Occlusion Boundary Detection and Figure/Ground Assignment from Optical Flow

Patrik Sundberg[1], Thomas Brox[2], Michael Maire[3], Pablo Arbeláez[1], and Jitendra Malik[1] *

[1]University of California at Berkeley
{sundberg,arbelaez,malik}@eecs.berkeley.edu

[2]University of Freiburg, Germany
brox@informatik.uni-freiburg.de

[3]California Institute of Technology
mmaire@caltech.edu

Abstract

In this work, we propose a contour and region detector for video data that exploits motion cues and distinguishes occlusion boundaries from internal boundaries based on optical flow. This detector outperforms the state-of-the-art on the benchmark of Stein and Hebert [24], improving average precision from .58 to .72. Moreover, the optical flow on and near occlusion boundaries allows us to assign a depth ordering to the adjacent regions. To evaluate performance on this edge-based figure/ground labeling task, we introduce a new video dataset that we believe will support further research in the field by allowing quantitative comparison of computational models for occlusion boundary detection, depth ordering and segmentation in video sequences.

1. Introduction

Vision systems make use of optical flow for a number of purposes, such as egomotion estimation and scene structure recovery, the latter including both metric depth estimates and ordinal relationships like figure/ground. In this paper, we focus particularly on the role of motion for grouping and figure/ground assignments. The importance of motion cues in these tasks is a classic point in the psychophysical literature. Koffka stated the Gestalt principle of "common fate" where similarly moving points are perceived as coherent entities [15], and grouping based on motion was emphasized by numerous other works including Gibson [12], who also pointed out occlusion/disocclusion phenomena. In contrast to color and stereopsis, which also help to separate different objects, motion is a cue shared by basically all visual species - a fact that emphasizes its importance in biological vision systems.

*This work was supported by the German Academic Exchange Service (DAAD) and ONR MURI N00014-06-1-0734.

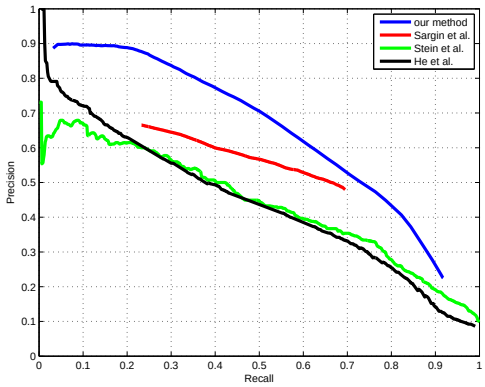

Figure 1. **Occlusion boundary detection benchmark.** Precision-recall curves for the occlusion boundary detection task reported by Stein and Hebert [24]. We show results reported by [24], Sargin et al. [21], He et al. [14] as well as our own results. For performance numbers, see Table 1.

The algorithm presented in this paper combines static boundary cues produced by the boundary detector from [17] with low-level motion cues and motion cues derived from optical flow [6]. For each point on the boundaries produced by the static detector, we compute the motion difference δ between the two regions adjacent to the boundary. The computation of δ involves both spatial and temporal aggregation of the optical flow. We then combine the δ feature with the static detector output into the final contour classifier f, which can be turned into a set of closed regions. In addition, by comparing the optical flow on boundary points with that of regions adjacent to the boundary, we can assign figure/ground labels to edges. Figure 2 illustrates this procedure. Compared to previous techniques, our approach is very simple and transparent. Hence, we find it particularly remarkable and rewarding that we perform so much better than previous, far more complex methodologies.

Figure 2. **Overview. Top:** Keyframe from the rocking horse sequence of [24], along with ground truth and the output of the static boundary detector of [2]. **Bottom Left:** Our motion feature δ, computed on boundaries. **Bottom Center:** Resulting boundaries combining δ and static cues. **Bottom Right:** Figure/ground classification results on detected regions, figure in green.

The present paper makes three distinct contributions. First, we extend the current state-of-the-art static boundary detector from [17] to exploit motion cues in videos, using both low-level motion cues and optical flow. We evaluate our algorithm on the current leading benchmark introduced in [24], and show that it improves the state of the art as measured by average precision from .58 to .72, as shown in Figure 1. This is a very significant performance improvement despite our very clean and transparent approach. Second, we introduce a new, larger, and more difficult dataset for occlusion boundary detection and figure/ground assignment. The dataset contains 102 HD video sequences with segmentations and depth ordering labels, separated into training and test sets. This dataset can be regarded as a video counterpart to the Berkeley Segmentation dataset on static images [19]. Third, we show how to assign figure/ground labels to detected occlusion boundaries based only on the motion of edges and their adjacent regions, getting 84% accuracy on the dataset from [24] and 69% on our new, more challenging dataset.

2. Previous Work

In the computer vision literature, many works have dealt with the problem of optical flow estimation over the past three decades, and there have been numerous approaches to make use of the optical flow field for grouping [10, 22, 23, 28, 9]. Most of them are similar to the work of Wang and Adelson, which proposes to partition the image into motion layers by clustering similar optical flow vectors according to a parametric motion model [26]. While this approach is attractive in the sense that it directly provides object regions, there are many cases that are not properly captured by parametric models. To deal with this shortcoming, [27] suggested a nonparametric version of layered motion, where each layer is described by a smooth flow field. Similar techniques based on level sets have been presented in [1, 7]. A shortcoming of these nonparametric layer models is the susceptibility of the EM procedure to local minima, particularly in areas of the image with little structure.

An alternative strategy is to detect occlusion edges and to infer layers from these edges afterwards, if needed. Such a strategy has been shown to work very well for segmentation of static images [2], and it makes even more sense for grouping based on motion cues, where additional difficulties due to the aperture problem limit the reliability of typical EM style procedures.

Only a few works have dealt with explicitly detecting occlusion boundaries based on motion cues and assigning figure/ground labels to both sides of these boundaries. [25] uses a low-level edge detector combined with optical flow information to detect occlusion edges, using the result to improve the overall quality of the optical flow field. In [4], initial motion boundaries are obtained from a motion estimator and then serve a probabilistic state model that can distinguish occlusion boundaries from internal boundaries and assign figure/ground labels to the regions on both sides

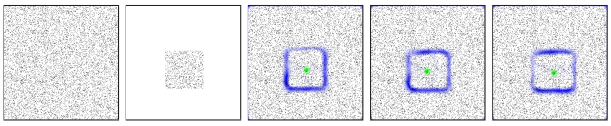

Figure 3. **Boundary detection in the absence of static cues.** Consider a textured background (left) and an identically textured foreground object (center-left) moving in front of it. Three frames of this motion sequence are shown (center to right). Humans easily detect the moving square from the video sequence despite the uniform appearance of any single frame. Our motion gradient cue (shown in blue) successfully detects the object boundary. Ground truth locations of the object center are marked in green.

of the edge. A particle filter approach is employed to deal with the complex, multi-modal distributions in the high-dimensional state space. [8] uses local motion information only to infer boundaries and direction-of-figure. In [11], tensor voting yields optical flow estimates together with an uncertainty measure based on the homogeneity of the votes. Occlusion boundaries are assumed to be maxima in the uncertainty measure. Neither [4], [8] nor [11] make use of static cues. Since the optical flow as a secondary feature requires integration over a spatial domain to deal with the aperture problem, edges based only on motion estimates are usually inaccurate and dislocated.

In contrast, [23] presents a method that relies only on static edges and their motion. Edges are computed by the Canny edge detector and an affine motion model for edge fragments is computed via block matching. The authors also provide a depth ordering of both sides of an edge by reasoning at t-junctions. The work of Stein and Hebert [24] also makes use of a static edge detector from [18] to obtain an initial set of potential occlusion boundaries. They then learn a classifier that distinguishes occlusion boundaries from internal edges based on both static and motion features. [21] shows improved performance on the same dataset using a probabilistic detection framework defined on spatio-temporal lattices, using joint analysis of multiple image frames. [14] presents an alternate approach based on pseudo-depth which performs roughly as well as [24]. We show in this paper that such an approach is less powerful that one which uses good motion cues directly.

3. Motion Cues for Boundary Detection

Edge detection was one of the earliest problems addressed in computer vision, yet current state-of-the-art boundary detectors [18, 17] are designed for analyzing single images and rely only on static cues such as brightness, color, and texture. When working with video, we have the additional cue of motion and would like to take advantage of it at all levels of processing involved in figure/ground labeling, starting with the boundary detection task itself.

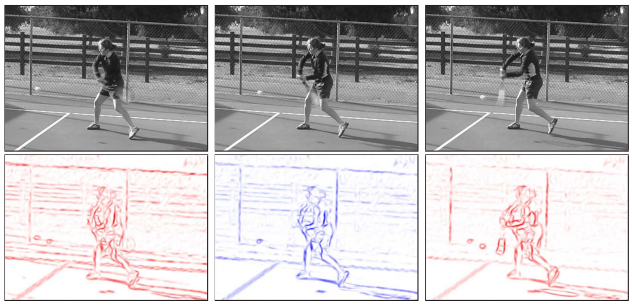

Figure 4. **Motion gradient. Top:** Three consecutive frames from video of a tennis player. The large frame to frame displacement of the tennis ball, tennis racket, and person's arms and legs make this a challenging sequence. **Bottom Left and Right (red):** Gradient operator applied to the temporal differences between the center frame and those immediately before and after. We show $MG^-(x,y) = \max_\theta\{MG^-(x,y,\theta)\}$ and $MG^+(x,y) = \max_\theta\{MG^+(x,y,\theta)\}$, respectively. Both MG^- and MG^+ contain double images of the moving boundaries. **Bottom Center (blue):** A motion boundary in the current frame should be detected as differing from both the previous frame and subsequent frame. We compute MG using the geometric mean of MG^- and MG^+ as a soft "and" operation and display $MG(x,y) = \max_\theta\{MG(x,y,\theta)\}$. Double boundaries are eliminated and those surviving in MG are correctly aligned with the current frame.

Figure 3 illustrates the extreme case of a uniformly textured object moving in front of a background with the same uniform texture. Here, there are no static cues, yet humans easily perceive the motion of the foreground object when viewing the video sequence. We compute a new cue, which we refer to as the motion gradient, that correctly detects the boundary of this moving object in each frame.

Our motion gradient signal can be thought of as a temporal analog to the brightness, color, and texture gradients used in the boundary detectors of [18] and [17]. To compute the motion gradient for grayscale video frame I_t, we first compute temporal derivatives with respect to the previous and subsequent frames: $D^- = I_t - I_{t-l}$ and $D^+ = I_t - I_{t+l}$. Then, we consider the gradient operator $G_r(x,y,\theta)$ defined in [18] which measures the χ^2 difference of the histograms of values in the two halves of a radius r disc centered at (x,y) and divided by a diameter at angle θ. We sample θ for 8 orientations in the interval $[0,\pi]$ and apply this gradient operator to D^- and D^+ to produce motion gradients $MG^-(x,y,\theta)$ and $MG^+(x,y,\theta)$, respectively.

Gradients MG^- and MG^+ both contain double responses to moving boundaries as there is a large temporal difference between any two consecutive frames at both the old and new locations of a boundary. However, these double responses occur at different spatial locations in MG^- and MG^+. Each moving edge is detected at three spatial locations, two of which appear in MG^- and two of which

appear in MG^+. The detection common to both MG^- and MG^+ is the true location of the edge at time t. By taking the geometric mean of these signals we suppress the spurious responses while preserving the correct one, resulting in the motion gradient

$$MG(x, y, \theta) = \sqrt{MG^-(x, y, \theta) \cdot MG^+(x, y, \theta)} \quad (1)$$

Figure 4 shows an example of the motion gradient applied to a real video sequence. Note that it is not our intention to detect occlusion boundaries at this stage. Rather, the motion gradient is designed to respond strongly at any edge which moves in the image. For this reason, MG picks up the surface markings on the tennis court. At the same time, it serves to enhance the actual occlusion boundaries on the person and provides robustness against cases in which static cues are weak or nonexistent (Figure 3). The main concern is to utilize the motion gradient to improve the quality and reliability of the boundary map upon which our occlusion reasoning machinery will depend.

Following the framework of [17], our boundary detector adds the motion gradient as an additional channel alongside the static brightness, color, and texture cues from [18], and passes these local cues through the same sequence of steps for combining them with the result of a spectral partitioning process. Adopting the notion from [17], we use gPb (global probability of boundary) to refer to contours created using only static cues and gPb+mg to indicate those created from both the static cues and our motion gradient.

We then utilize the machinery of [2] to transform the output of our boundary detector into an Ultrametric Contour Map (UCM). The UCM is a weighted contour image that can be thresholded at any level to produce a set of closed curves. The corresponding sets of regions bounded by these contours define a hierarchical segmentation. This data structure determines the edge fragments and associated regions that we classify in occlusion reasoning and figure/ground stages.

4. Occlusion Boundary Detection

In contrast to other recent work [24, 21], our occlusion reasoning stage utilizes a straightforward classification technique whose power lies in its ability to exploit two key components: the boundary detector discussed in the previous section, and the variational optical flow method of [6]. Given both a reliable set of prior boundaries gPb+mg and a reliable, dense optical flow field $(u, v)^\top := \mathbf{w} : (\Omega \subset \mathbb{R}^2) \rightarrow \mathbb{R}^2$, we are able to judge whether an edge is an occlusion boundary by looking at the difference between the flows in the regions on either side.

We begin by estimating motions of the two adjacent regions at each point on every edge fragment in the UCM. We need to ensure that the estimate is not polluted by the motion of the edge itself, or by the motion of the opposing region. In order to do this, we define a weighted filter $w_i(x, y)$ for each edge point $\mathbf{x}_i = (x_i, y_i)$ and region R neighboring $\mathbf{x}_i$ as follows:

$$w_i^R(x, y) = \frac{1}{Z} \exp^{-\frac{(x-x_i)^2+(y-y_i)^2}{2\sigma_w^2}} \delta(r(x, y), R) i_d(x, y) \quad (2)$$

where δ is the Kronecker delta function, $r(x, y)$ is the region assignment for pixel location (x, y), and $i_d(x, y)$ is 1 for pixels internal to any region, defined as being at least d pixels away from any region boundary. In this work, we use $d = 2$. While it is reasonable to vary σ_w, combining windowing functions of different scales, in our experiments we did not observe a significant change when doing so. This is most likely due to the optical flow method from [6] already considering multiple scales. Experiments reported in this paper all use $\sigma_w = 3.5$, and a circular window of radius $r = 2\sigma_w = 7$ pixels.

Given the weights w_i, we use weighted least squares to fit a plane to the u and v flow components as functions of x and y, minimizing

$$e_u^R = \sum_{(x,y)} w_i^R(x, y)(A_u^R x + B_u^R y + C_u^R - u(x, y))^2 \quad (3)$$

$$e_v^R = \sum_{(x,y)} w_i^R(x, y)(A_v^R x + B_v^R y + C_v^R - v(x, y))^2 \quad (4)$$

and describe the flow of a region R by an affine model:

$$(u_i^R(x, y), v_i^R(x, y)) = (A_u^R x + B_u^R y + C_u^R, A_v^R x + B_v^R y + C_v^R). \quad (5)$$

As a final step, we evaluate these two flows at the edge point itself, computing the flow vectors $\mathbf{w}_i^+ := (u_i^+, v_i^+)$ and $\mathbf{w}_i^- := (u_i^-, v_i^-)$. This step is critical, since we do not measure the region flow at any points closer than d pixels away from the boundary, and many common camera and object motions lead to flow vectors which vary spatially.

Intuitively, when the optical flow in two regions is significantly different, the edge between them is likely to be an occlusion boundary. To capture this notion, we define

$$\delta = |(u_i^+ - u_i^-, v_i^+ - v_i^-)|. \quad (6)$$

In addition, we also measure the value of gPb at the edge point, and combine the two using a simple linear classifier, with the single weight trained using SVM training:

$$f = \rho * \delta + (1 - \rho) * gPb \quad (7)$$

In this paper, we use $\rho = 0.7$. To produce precision/recall curves, we simply vary the cutoff threshold used for f.

4.1. Temporal Scale

In most cases, we have multiple frames of video to assist in deciding whether or not an edge is an occlusion boundary. Intuitively, we'd expect that using more frames would

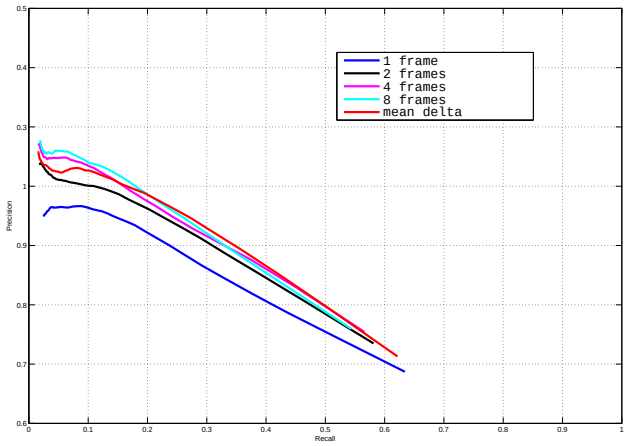

Figure 5. **Effect of temporal scale.** Using multiple frames when computing the optical flow based feature delta helps, but only to a point. If using too many frames, performance degrades due to changes in appearance and errors in the optical flow field. The best performance is achieved by combining results across temporal scales.

allow us to make better decisions, especially when the motions involved are small. On the other hand, errors in the optical flow and changes in object appearance and motion may become significant at large temporal scales. Figure 5 shows the performance on our dataset for flows spanning 1, 2, 4 and 8 frames in the forward and backward direction from a keyframe. By averaging results across temporal scales, we achieve the best performance. Training a classifier using logistic regression did not improve performance on top of a simple linear average.

It should be noted that normalization of the optical flow is quite important. We normalize all flows by their variance across all pixels in the frame. Note that this normalization scheme accounts for changes in image scale and sample rate, while keeping our feature δ invariant to image translations.

5. Region Detection

In order to produce closed regions from the output of our occlusion boundary classifier, we adapt to our framework the method of [2], that converts any oriented contour signal into a hierarchy of regions. That method considers an arbitrary contour detector, whose output $E(x, y, \theta)$ predicts the probability of an image boundary at location (x, y) and orientation θ. By applying watersheds on the maximal response of E over orientations, an over-segmentation of the image is initially computed, and each point in the watershed arcs is then weighted by the value of the contour signal in the orientation of the arc. In our case, the value of the contour signal is given by a linear combination of the static cue gPb and the motion classifier output: $c = \alpha \cdot gPb + (1-\alpha)f$. In all the experiments reported, we use a fixed value of $\alpha = 0.2$. Notice that one could use the raw output of δ rather than our combined classifier f. However, we empirically found the estimation of the boundary strength given by f to be more reliable.

6. Figure/Ground Assignment

Once we know that a moving edge is an occlusion edge, we can determine which side of the edge is the occluding foreground object and which side belongs to the background, thereby obtaining a depth ordering of the attached regions. There have been previous attempts on assigning figure/ground labels to both sides of an edge based on static cues [20, 13], which is a difficult task with many ambiguities. Motion cues provide rich information to decide upon figure/ground. Obviously, an occluding edge moves the same way as the occluding region next to it. Thus, a decision can be inferred from the motion of the two regions as well as the motion of the occluding edge.

In the previous section, we discussed how the motion of a region can be obtained from an optical flow field without significant interference from edge motion. Thus, all we need here for the figure/ground assignment is the motion directly on the edge. [23] and [16] both show how to compute the motion of an edge. In particular [16] takes some effort to estimate the edge motion from edge cues only. Moreover, they deal with special cases such as illusory boundaries. Here, we are interested in a methodology that is most consistent with the way we estimated the motion of the regions, in order for the comparison to be fair. Thus, it is natural to simply consider the optical flow on the edge in order to obtain the edge motion (u_i^0, v_i^0) at each edge pixel i. Since the variational optical flow is most accurate on edges, this is a reasonable thing to do.

For each edge fragment we therefore compute two distances, between the motion of the edge and each of the two neighbouring regions. Again calling the two regions $+$ and $-$, we compute

$$\delta^+ = \frac{1}{n}\sum_i^n \sqrt{(u_i^+ - u_i^0)^2 + (v_i^+ - v_i^0)^2} \quad (8)$$

$$\delta^- = \frac{1}{n}\sum_i^n \sqrt{(u_i^- - u_i^0)^2 + (v_i^- - v_i^0)^2} \quad (9)$$

where n is the number of edge pixels on this edge fragment. Finally, we assign the edge to the region with the smaller distance.

7. Experimental Results

7.1. Datasets

We tested our occlusion boundary detector on the dataset introduced by Stein and Hebert [24], consisting of 30 se-

Figure 6. **Our dataset.** Keyframe and segment annotation pairs. Depth ordering is also available, but not shown here.

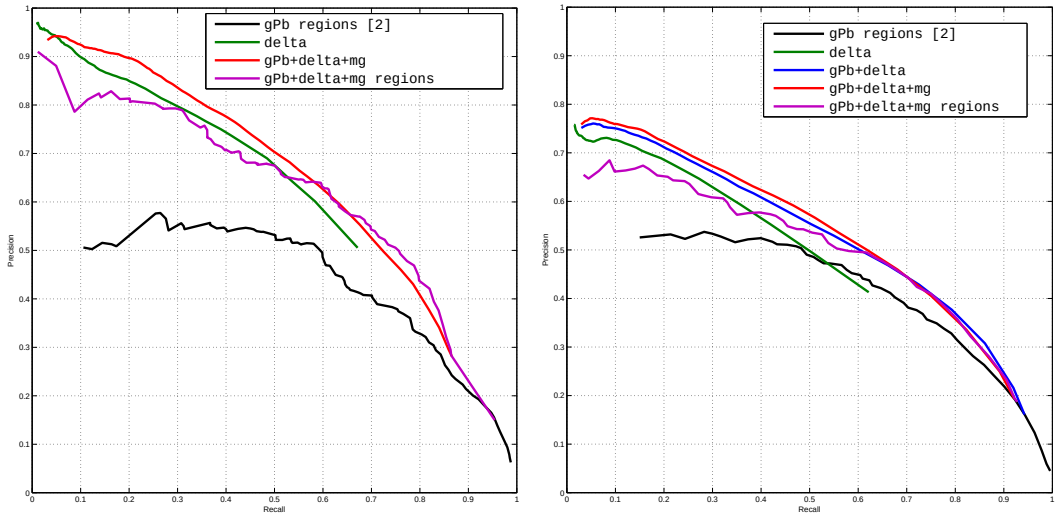

Figure 7. **Benchmark performance.** Precision-recall curves for the baseline algorithm from [2], using only static cues, as well as our algorithm as a boundary detector using δ, $gPb + \delta$ and $gPb + \delta + mg$, and the closed region version of our algorithm. Both our motion gradient cue mg and optical flow feature δ contribute to improving performance. **Left:** Results on the CMU dataset. **Right:** Results on our dataset.

quences. However, we found this dataset to be somewhat limiting. The resolution and visual quality of many sequences is suboptimal, many scenes are somewhat artificial, and the number of frames per sequence can be as low as 5. There is also no official split into training and test, and the number of sequences is too low. To address these issues, and to help set a new gold standard for evaluating figure/ground assignment on video, we created a new dataset consisting of 102 challenging image sequences exhibiting a large variety of scenes and actions. For the experiments presented, this dataset was divided into 42 train and 60 test sequences. One keyframe in each sequence was manually labeled with occlusion boundaries, object instance labels and region depth ordering. We intend to extend and release our dataset to promote the quantitative evaluation of occlusion boundary detection, segmentation and figure / ground assignment in video data. An example of images and annotations can be seen in Figure 6.

7.2. Occlusion Boundary Detection

Figure 7 presents the results on occlusion boundary detection, separating the various components of the algorithm, on both the CMU dataset [24] and our own. On both datasets, our occlusion boundary detectors provide a large improvement in precision when compared with the method of [2], which is currently the state of the art on static images. These results demonstrate empirically the strength of motion as a cue for perceptual grouping. Qualitative results can be observed in Figure 8. Figure 1 shows our results on the CMU dataset compared to existing approaches. Despite its simplicity, our technique significantly outperforms the state of the art on this dataset.

7.3. Figure/Ground

Table 2 shows the performance of the figure / ground algorithm on the datasets used in this paper, and Figure 9

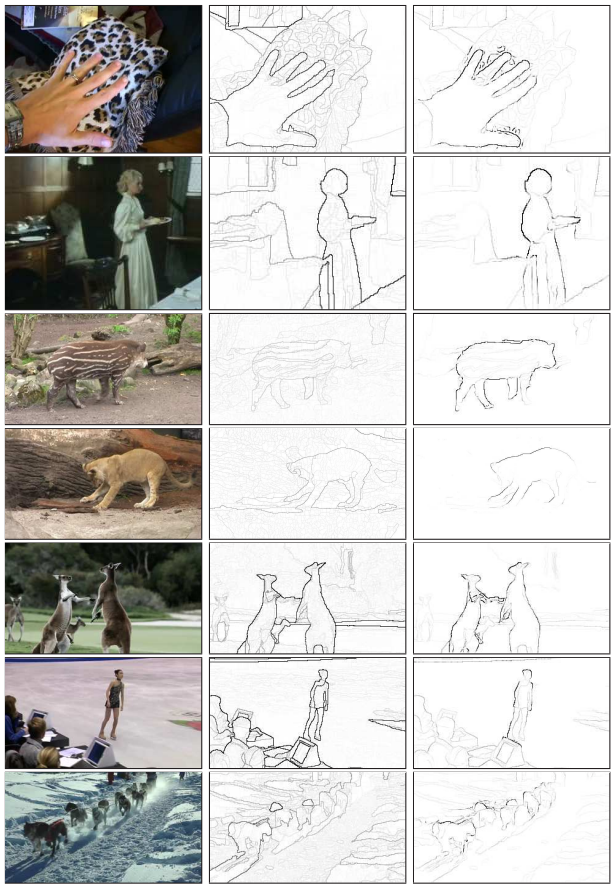

Figure 8. **Occlusion boundary detection results. From left to right:** Keyframe from the video sequence, boundary results using gPb+mg only, and boundary results using gPb+mg+δ.

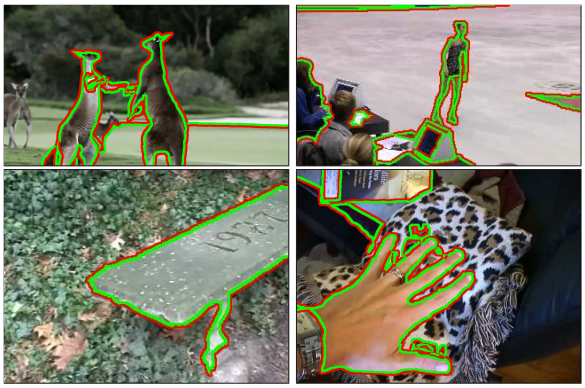

Figure 9. **Figure/ground classification results.** We show figure/ground classification results for detected occlusion boundaries with strength above 0.3 for a subset of frames from Figure 8. Green and red markings indicate the figure and ground sides of each boundary, respectively.

shows results on some keyframes. The results are far above the chance level of 0.5, especially on the CMU dataset.

	F-measure	AP
Stein et al.	.48	.47
Sargin et al.	.57	.58
He et al.	.47	.43
Our contours	**.61**	**.72**
Our regions	**.62**	**.69**

Table 1. **Results on the CMU benchmark.** We report maximal F-measure and average precision for our methods and those of [24], [21] and [14]. Average precision is computed only in the recall interval [.25,.70], since the method of [21] does not produce results outside this interval.

Dataset	per-pixel	aggregated
CMU	65.7 %	83.8 %
ours	63.0 %	68.6 %

Table 2. **Figure/Ground classification results.** Results on the figure / ground classifier as evaluated on ground truth boundaries. We benchmark two algorithms per dataset. In the first, each pixel independently choses which side is figure. In the other, results for all pixels in each edge fragment are combined, and the entire edge fragment choses which side is figure, as described earlier.

8. Conclusions

In this paper we presented a framework for detecting occlusion boundaries and assigning figure/ground labels to both sides of those boundaries. To this end, we extended the state-of-the-art boundary detector from [17] designed for static images to make use of basic motion cues in video. Our experiments on occlusion boundary detection clearly show the improvement obtained due to this extension. We also provide empirical evidence showing that optical flow and these motion cues provide largely complementary information.

Our final results also clearly outperform all previous works on occlusion boundary detection. This shows that a good boundary detector and a reliable optical flow field are powerful features for success in this task. The optical flow further allows for assigning the edge to one of the two attached regions, in a natural and intuitive way. In addition to demonstrating the power of our approach on the existing standard dataset, we have also collected a new, larger, more diverse and challenging dataset to help advance the state of the art in the field further. Our results on this dataset provide an excellent baseline for future work.

We think that occlusion boundaries and the knowledge about which side is figure are a very promising way to extract objects from video in an unsupervised fashion. Unsupervised segmentation from static images is not currently possible, but motion is a very reliable cue for this task. Fully

utilizing that cue would make learning of models for visual recognition significantly easier and more natural.

References

[1] T. Amiaz and N. Kiryati. Piecewise-smooth dense optical flow via level sets. *International Journal of Computer Vision*, 68(2):111–124, 2006. 2

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: an empirical evaluation. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2009. 2, 4, 5, 6

[3] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *Proc. International Conference on Computer Vision*, 2007.

[4] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion boundaries. *International Journal of Computer Vision*, 38(3):231–245, 2000. 2, 3

[5] G. J. Brostow, J. Shotton, J. Fauqueur and R. Cipolla, Segmentation and Recognition Using Structure from Motion Point Clouds, In *Proc. European Conference on Computer Vision*, pages 44–57. 2008.

[6] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3024 of *LNCS*, pages 25–36. Springer, May 2004. 1, 4

[7] T. Brox, A. Bruhn, and J. Weickert. Variational motion segmentation with level sets. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Proc. European Conference on Computer Vision*, volume 3951 of *LNCS*, pages 471–483. Springer, 2006. 2

[8] G. T. Chou. A model of figure-ground segregation from kinetic occlusion. In *Proc. International Conference on Computer Vision*, 1995. 3

[9] D. Cremers and S. Soatto. Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005. 2

[10] T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):474–487, 1995. 2

[11] L. Gaucher and G. Medioni. Accurate motion flow estimation with discontinuities. In *Proc. International Conference on Computer Vision*, pages 695–702, 1999. 3

[12] J. J. Gibson The perception of the visual world. Houghton Mifflin, 1950. 1

[13] D. Hoiem, A. Stein, A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. In *Proc. International Conference on Computer Vision*, 2007. 5

[14] X. He, and A. Yuille Occlusion Boundary Detection Using Pseudo-depth. In *Proc. European Conference on Computer Vision*, 2010. 1, 3, 7

[15] K. Koffka. *Principles of Gestalt Psychology*. Hartcourt Brace Jovanovich, New York, 1935. 1

[16] C. Liu, W. T. Freeman, and E. H. Adelson. Analysis of contour motions. In *Proc. Neural Information Processing Systems*, 2006. 5

[17] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2008. 1, 2, 3, 4, 7

[18] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004. 3, 4

[19] 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 *Proc. International Conference on Computer Vision*, pages 416–425, 2001. 2

[20] X. Ren, C. Fowlkes, and J. Malik. Figure/ground assignment in natural images. In *Proc. European Conference on Computer Vision*, volume 3952 of *LNCS*, pages 614–627. Springer, 2006. 5

[21] M. E. Sargin, L. Bertelli, B. S. Manjunath and K. Rose. Probabilistic Occlusion Boundary Detection on Spatio-Temporal Lattices. In *Proc. International Conference on Computer Vision*, 2009. 1, 3, 4, 7

[22] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proc. 6th International Conference on Computer Vision*, pages 1154–1160, Bombay, India, Jan. 1998. 2

[23] P. Smith, T. Drummond, and R. Cipolla. Layered motion segmentation and depth ordering by tracking edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):479–494, 2004. 2, 3, 5

[24] A. N. Stein and M. Hebert. Occlusion boundaries from motion: low-level detection and mid-level reasoning. *International Journal of Computer Vision*, 82(3):325–357, 2009. 1, 2, 3, 4, 5, 6, 7

[25] W. B. Thompson. Exploiting Discontinuities in Optical Flow. *International Journal of Computer Vision*, 30(3):163–173, 1998. 2

[26] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, 1994. 2

[27] Y. Weiss. Smoothness in layers: motion segmentation using nonparametric mixture estimation. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 520–527, 1997. 2

[28] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1644–1659, 2005. 2