Combining Local Appearance and Motion Cues for Occlusion Boundary Detection

Andrew Stein[*] and Martial Hebert
The Robotics Institute, Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
anstein@cmu.edu

Abstract

Building on recent advances in the detection of appearance edges from multiple local cues, we present an approach for detecting occlusion boundaries which also incorporates local motion information. We argue that these boundaries have physical significance which makes them important for many high-level vision tasks and that motion offers a unique, often critical source of additional information for detecting them. We provide a new dataset of natural image sequences with labeled occlusion boundaries, on which we learn a classifier that leverages appearance cues along with motion estimates from either side of an edge. We demonstrate improved performance for pixelwise differentiation of occlusion boundaries from non-occluding edges by combining these weak local cues, as compared to using them separately. The results are suitable as improved input to subsequent mid- or high-level reasoning methods.

1 Introduction

Occlusion boundaries are a rich source of information in images. Not only do they provide boundary conditions for almost *any* process which reasons spatially within an image (*e.g.* optical flow, shape-from-X methods, feature extraction, filtering, *etc.*), but they also capture important perceptual information about the 3D scene [2]. Rather than being considered merely a nuisance to be “handled” or outliers to be avoided, as is often the case, these boundaries offer *opportunities* for segmentation and object discovery [3, 12, 14], and for reasoning about shape and structure [18].

Since occlusion boundaries correspond to locations where one object or surface is closer to the camera than another, we can exploit the resulting depth discontinuity as an indication of their existance. Noting that in many applications, video rather than single isolated images may be available, we can use local *motion* estimates as evidence of those depth discontinuities. In addition, most occlusion boundaries are also visible as appearance edges (though we note that many appearance edges arise solely due to surface markings or illumination effects). Neither motion nor appearance alone, however, is sufficient for the detection of occlusion boundaries. Accurate local motion estimates may be hard to obtain near occlusion boundaries, and appearance edges do not always correspond to occlusions. Thus we will combine multiple appearance cues, captured by state-of-the-art edge detectors, with local motion cues to show that *together* these distinct sources of information produce superior results to using either cue alone. In particular, our goal is to determine the subset of appearance edges that correspond to occlusion boundaries, thereby framing our problem as one of classification.

[*]This work was partially supported by a National Science Foundation Graduate Fellowship.

After explaining in Sections 2 and 3 the specifics of extracting our motion and appearance cues and their classification, we will describe our experiments in Section 4, demonstrating improved occlusion boundary detection when combining these cues. These experiments provide quantitative as well as anecdotal results on a novel dataset labeled for this task.

2 Local Occlusion Boundary Features

Edge detectors generally assign a "strength" to each pixel, which captures the degree to which an edge exists there, based on the contribution of various perceptual cues. At occlusion boundaries, there is often an additional cue in the form of inconsistent image motion. This motion may be caused by camera movement, which induces parallax at depth discontinuities, or it may be a result of dynamic objects in the scene. Our approach handles either situation equivalently and is thus more general than motion *detection* work that relies on a static camera for background subtraction, *e.g.* [14, 19]. In the following sections, we will describe our methods for extracting each of these features, which will then be used as cues for an occlusion boundary classifier described in Section 3.

2.1 Oriented Edge Detection

While classical edge detectors based on filtering are popular, most notably the Canny detector, they rely on rather simple models of image intensity at edges. Even moving beyond simple step edges to more complex edge types [13], linear filtering approaches still perform poorly on edges which exist between cluttered or textured regions. This is a serious concern for our work since we hope to extract motion in the vicinity of detected edges (as described in the Section 2.2 below). Motion is only observable when there is sufficient intensity gradient due to texture or clutter, so we need an edge detection approach which works well in such cases.

Thus we seek a detector capable of combining multiple cues which does not rely on overly simplistic edge models. An increasingly popular approach to achieve these goals computes edge strength using statistical comparisons of non-parametric distributions of cues on either side of a sample image patch at various orientations [8, 10, 11, 15, 20]. These detectors produce good results even on edges in texture and clutter and are therefore more appropriate for our task. Furthermore, they were extended to the spatio-temporal domain in [17], yielding a detector also capable of estimating an edge's normal *speed*. Though potentially useful for future work, here we focus instead on integrating *multiple appearance cues*, whereas [16, 17] only use intensity information.

Thus, we have chosen to use the popular Berkeley "*Pb*" detector for our experiments [10], which already incorporates three appearance cues (brightness, color, and texture) and offers a publicly available implementation. As an added benefit, the *Pb* detector's default parameters were learned on a large set of human-segmented data [9], allowing us to avoid tedious parameter tuning. At each location in the image, we interpolate better estimates for both orientation (θ) and edge strength (e) by fitting parabolas around the peak *Pb* response over the set of sampled orientations. Then we suppress those responses which are not local maxima along the edges' normal directions [13]. All edges which survive this suppression are kept for the classification step, *i.e.* we ignore edge strength at this stage (effectively thresholding at zero) to avoid prematurely ruling out edges simply because of low strength before also considering motion cues.

In Figure 1, we provide an example of edges detected using a traditional linear filtering approach (b), which is based on response to a quadrature pair of oriented filters [1, 5, 13],

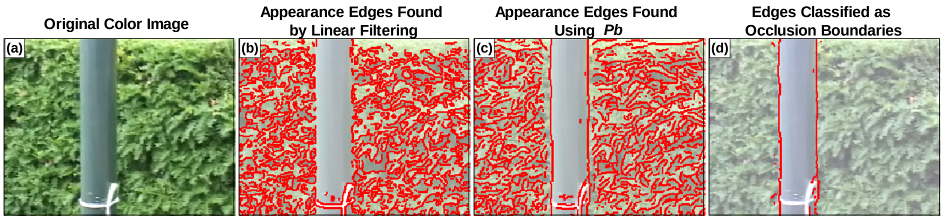
Original Color Image
Appearance Edges Found by Linear Filtering
Appearance Edges Found Using *Pb*
Edges Classified as Occlusion Boundaries
(a)
(b)
(c)
(d)

Figure 1: For an input image (a), we compare (b) classical edge detection using a quadrature pair of filters to (c) the Berkeley *Pb* detector (all non-zero responses after non-local maxima suppression are shown). Only the *Pb* detector fires consistently on the edges which lie on occlusion boundaries of the pole, giving subsequent classification a chance of succeeding. The goal of this work, then, is to utilize appearance and motion cues in order to classify which of those edge detections are also occlusion boundaries, as shown in (d).

as compared to the output of the *Pb* detector (c). Each shows all non-zero responses after non-local maxima suppression. Note how the *Pb* detector finds more consistent edges at the occlusion boundaries on the sides of the pole despite the background clutter. At this stage, we are most interested in providing all *potential* occlusion boundaries to the subsequent classifier (*i.e.* we can tolerate false positives but not false negatives). Therefore *Pb* is much better suited to our classification task, an example of which is shown in (d).

2.2 Local Multi-Frame Motion Estimation

As with edge detection, the estimation of image motion, *i.e.* optical flow, is a classical problem in computer vision (see [4] for a recent tutorial). Here, we will consider several consecutive frames of video and compute a *multi-frame* motion estimate. As compared to using only two frames, we find that using multiple frames produces substantially more robust estimates that are more discriminative for our classification task.

Given a set of frames $\{I^{(n)}\}_{n=-N}^{N}$ our goal is to find the translational motion, with components u and v, which best matches a patch of pixels P in the central reference image, $I^{(0)}$, with its corresponding patch in each of the other images, $\{I^{(n)}\}_{n\neq 0}$:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \arg\min \sum_{n=-N}^{N} h(n) \sum_{(x,y)\in P} w(x,y) \underbrace{\left(I^{(n)}(x+nu, y+nv) - I^{(0)}(x,y)\right)^2}_{I_t(u,v,n)} \qquad (1)$$

This implicitly assumes constant translation for the duration of the set of frames, which we find to be reasonable over brief time periods.

We employ Gaussian-shaped weighting functions, $w(x,y)$ and $h(n)$ (with associated bandwidths σ_h and σ_w), to decrease the contribution spatially and temporally of pixels distant from the center of the reference patch. We iteratively estimate u and v using a multi-frame, Lucas-Kanade style differential approach. This amounts to solving iteratively the following least squares problem for new translation estimates (at iteration $k+1$), given the previous ones (at iteration k), based on spatial derivatives of the reference patch, I_x and I_y, and temporal derivatives, I_t:

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t(u_k, v_k, n) \\ \sum I_y I_t(u_k, v_k, n) \end{bmatrix}, \qquad (2)$$

where the sums are taken over all pixels within the patch, across all frames. (For clarity, we have omitted the weights, $w(x,y)$ and $h(n)$, in this formulation.) In practice, we initially consider only $I^{(0)}$ and its two immediate neighbors. We then gradually increase the temporal window, initializing with the previous translation estimate, until finally considering all frames from $-N$ to N. This prevents frames at extremes of the temporal window from pulling us to poor local minima of (1)[1].

Aggregation of patches of data near occlusion boundaries is problematic and addressing this problem specifically for optical flow estimation is the subject of extensive research, including multiple motion estimation, robust estimators, line processes, and parametric models [2, 4]. Recently, impressive results computing dense flow fields in spite of significant occlusion boundaries by using a variational approach and bilateral filtering were demonstrated in [21].

For our purposes, since we are interested only in motion estimates near edges (rather than a dense flow field), we will choose patches of data P_L and P_R on either side of each detected edge pixel, as shown in Figure 2. In addition, because we have an estimate of each edge pixel's orientation, θ, we can align those windows to the edge in order to prevent the collection of information across a potential occlusion boundary. This technique is related to adaptive/multiple-window techniques, *e.g.* in stereo vision [6, 7], and was also recently used in occlusion reasoning [16]. (Spatio-temporal alignment to moving edges is also performed in [16], which could be used to augment our approach as well.) Computing the necessary derivatives within each window (via standard finite differencing), we can then estimate the motions ($\mathbf{u}_L = [\ u_L \quad v_L\ ]^T$ and $\mathbf{u}_R = [\ u_R \quad v_R\ ]^T$) of the patches on either side of each edge using the least squares approach outlined above. We then compute the difference in motion between the left and right patches, $\mathbf{u}_d = \mathbf{u}_L - \mathbf{u}_R$. Finally, we use the Euclidean norm of the $\mathbf{u}_d$ vector to capture the relative motion between the surfaces on either side of a potential occlusion boundary. This metric serves as the second feature, or cue, used by the classifier described in the next section.

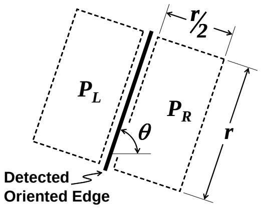

Figure 2: Patches for motion estimation aligned to an oriented edge.

In our experiments, this Euclidean metric proved to be just as useful as a Mahalanobis distance. This is likely due to the difficulty in obtaining good estimates of the necessary covariance information on the motion components (*e.g.* by using the Hessian in (2), which is not sufficient), without resorting to expensive sampling techniques [2]. More advanced motion estimation methods and distance metrics are possible avenues of continued research. For example, it may be useful to use an affine motion model or to consider separately the estimated components of motion normal and tangential to the edge's orientation.

3 Classification

Our goal is to label edges as occlusion boundaries or not. We do so by using the posterior probability of the existence of an occlusion boundary given our features, $\Pr(B|f)$, where f may represent the motion difference d, the edge strength e, or both $\{d,e\}$. Given the substantial, scene-dependent variation in the fraction of appearance edges that are also occlusion boundaries, we assume a uniform prior on $\Pr(B)$ and use Bayes' Rule to estimate

[1] This is equivalent to gradually increasing the bandwidth of $h(n)$.

this posterior (note that estimating a prior from the training data was not helpful):

$$\Pr(B|f) = \frac{p(f|B)}{p(f|B) + p(f|\neg B)}. \tag{3}$$

Given training data, we can sample our edge strength and motion difference freatures to estimate the necessary data likelihoods, $p(f|B)$ and $p(f|\neg B)$, as described in the next section. Thresholding this ratio yields the classifier used for our experiments. In the future, it may be possible to achieve better performance by learning adaptive priors for a given image sequence.

4 Experiments

We first need a dataset with labeled occlusion boundaries in order to learn the likelihoods for the classifier. Such a dataset currently does not exist[2]. Thus we have constructed a new dataset for this task, which will also be made available online for other researchers. It contains 30 short image sequences, approximately 8-20 frames in length with the ground truth occlusion boundaries labeled in the reference (*i.e.* middle) frame of each sequence. Some example scenes from this dataset are depicted in Figure 3 with their ground truth occlusion/object boundary labels overlaid. The dataset is quite challenging, with a variety of indoor and outdoor scene types, significant noise and compression artifacts, unconstrained handheld camera motions, and some moving objects. We plan to augment this dataset with further examples in the future.

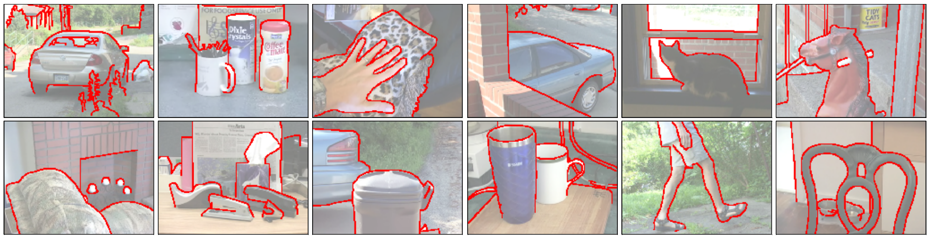

Figure 3: Ground truth occlusion boundaries labeled for 12 of the 30 scenes from our dataset. Each example is the reference (middle) frame of a short sequence, usually 8-20 frames. The images have been lightened for clarity. The scene in Figure 1 is also in the dataset.

For our experiments, we first extract our edge strength feature by applying the Berkeley *Pb* code to the reference frame of each sequence, using all default parameters (*i.e.* those learned from the BSDS training data). Next we align each frame of the sequence to the reference frame using a global translational motion estimate, as suggested in [16]. This stabilization step removes gross camera motions, allowing us to focus on the (potentially small) *relative* patch motions which are most important for our task. In addition, the stabilized sequence better adheres to our constant velocity assumption. Then, as described in Section 2.2, we align small patches ($r = 12$ pixels) on either side of each edge according to the edges' detected orientations (see Figure 2). Using (1) and (2), we estimate the translational motion of each patch separately and compute the Euclidean distance between the two estimates. We use a temporal window radius of $N = 3$ frames and weighting function

[2] The popular Berkeley Segmentation Data Set (BSDS) [9] does not provide image *sequences* necessary for estimating motion, nor do the human-labeled edges necessarily correspond strictly to occlusion boundaries.

bandwidths of $\sigma_h = N$ and $\sigma_w = r$. As shown by the distribution in Figure 4, most relative motions $\mathbf{u}_d$ are quite small, with a mean of 0.14 pixels/frame. This supports our claim that the motion cue available for our task is quite subtle.

4.1 Training

We randomly select half of our dataset to use for training. We first determine the correct label for all detected edge pixels in an image by matching them to occlusion boundary pixels from the ground truth data. Because of localization inaccuracies (in labeling and detection), we use an approach similar in spirit to the one outlined in Appendix B of [10], which seeks to find a one-to-one correspondence between detected edge pixels and nearby hand-labeled boundary pixels. A given training set consists of 15 scenes, yielding a total of approximately $80,000$ individual examples of edge pixels for training. Unfortunately, these examples are taken from contiguous edges and therefore the patches used in generating their appearance and motion cues overlap significantly. Thus they are highly dependent samples, making it inappropriate to use them all for training.

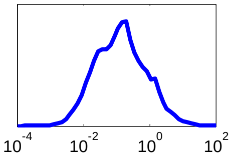

Figure 4: Empirical distribution of relative motions $\mathbf{u}_d$.

To alleviate this problem somewhat, we consider only a random subset of the edges available in the training set. This subset is selected such that no two samples which come from the same image could have utilized overlapping patches of data in estimating motion or computing *Pb*. Thus, for these experiments, we sample edges that are at least $r = 12$ pixels apart. The resulting subset contains approximately 6000 examples, which we use for the training described below. (For testing in Section 4.2, we classify *all* edges detected in a given image.)

Using the edge strength and motion features for all edge pixels corresponding to ground truth occlusion boundaries, we construct kernel density estimates of each cue likelihood independently, $p(e|B)$ and $p(d|B)$, as well as their joint likelihood, $p(e,d|B)$. Similarly, we use any detected edges that are *not* occlusion boundaries as negative examples to learn $p(e|\neg B)$, $p(d|\neg B)$, and $p(e,d|\neg B)$. We use a Gaussian kernel with $\sigma = 1$ bin, and $\pm 3\sigma$ support. For each cue, we use 50 bins (and thus the joint likelihood estimate contains 50×50 bins). In our experience, using a kernel does offer improved results, despite the fairly coarse binning, particularly in terms of generalization from training to test data. To emphasize the importance of distinguishing the very small motion differences (Figure 4), the bins used for estimating the likelihood of the motion-difference cue are logarithmically spaced between 10^{-3} and 10^{2} (where very large motion is indicative of noise or lack of texture). The bins for edge strength are linearly spaced between 0 and 1.

The resulting independent cue likelihoods are shown in Figure 5. As evidenced by the separation of the distributions for each class, these cues do contain some distinct information for our classification task. The distributions also make intuitive sense: higher edge strength and larger motion differences more commonly correspond to occlusion boundaries. It is worth noting that the motion difference cue is fairly weak (*i.e.* the distributions overlap significantly). While improved motion estimation techniques may help, this further supports our claim that the use of optical flow alone for finding occlusion boundaries, as is common practice in segmentation schemes based on motion, could produce poor results on natural scenes which lack texture at many true occlusion boundaries.

The estimated joint likelihoods are shown in Figure 6. We have estimated the full two-dimensional joint distributions $p(e,d|B)$ and $p(e,d|\neg B)$ as well as approximate joint distributions $p(e|B)p(d|B)$ and $p(e|\neg B)p(d|\neg B)$, which assume our two cues are inde-

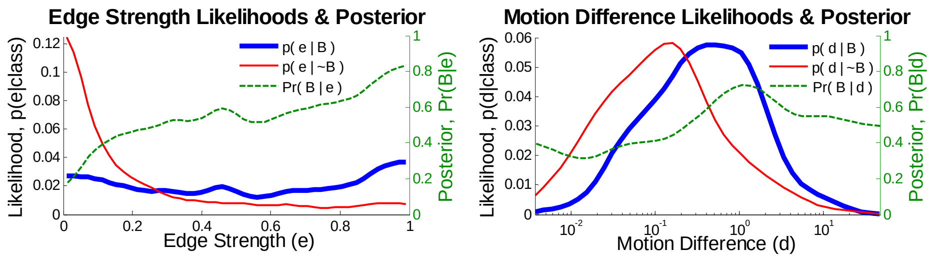

Figure 5: Independent distributions and ratio scores for our two cues.

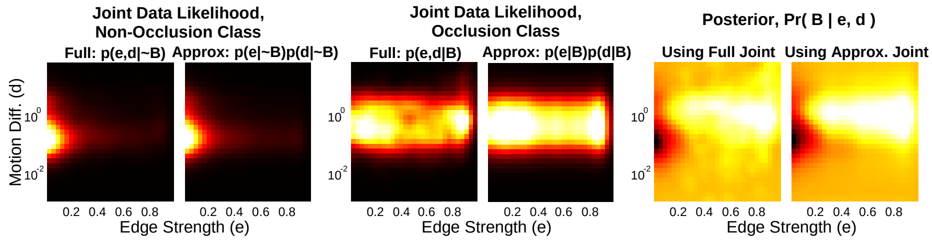

Figure 6: Learned joint likelihood distributions and ratio scores for our two cues. Left and right of each pair shows the result using the full and approximated joint, respectively.

pendent. Given the visually similar estimates, it would appear safe to make such an independence assumption and approximate the joint in this manner. We will test our classifier with both versions below.

Next we compute the posterior probability according to (3). For the separate cues, the result is overlaid on the likelihoods in Figure 5. For the combined cues, the posterior estimates are found in the rightmost pair of Figure 6. Rather than fitting an arbitrary model to the posterior, we have chosen to use the estimates as non-parametric lookup tables.

Finally, we evaluate the learned classifier on the training data itself. After estimating $\Pr(B|f)$ at each edge pixel, we generate Precision vs. Recall curves by varying the threshold on that posterior estimate and counting the number that were correctly labeled. As seen in the left plot of Figure 7, each cue separately provides some information, but the two together perform better, with the full joint providing the best result. The precision levels of these curves also capture a notion of the difficulty of our task and dataset.

We can repeat the entire training process with a different randomly-selected set of sequences for training. Doing so allows us to compute the error bars on the precision recall curves show in Figure 7. These error bars represent plus/minus one standard deviation $(\hat{\sigma})$ for $n = 50$ trials. Thus they indicate the typical distribution of the curves for various divisions of the data. The confidence intervals based on standard errors $(\hat{\sigma}/\sqrt{n})$ are very tight and visually imperceptible from the mean (and thus are not shown). This indicates a statistically significant difference between the mean curves in the plots.

4.2 Testing

For testing, we use the remainder of the dataset, extracting motion and edge strength cues as before. This includes the other half of the scenes, again with approximately $80,000$ examples to be classified. We classify each edge pixel by thresholding the estimated posterior. We can vary this threshold to produce the Precision vs. Recall curves shown in the right plot of Figure 7. Here we see confirmation that the learned classifier can generalize

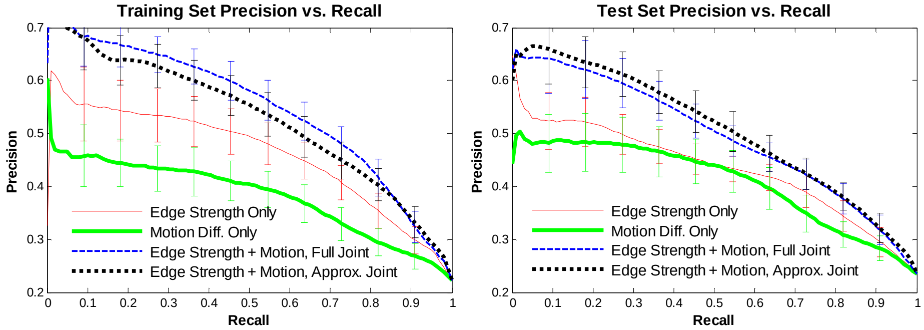

Figure 7: Precision vs. Recall curves for the training and test sets (left and right, respectively), using various combinations of cues. Error bars indicate plus/minus one standard deviation of the curves for 50 randomly selected divisions of the dataset.

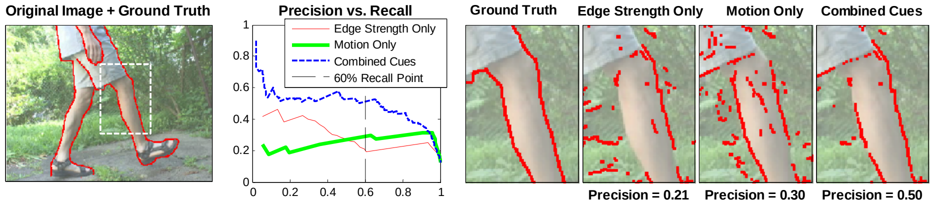

Figure 8: Example classification result at a chosen operating point of 60% recall. Combining appearance and motion cues produces superior precision than either cue alone. Note in the combined result the increased detection on the left of the leg as compared to using edge strength alone, and the decreased spurious detections as compared to using motion alone.

to novel scenes. We see similar performance between the full and approximated joint distributions, with marginal improvement using the approximation. This may indicate that the full joint estimate is slightly overfitting the training data. And once again, by repeating the experiment with different test sets, we can generate the displayed error bars.

Aggregated results as provided in Figure 7 give a general sense of performance, but here we also provide a few anecdotal examples from the dataset to exhibit more concretely the information sometimes hidden in such cumulative comparisons. Figure 8 shows a scene with ground truth overlaid. To illustrate the improvement when using both cues together, we have selected the threshold for each classifier that results in 60% recall, as indicated on the Precision vs. Recall plot. For the indicated window of the original scene, the right four boxes compare the ground truth labeling and the classification results using the cues individually and together. As shown, the best result (with significantly higher precision) is achieved using both cues. For example, combined cues yield improved detection with fewer false positives on the left side of the leg as compared to the result using individual cues alone. Similarly, the examples in Figure 9 demonstrate classification improvement using combined cues.

5 Discussion & Conclusion

Because the performance of any local edge detector is limited, some edges will always be missed. By restricting ourselves to the classification of only the appearance edges

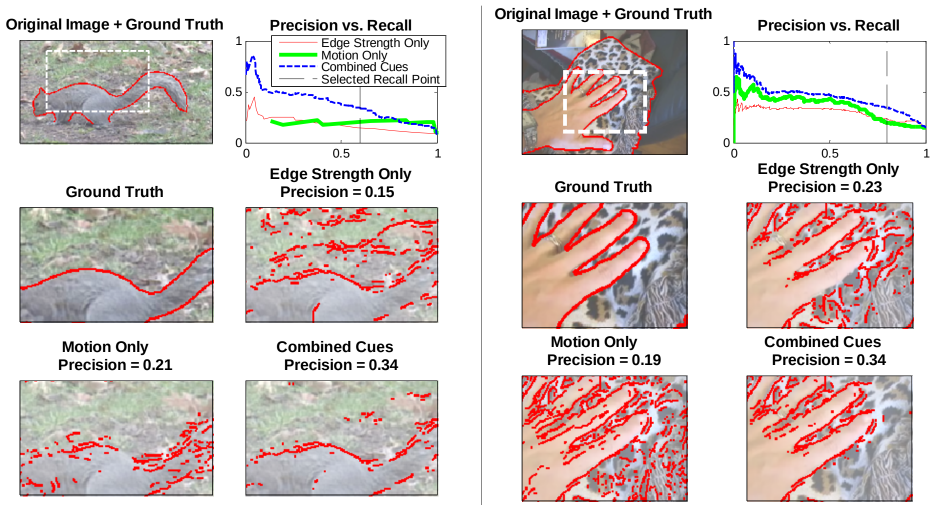

Figure 9: Two additional example classification results. Combining appearance and motion cues produces superior precision at the selected recall operating point than using either cue alone. Note the decreased false positives when using the combined cues.

which *are* detected, we therefore inherit those limitations. As detectors improve (*e.g.* in detecting very weak edges), so too will our approach. For our dataset, however, the edge detector fires with non-zero strength on 83.5% of the ground truth boundaries, indicating that our technique is viable in practice. A complementary approach may include finding motion boundaries *first* and subsequently incorporating appearance reasoning. This may allow the detection of occlusion boundaries visible *only* due to motion, but these cases are relatively rare and such an approach could come at high computational cost.

Local estimates of any kind, including the *Pb* detector and our motion difference feature, are inherently noisy and ambiguous. They are most useful when incorporated into more global reasoning, *e.g.* using a graphical model. Rather than blindly using local estimates for mid- and high-level tasks, however, we believe it is important, if not crucial, to evaluate the utility of these low-level cues themselves (separately and in combination). Having verified here the benefit of using motion, we are currently developing methods of globally reasoning about object/occlusion boundaries and object segmentation which build on the combined local cues described in this work.

Our goal of detecting occlusion boundaries could potentially benefit many computer vision methods, which often rely on spatial aggregation. In this work, we have presented experiments demonstrating anecdotal and quantitative results for two local, low-level feature types useful for future research into globally reasoning about occlusion boundaries. While further investigation into augmenting and strengthening each of our chosen features is warranted, particularly in the better estimation and comparison of local motion, we have demonstrated that considerable improvement in classifying occlusion boundaries is possible when combining these two distinct, individually weaker cues. We have also provided a novel, labeled dataset as an additional resource for future research on occlusion boundary detection.

Acknowledgements

The authors thank David Fleet for his many helpful comments and suggestions.

References

[1] Edward H. Adelson and James 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] Michael J. Black and David J. Fleet. Probabilistic detection and tracking of motion discontinuities. *IJCV*, 38(3):231–245, 2000.

[3] Doron Feldman and DaphnaWeinshall. Motion segmentation using an occlusion detector. In *Worksop on Dynamical Vision at ECCV*, 2006.

[4] David J. Fleet and Yair Weiss. Optical flow estimation. In N. Paragios, Y. Chen, and O. Faugeras, editors, *Mathematical models for Computer Vision: The Handbook*. Springer, 2005.

[5] W. T. Freeman and E. H. Adelson. The design and user of steerable filters. *PAMI*, 13(9):891–906, September 1991.

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

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

[8] Scott Konishi, Alan L. Yuille, James M. Coughlan, and Song Chun Zhu. Statistical edge detection: Learning and evaluating edge cues. *PAMI*, 25(1):57–74, January 2003.

[9] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, pages 416–423, July 2001.

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

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

[12] Abhijit S. Ogale, Cornelia Fermüller, and Yiannis Aloimonos. Motion segmentation using occlusions. *PAMI*, 27(6):988–992, June 2005.

[13] Pietro Perona and Jitendra Malik. Detecting and localizing edges composed of steps, peaks, and roofs. In *ICCV*, pages 52–57, 1990.

[14] Michael G. Ross and Leslie Pack Kaelbling. Learning static object segmentation from motion segmentation. In *AAAI*, 2005.

[15] Mark Ruzon and Carlo Tomasi. Color edge detection with the compass operator. In *CVPR*, pages 160–166, June 1999.

[16] Andrew N. Stein and Martial Hebert. Local detection of occlusion boundaries in video. In *BMVC*, 2006.

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

[18] Regis Vaillant and Olivier D. Faugeras. Using extremal boundaries for 3-D object modeling. *PAMI*, 14(2):157–173, 1992.

[19] Thomas Veit, Frédéric Cao, and Patrick Bouthemy. An a contrario decision framework for region-based motion detection. *IJCV*, 68(2):163–178, June 2006.

[20] Lior Wolf, Xiaolei Huang, Ian Martin, and Dimitris Metaxas. Patch-based texture edges and segmentation. In *ECCV*, 2006.

[21] Jiangjian Xiao, Hui Cheng, Harpreet Sawhney, Cen Rao, and Michael Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *ECCV*, volume I, pages 211–224, 2006.