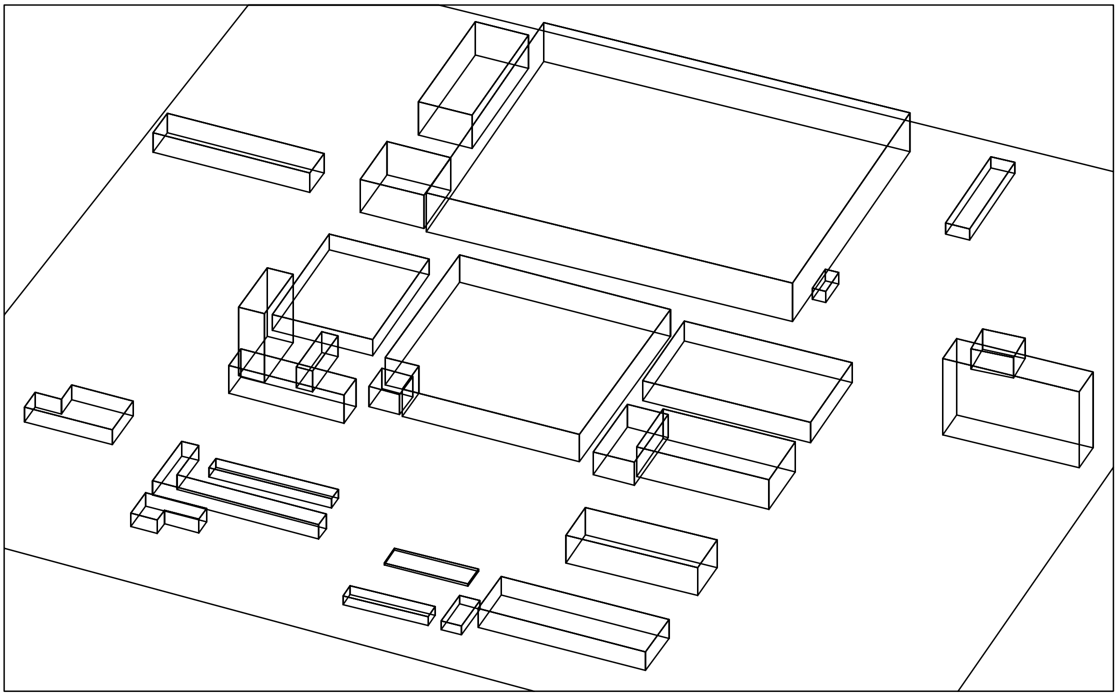

Figure 2: A CAD display of the acquired site model

to determine usable portions of individual texture maps. Section 4 describes how a single, consistent representation of the texture map for a facet is obtained from the tagged library of facets. Section 5 discusses related issues in model refinement and scene rendering, and Section 6 describes future work. Potential uses of the presented approach are not confined to RADIUS. It can be applied to any scene containing objects approximated by polygonal facets of arbitrary shape and orientation.

We note in passing that the correctness of the given site model and camera parameters is important for transforming between an image and a surface in the 3D site, which is the basic computation in texture mapping. Since the acquisition of site model and camera parameters are based on low-level image processing, errors arise due to image noise and unmodeled distortion. In the paper, we assume that those errors are negligible at the scale of objects like buildings.

2 Orthographic Facet Image Library

Goals like model refinement and scene rendering require a convenient method for accessing sensed intensity data. The only sensed data available are multiple perspective images (site images) captured over different days, from various viewing positions, and under variable weather and light conditions. A simple mechanism for accessing this data is to keep the site images in a database and to retrieve surface intensity information by reading pixel values directly from the original images themselves. This mechanism has a number of problems: the prohibitive amount of storage space used, the necessity of arbitrating between multiple images that all contain the same object surface, and the perspective distortion of the the object surfaces in the original images, which causes difficulty for further image analysis.

The central idea in the architecture we present for managing site model intensity information is the *orthographic facet image library* (OFIL). A *facet* in the site model is a modeled polygonal surface of a building, such as a wall or a roof. An *orthographic facet image* (or concisely, *facet image*) is an intensity image of the facet, as seen under *orthographic projection.* Orthographic projection [1] of a facet is a special perspective transformation in which the image plane of the camera is parallel to the facet and its focal point is at infinity, resulting in an image free from perspective distortion. An OFIL for a 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 traditional, model-directed texture mapping algorithm. 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 need to be 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. Besides availability, information like localization (how the facet is aligned in the orthographic image) and visibility (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 building facet images together with relevant information to aid retrieval and analysis of these images.

Construction of an OFIL as an intermediate-level representation has immediate advantages over the mechanism of storing raw site image intensities. First, individual facets are stored separately so that specific surface structure extraction techniques can be applied only to relevant surfaces: window extraction on wall images, roof vent computations on roof images, etc. Second, many man-made structures related to buildings have rectilinear, repetitive patterns, like the lattices of windows on building walls. The orthographic facet image provides a view that is free from perspective distortion, which is critical to the development of efficient techniques for extracting these patterns. Third, the collection and alignment of all the visible versions of a building facet provides a mechanism for comparing and combining intensities from multiple views to produce a better, or clearer, view of each facet (see Section 4). Finally, a set of separately stored facet images is a natural and convenient component of a system for rendering texture-mapped 3D perspective views (see Section 5).

The texture mapping algorithm we use for creating the OFIL is *inverse mapping*, which involves the computation of transformations between three coordinate systems: the *orthographic image coordinate system* for the facet, the *world coordinate system* for the 3D site model, and the *camera coordinate system* for the site image. Techniques for inverse mapping and representations of the coordinate systems involved can be found elsewhere [10]. A 3D site model and the camera parameters for each site image are needed by the texture mapping algorithm. The mapping is computed pixel by pixel in scanline order on the orthographic facet image. Each pixel square in the facet image is orthographically projected onto an object face in the world coordinate system to determine a corresponding square on that surface. The new square in the world coordinate system is then perspectively projected into a site image as a quadrilateral that totally or partially covers a number of pixels (Figure 3). The intensity of the pixel in the facet image is computed as the average of the intensities of the covered pixels in the site image, each weighted by area of coverage.

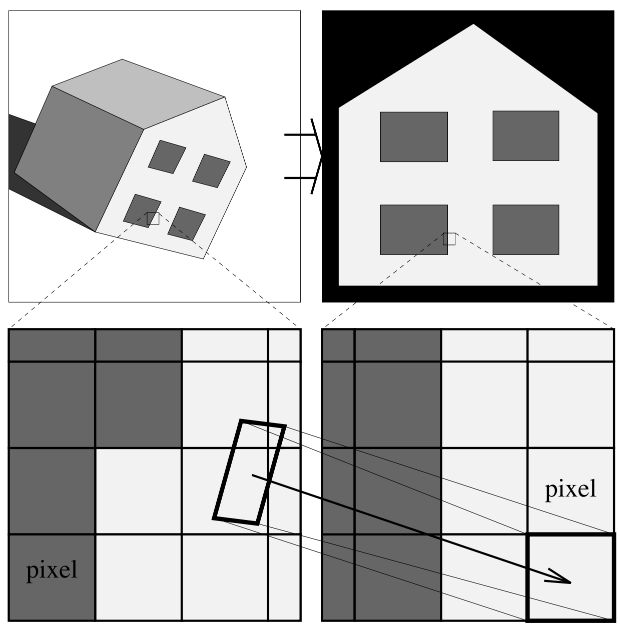

Figure 3: Orthographic texture mapping of object surface (left: site image, right: orthographic facet image)

3 Occlusion and Shadow Modeling

One important feature of the OFIL architecture is that it provides a convenient way to handle *occlusions* and *shadows* that arise on the object surfaces in a scene. Occlusions in a site image occur when objects stand between the camera and the object surface of interest. During texture mapping, intensity values from the occluding objects get mapped onto the orthographic facet image as well. Shadow areas occur on the surface due to objects standing between the light source and the surface of interest. Generally speaking, occlusions do not provide useful intensity information for surface texture analysis, while shadow areas may still be useful, provided that enough dynamic range exists in the shadow area to reconstruct the texture of the surface. Typically, unocclude, sunlit parts of the surface are the best sources of intensity information.

To avoid the negative effects of occlusions and shadows on subsequent facet image analysis, an extra record is associated with each facet image, explicitly indicating which pixels in the image are occluded and which are in shadow. In the OFIL, each orthographic facet image version is associated with an *orthographic labeling image* of the same size, in which each pixel is composed of a number of "attribute" bits that record whether the corresponding pixel on the facet image is occluded or in shadow. Figure 4 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 [10]. Basically speaking, viewing pyramids (Figure 5) are constructed from the camera viewpoint to the object facets. Suppose Facet F in the site model is the current facet of interest. To determine occlusions, every other facet in the model, say Facet i, is taken as a test facet for checking

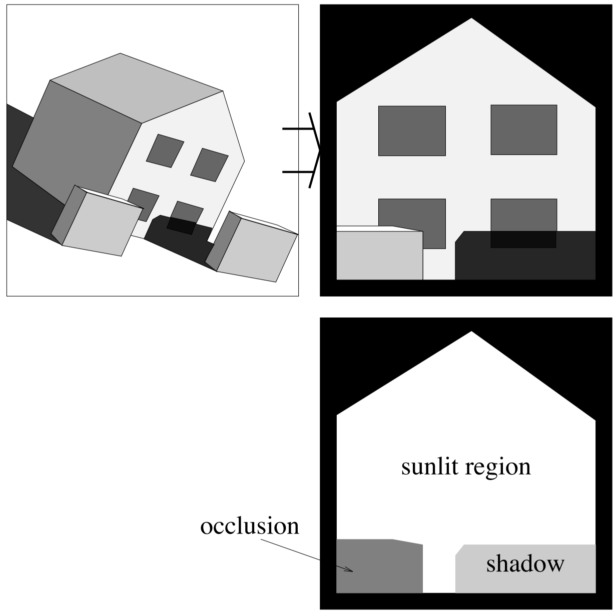

Figure 4: Occlusion and shadow labeling (left: site image, upper right: facet image, lower right: facet labeling image)

the positional relationship of Facet F, Facet i and the camera. A pyramid is constructed by connecting the camera viewpoint with each vertex of Facet i. The infinite plane containing Facet F intersects with the pyramid to produce a new polygon on F's plane. We call the new polygon F_i. It is easy to calculate which polygon is closer to the viewpoint, Facet i or polygon F_i. If Facet i is closer, the intersection of F and F_i is an area occluded by Facet i and is subject to marking with the occlusion attribute in the labeling image. After all the facets in the site model are tested, we have a complete set of occlusion areas for Facet F in the labeling image. The algorithm for shadows works in the same way, except that the pyramids are extended from the light source rather than from the camera viewpoint, and since the sun is treated as parallel light source, the pyramid becomes a prism.

It is worth noting that occlusion and shadow computation is valid only when the following assumptions hold: 1) the geometric site model, camera and light source parameters are known to sufficient accuracy, 2) the perspective camera model is valid at the

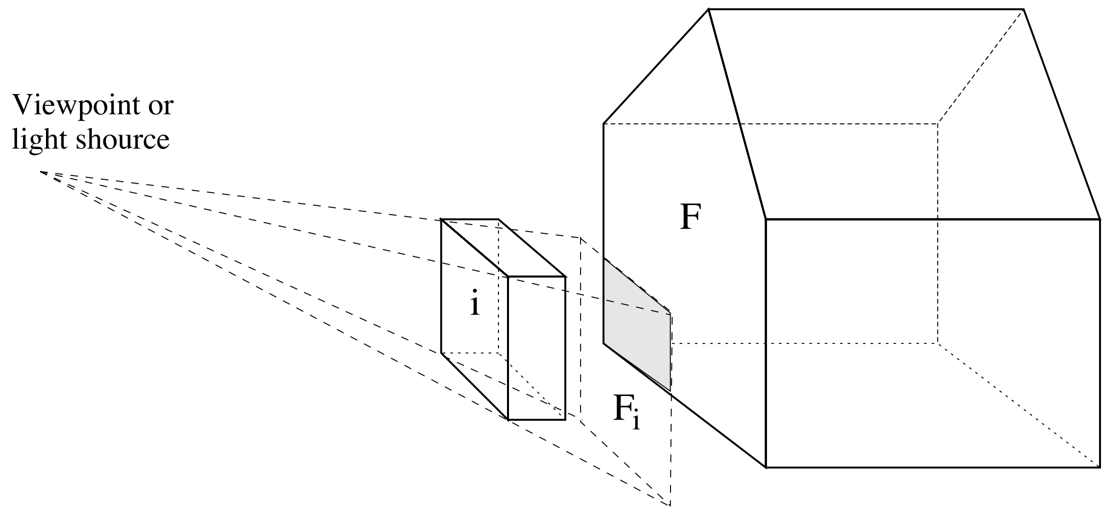

Figure 5: The pyramid for occlusion and shadow computation

scale of objects such as buildings, and 3) all objects in the site are present in the site model. Our building detection experiments on the Model Board 1 dataset give encouraging results regarding assumption 3. A data-driven processing step is under development (see Section 6) to solve problems caused by geometric inaccuracy regarding assumptions 1 and 2.

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

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

To provide a glimpse of the OFIL, Figure 6(a) shows a set of orthographic facet images, with labelings, for a particular building facet in Model Board 1. This rectangular facet is the right wall of the largest building shown on 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 obliquity of the viewing angle. From this example we can see that the OFIL has collected and organized the available information about this wall facet, and that multiple images are necessary to see all the portions of this particular building face.

4 Unique Intensity Representation

The OFIL collects all the intensity information about each building facet in the form of separate, but aligned, orthographic image versions. For many tasks it is desirable to produce a single, unique intensity representation of the 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 6, the version that contains the least occlusions and shadows is the one from site image J2, but that version is too blurred to be a good representation due to the

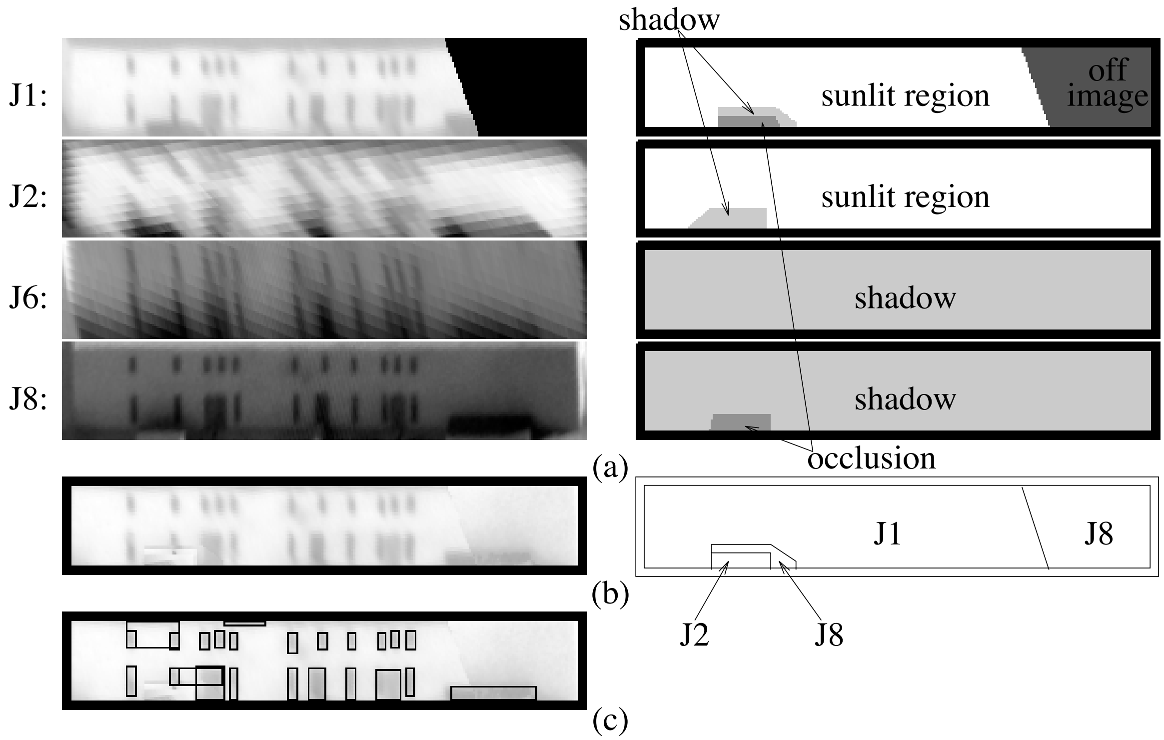

Figure 6: The application of OFIL architecture to a building facet

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

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

(c) the result of symbolic rectangle extraction toward model refinement

obliqueness of the viewing angle.

In this section we present 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 in scanline order, 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 piece p on facet image version v, we evaluate the piece using the value given by the function

$$f(p) = \alpha(p)A(v),$$

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

was obtained, and

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

The area factor $A(v)$ reflects the combined effects of surface-camera distance and viewing obliqueness. The weighting factor $\alpha(p)$ is set according to the attribute bits of the piece. 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 in the BPR image comes from the associated piece with the highest value of $f(p)$.

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 designed for selecting the exemplar piece. For any piece p on the facet image version v,

$$g(p) = \alpha(p)A(v)S(p) = f(p)S(p),$$

where $\alpha(p)$, $A(v)$ and $f(p)$ are as described above, and $S(p)$ is the area percentage of piece p to the whole facet. The meaning of $A(v)S(p)$ is the area of piece p 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.

Figure 6(b) shows the BPR image synthesized using the facet images in Figure 6(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 as well.

5 Model Refinement and Visualization

A complete orthographic facet image library has been built up in the RADIUS texture mapping system at The University of Massachusetts. For the 25 modeled buildings, 133 facet images are created automatically using the BPR algorithm as surface texture maps. Among them, there are 108 rectangular walls, 21 rectangular roofs (corresponding to the 21 rectangular prism-shaped buildings), and 4 L-shaped roofs (corresponding to the 4 L-shaped buildings). The site images, from which the BPR collects intensities, are J1-J8 from Model Board 1 imagery. In this section we show sample applications of the OFIL architecture in model refinement and scene rendering.

After the initial model acquisition stage of the RADIUS project, the site model only contains 3D geometric descriptions of building shapes and locations [5, 4]. The task of model refinement is to fill in detailed surface descriptions of the site. As one example, we have developed a generic algorithm for detecting dark, oriented rectangular patterns, as a precursor for extracting windows and doors on wall surfaces. Figure 6(c) shows the results of applying this algorithm to the BPR image in Figure 6(b). The algorithm first scans the image looking for local "intensity dips" or blobs, and marks their centers as *seeds*. A rectangular region growing method is then used, starting from each seed, to grow the region outwards in all four directions. The growing is stopped by an intensity derivative criterion designed to localize the rectangle's borders. Twenty dark regions in Figure 6(c) are correctly extracted as hypothesized windows and doors and written into a new, refined model. There are also 4 false detections, which could potentially be removed automatically using knowledge-based constraints regarding the size and shape (range of aspect ratios) of true windows and doors. An interesting result in Figure 6(c) is that the detected rectangle on the right bottom of the image crosses over two regions whose intensities come from different image versions, suggesting that our BPR algorithm maintains good consistency of image intensities from different sources. It is worth noting that the OFIL architecture plays an important role in simplifying the rectangle detection algorithm, as it ensures that the windows and doors are rectangular and orthogonally aligned.

The purpose of model visualization is to provide a visually realistic rendering of the scene. Given a desired camera pose, we want to render a 3D view of the site with textures mapped onto the modeled facets of the objects. The *Z-buffer* algorithm is the most commonly used method for determining surface visibility. This algorithm, as well as other techniques needed in 3D view rendering, can be found in a graphics text [10]. Similar work is shown by Forsyth and Rothwell [6]. They backprojected intensity information from perspective images of extruded objects into a canonical reference frame for use in rendering and object recognition. However, they do not consider issues such as removal of shadows and occlusions from the acquired intensity maps, and do not consider the prospect of fusing information from multiple intensity maps to synthesize a more complete surface representation. In our visualization system, the textures of the object surfaces are mapped from the BPR facet representatives constructed as described earlier. Consequently, the rendered scenes are free of defects, like shadows and occlusions, caused in the original perspective images. A sample 3D site rendering is shown in Figure 7. The big building on top of site image J1 in Figure 1 is at the right side of the new rendered image, as seen from this angle, and clearly shows the BPR synthesized wall from Figure 6(b). Sequences of rendered 3D views are constructed in this way for site visualization and familiarization. They are used for generating animated fly-throughs, when played back at video speeds.

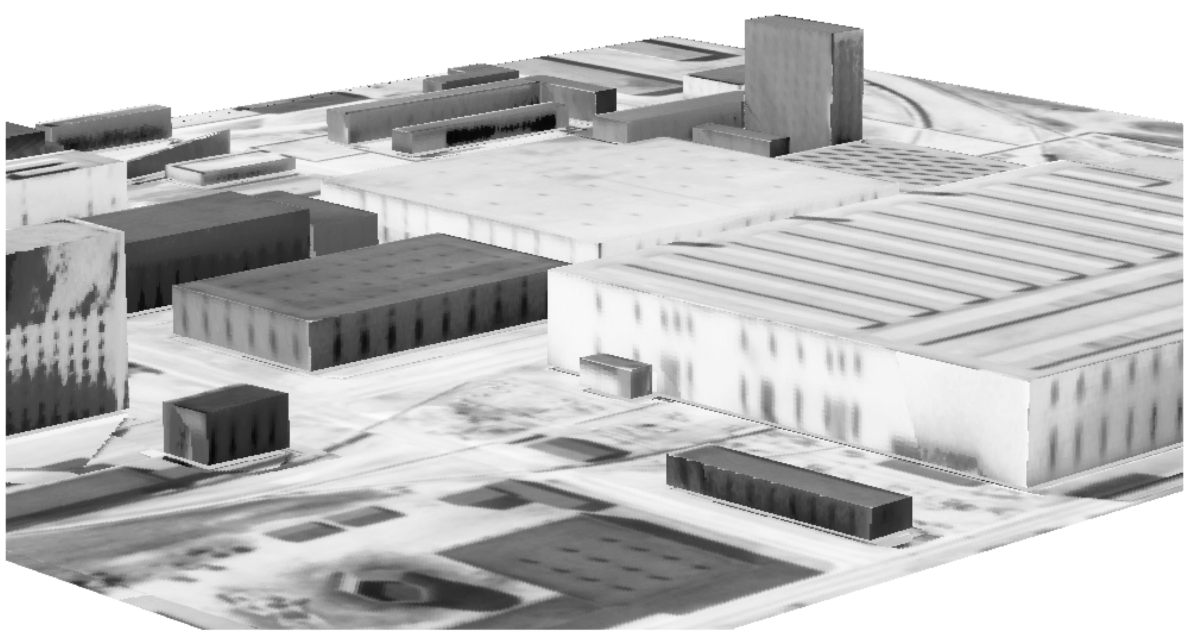

Figure 7: Model visualization using BPR texture maps

6 Discussion and Future Work

Efficient intermediate representations play an important role in complex image understanding. In this paper we have proposed an orthographic facet image library architecture to aid image understanding of buildings from aerial site images. With the assumption that camera parameters, lighting conditions, and a geometric site model are known, facet intensity information is collected systematically from the original set of perspective images into an organized orthographic library. Occlusions and shadows on each facet are recorded in the library. A best piece representation algorithm is proposed to combine intensity information from multiple image versions to give a complete and consistent intensity representation of each facet, which substantially simplifies later work in model refinement and visualization. This paper has presented experimental results of the application of the proposed architecture to model refinement and scene rendering.

High-end graphics engines are now capable of generating texture-mapped, real-time, scene fly(walk)-throughs suitable for use in flight simulation, site visualization, and mission planning. These "virtual-reality" environments are currently built by hand, and are expensive and time consuming to produce. The techniques presented in this paper form one component of a system for automatically acquiring environmental models of great geometric and photometric realism. In addition, generation of a viewpoint-independent surface intensity map yields a rich surface description that can greatly simplify vision tasks such as object recognition and the analysis of surface markings.

The drawback of the proposed architecture is that it is almost purely top-down, or model-driven. Inaccuracy and incompleteness in the 3D model can cause problems. For example, undetected objects (such as smoke stacks) in the site will not only be missed in the library, hence in any reconstructed view, but their absence also leads to misidentification 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 mend 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 refinement as directed by the comparison between the model-driven predicted and data-driven extracted shadows. The boundaries of shadows in images are often blurred, while model-driven shadow computation always gives sharp edges. This phenomenon further corrupts the quality of intensity fusion by the BPR algorithm. An enhanced BPR method using antialiasing techniques is being developed to avoid this problem.

The BPR algorithm selects pixel intensity values from the most competitive pieces in the OFIL, but it ignores completely the less competitive versions of a facet. Another possible approach is to generate a superresolution representation of the facet images, which combines intensities from all facet image versions and is expected to utilize as much information as possible from the OFIL. Some of the requirements of a superresolution method are: a more accurate registration of the facet image versions than required by BPR, a more strict comparability of the intensities from different versions than required by BPR, and a mechanism at the sub-pixel level to fuse the intensities from different versions into one image.

Acknowledgements

We would like to thank Professor Edward Riseman for his discussions and comments, and Jonathan Lim and Fred Weiss for their technical support.

References

[1] D. Ballard and C. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[2] Y. Cheng, R. Collins, A. Hanson and E. Riseman, "Triangulation without Correspondences," *Proc. Arpa Image Understanding Workshop*, Monterey, CA, 1994, pp. 993-1000.

[3] D. Climenson and T. Strat "RADIUS: Site Model Content," *Proc. Arpa Image Understanding Workshop*, Monterey, CA, 1994, pp. 277-286.

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

[5] R. Collins, A. Hanson and E. Riseman, "Site Model Acquisition under the UMass RADIUS Project," *Proc. Arpa Image Understanding Workshop*, Monterey, CA, 1994, pp. 351-358.

[6] 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.

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

[8] C. Jaynes, F. Stolle and R. Collins, "Task Driven Perceptual Organization for Extraction of Rooftop," *Proc. Arpa Image Understanding Workshop*, Monterey, CA, 1994, pp. 359-365.

[9] E. Riseman, A. Hanson, R. Collins, B. Draper and R. Weiss, "Progress in Computer Vision at the University of Massachusetts," *Proc. Arpa Image Understanding Workshop*, Monterey, CA, 1994, pp. 43-51.

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