Local Detection of Occlusion Boundaries in Video

Andrew N. Stein* and Martial Hebert
The Robotics Institute, Carnegie Mellon University
Pittsburgh, Pennsylvania
anstein@cmu.edu, hebert@ri.cmu.edu

Abstract

Occlusion boundaries are notoriously difficult for many patch-based computer vision algorithms, but they also provide potentially useful information about scene structure and shape. Using short video clips, we present a novel method for scoring the degree to which edges exhibit occlusion. We first utilize a spatio-temporal edge detector which estimates edge strength, orientation, and normal motion. By then extracting patches from either side of each detected (possibly moving) edglet, we can estimate and compare motion to determine if occlusion is present. This completely local, bottom-up approach is intended to provide powerful low-level information for use by higher-level reasoning methods.

1 Introduction

Occlusion boundaries provide strong cues about the 3D structure of a natural scene. Their detection has particular application in scene segmentation, figure-ground separation, and shape extraction, all of which can improve object recognition and detection (*e.g.* [21]). Such boundaries correspond to locations in an image where one physical surface is closer to the camera than another. Usually, these boundaries are also visible as edges. In this paper, we explicitly distinguish between the detection of typical *edges*, which may result from changes in intensity, color, or texture, and *boundaries* which additionally correspond to 3D scene structure. Indeed, we will consider occlusion boundaries to be a subset of the edges in a scene. The goal of this work is to identify those edges in an image which are also occlusion boundaries. As with general scene segmentation, measuring the performance of edge detectors is fairly ill-defined beyond the use of semantic labels provided by humans. Occlusion boundaries, however, have a physical meaning for which the notion of a "correct" answer is more easily defined.

Occlusion cannot be detected from a single image. Thus, while the use of sophisticated edge detectors that can handle texture and color ([14, 15, 17]) may offer improved results over a simple Canny edge detector [5], they still (correctly) respond strongly to edges which do not correspond to any physical occlusion. Occlusion can, however, be observed through motion – either the scene's, the camera's, or both. Without motion (and without non-trivial higher level knowledge), it is impossible to distinguish between edges and boundaries in a single image.

*Partial support provided by a National Science Foundation Graduate Research Fellowship, with additional support from the Intelligent Robotics Development Program, a 21st Century Frontier R & D Program funded by the Korean Ministry of Commerce, Industry, and Energy.

When one object occludes another, due to either object's motion or to parallax from camera movement, pixels may disappear or become visible. This occlusion and disocclusion are the source of significant difficulty for many computer vision methods, which often rely on image patches that may overlap the boundaries. Since it is generally assumed that all the pixels within a patch "belong together" (*e.g.* are from the same object, motion layer, *etc.*), patches overlapping occlusion boundaries violate this assumption and muddy results. For this reason, patches near boundaries are often treated as outliers, or multiple/adaptive-windowing techniques are employed [6, 10, 11]. By contrast, the work of this paper will focus precisely on these boundaries themselves and will attempt to detect them directly by augmenting standard single-image edge detection with motion information.

Focusing explicitly on motion at the edges in a scene will allow us to avoid dense motion estimation and reasoning via full-blown optical flow followed by region-growing or clustering approaches. In addition, we can detect *local* occlusion directly using a bottom-up approach which is complimentary to top-down approaches that rely on higher-level reasoning. Such methods often impose the restrictive assumption that the scene consists of a set of distinct layers moving separately. The challenge of a purely local, bottom-up approach is estimating and utilizing motion information at exactly those locations (occlusion boundaries) at which it arguably the most difficult to obtain.

Using a few frames of video (typically 6-10), we will leverage space-time edge detection [1, 4, 8, 22], which simultaneously estimates local edge strength, orientation, and crucially, edge *speed* in the direction normal to its orientation. Given these quantities, we can then safely extract spatio-temporal patches of intensity data from either side of a moving edge. By analyzing and comparing the motion in these two patches, we can estimate the degree to which we believe occlusion is occurring there.

After some discussion of related work, the above approach is explained in more detail in the remainder of the paper, followed by results on natural image sequences.

2 Related Work

Martin *et al.* [14] have designed an excellent edge detector which has been trained (using human-labeled data) to respond to those local gradients of brightness, color, or texture which people generally seem to label as edges. (Note that the authors' use of the term "boundaries" does not directly correspond to the extraction of the occlusion boundaries sought in this work). Their comparison of histograms on either side of proposed oriented edges allows the detection of difficult complex edges, such as those between textured regions or in clutter. The Compass Edge Detector uses a very similar histogram-based approach [15, 17] but without learning all the parameters from human-labeled data. None of these approaches, however, incorporate motion information or seek specifically to identify occlusion boundaries.

Smith *et al.* [20] track edge fragments and use Expectation Maximization to then segment the scene into regions with consistent motion. Their approach is one of the few that tracks edges/boundaries directly, but it still assumes layered scene structure, and the method seems to work best on two-layer sequences (computation increases exponentially with the number of layers). Our completely local approach, on the other hand, makes no assumption that the scene is layered and should therefore be applicable to a more general set of scenes. Also, the initial step in [20] of linking edge pixels into chains which are part of a single surface is non-trivial, but quite crucial since it is a hard decision imposed on the remainder of the system.

Black and Fleet [3] also attempt to estimate local evidence of occlusion by analyzing motion. They build a parametric model of local occlusion within sampled circular regions and then estimate the posterior probability of this model using particle filtering. Demonstrated results are quite limited, possibly due to the significant computational expense of evaluating the thousands of particles necessary to sample the parameter space for each region.

Many approaches segment dense motion estimates derived from optical flow into distinct regions or layers (*e.g.* [12, 19], to name just a few), usually treating the erratic results at boundaries as outliers to an underlying smooth process. The subsequent delineation of precise motion boundaries, if performed at all, is generally of secondary importance. A notable exception, however, is found in [9], where vertical and horizontal between-pixel motion boundaries plus their interactions with nearby dense optical flow vectors are considered in an MRF framework. Stereo or structure from motion techniques also have trouble near occlusion boundaries and usually focus on the interiors of regions while handling data near occlusions as complex special cases [6, 10, 11]. Our work, on the other hand, is not concerned with precise dense motion estimation or full 3D scene reconstruction; we seek only to *identify* oriented boundary locations that correspond to visible occlusion.

Also related to occlusion detection is the classical problem of T-junction detection, recently addressed in a discriminative framework by [2] (see references therein for substantial prior work). That work utilizes spatio-temporal slices (not volumes) and demonstrates limited extraction of occlusion boundaries. For higher level reasoning, note that T-junction detection may be a complimentary source of information to the motion *boundary* detection presented here.

As discussed above, our approach will compare motions from two patches of spatio-temporal data to determine their consistency. This exact problem was recently addressed, though in a rather different context, in the work of [18]. That approach utilizes a continuous rank-increase measure between Gram matrices constructed from spatio-temporal derivatives within each patch. This measure provides a motion inconsistency score without explicitly estimating the patches' motion vectors.

3 Properties of Local Occlusion

Assume for a moment that we have detected a small edge fragment (or "edgelet") in an image. If that edgelet corresponds to simple texture on the surface of a single object, we would expect the patches on either side to exhibit motion consistent with the edglet and with each other. If the edglet's appearance is the result of an occlusion boundary, we would expect, in general, the patch on the foreground side of that occlusion to move consistently with the edgelet while the background patch moves in a manner which contradicts the motion of the foreground patch and/or the edge.

A motion estimate for a patch of intensity data may be computed from the patch's spatio-temporal derivatives [13, 18, 23]. This idea is based on the brightness constancy assumption and forms the fundamental building block for many optical flow, tracking, and registration methods. But as discussed, the underlying assumption when using a patch of data is that all the data within that patch belongs to the same surface and therefore moves together. If we naïvely extract a spatio-temporal patch in the vicinity of a moving edgelet corresponding to an occlusion boundary, however, we run the risk of including data from two different surfaces or objects in the scene. Our derivatives will then be corrupted, and our motion estimates will be incorrect. To extract a more appropriate patch which does not cross the occlusion boundary, we first need a method for detecting oriented edgelets in an image sequence and determining their normal velocity.

4 Spatio-Temporal Edge Detection

In this section we discuss the detection of oriented, moving edges in short video clips. We use the histogram-based approach of [22], summarized here. By using a more general, "non-parametric" model of edges, this method offers superior performance to the classical filtering approaches of [1, 4, 8], which effectively consider only mean intensities on either side of an edge. Improvement is most apparent for edges bordering textured or cluttered regions, which are precisely those edges at which occlusion is most likely to be visible. Thus we have chosen an initial edge detection method which works well in such cases.

We define an edge to be a position in the image for which the distribution of intensity (or textons, color, *etc.*) is significantly different on either side of an oriented dividing line. To estimate oriented edge strength in a single image, one can extract a circular patch of data, divide that patch into a set of "pie slices," and efficiently compute histograms of the intensity in each half of the patch for a given set of orientations [14, 15, 17]. The histograms from two patch halves at a given orientation are compared using a χ^2 distance metric, and the largest distance is returned as that position's edge strength, while the corresponding orientation is used as that edge's spatial orientation.

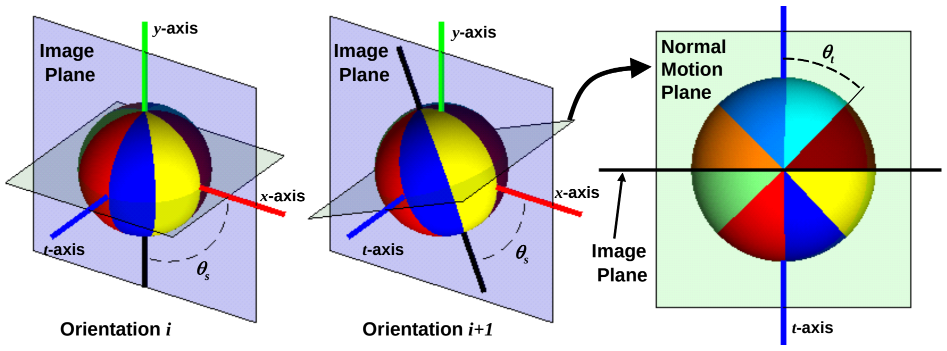

Figure 1: We use a volumetric patch to simultaneously detect edges and estimate both their spatial orientation and normal velocity. The spherical patch can be divided into histogram sections and oriented to produce a detector for a given set of spatial orientations and normal motions. (Figure reproduced from [22].)

Similarly, the angle with respect to the *temporal* axis at which an edge sweeps through a sequence of images corresponds to its normal velocity. (For example, a simple vertical step edge moving left to right produces an *x*-*t* slice of intensities as shown in Figure 2 (a).) As shown in Figure 1, we can extract a spherical (or ellipsoidal) patch from the spatio-temporal volume of data and apply the same approach as above to pick the *temporal* orientation at which histograms computed from each half of the hemisphere differ the most. The sphere can be divided into sections (like those of an orange), analogous to the pie slices above. This sectioned sphere is first oriented to a given spatial orientation, θ_s, and then the sections can be used to efficiently construct the required histograms to find the best temporal orientation θ_t which corresponds to the normal velocity at that θ_s.

The best pair (θ_s, θ_t) specifies the normal for the best dividing *plane* for the patch around a given (x, y, t) position in the image sequence. When extracting patches of data to compare motion on either side of a moving edgelet, that plane is precisely the boundary we want to avoid crossing. This is illustrated in Figure 2. In (a), we see an *x*-*t* slice of data from a spatio-temporal volume in which a vertical edge moves from left to right. If

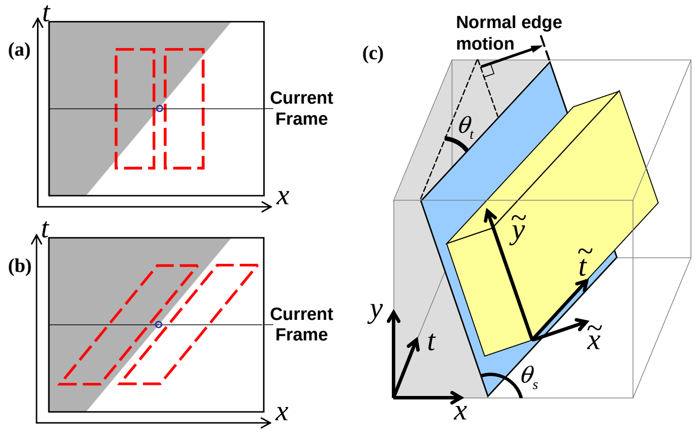

Figure 2: Extraction of spatio-temporal patches aligned to detected moving edges. (a) Simple patches in an *x*-*t* slice will cross a moving edge, but (b) using a spatio-temporal moving edge detector we can extract patches skewed to be aligned with the edge. (c) A volumetric patch is shown for an oriented, moving edge, with its associated coordinate frame.

we detect only that edge's position (depicted by the small circle) and orientation within a single frame (which is into the page for a vertical edge), extracted patches will contain data from both sides of the edge over time. Using a spatio-temporal edge detector as depicted in (b), we can also estimate the normal motion of the edge, allowing us to extract patches which respect the edge's motion and only consider data from one side of the (potential) occlusion boundary at a time.

5 Comparing Motions

We can now safely extract spatio-temporal patches of data on either side of a moving edge which are consistent with that edge's normal motion. By using oriented, skewed patches as in Figure 2 (b), we effectively remove the local normal motion component. We can then estimate and compare only the *residual* tangential and/or normal motions in either patch. Note that the patches' coordinate frames correspond exactly to the directions of motion in which we are interested; their bases are aligned to the edge's normal and tangent directions spatially ($\tilde{x}$ and $\tilde{y}$), and temporally to the direction of normal motion ($\tilde{t}$). So we can detect occlusion boundaries by comparing the differences in normal and tangential motion from the two patches. At a boundary, we expect to see a significant difference in these estimated motions. (In the remainder of the paper, we will adopt this $(\tilde{x},\tilde{y},\tilde{t})$ notation to emphasize that all analyses are performed in a coordinate frame relative to the current edge under consideration.)

Since we are assuming small motion between frames, we will only attempt to estimate and compare normal and tangential *translation* $(\tilde{u},\tilde{v})$ within the two extracted patches. We use the classical approach of [13] extended to multiple frames, similarly to [18], since our patches also have temporal extent.

Leveraging the standard brightness constancy assumption from optical flow at each of the *M* pixels within a patch, we stack the patch's spatio-temporal intensity derivatives, again taken along the $\tilde{x}$, $\tilde{y}$, and $\tilde{t}$ directions within the patch:

$$\underbrace{\begin{bmatrix} P_{\tilde{x}_1} & P_{\tilde{y}_1} \\ P_{\tilde{x}_2} & P_{\tilde{y}_2} \\ \vdots & \vdots \\ P_{\tilde{x}_M} & P_{\tilde{y}_M} \end{bmatrix}_{M\times 2}}_{A} \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \underbrace{\begin{bmatrix} P_{\tilde{t}_1} \\ P_{\tilde{t}_2} \\ \vdots \\ P_{\tilde{t}_M} \end{bmatrix}_{M\times 1}}_{b} . \tag{1}$$

This leads to the following least squares problem to solve for the desired motion components $(\tilde{u}, \tilde{v})$:

$$A^T A \begin{bmatrix} u_{\tilde{x}} \\ u_{\tilde{y}} \end{bmatrix} = A^T b \Longrightarrow \underbrace{\begin{bmatrix} \sum P_{\tilde{x}}^2 & \sum P_{\tilde{x}} P_{\tilde{y}} \\ \sum P_{\tilde{y}} P_{\tilde{x}} & \sum P_{\tilde{y}}^2 \end{bmatrix}}_{G} \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} \sum P_{\tilde{x}} P_{\tilde{t}} \\ \sum P_{\tilde{y}} P_{\tilde{t}} \end{bmatrix}, \tag{2}$$

where all summations are taken over the M pixels within the patch. Note that this is a slightly different formulation than the one given in [18], as we are solving for two-dimensional motion vector in the image plane instead of a three-dimensional vector describing the direction of motion through space-time. Normalizing by the temporal motion component (*i.e.* "w" in [18]) makes the approaches equivalent for our purposes.

While we can always find a solution to this system of equations, our "confidence" in the resulting estimated motion vector depends on the spatial gradients within the patch. Loosely speaking, in a patch taken from a nearly-uniform region of the image, the covariance on the estimated motion will be much higher than the covariance for a patch containing strong gradient information which provides more meaningful evidence of motion. It is critical that we take these covariances into account when comparing motion estimates from different patches, especially since it is very common that one side of an occlusion boundary is nearly uniform. Luckily, the matrix G in (2), which is the the same as that used in the Harris detector [7] (except that it involves a spatio-temporal patch), exactly captures the necessary spatial covariance information. Since we have two G matrices, one from each side of the patch, we will consider two motion estimates to be consistent if their difference according to *either* side's covariance estimate is small. Thus, we will take the maximum of the two consistency scores to accomplish this logical OR operation, as will be seen below.

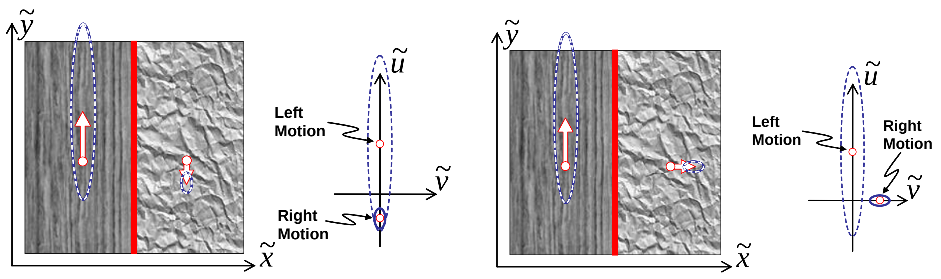

Figure 3: For patches of data on either side of an edge, the covariance on the estimated motion influences the occlusion scoring. The motions for the left example more consistent because we are less sure of the tangential motion. We are fairly sure that the right example is an occlusion boundary because the confident normal motion estimates disagree.

To make this discussion more concrete, consider the patch of data taken around a vertically-oriented edge shown in the left half of Figure 3. Assume we have used Eq. (2) to estimate the indicated motions on either side of the edge. Because of the lack of significant horizontal texture on the left side of the edge, we cannot be very sure of the estimated tangential motion, but we are fairly certain about the normal motion. On the highly-textured right side, we are confident about both components of the motion. These confidences are indicated by covariance ellipses. If we consider the left and right motions as 2D points in $(\tilde{u},\tilde{v})$ space, we can compare them by evaluating the distance between them. Without taking the covariance into account, the difference in motion appears to be significant, which would indicate occlusion. But using G to normalize the distance computation (*i.e.* computing a Mahalanobis distance) allows us to be more “forgiving” along the tangential direction since we are less sure about the left side’s motion in that direction. Thus, this edge will receive a lower (albeit non-zero) occlusion score, as desired.

In the right half of Figure 3, we see the same patch of data with different estimated motion for the right patch. In this case, we are still fairly confident that there is no residual normal motion in the left half (*i.e.* the initial normal speed estimate is correct) and we are also confident that the right half did move normal to the edge. In this case, the two points’ motions plotted in $(\tilde{u},\tilde{v})$ space are roughly the same (simple Euclidean) distance apart as they were for the previous example, so without taking variances into account, either example would have the same degree of motion inconsistency (and thus the same occlusion score). But because the variance on the normal component of our motion estimate in the left patch is small, we will trust the disagreement in normal motions between the two sides and give the correct high occlusion score for this example. Note that this is true despite the fact that the difference in the normal components is actually *smaller* than the difference in the tangential components. These examples demonstrate that the covariance estimates allow us to rely on the most informative motion component when determining occlusion.

For each edgelet identified in a scene, we would like to define a score which will be near zero for edgelets at which no occlusion is visible and one for edgelets where there is clear evidence of occlusion. From two patches, P_L and P_R, aligned to either side of the edgelet, we estimate motion vectors $\mathbf{u}_L = [\ \tilde{u}_L \quad \tilde{v}_L\ ]^T$ and $\mathbf{u}_R = [\ \tilde{u}_R \quad \tilde{v}_R\ ]^T$ as described above and compute their difference $\mathbf{u}_d = \mathbf{u}_L - \mathbf{u}_R$. We can then compute the consistency of the left and right motion vectors according to each patch’s spatial information, encoded in G_R and G_L. As discussed, we will keep the maximum of the the two scores. Finally, to convert to a measure of **in**consistency, and thus a measure of occlusion, we will substract the result from one. The following equation captures this process:

$$score = 1 - \max\left\{\exp\left(-\frac{\mathbf{u}_d^T G_L \mathbf{u}_d}{2}\right), \exp\left(-\frac{\mathbf{u}_d^T G_R \mathbf{u}_d}{2}\right)\right\} \qquad (3)$$

We can use this equation to score each edgelet as also being an occlusion boundary. Results using this approach are provided in the next section.

6 Results

Given a few frames of an image sequence, we can first find edges and then estimate whether those edges exhibit occlusion within the sequence. It is important to note that this approach is purely local and can therefore only reason about occlusion at an edge if visible occlusion actually occurs at that location within the observation’s timespan. We may observe edges which lie in front of a uniform background for the duration of the video clip. These edges will not exhibit any visual cues of occlusion locally, and thus they will not be detected as occlusion boundaries by our method. Higher-level reasoning and edge grouping or chaining, which may handle such sitations, are left to future work.

We first pre-process the data to remove dominant motion. This prevents large observed normal motions due to camera movement from overshadowing the relative motion between surfaces (*i.e.* the parallax), which provides the evidence of occlusion and is usually relatively subtle. This step simply consists of a full-frame affine registration of each frame in the sequence to a selected reference frame (*e.g.* the middle frame of the sequence).

Next we compute edge strength, orientation, and normal motion at each spatial location according to the method outlined above and in [22], including the use of sub-pixel edge localization and interpolation of orientation and normal speed. Non-local maxima of edge strength are also suppressed [16], and the result is hysteresis thresholded. A patch of data is then extracted around each edglet, aligned to its orientation and normal motion. The residual normal and tangential motions are computed for either half of the patch as described. Finally, the measure of occlusion given by (3) is computed.

As noted in Section 2, Shechtman and Irani have recently proposed an efficient method for computing motion (in)consistency from two spatio-temporal intensity patches *without* explicitly estimating motion itself [18]. Their work, however, was in the context of behaviour recognition. We compare their rank-based score to our explicit motion comparison score in the results that follow. Each score is computed using the same initial edge detections and extracted patches.

In order to quantitatively evaluate results, we hand-labeled the occlusion boundaries in each test sequence below. For each occlusion scoring method (ours and the rank-based score from [18]), we vary the threshold on the score and count true and false positives. We can then generate and compare plots of pixel-wise precision vs. recall for each example using both scoring methods. (To mitigate difficulties in precisely localizing boundaries when labeling images by hand, we dilate the ground truth label image by one pixel when comparing it to the score images.)

Figure 4 shows the middle frame of an 8-frame synthetic sequence depicting a square translating up and right in front of a textured background. The hand-labeled ground truth occlusion boundaries are overlaid in red, and the edge strength is also provided. Using our method, we see that the boundaries of the square – and *not* the background's texture edges – correctly receive a high occlusion score. Using the rank-based score provides visually similar discrimination between edges and occlusion boundaries. When we compare the scores' precision-recall plots in Figure 6 (a), we see that ours is slightly superior.

The results for three real sequences from a handheld video camera, each 8-10 frames in length, are provided in Figure 5. Each shows qualitatively that we are generally able to distinguish occlusion boundaries from texture edges using the described approach. Furthermore, we see in Figure 6 (b)-(d) significantly better performance using our scoring approach, regardless of choice of threshold.

7 Conclusion and Future Work

We have described a novel, bottom-up procedure for detecting local occlusion boundaries in short video clips. The notion of locality here applies not only to relatively small spatial patches, but also to the fact that we rely on short-term *temporal* information rather than long-term edge tracking or higher-level reasoning. Our approach is also quite general: it does not require a strict layered scene structure, the camera can be moving or stationary, and the scene can be static or dynamic.

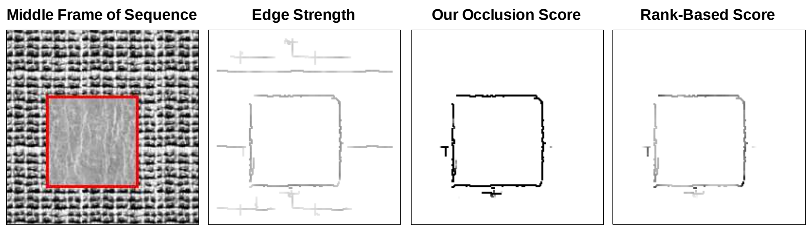

Figure 4: Results for a 6-frame synthetic sequence depicting a square moving 2 pixels up and 2 pixels to the right in each frame. Ground truth occlusion boundaries are displayed in red. For edge strengths and scores in this figure and in Figure 5, darker color corresponds to larger values.

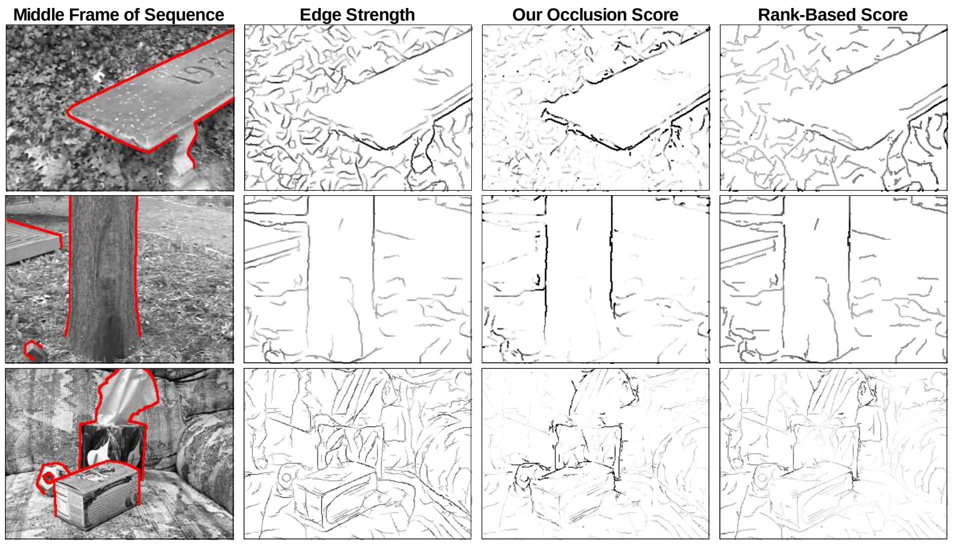

Figure 5: For handheld video sequences observing a bench in front of ivy (top) and a tree trunk (middle), and a set of objects on a textured couch (bottom), we see a representative frame of data with ground truth occlusion boundaries labeled in red, the detected edge strengths, and the two different occlusion scores, from left to right respectively.

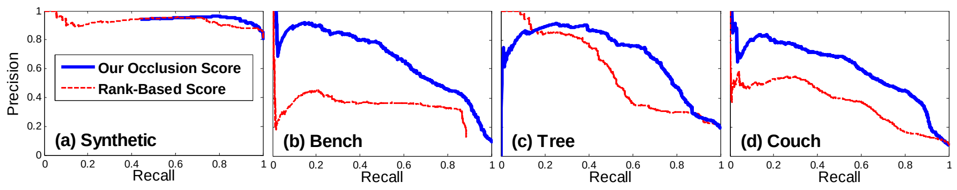

Figure 6: Precision vs. Recall plots for each example, showing significant performance gain when using our motion-estimation score over the rank-based motion inconsistency measure from [18].

The results provided here only utilize intensity (grayscale) information, but we plan to extend the entire method to utilize color information as well. We believe this will improve the edge detection process as well as the motion extraction, both of which should result in improved occlusion detection. In addition, it may be worthwhile to explore the estimation and comparison of more complex motion models (*e.g.* affine) between the two patches.

Most interestingly, we now have a method for extracting substantial low-level edge information, including orientation, motion, and occlusion, which we can leverage for higher-level inference and reasoning. For example, we could include occlusion and motion information in the classical perceptual organization problem of salient contour extraction (which generally relies heavily on orientation estimates alone). We believe occlusion information will also be useful for figure-ground segmentation, cueing for object recognition, and other high-level computer vision tasks.

References

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(2):284–299, February 1985.

[2] N. Apostoloff and A. Fitzgibbon. Learning spatiotemporal t-junctions for occlusion detection. In *CVPR*, 2005.

[3] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion discontinuities. *IJCV*, 38(3):231–245, 2000.

[4] P. Bouthemy. A maximum likelihood framework for determining moving edges. *PAMI*, 11(5):499–511, May 1989.

[5] J. Canny. A computational approach to edge detection. *PAMI*, 8:679–698, 1986.

[6] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. In *CVPR*, pages 858–863, June 1997.

[7] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[8] D. J. Heeger. Optical flow using spatiotemporal filters. *IJCV*, 1:270–302, 1988.

[9] F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using markov random fields. *PAMI*, 15(12):1217–1232, December 1993.

[10] H. Hirschmüller, P. R. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *IJCV*, 47(1-3):229–246, April-June 2002.

[11] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *PAMI*, 16(9):920–932, September 1994.

[12] Q. Ke and T. Kanade. A robust subspace approach to layer extraction. In *MOTION*, pages 37–43, December 2002.

[13] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.

[14] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, May 2004.

[15] B. A. Maxwell and S. J. Brubaker. Texture edge detection using the compass operator. In *BMVC*, volume II, pages 549–558, September 2003.

[16] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks, and roofs. In *ICCV*, pages 52–57, 1990.

[17] M. Ruzon and C. Tomasi. Color edge detection with the compass operator. In *CVPR*, pages 160–166, June 1999.

[18] E. Shechtman and M. Irani. Space-time behavior based correlation. In *CVPR*, 2005.

[19] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, 1998.

[20] P. Smith, T. Drummond, and R. Cipolla. Layered motion segmentation and depth ordering by tracking edges. *PAMI*, 26(4):479–494, April 2004.

[21] A. Stein and M. Hebert. Incorporating background invariance into feature-based object recognition. In *WACV*, 2005.

[22] A. N. Stein and M. Hebert. Using spatio-temporal patches for simultaneous estimation of edge strength, orientation, and motion. In *Beyond Patches Workshop at CVPR*, 2006.

[23] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.