# OCCLUSION-SENSITIVE MATCHING

*William B. Thompson* *Rand P. Whillock*

Computer Science Department
University of Minnesota
Minneapolis, MN 55455

## Abstract

Model-based recognition of partially occluded objects is a difficult task because of the need to accept matches in which only a subset of model features correspond to image features. Most approaches to implementing these partial matches are subject to serious problems due to ambiguity. Improvements in performance are possible by directly exploiting evidence for occlusion in the image. Once a potential match has been hypothesized, occlusion cues can be used to predict portions of an object model that are not likely to be visible in the image. We describe both an algorithm for matching using occlusion cues, and a method for determining the presence of occlusion based only on image properties. Occluding surfaces are recognized with an approach that combines motion and contrast information. The method accurately localizes edges, detects only those edges likely to correspond to surface boundaries, and provides an indication of which side of an edge corresponds to the occluding surface.

## 1 Introduction.

Many computational models for object recognition depend in some way on matching two-dimensional object models to image features. 2-D matching is not limited to template matching algorithms. Recently, many recognition approaches have been developed which use three-dimensional part/object models and sophisticated 3-D matching strategies. Because of the highly ambiguous nature of the problem, the final stage in such methods is typically a verification step in which hypothesized information about identification, position, and orientation is used to project a model back into the image to be matched against the actual image features.

Two significant problems plague matching operations. First of all, image features (lines, corners, holes, etc.) cannot be determined in a highly reliable manner. Model features are often missing in the image. Many patterns detected as image features either do not correspond to actual object properties or are not contained within the models. Secondly, in complex scenes objects are often partially occluded. Dealing with occlusion by accepting partial matches increases computational complexity while reducing the reliability of the matching process.

This work was supported by AFOSR contract AFOSR-87-0168 and NSF Grant DCR-8500899

This paper outlines two methods for improving the reliability of matching in the presence of partial occlusion. First, we describe a technique in which visual motion can be combined with static edge cues to improve the effectiveness of the edge detection process. Our technique recognizes that static and dynamic edge cues provide different sorts of information about a boundary. Static cues such as contrast edges give good spatial localization, but are subject to highly ambiguous interpretations. Visual motion is a robust indicator of surface boundaries, but does not yield precise information on the location of the boundary. The approach given here accurately locates edges due to surface boundaries, without generating many "false" edges. Even more importantly, the method gives a direct indication of which side of an edge corresponds to the occluding surface generating the edge.

The second technique uses information about occlusion to aid in the matching process. Most existing matching algorithms that are tolerant of occlusion look for a partial correspondence between model and image features. If a partial match is found, unmatched model components are assumed to be hidden by an occlusion. This approach leads to difficulties because of the chances for partial matches occurring coincidentally. In our method, information about occlusion boundaries is used to explicitly identify model features that will not be visible in the image. Most of the remaining model features should be findable if the match is in fact correct. Occluded model features are determined based directly on image properties at boundaries, rather than just on the absence of an image feature at some expected location. The result is a significant decrease in ambiguity.

## 2 Background.

### 2.1 Combining motion and contrast information for edge detection.

Segmentation schemes which combine motion and contrast information date back to at least to the work of Jain, Martin, and Aggarwal [1]. This approach used a difference operator between two frames to find areas in the image that had changed due to motion. A static segmenter was then run within these areas to find the boundaries of the moving regions. Thompson used a region merger approach that grouped pixels into regions based on similarities in contrast and motion information [2]. Hayes and Jain developed an edge detector based on a product of the spatial gradient and a temporal operator [3]. The purpose was to limit sensitivity to areas signaled by both static and dynamic effects. More recently, Gamble and Poggio have developed a Markov Random Field model for recovering optical flow in a manner that integrated contrast boundaries with visual motion [4]. Their approach constrained discontinuities in flow to occur only at intensity edges.

Relatively little work has been done on differentiating between occluding and occluded surfaces without resort to fitting object or part models. Waltz used constraints associated with line drawing vertices to identify extremal contours and to determine which side of such a contour corresponded to an occluding surface [5]. Smitley and Bajcsy identified occluding surfaces in stereo imagery by comparing correlations between frames for images patches on either side of a boundary [6]. If the correlations differed substantially, the boundary was assumed to be due to occlusion and the region with the highest correlation between views was assumed to correspond to the occluding surface. Thompson, Mutch, and Berzins showed how edges in optical flow could be used to recognize occluding surfaces [7]. Their approach is discussed in more detail in section 3.

### 2.2 Matching.

Template matching was one of the first methods proposed for the visual recognition of objects. Template matching utilizes a correlation measure between one or more model patterns and images to be analyzed. Invariance to translation and/or rotation can be obtained by appropriate scanning of the template pattern over an image. While useful in some applications, template matching suffers from problems due to computational complexity and is unable to deal effectively with the matching of three-dimensional models to two-dimensional imagery.

Recognition of three-dimensional objects is often done by using configurations of image features to estimate how a three-dimensional object is being projected into the two-dimensional image. The object model is subjected to the appropriate projection, resulting in a prediction of the objects appearance in the image. A verification process is used to determine if the predicted configuration of object features actually appears in the image (e.g., [8,9,10]). Such methods avoid many of the problems associated with straightforward template matching.

Recognition of partially occluded objects has been a major challenge for many years. Most approaches attempt to find good partial matches between subsets of object models and image features (e.g., [11,12,13]). Allowing for partial matches increases the likelihood of false positive classification errors. In addition, the extraneous configurations of boundaries generated by overlapping objects causes additional confusion.

Some preliminary attempts have been made to directly incorporate occlusion information into the matching process. Fisher developed evidence for extraneous or missing image features based on boundary topology and other information about the depth ordering of surfaces [14]. Specialized heuristics were used to discount the irrelevant mismatches during a verification stage. Castan used the results of a partial matching process to determine estimates of model features likely to be hidden by occlusions [15]. Evidence for visibility and occlusion came from a presumption that visible features were spatially adjacent, rather than from any three-dimensional analysis of the imagery.

## 3 Motion-based Segmentation.

Thompson, Mutch, and Berzins develop an edge detector for optical flow fields [7]. One important aspect of this work is that motion-based edge detection directly yields information about which side of the edge corresponds to the occluding surface. This identification is based on a comparison between the optical flow on either side of the boundary and the visual motion of the boundary itself. (Aperture effects usually require that all image flows be projected onto an axis parallel to the normal to the edge.) The principle underlying the identification of occluded surfaces is summarized in the *boundary flow constraint*:

> *At a surface boundary, the visual motion of the boundary itself is the same as the visual motion of the surface generating the boundary.*

At a boundary, we need only look at the image-plane motion of the boundary (the *boundary flow*) and the optical flow immediately to either side. Optical flow inconsistent with the boundary flow corresponds to an occlud*ed* surface.

One problem with exploiting the boundary flow constraint is the apparent need to determine the actual motion of the boundary. In many circumstances, this can result in a difficult correspondence problem. [7] demonstrated how the motion of optical flow edges can be related to the boundary flow constraint in a manner that does not explicitly compute boundary motion. In that work, the boundary that was moving was itself indicated by a motion cue. Here, we extend the result to show how *any* zero-crossing style edge operator can be easily used to distinguish between occluding and occluded surface. As shown in [7], with an appropriate change of coordinate systems it is sufficient to consider only two cases. In one, two surfaces are moving towards one another with equal but opposite optical flows. In the second case, the surfaces are moving away from one another with equal but opposite flows. Over time, the Laplacian pattern at the boundary will move with the surface to which it is attached. If a zero-crossing edge detector is applied to an optical flow pattern, all that is necessary to classify the edge is to observe the sign of the Laplacian pattern as it translates.

The situation is somewhat more complicated if edges are signaled by some feature other than optical flow. In such cases, it is necessary to consider both the contrast orientation of the edge and the pattern of motion to either side. The sign of the Laplacian function can be used to determine the direction of boundary movement relative to the direction of the gradient at the boundary. If we observe the value of the Laplacian at the zero crossing and that value goes negative, then we know that the edge has moved in the direction of the gradient. If the value of the Laplacian goes positive, then the edge motion is in the direction opposite to the gradient. It is still necessary to compare edge motions and surface motions. Again using the coordinate system transform, we need only determine whether the two surfaces are moving towards or away from each other. It is not necessary to quantitatively estimate actual surface and boundary flows.

The following algorithm implements this process:

1. Find an edge point, $\vec{x}_0$, in frame $t_0$. Compute the gradient $\nabla i(\vec{x}_0)$, where $i(...)$ is any perceivable function of $x$ that corresponds to surface properties.
2. Project all optical flow values onto an axis parallel to $\nabla i(\vec{x}_0)$.
3. Normalize coordinates by locating an evaluation point $\vec{x}_1 = \vec{x}_0 + f_a$ in frame $t_1$, where $f_a$ is the average inter-frame flow in the neighborhood of $\vec{x}_0$.

4a. The direction of $\nabla i(\vec{x}_0)$ points towards the side of the boundary corresponding to the occluding surface if $\nabla^2 G\ i(\vec{x}_1)$ is negative and the two surfaces are approaching one another or if $\nabla^2 G\ i(\vec{x}_1)$ is positive and the two surfaces are separating.

4b. The direction of $\nabla i(\vec{x}_0)$ points towards the side of the boundary opposite the occluding surface if $\nabla^2 G\ i(\vec{x}_1)$ is positive and the two surfaces are approaching one another or if $\nabla^2 G\ i(\vec{x}_1)$ is negative and the two surfaces are separating.

Note that if surface motion is parallel to the boundary, no determination of occluding and occluded surfaces is made. In fact, in this situation no definitive determination is possible based only on visual motion.

One advantage of this particular algorithm is that it directly provides a mechanism for combining motion-based boundary detection with static edge cues. Discontinuities in optical flow can only occur due to discontinuities in depth and/or due to two surfaces moving relative to one another. Thus, flow edges can arise from far fewer causes than edges due to changes in intensity, texture, color, etc. Unfortunately, flow edges are difficult to localize precisely. The above algorithm can be used to filter out all static edges that are not associated with a change in optical flow over the neighborhood of the edge. The effect is to use motion to reduce ambiguity, while using the static cues to preserve localization. In our current algorithm, we are only interested in boundary points at which we can differentiate between occluding and occluded surfaces. As a result, we delete all edge elements that do not have some differential optical flow along an axis perpendicular to the edge. This is easily done by modifying the above algorithm as follows:

3b. If the magnitude of $\nabla^2 G\ i(\vec{x}_1)$ is close to zero, delete the edge element at $x_0$ from further consideration.

Only a bit more complexity is required in order to recognize edges with differential motion only tangential to the edge orientation. Such edges signal surface boundaries, but it is not possible to distinguish between the occluding and occluded sides.

# 4 Occlusion-Sensitive Matching.

We have developed a simple model of how occlusion information might be used to aid in recognition. The model uses occlusion cues arising from the boundary flow constraint to reduce ambiguity in template matching applied to partially occluded objects. In presenting this model, our aim is to demonstrate the utility of incorporating occlusion information directly into the recognition process. The specifics of the algorithm are for purposes of illustration only. The approach will work for verification as well as standard template matching. Any occlusion cue can be used; the method is not limited to using just motion information. More efficient and reliable implementations are possible. The basic principles of our approach can be summarized as follows:

- *Determine a matching "score" based on searching for model features in the image.*
- *Introduce penalities for model features not in the image, but only if there is* not *evidence for the features being hidden by an occlusion.*
- *Do* not *introduce penalties for image features not accounted for in the model.*

In the examples presented below, we define the matching score to be the percentage of model features found in the image. This is done by computing the ratio of matched model features to potentially matchable model features. The features used in our simple example are silhouette edge elements. Only image edges with differential motion across the edge are used. A small distance toleration is allowed for to accommodate noise and other distortions. Information about occluding edges in the image is used in two ways. First of all, the model/non-model sides of the template edge must be compatible with the occluding/occluded sides of the image edges. (Note that this is a stronger requirement than just orientational compatibility.) Secondly, a model edge element is considered potentially matchable if it is not *masked.* When a model is being matched at a particular image location, masking occurs if there are significant occlusion edges in the image within the interior of the model. Masking regions are "grown" outward from the occluding side of any interior image edges. To assure that it will not extend beyond any occluding surface, the masking region ends at the first image edge reached. In our current implementation, matching is first done without using the masking operation. Areas of partial match are then reevaluated using the masking procedure.

A set of simple examples was created to test our approach of occlusion sensitive matching. We used artificially created objects to better control for ambiguity in matching. However, the examples all involve real imagery and automatically determined optical flow. Figure 1 shows a set of fourteen object models. Two actual objects were used, one **T** shaped, the other **L** shaped. Figure 2 shows one frame from a sequence in which the **T** is moving behind a wall to the right. The wall is partially occluding the **T**. As a result, simple template matching may not be effective for recognition. Figure 3 shows contrast edges in the **T** sequence. The edges were determined using a large kernel zero-crossing operator. Figure 4 shows motion/contrast edges determined by deleting edge

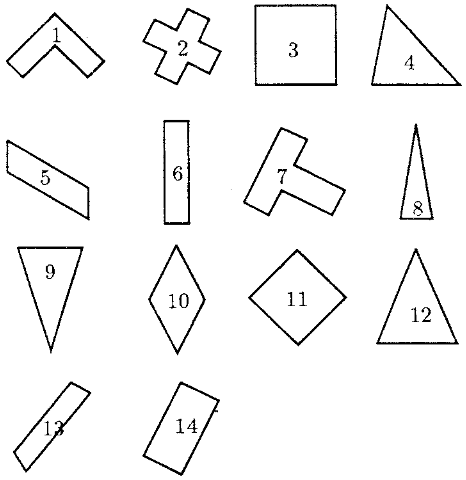


Figure 1: Model set.

Figure 2: Frame from **T** image sequence.

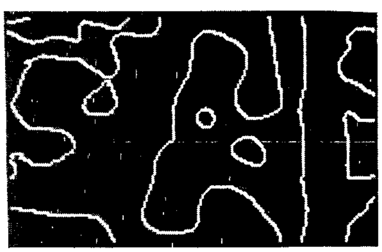

Figure 3: **T** contrast edges.

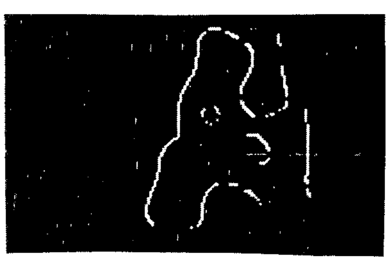

Figure 4: **T** motion/contrast edges.

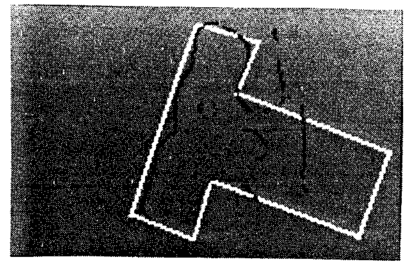

Figure 5: Best fit location for model.

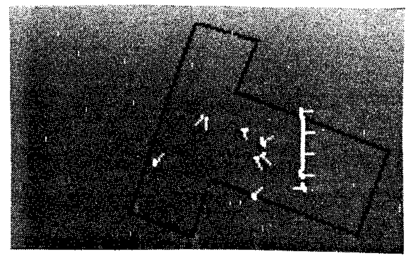

Figure 6: Unmatched edges.

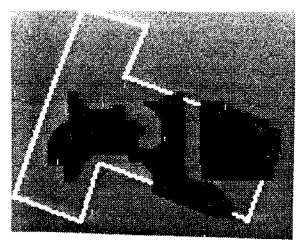

Figure 7: Masked portions of **T** model.

elements in figure 3 that are not associated with differential optical flow across the edge. Figure 5 shows the position of the T model in the image resulting in the highest matching score. Figure 6 shows the unmatched edges within the **T** model when applied to the image at the location shown in figure 5. The hash marks along the edges point to the occluding surface, as indicated by the boundary flow constraint. Finally, figure 7 shows the portions of the **T** model which have been masked as a result of the internal edges shown in figure 6.

Table 1 shows the matching scores for all model types evaluated against the **T** and **L** sequences. The highest scores in each column have been italicized. The models are matched against the raw contrast edges, the motion/contrast edges, the motion/contrast edges using the model/non-model orientational compatibility constraint, and finally using all of the matching constraints described above (differential motion, model/non-model edge orientation, and masking). The data, while currently limited to a few test cases, suggests that using occlusion information can reduce ambiguity in matching. Using all of the available matching constraints, both examples are correctly classified. Using either traditional template matching or using only a subset of the matching constraints causes one or both of the images to be misclassified.

## 5 Summary.

Edge detection is possible based on both contrast and motion information. Contrast edges can arise from a large number of causes and thus are difficult to accurately interpret. Motion edges are always associated with depth and/or surface boundaries, but are difficult to localize precisely. The motion-based segmentation technique described above combines motion and contrast cues in an integrated edge detection process. Localization is based on contrast edges, while motion information is used to filter out edges not likely to correspond to surface boundaries. The method further gives a direct indication of the side of the boundary corresponding to the occluded surface.

Identification of occluded and occluding surface can significantly aid in recognition tasks. We have presented a simple matching algorithm in which the presence of occlusion boundaries is used to avoid penalizing matches for situations in which model features are hidden from view by other objects. While our algorithm has been described within the context of template matching, it is equally appropriate when verifying hypothesized matches suggested by more complex three-dimensional reasoning processes.

## References

[1] R. Jain, W.N. Martin, and J.K. Aggarwal, "Segmentation through the detection of changes due to motion", *Computer Graphics and Image Processing*, 11:13–34, 1979.

[2] W.B. Thompson, "Combining motion and contrast for segmentation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-2:543–549, November 1980.

[3] S.M Haynes and R. Jain, "Detection of moving edges", *Computer Vision, Graphics and Image Processing*, 21:345–367, March 1983.

[4] E. Gamble and T. Poggio, *Visual Integration and Detection of Discontinuities: The Key Role of Intensity Edges*, AI Memo 970, MIT, 1987.

[5] D. Waltz, "Understanding line drawings of scenes with shadows", In P.H. Winston, editor, *The Psychology of Computer Vision*, McGraw-Hill, New York, 1975.

[6] D.L. Smitley and R. Bajcsy, "Stereo processing of aerial, urban images", *Proc. Seventh Int. Conference on Pattern Recognition*, 433–435, 1984.

[7] W.B. Thompson, K.M. Mutch, and V.A. Berzins, "Dynamic occlusion analysis in optical flow fields", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-7:374–383, July 1985.

[8] L.G. Roberts, "Machine perception of three-dimensional solids", In J.T. Tippett et al., editors, *Optical and Electro-Optical Information Processing*, MIT Press, Cambridge, MA, 1965.

[9] W.E.L. Grimson, "Recognition of object families using parameterized models", *Proc. First International Conference on Computer Vision*, 93–101, 1987.

[10] D.P. Huttenlocher and S. Ullman, "Object recognition using alignment", *Proc. First International Conference on Computer Vision*, 102–111, 1987.

[11] W.A. Perkins, "Model-based vision system for scenes containing multiple parts", *Proc. Fifth International Joint Conference on Artificial Intelligence*, 678–684, 1977.

[12] J.W. McKee and J.K. Aggarwal, "Computer recognition of partial views of curved objects", *IEEE Trans. on Computers*, C-26:790–800, 1977.

[13] R.C. Bolles, "Robust feature matching through maximal cliques", *Proc. SPIE Technical Symposium on Imaging Applications for Automated Industrial Inspection and Assembly*, 1979.

[14] R. Fisher, "Using surfaces and object models to recognize partially occluded objects", *Proc. Eighth International Joint Conference on Artificial Intelligence*, 989–995, 1983.

[15] S. Castan, J. Shen, and N.Q. He, "A method for recognition and positioning of partially observed objects", *Proc. Eighth Int. Conference on Pattern Recognition*, 1986.

| Model | T image sequence | | | | L image sequence | | | |
|---|---|---|---|---|---|---|---|---|
| | contrast edges | motion/ contrast | model/ non-model | occlusion masking | contrast edges | motion/ contrast | model/ non-model | occlusion masking |
| $M_1$ (L) | .635 | .422 | .345 | .439 | *.847* | .594 | .550 | *.637* |
| $M_2$ (Cross) | .659 | .562 | .511 | .542 | .754 | .517 | .448 | .454 |
| $M_3$ (Square) | .642 | .248 | .215 | .296 | .512 | .260 | .192 | .192 |
| $M_4$ (Asymmetric triangle) | .628 | .520 | .456 | .603 | .416 | .321 | .257 | .277 |
| $M_5$ (Quadrilateral) | .652 | .380 | .295 | .526 | .761 | .377 | .338 | .338 |
| $M_6$ (Rectangle) | *.800* | .548 | .388 | .638 | .704 | .504 | .356 | .356 |
| $M_7$ (T) | .670 | .532 | .494 | *.667* | .543 | .327 | .270 | .286 |
| $M_8$ (Narrow triangle) | .665 | .543 | .520 | .520 | .715 | .498 | .412 | .412 |
| $M_9$ (Inverted triangle) | .769 | .474 | .446 | .475 | .713 | .478 | .430 | .430 |
| $M_{10}$ (Narrow diamond) | .621 | *.571* | *.566* | .606 | .797 | *.648* | .571 | .571 |
| $M_{11}$ (Standard diamond) | .583 | .456 | .406 | .437 | .772 | .594 | .556 | .559 |
| $M_{12}$ (Broad triangle) | .563 | .540 | .398 | .525 | .716 | .425 | .372 | .380 |
| $M_{13}$ (Tilted trapezoid) | .635 | .450 | .375 | .551 | .745 | .625 | *.590* | .590 |
| $M_{14}$ (Tilted rectangle) | .574 | .426 | .413 | .439 | .702 | .603 | .554 | .561 |

Table 1: Matching scores – all models applied to **T** and **L** sequences.