Detecting, localizing and grouping repeated scene elements from an image

Thomas Leung and Jitendra Malik

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley, Berkeley, CA 94720
email: leungt@cs.berkeley.edu, malik@cs.berkeley.edu

Abstract. This paper presents an algorithm for detecting, localizing and grouping instances of repeated scene elements. The grouping is represented by a graph where nodes correspond to individual elements and arcs join spatially neighboring elements. Associated with each arc is an affine map that best transforms the image patch at one location to the other. The approach we propose consists of 4 steps: (1) detecting "interesting" elements in the image; (2) matching elements with their neighbors and estimating the affine transform between them; (3) growing the element to form a more distinctive unit; and (4) grouping the elements. The idea is analogous to tracking in dynamic imagery. In our context, we "track" an element to spatially neighboring locations in one image, while in temporal tracking, one would perform the search in neighboring image frames.

1 Introduction

This paper presents an algorithm for detecting, localizing and grouping instances of repeated scene elements. The notion of scene elements is intended to be quite general and includes as special cases:

- texels on a surface as shown in Figure 1(a). Other examples might be spots on a leopard's skin or the pattern of brickwork on a pavement.
- discrete structures such as the eyes on a person's face.

The goal of our approach is to compute a representation where the individual elements are detected, localized and grouped. The grouping is represented by a graph where nodes correspond to individual elements and arcs join spatially neighboring elements. With each arc r_{ij} is associated an affine map A_{ij} that best transforms the image patch $I(\boldsymbol{x}_i)$ to $I(\boldsymbol{x}_j)$. This affine transform implicitly defines a correspondence between points on the image patches at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

What is this representation good for? The two short term objectives are:

1. Grouping of these repeated elements is a useful goal in itself. Certainly this provides an operationalization of one of the Gestalt laws of grouping, that based on similarity.
2. Knowledge of the affine transforms between corresponding patches, or equivalently knowledge of the point correspondences between fiducial points on the image patches, can be used to recover 3D scene structure. The mathematics for shape recovery can either follow analysis of shape from texture [4, 10, 11] or structure from motion, or variations of the ideas therein.

In the long term, our primary driving application is recognition. For that the vital importance of grouping has long been noted, and recovery of 3D structural relationships can be useful as well. A more novel aspect is that given a knowledge of the 3D scene structure, we can also hope to recover what one of the scene elements looks like from a canonical view (say fronto-parallel). The appearance from such a canonical view is distinctive and invariant (up to a scale factor) to camera pose and could be used in an iconic matching scheme to provide another visual cue for recognition.

2 Relevant previous work

The most relevant previous work is that on texture processing and on reconstruction from repeated structure.

2.1 Texture processing

Texture has often been defined to be some repeating pattern, so it is natural to think of the problem of finding repeated elements as being a problem of finding regions with coherent spatial texture. This is simplistic–we will discuss below the similarities and differences between texture processing and our work.

First, we talk about the similarities. As pointed out by Malik and Rosenholtz [10, 11], shape from texture is a cue to 3D shape very similar to binocular stereopsis and structure from motion. All of these cues are based on the information available in multiple perspective views of the same surface in the scene. In the context of shape from texture, two nearby patches appear slightly differently because of the slightly different geometrical relationships that they have with respect to the observer's eye or camera. We thus get multiple views in a *single* image. This naturally leads to a two-stage framework (1) computing the "affine texture distortion" from the image, and (2) interpreting the "texture distortion" to infer the 3D orientation and shape of the scene surface. When we deal with repeated scene elements, clearly the idea of multiple views of the "same" element being available in one image carries over.

The differences are enumerated as follows:

1. The notion of texture includes very irregular arrangements where one may not meaningfully speak of a basic unit, a texel, that is repeated across the image. Some of the techniques in texture analysis, such as looking at the response of multi-scale, multi-orientation filter outputs are of course applicable. However we exclude such textures from our consideration of arrangments of repeated scene elements.
2. We want to deal with the cases where there are only a small number of repeated elements (e.g. two eyes). In this case, the notion of texture is usually not appropriate and coherence of filter outputs is too weak a grouping cue.
3. In the case of texture analysis, one usually ignores the precise positional relationships. A crude summary of the psychophysics results of Julesz, Beck, Treisman and others is that in preattentive vision, the positional relationships between the textons do not matter, only the densities need to be considered. This insight has carried over to the quasilinear filtering models that now dominate the field—spatial averages of filter outputs are considered [9].

In our context of finding repeated scene elements, *the precise positional relationships between the features on a single element do matter: we can talk about precise point to point correspondences between features on one element and another.*

In summary, we have a problem formulation that excludes a certain class of textures—stochastic textures with ill defined texture elements—but considering only repeated elements enables a notion of point to point correspondence that is now *exactly* like that in motion or stereopsis.

2.2 Reconstruction from repeated structure

It has been pointed out [1, 6, 7, 12] that structures that repeat in a single image of a scene are equivalent to multiple views of a single instance of a structure. If there are many identical objects available, then one can even calibrate the camera and recover structure up to a similarity ambiguity. Though arrived at independently, fundamentally this represents a variation of the same idea as expressed by Malik and Rosenholtz in the texture context [10, 11]. But, here, the notion of point-to-point correspondence is exactly the same as in the motion/stereopsis context.

However, in previous work on this topic, the early vision operations of finding image correspondences have not been addressed. The work in this paper makes this process feasible.

3 The Algorithm

An intuitive understanding of the algorithm can be obtained rather simply by an analogy to tracking in dynamic imagery. Suppose, one has identified one of the elements somehow. To identify others like it, we "track" it by searching for it in neighboring locations in the image plane, just as one would in the temporal tracking context where we search in neighboring image frames. A temporally tracked object would be expected to change in size and shape because of its differing pose with respect to the camera. That change, if small enough, can be conveniently modeled as an affine transform. For finding the best match,variations on the theme of maximizing cross-correlation or minimizing SSD (sum of squared differences) have been demonstrated [1] to be simple and robust. Once the object has been located in the next time frame, one can then use a new template based on its new appearance to search for it in the subsequent frame and so on. Reliable, robust temporal tracking has now become a core technology in dynamic computer vision.

The success of temporal tracking suggested that a multi-directional spatial tracking approach to finding repeated scene elements should work well. The difficult part is that of finding a good initialization—it is not easy as in the temporal tracking context. Our approach was to look for "interesting" windows–ones with enough image intensity variation. The algorithm consists of the following steps:

1. Detection of interesting windows in the image. These windows constitute the possible candidates for the repeating elements. The criterion is that they are distinct enough so that matching can be done easily and unambiguously.

[1] Weaknesses of correlation/SSD approaches for very large changes in foreshortening as in wide-baseline stereo are not relevant in this setting.

2. For each window, we look at the neighboring regions to search for similar structures. This search allows for small affine deformations.
3. The size of the unit is allowed to change to maximize the distinctiveness of the element.
4. The whole repeating structure is grouped together with the individual elements marked.

3.1 Detection of distinctive elements

The first step of our algorithm is to look for distinctive elements in the image. We break up the whole image into overlapping windows. At each window, we compute the second moment matrix:

$$\Sigma_{\mathcal{W}} = \sum_{\mathcal{W}} \nabla I(\boldsymbol{x}) \nabla I^T(\boldsymbol{x}) \tag{1}$$

where ∇ is the gradient operator.

The second moment matrix tells us how much spatial intensity variation is present in the window. It has been used before in early vision processing as in [3, 5]. The eigenvalues, k_1 and k_2 (assume $k_1 > k_2$), of $\Sigma_{\mathcal{W}}$ represent the amount of "energy" in the two principal directions. For 2D structures, k_1 and k_2 will be large and comparable in magnitude; k_1 will be large and k_2 will be close to zero for 1D structures; both k_1 and k_2 will be small if little variation is present.

Using the two eigenvalues, we can give further characterization for each window. For a 2-D pattern, k_1 and k_2 will be similar:

$$\frac{k_1}{k_2} < r_1 \tag{2}$$

Here, r_1 is a constant. It describes the ratio of the energy in the two directions of the spatial structure. This tells apart a 2D from a 1D pattern.

In general, there are two types of 1D structures. They can be an edge or a pattern in one direction (a flow), e.g. the vertical stripes on a shirt. However, they can be distinguished by looking at the sign of the gradient vectors. Consider an edge versus a bar. The gradient vectors in the neighborhood of an edge will all point in the same direction, while the gradient vectors point in opposite directions on the two sides of a bar. Making use of this observation, we can tell a flow from an edge. Let $\boldsymbol{v}$ denote the dominant orientation in the window. This is the direction of the eigenvector corresponding to the large eigenvalue of $\Sigma_{\mathcal{W}}$. Let $\boldsymbol{x}_1$ be the pixel locations where the gradient makes an acute angle with the dominant direction, i.e. $\boldsymbol{v}^T \nabla I(\boldsymbol{x}_1) > 0$. Similarly, let $\boldsymbol{x}_2$ be the locations where the gradient makes an obtuse angle with $\boldsymbol{v}$, i.e. $\boldsymbol{v}^T \nabla I(\boldsymbol{x}_2) < 0$. Define

$$\Sigma_1 = \sum_{x_1} \nabla I \nabla I^T \quad \text{and} \quad \Sigma_2 = \sum_{x_2} \nabla I \nabla I^T \tag{3}$$

For a flow, Σ_1 and Σ_2 will have similar energy, while for an edge, most of the energy will be concentrated in one of them. Therefore, an 1-D pattern has to satisfy the following requirements:

$$\frac{k_1}{k_2} > r_1 \quad \text{and} \quad \frac{\max(\sum \sigma(\Sigma_1), \sum \sigma(\Sigma_2))}{\min(\sum \sigma(\Sigma_1), \sum \sigma(\Sigma_2))} < r_2 \tag{4}$$

Here, $\sigma(A)$ refers to the spectrum of the matrix A. r_1 is the same as in Equation 2. It tells us that we are dealing with a 1D pattern. r_2 is a constant describing the ratio of energy in Σ_1 and Σ_2. It distinguishes a 1D flow pattern from an edge.

In all our experiments, r_1 and r_2 are fixed. We set $r_1 = 4$, meaning that a pattern is defined to be 2D if the energy in the dominant direction is no more than 4 times larger than the energy in the orthogonal direction. The value of r_2 is fixed to be 2, meaning that when the ratio of the energy in Σ_1 and Σ_2 is no more than 2, the pattern is declared to be 1D flow, as opposed to an edge.

3.2 Finding Matches

After identifying possible candidates, we examine each of them to determine if they belong to a repeating structure. This is done by searching for similar patterns in the neighborhood. A small amount of affine transform is allowed to take into account surface shapes [2]. A rough registration is obtained by finding the neighboring patches where the SSDs to our candidate are small. We then use a differential approach [3, 8] to estimate the affine transform which will bring the two patches in better correspondence:

$$Err = \sum (I(\boldsymbol{x}) - \tilde{I}(A\boldsymbol{x} + d))^2 \tag{5}$$

$$\{A, d\} = \arg \min_{\{A,d\}} Err \tag{6}$$

$\tilde{I}$ is the intensity at the neighboring location. Linearizing,

$$A \approx I + \Delta A \quad \text{and} \quad d \approx 0 + \delta d \tag{7}$$

$$Err \approx \sum (I(\boldsymbol{x}) - \tilde{I}(\boldsymbol{x}) - \nabla \tilde{I}(\boldsymbol{x})^T \Delta A \boldsymbol{x} + \nabla \tilde{I}(\boldsymbol{x})^T \delta d)^2 \tag{8}$$

ΔA and δd can be estimated by solving a system of linear equations. See [8] for details. We then iterate by warping the image and refining the affine transform.

Similarity of the two patches are measured in a normalized way. Define

$$D(\boldsymbol{x}) = I(\boldsymbol{x}) - \tilde{I}(A\boldsymbol{x} + d) \tag{9}$$

$$\sigma_I = \sqrt{\sum (I(\boldsymbol{x}) - \overline{I}(\boldsymbol{x}))^2} \tag{10}$$

$$\sigma_e = \sqrt{\sum (D(\boldsymbol{x}) - \overline{D}(\boldsymbol{x}))^2} \tag{11}$$

Two patches are declared to be similar if

$$E_{sim} = \frac{\sigma_e}{\sigma_I} < \tau_e \tag{12}$$

[2] The repeated elements are related by a projective transformation, but an affine transform is a very good approximation for neighboring elements.

σ_I can be interpreted as the contrast of the image [3] and σ_e the variation of the error. $\frac{\sigma_e}{\sigma_I}$ can then be thought of as the inverse of the signal-to-noise ratio. τ_e is a constant threshold taken to be 0.2 in all our experiments. Notice that this error measure is invariant to both constant scaling of image brightness and an illumination offset.

The output of this procedure is a list of elements which form units for repeating structures in the image. Associated with each element are the neighboring patches which match well with the element, together with the affine transform relating them.

3.3 Growing the Pattern

The initial size of the window is chosen quite arbitrarily [4], and thus each element may not correspond to the most distinctive unit. Therefore, we allow the elements to change in size. The criterion for change is that it will decrease the matching error (Eq. 12). This is measured against the neighboring patches that we obtained in the previous step. With the affine transform, we can warp each individual neighboring patch to one which registers with the element. We iteratively increase the height and width of the window and measure the total error. If the total error decreases, we enlarge the window and continue the process until the error stops decreasing or a maximum allowed size is reached. The outcome would be a set of elements which are more representative to the underlying repeating structure.

3.4 Grouping Elements

The final step is to group the elements together. We start at each basic unit, and look inside its 8 neighboring windows for similar elements. In each of the 8 windows, we find the patch which matches the element best. The affine transform between them is also computed. If the error between the neighboring patch and the element is smaller than τ_e (Eq. 12), we group the two patches together.

When we propagate the growing procedure outward, the size and shape of the basic element in the image will change. This is very well illustrated in Figure 1. Owing to the slanting of the surface, the glass windows look smaller when we go towards the far end of the building. The affine transform tells us exactly how the size and shape change. So, if we initially have a rectangular window, the neighboring elements are best described by a parallelogram in the image plane. We approximate the parallelogram by a rectangle, defined by the mid-points of the 4 edges of the parallelogram. This notion of changing the window size is very important because we are grouping *scene* elements, not image elements. Because of the surface shape, the scene elements will "look" different in the image.

4 Results

The results of applying the algorithm on a number of images are presented in this section. In all our experiments, we set $r_1 = 4$ (Eq. 2), $r_2 = 2$ (Eq. 4) and $\tau_e = 0.2$ (Eq. 12).

[3] In psychophysics, contrast is usually defined as $\sigma_I/\overline{I}$. This has no effect on our similarity measure, because σ_e will be scaled by the same factor and the ratio will not change.

[4] A multi-scale analysis may help in choosing the optimal window size.

Figure 1 shows an image of a building. Architectural objects usually have a very large number of repeating elements, like windows or doors. The rectangle marked in (a) indicates where the initial element is. This unit is found automatically by the "interesting-window" search described in Section 3.1 and the size automatically adjusted to give a more distinctive element. The crosses are the locations of the elements obtained by the grouping process. Notice that the spacing of the crosses are smaller as the units get thinner when we propagate to the far end of the building. This is where the importance of changing the window size and shape by the affine transform shows up. In (b), the grouped region is displayed. This provides us with a segmentation of the surface patch.

The algorithm also works well on curved surfaces, where affine distortion between neighboring elements is significant. Figures 2 and 3 are some textile images. In Figure 2, the repeating unit chosen by the algorithm was 2 units of the actual pattern; in Figure 3, it was one unit. Note that there is an inherent ambiguity here because the 2-unit element chosen in Figure 2 is also a repeating pattern. In Figure 3, note that the front of the shirt is isolated from the arms. This is to be expected because the grouping process assumes small changes from one element to its neighbor. This offers us a useful segmentation technique where the shirt front, left arm and right arm would emerge as separate groups. Work on human figure recognition [2] have shown the usefulness of such segmentation into parts.

Figure 4 shows another type of grouping. Here, we have a simple repeating pattern. Traditionally, this would not be regarded as texture as it does not have significant 2D spatial extent. Instead, it is a distinctive element in the scene. The grouping works well too and the repeating units are found.

Figure 5 is an example of a yet different kind of image. Here the pattern is that of an oriented flow field. Note that repeating units are found though obviously the breakup into individual elements is somewhat artificial. However, the knowledge that these have predominantly 1D structure could be used to perform a more suitable grouping operation in a post-processing stage.

5 Conclusion

We have demonstrated a technique to detect, localize and group instances of repeated scene elements in an image. The result is a graph where nodes correspond to individual elements and arcs join spatially neighboring elements. Associated with each arc is an affine map that best transforms the image patch at one location to the other. Results on a variety of images are shown. They include 2D structures on planar and curved surfaces and 1D flows on curved surfaces.

A number of extensions of this work follow immediately. A multi-scale analysis would be useful to group elements of different scales. At present, the initial partition into windows is arbitrary. In principle, this should be chosen based on the scale of the features we are looking for.

Another direction of work that will be done is to use the affine transforms and the point correspondences in the patches to infer surface shape. Techniques from shape from texture [4, 10, 11] or structure from motion will be useful.

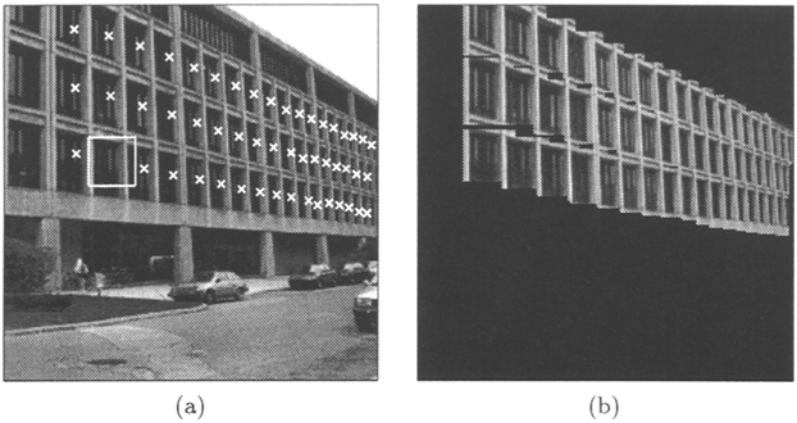

Fig. 1. Architectural image. The rectangle in (a) is the unit we found and where the growing process starts. The crosses are the locations of the units grouped together. The segmented region is shown in (b). Notice that the spacings between the crosses become smaller as we propagate to the far end of the building. This is capturing well the change in size and shape of the units in the scene.

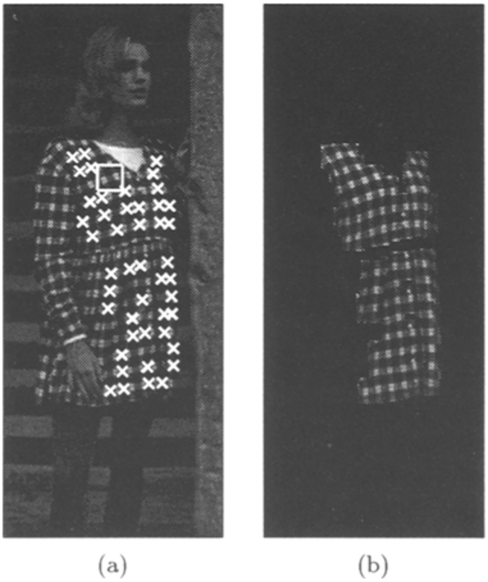

Fig. 2. A textile image on a curved surface. The rectangle in (a) is the initial unit we found. The crosses are the locations of the units grouped together. In (b), the segmented region is shown. Notice that the rectangle includes two units in the actual pattern. This is due to the inherent ambiguity in defining a repeating unit, namely, 2 units together is still a repeating element.

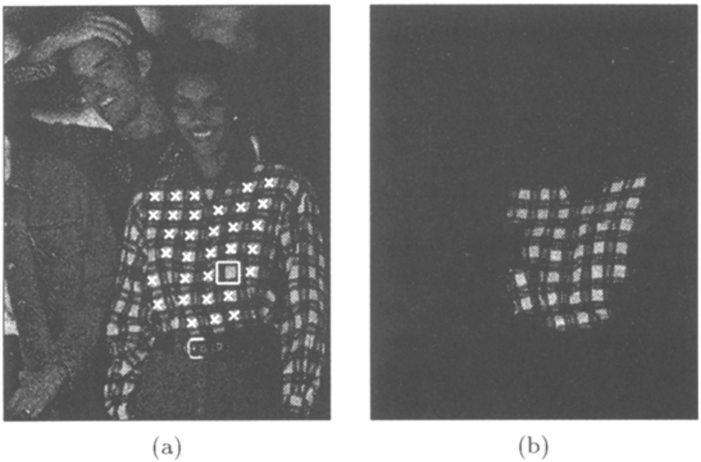

Fig. 3. Another example of a textile image on a curved surface.

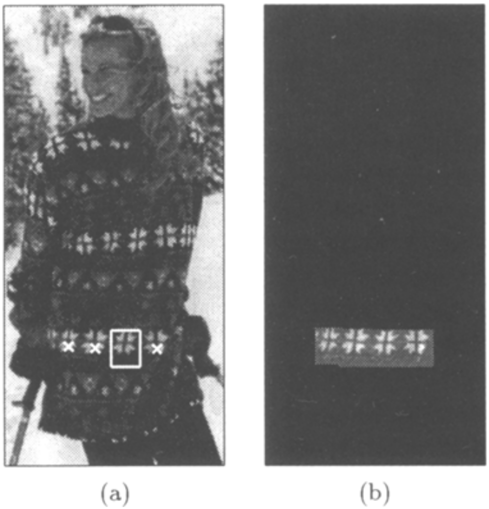

Fig. 4. A different type of grouping is shown in this figure. The rectangule in (a) shows a distinctive element in the image. Traditionally, this would not be considered as texture because there is no significant 2D spatial extent. Our algorithm works well in this situation too. The crosses in (a) mark the instances of the unit.

As we have pointed out, the ultimate goal of our work is object recognition. This algorithm can provide grouping and 3D structural information about the scene. Given the 3D structural relationships, we can recover what one of the scene elements looks like from a canonical view (say fronto-parallel). The appearance from such a canonical view is distinctive and invariant (up to a scale factor) to camera pose and could be used in an iconic matching scheme to provide another visual cue for recognition.

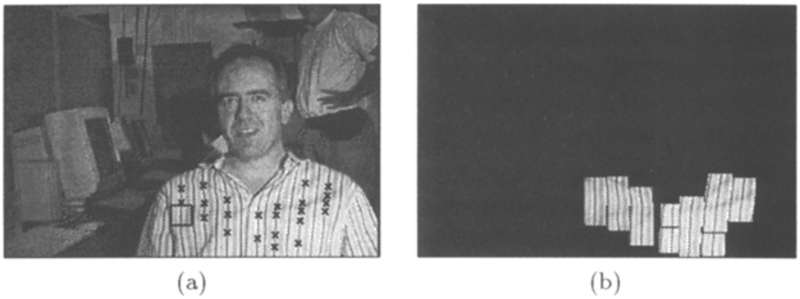

Fig. 5. An oriented 1D flow field. Notice that repeating units are found though obviously the breakup into individual elements is somewhat artificial. However, the knowledge that these have predominantly 1D structure could be used to perform a more suitable grouping operation in a post-processing stage.

Acknowledgement

This work was supported by the Digital Library Grant IRI-9411334 and a Berkeley Fellowship.

References

1. M. Armstrong, A. Zisserman, and P. Beardsley. "Euclidean Structure from Uncalibrated Images". In *Proc. British Machine Vision Conference*, 1994.
2. M. Fleck, D. Forsyth, and C. Bregler. "Finding Naked People". *To appear in ECCV*, 1996.
3. W. Förstner. "Image Matching". In R. Haralic and L. Shapiro, *"Computer and Robot Vision"*, chapter 16. Addison-Wesley, 1993.
4. J. Gårding. "Surface orientation and curvature from differential texture distortion". In *Fifth Intl. Conf. Computer Vision*, pages 733–739, 1995.
5. J. Gårding and T. Lindeberg. "Direct Estimation of Local Surface Shape in a Fixating Binocular Vision System". In *Proc. Euro. Conf. Computer Vision*, 1994.
6. J. Liu, J. Mundy, and E. Walker. "Recognizing Arbitrary Objects from Multiple Projections". In *Proc. Asian Conf. Computer Vision*, 1993.
7. J. Liu, J. Mundy, and A. Zisserman. "Grouping and Structure Recovery for Images of Objects with Finite Rotational Symmetry". In *Proc. Asian Conf. Computer Vision*, 1995.
8. B.D. Lucas and T. Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision". In *Proc. 7th Intl. Joint Conf. on Art. Intell.*, 1981.
9. J. Malik and P. Perona. "Preattentive Texture Discrimination with Early Vision Mechanisms". *J. Opt. Soc. Am. A*, 7(5):923–932, 1990.
10. J. Malik and R. Rosenholtz. "Recovering Surface Curvature and Orientation from Texture Distortion: a Least Squares Algorithm and Sensitivity Analysis". *Proc. Third Euro. Conf. Computer Vision*, pages 353–364, Sweden, 1994.
11. J. Malik and R. Rosenholtz. "Computing Local Surface Orientation and Shape from Texture for Curved Surfaces". *To appear in the International Journal of Computer Vision*, 1996.
12. J.L. Mundy and A. Zisserman. "Repeated structures: image correspondence constraints and 3D structure recovery.". In J.L. Mundy, A. Zisserman, and D.A. Forsyth, editors, *Applications of invariance in computer vision.* 1994.