Occlusion Boundary Detection using Pseudo-Depth

Xuming He and Alan Yuille

Department of Statistics, UCLA,
8145 Math Science Building, Los Angeles, CA, USA
{hexm,yuille}@stat.ucla.edu

Abstract. We address the problem of detecting occlusion boundaries from motion sequences, which is important for motion segmentation, estimating depth order, and related tasks. Previous work by Stein and Hebert has addressed this problem and obtained good results on a benchmarked dataset using two-dimensional image cues, motion estimation, and a *global boundary model* [1]. In this paper we describe a method for detecting occlusion boundaries which uses depth cues and local segmentation cues. More specifically, we show that crude scaled estimates of depth, which we call *pseudo-depth*, can be extracted from motion sequences containing a small number of image frames using standard SVD factorization methods followed by weak smoothing using a Markov Random Field defined over super-pixels. We then train a classifier for occlusion boundaries using pseudo-depth and local static boundary cues (adding motion cues only gives slightly better results). We evaluate performance on Stein and Hebert's dataset and obtain results of similar average quality which are better in the low recall/high precision range. Note that our cues and methods are different from [1] – in particular we did not use their sophisticated global boundary model – and so we conjecture that a unified approach would yield even better results.

Key words: Occlusion boundary detection, Depth cue, Markov Random Field

1 Introduction

Occlusion boundary detection, which detects object boundaries that occludes background in motion sequences, is important for motion segmentation, depth order estimation, and related tasks. For example, although there has been much recent progress in estimating dense motion flow [2–5] the estimation errors typically occur at the boundaries. Recently Stein and Hebert [1] developed a method for occlusion boundary detection using machine learning methods which combines two-dimensional image cues, motion estimation, and a sophisticated global boundary model. Their method gave good quality results when evaluated on a benchmarked database.

In this paper we argue that occlusion boundaries often occur at depth discontinuities and so depth cues can be used to detect them. We use a motion

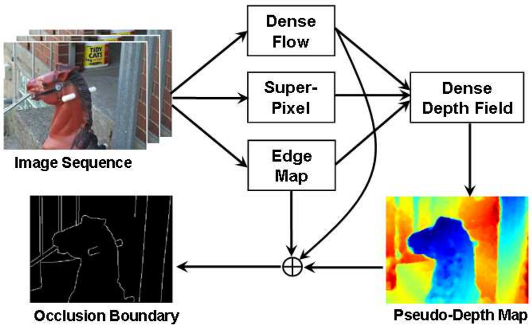

Fig. 1. Overview of our approach. Step 1 computes the dense motion flow. Step 2 estimates the pseudo-depth using SVD and weak smoothness. Step 3 trains a classifier for detecting occlusion boundaries in terms of the local edge map, the motion-flow, and the pseudo-depth. Best seen in color.

estimation algorithm to find the correspondence between different image frames and hence estimate crude scaled estimates of depth, which we call *pseudo-depth*. The discontinuities in pseudo-depth typically occur at occlusion boundaries and so provide detection cues which can be combined with local image segmentation cues. We note that the relationship of occlusion boundaries to depth has long been realized in the binocular stereo community [6, 7] and indeed earlier work has described how it can apply to motion sequences (e.g., [8]). More recently, some motion estimation algorithms [5] do introduce some depth knowledge in an implicit form of motion smoothing.

In this paper, see Figure (1), we proceed in the following steps. *Step 1:* estimate the dense motion flow from the input image sequence. We perform this estimation using a novel algorithm (submitted elsewhere) but other motion flow algorithms that perform well on the Middlebury dataset [2] would probably be sufficient. *Step 2:* estimate *pseudo-depth* by the Singular-Value-Decomposition (SVD) technique [9, 10] from the motion flow. We call this pseudo-depth since it is: (a) very noisy, (b) only known up to a scaling factor, and (c) only valid as depth if there is a single rigid motion in the image. We perform weak smoothing of the depth using a Markov Random Field (MRF) defined over super-pixels [11] (extending a method reported in [12]). We observe, see figure (1), that the pseudo-depth captures the rough depth structure and, in particular, tends to have discontinuities at occlusion boundaries. *Step 3:* train a classifier for occlusion boundaries which takes as input the motion flow, the local edge map, and the pseudo-depth map. In practice, we obtain good results using only the local edge map and the pseudo-depth map.

The contribution of this paper is to show that we can obtain results comparable to Stein and Hebert's [1] using only pseudo-depth cues and static edge cues (i.e. the Berkeley edge detector [13]). We describe the background material in section (2) and how we estimate motion flow in section (3). Section (4) describes how we estimate pseudo-depth which is our main cue for occlusion boundary detection. Section (5) describes how we train a classifier to detect occlusion boundaries using pseudo-depth, static edge cues, and motion cues. Section (6) shows that our classifier achieves state of the art results, based on pseudo-depth and static edge cues alone, and gives comparisons to the results in [1].

2 Background

There is an enormous computer vision literature on motion estimation that can mostly be traced back to the classic work of Horn and Schunk [14]. Most of them uses a measurement term based on the optical flow constraint and smoothness terms on the velocity field to resolve the ambiguities in local measurement. The effectiveness and efficiency of algorithms for estimating velocity were improved by the use of coarse-to-fine multi-scale techniques [3] and by the introduction of robust smoothness [15] to improve performance at velocity boundaries.

Earlier researchers have discussed how motion and depth cues could be combined to detect surface discontinuities (e.g., [8]) but their technique relies on pixel-level MRF and line processes. The importance of occlusion boundaries has been realized in the binocular stereo community [7, 6]. In visual motion analysis, many efforts have been made to address the problem of modeling motion boundaries in motion estimation (see [16, 17] and reference therein). Some work (e.g., [18]) has attempted to estimate motion boundaries using image segmentation and explicitly modeling regions that appear or disappear due to occlusion, but they have not been systematically evaluated on benchmark datasets. Stein and Hebert [1] proposed methods for detecting occlusion boundaries but do not use explicit depth cues.

There is an extensive literature on methods to estimate depth from sequences of images [19]. This is a highly active area and there has been recent successful work on estimating three-dimensional structure from sets of photographs [20], or dense depth estimation from optical flows [21–23]. In this paper, our goal is only to obtain rough dense depth estimates from motion so we rely on fairly simple techniques such as the SVD factorization method [9, 10] and a scaled orthographic camera model instead of the more sophisticated techniques and camera models described in those approaches.

There has also been recent work on estimating depth from single images [12, 24, 25], some of which explicitly address occlusion [25]. There seems to be no direct way to compare our results to theirs. But we adapt techniques used in these papers, for example performing a pre-segmentation of the image into superpixels [11] and then smoothing the depth field by defining a Markov Random Field (MRF) on superpixels [12].

3 Step 1: Motion Flow Estimation

We compute motion flow between image sequences using our own motion flow algorithm (submitted elsewhere and to be publicly available). But the rest of

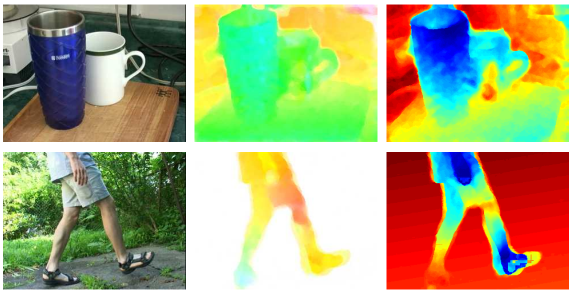

Fig. 2. Two examples of dense flow estimation (based on Middlebury flow-color coding, middle panel) and dense pseudo-depth estimation (right panel). Observe that the pseudo-depth has discontinuities at the boundary of the walking figure (lower panels) even though the images contain multiple motions. Best seen in color.

this paper does not critically depend on which motion flow algorithm is used. We obtained similar results using motion code publicly available (e.g., [5]). So we believe that motion flow results by, for example, other algorithms that perform well on the Middlebury dataset [2] may yield motion flow that is sufficiently accurate to be used as input to our approach.

More specifically, we compute dense motion flow for a sequence of three images $\{\mathbf{I}_{-m}, \mathbf{I}_0, \mathbf{I}_m\}$, where $m \geq 3$ indexes the image frame. The middle image $\mathbf{I}_0$ is the reference image for which we will evaluate the pseudo-depth and estimate the occlusion boundaries. The size of m is determined by the following considerations. We require that the images $\mathbf{I}_m, \mathbf{I}_{-m}$ must be sufficiently far apart in time to enable us to estimate the pseudo-depth reliably but they must be close enough in time to ensure that we can obtain the motion flow accurately. In practice, our default setting for the Stein and Hebert database [1] was $m = 7$ but we reduced m for short image sequences. We show two typical results of the motion flow in Figure (2)(middle panels).

We compute the motion flow $\mathbf{V}_b$ from $\mathbf{I}_0$ to $\mathbf{I}_{-m}$ (backwards) and $\mathbf{V}_f$ from $\mathbf{I}_0$ to $\mathbf{I}_m$ (forwards). These motion flows $\mathbf{V}_b$ and $\mathbf{V}_f$ are used to specify the correspondence between pixels in the three images. We represent the pixel coordinates in the reference image $\mathbf{I}_0$ by $\mathbf{x}$ with corresponding pixels coordinates $\mathbf{x} - \mathbf{V}_b$ and $\mathbf{x} + \mathbf{V}_f$ in $\mathbf{I}_{-m}$ and $\mathbf{I}_m$ respectively.

4 Step 2: Dense Pseudo-Depth Estimation

We use the motion flow field to estimate dense pseudo-depth by a two-stage process. Firstly, we formulate the problem in terms of quadratic minimization which can be solved by the standard Singular Value Decomposition (SVD) approach, yielding a noisy estimation of pseudo-depth. Secondly, we obtain a smoothed estimate of pseudo-depth by decomposing the image into super-pixels [11], defining a Markov Random Field (MRF) on the pseudo-depth for the super-pixels and imposing a weak smoothness assumption.

Our method depends on three key assumptions that: (i) occlusion (and pseudo-depth) boundaries can only occur at the boundaries of super-pixels, (ii) the pseudo-depth within each super-pixel can be modeled as planar, and (iii) the parameters of the pseudo-depth planes at neighboring super-pixels are weakly smooth (i.e. is usually very similar but can occasionally change significantly – for example, at occlusion boundaries). Figure 3 shows two examples of smoothed pseudo-depth fields.

4.1 Pseudo-Depth from motion

We use corresponding pixels $\mathbf{x} - \mathbf{V}_b$, $\mathbf{x}$, and $\mathbf{x} + \mathbf{V}_f$, supplied by the motion-flow algorithm, to estimate pseudo-depth for all pixels in the reference image $\mathbf{I}_0$. We assume that the camera geometry can be modeled as scaled-orthographic projection [19] (This is a reasonable assumption provided the the camera has the same direction of gaze in all three images).

We also assume that the motion of the viewed scene can be modeled as if it is perfectly rigid. This rigidity assumption is correct for many images in the dataset [1] but is violated for those which contain moving objects such as cats or humans, for example see Figure (2)(lower panels). Interestingly the pseudo-depth estimation results are surprisingly insensitive to these violations and, in particular, discontinuities in the pseudo-depth estimates often occur at the boundaries of these moving objects.

More formally, we assume that the pixels $\mathbf{x} = \{(x_\mu, y_\mu) : \mu \in \mathbf{L}\}$ in the reference image (where $\mathbf{L}$ is the image lattice) correspond to points $\mathbf{X} = \{(x_\mu, y_\mu, z_\mu) : \mu \in \mathbf{L}\}$ in three-dimensional space, where the $\{z_\mu : \mu \in \mathbf{L}\}$ are unknown and need to be estimated. We assume that the other two images $\mathbf{I}_{-m}$ and $\mathbf{I}_m$ are generated by these points $\mathbf{X}$ using scaled orthographic projection with unknown camera projection matrices $\mathbf{C}_{-m}, \mathbf{C}_m$. Hence the positions of these points $\mathbf{X}$ in images $\mathbf{I}_{-m}, \mathbf{I}_m$ is given by $\Pi(\mathbf{X}; \mathbf{C}_{-m})$ and $\Pi(\mathbf{X}; \mathbf{C}_m)$ respectively, where $\Pi(\mathbf{X}; \mathbf{C}) = \mathbf{CX}$ is the projection.

Our task is to estimate the projection parameters $\mathbf{C}_{-m}, \mathbf{C}_m$ and the pseudo-depths $\{z_\mu\}$ so that the projections best agree with the correspondences between $\mathbf{I}_{-m}, \mathbf{I}_0, \mathbf{I}_m$ estimated by the motion flow algorithm. It can be formulated as minimizing a quadratic cost function:

$$E[\{z_\mu\}; \mathbf{C}_{\{-m,m\}}] = \sum_\mu |\mathbf{x}_\mu + \mathbf{v}^f_\mu - \Pi(\mathbf{X}_\mu; \mathbf{C}_m)|^2 + |\mathbf{x}_\mu - \mathbf{v}^b_\mu - \Pi(\mathbf{X}_\mu; \mathbf{C}_{-m})|^2. \quad (1)$$

As is well known, this minimization can be solved algebraically using singular value decomposition to estimate $\{z^*_\mu\}$ and $\mathbf{C}^*_{-m}, \mathbf{C}^*_m$ [9, 10]. For the scaled orthographic approximation there is only a single ambiguity $\{z^*_\mu\} \mapsto \{\lambda z^*_\mu\}$ where λ is an unknown constant (but some estimate of λ can be made using knowledge of likely values of the camera parameters). We do not attempt to estimate λ and instead use the method described in [10] which implicitly specifies a default value for it.

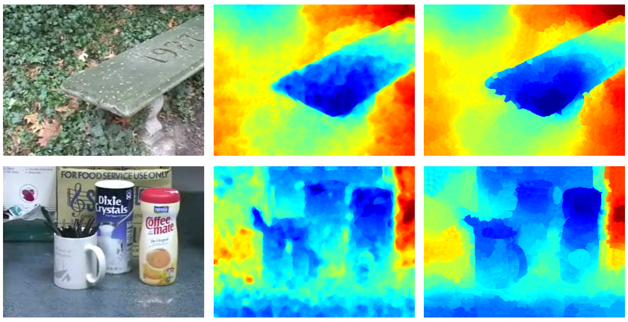

Fig. 3. Two examples of the estimated pseudo-depth field. Left panels: the reference images. Middle panels: the pseudo-depth estimated using SVD. Right panels: the pseudo-depth after weak smoothing. Best seen in color.

4.2 Weak Smoothness to improve the pseudo-depth estimates

The pseudo-depth estimates provided by SVD are noisy, particularly at places where the motion flow is noisy. We improve these estimates by weak smoothing using a Markov Random Field (MRF) model which discourages smoothing across depth discontinuities. This smoothing must be "weak" in order to avoid smoothing across the occlusion boundaries.

To define this MRF we first decompose the reference image into super-pixels using a spectral clustering method [11]. This gives roughly 1000 superpixels for the reference image (which is usually of size 240 × 320). We assume that each super-pixel corresponds to a planar surface in pseudo-depth. This assumption is reasonable since the size of the super-pixels is fairly small (also, by definition, the intensity properties of super-pixels are fairly uniform so it would be hard to get more precise estimates of their pseudo-depth). We also assume that neighboring super-pixels have planar surfaces which have similar orientations and pseudo-depth except at motion occlusion boundaries. This method is similar to one reported in [12] who also used a MRF defined on super-pixels for depth smoothing.

More precisely, let $\mathbf{X}_i = \{(x_{ir}, y_{ir}, z_{ir})\}$ be the set of points (indexed by r) in the i^{th} superpixel (with their pseudo-depths estimated as above). We assume a parametric planar form for each super-pixel – $a_i(x_{ir} - x_{i0}) + b_i(y_{ir} - y_{i0}) + z_{ir} - c_i = 0$ – and express the parameters as $\mathbf{d}_i = (a_i, b_i, c_i)$.

Next we define an MRF whose nodes are the super-pixels and whose state variables are the parameters $D = \{\mathbf{d}_i\}$ (i.e. a super-pixel i has state $\mathbf{d}_i$). The MRF has unary potential terms which relate the state $\mathbf{d}_i$ to the three-dimensional position estimates $\mathbf{X}_i$ and pairwise potential terms which encourage neighboring super-pixels to have similar values of $\mathbf{d}$, but which are robust to discontinuities (to prevent smoothing across occlusion boundaries). This gives an MRF specified by a Gibbs distribution with energy:

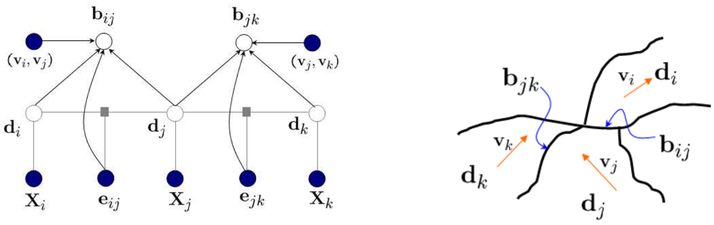

Fig. 4. Left panel: A graphical representation of the Markov random field model for the depth field and occlusion boundary detector. Circular nodes are random variables, rectangular nodes are data-dependent functions, and shaded nodes are observed. Right panel: An illustration of neighboring superpixels and the corresponding random variables defined on the superpixels and their boundaries.

$$E(D|\mathbf{X},\mathbf{e}) = \sum_i E_u(\mathbf{d}_i, \mathbf{X}_i) + \sum_{i,j:\ j \in N(i)} E_p(\mathbf{d}_i, \mathbf{d}_j, e_{ij}), \tag{2}$$

where $\mathbf{e} = \{e_{ij}\}$ is a static edge cue [13] and e_{ij} is the static edge probability between super-pixels i and j. $N(i)$ is the neighborhood of node i. Figure 4 shows the graphical representation of our model. The unary and pairwise terms are defined as follows.

The unary term at super-pixel i depends on the 3D position estimates $\mathbf{X}_i$ at the points within the super-pixel. We use an L1 norm to penalize the deviation of these points from the plane with parameters $\mathbf{d}_i$ (L1 is chosen because of its good robustness properties) which gives:

$$E_u(\mathbf{d}_i, \mathbf{X}_i) = \alpha \sum_r ||\mathbf{c}_{ir}^T \mathbf{d}_i + z_{ir}||_{l_1} \tag{3}$$

where $\mathbf{c}_{ir} = (x_{ir} - x_{i0}, y_{ir} - y_{i0}, -1)^T$ is a constant vector for each point in the super-pixel i. The pairwise energy function also uses the L1 norm to penalize the differences between the parameters $\mathbf{d}_i$ and $\mathbf{d}_j$ at neighboring pixels, but this penalty is reduced if there is a strong static edge between the super-pixels. This gives:

$$E_p(\mathbf{d}_i, \mathbf{d}_j, e_{ij}) = (1 - \beta e_{ij})||\mathbf{d}_i - \mathbf{d}_j||_{l_1}. \tag{4}$$

where β is a coefficient modulating the strength of the edge probability e_{ij}.

4.3 Inferring Pseudo-Depth using the weak smoothness MRF

We now perform *inference on the MRF* to estimate the best state $\{\mathbf{d}_i^*\} = \arg\min E(D|\mathbf{X},\mathbf{e})$ by minimizing the energy. This energy is convex so we can solve for the minimum by performing coordinate descent using Linear Programming (LP) [26] to compute each descent step. At each step, a superpixels's depth

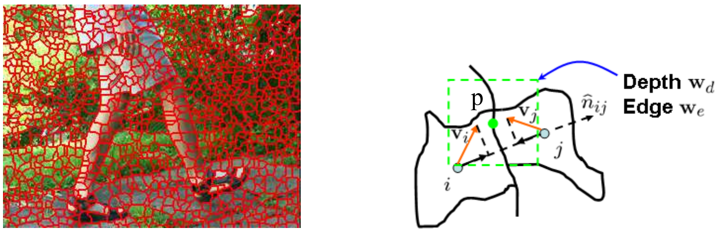

Fig. 5. Left: An instance of super-pixeled image. Right: Local cues for occlusion boundary detection. Best seen in color.

variables are updated given its neighbor information and we sequentially update all the nodes in the field until the changes are below a fixed threshold. A few iterations, typically about 5, suffice for convergence.

We keep updating the random field for several iterations until the change is small. More specifically, we solve the following minimization problem using LP at each node:

$$\mathbf{d}_i = \arg\min_{\mathbf{d}_i} \alpha \sum_r ||\mathbf{c}_{ir}^T \mathbf{d}_i + z_{ir}||_{l_1} + \sum_{j \in N(i)} (1 - \beta e_{ij}) ||\mathbf{d}_i - \mathbf{d}_j||_{l_1} \qquad (5)$$

where $\mathbf{c}_{ir}$ are constants in E_u as in Eqn (3).

5 Step 3: Occlusion boundary detection

We now address the final task of occlusion detection. We use the super-pixel map of the image and attempt to classify whether the boundary between two super-pixels is, or is not, an occlusion boundary. To achieve this we train a classifier whose input is the static edge map, the estimated motion flow field, and the estimated pseudo-depth. The ground truth can be supplied from a labeled dataset, for example [1].

Three types of local cues are evaluated in our method: 1) the pseudo-depth estimates; 2) the static boundary/edge map; 3) the averaged motion estimates within each superpixel. More specifically, for a pixel $\mathbf{x}_p$ on the superpixel boundary b_{ij}, the pseudo-depth cue is a patch $w_d(p)$ of the pseudo-depth map centered at $\mathbf{x}_p$, the edge cue is a patch $w_e(p)$ of the static edge probability map at the same location (e.g. supplied by Berkeley boundary detection [13] or [27]). For the motion cue, we compute the relative motion in the following way. For superpixels i and j, their average velocity v_i and v_j are computed first. Then they are projected onto the unit vector connecting the centers of mass of the two superpixels, denoted by $\hat{n}_{ij}$. See Figure 5 for an illustration of those cues.

The output of the classifier is a probability that the pixel $\mathbf{x}_p$ is on occlusion boundary. Let $b(\mathbf{x}_p)$ denote this event. The classifier output can be written as $P(x_p|w_d(p), w_e(p), \hat{n}_{ij}^T v_i, \hat{n}_{ij}^T v_j)$. To decide if a superpixel boundary b_{ij} is an occlusion boundary, we apply the classifier to all the points $\mathbf{x}_p \in b_{ij}$ and average the outputs of the classifier. More specifically, we set

$$P(b_{ij}|D, \mathbf{e}, v_{i,j}) = \frac{1}{|\mathbf{x}_{b_{ij}}|} \sum_{\mathbf{x}_p \in \mathbf{x}_{b_{ij}}} P(b(\mathbf{x}_p)|w_d(p), w_e(p), \mathbf{v}_{i,j}) \qquad (6)$$

where $\mathbf{x}_{b_{ij}}$ is the set of pixel sites on the boundary b_{ij}, and $\mathbf{v}_{i,j}$ denotes $(\hat{n}_{ij}^T v_i, \hat{n}_{ij}^T v_j)$. The classifier can be any binary classifier with probability output. In our experiments we report results using a Multilayer Perceptron (MLP) with 15 hidden units. But we also experimented with a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel, which gave similar results. We refer to [28] for details of these techniques.

6 Experiments

6.1 Dataset and setup

We evaluated our approach on the CMU dataset, which includes 30 image sequences of real-world scenes [1]. We use half of the sequences as training data and the other half as test data (selected randomly). We swap the training and test sets to obtain test results on all the images. In each case, we treat a third of the training data as a validation set.

We segment each reference image into approximately 1000 superpixels using spectral clustering [11]. We align the ground truth occlusion boundaries to the superpixels using the method described in [1]. This step introduces a small amount of errors, particularly on objects with detailed structure. We treat this aligned superpixel labeling as the ground truth in our evaluation (like [1]). We set the parameters of the pseudo-depth random field using the validation set and searching over the range $[0, 1]$ with step size 0.1. This yields values $\alpha = 1.0$ and $\beta = 0.9$ which we use in the following evaluation. We use a Multilayer Perceptron with 15 hidden nodes (we also tested an SVM with RBF kernel – both give similar performance), trained using all the positive examples and 1/3 of the negative examples selected randomly.

6.2 Experimental results

The experimental results are summarized in Figure 6, which shows the precision-recall curves of our occlusion boundary detector with different settings and the state of the art. The precision (Pr) and recall (Rc) are computed in terms of the superpixel boundary segments. We also show the error rates with threshold 0.5, and F measure computed at 50% recall rate in Table 1. The F measure is defined by the harmonic mean of the precision and recall rate, i.e., $F = 2/(1/Pr+1/Rc)$.

The left column in Figure 6 shows the detection results with the static edge cue and pseudo-depth cue. Observe that the pseudo-depth cue by itself is extremely useful at low recall values hence validating our use of it. The pseudo-depth is also complimentary to the static edge cue: it has high precision at

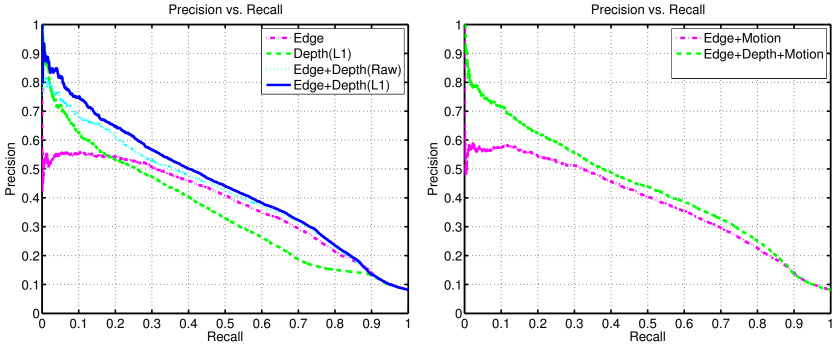

Fig. 6. Left panel: The precision-recall curve of our method with pseudo-depth and edge cues separately and in combination. Right panel: The precision-recall curve of our models including the motion cues. Observe that the motion cue does not contribute much if the other cues are used by comparing two plots. Best seen in color.

Table 1. A summary of average error rates and F measures for occlusion detection using different combinations of cues. Depth(raw) is the direct pseudo-depth output of SVD. Depth(L1) is the weakly smoothed pseudo-depth estimate. Adding motion cue to ”Edge+Depth” does not provide significantly different results.

	Edge only	Depth(L1)	Edge+Motion	Edge+Depth(raw)	Edge+Depth(L1)
Error Rate	8.29	8.30	8.27	8.20	7.98
F-Score	44.93	39.78	44.73	46.01	46.89

low recall region while the static edge cue works better in the higher recall region. Combining both achieves the best performance. The smoothed pseudo-depth information provides better performance than the raw pseudo-depth map (provided by SVD), which demonstrates the benefits of weakly smoothing the pseudo-depth. The right column in the plot examines the improvements in performance due to the motion cues. We notice that adding the motion cue achieves only slightly better results by comparing two plots in Figure 6, and performance is similar to the model without the motion cue, as shown by the precision-recall curve. This might be caused by the relatively simple motion cue we use. We note that direct comparisons with the methods of Stein and Hebert’s performance [1] is not completely precise because we used a different procedure for estimating super-pixels, but visual inspection suggests that their super-pixels are very similar to ours.

We can compare our results to those of Stein and Hebert’s shown in Figure 7. Observe that our method gives generally comparable results to their ”state-of-the-art” and outperforms them in the high precision regime. Moreover, their performance is significantly helped by their sophisticated global boundary model, which we do not use. Figure 8 illustrates the differences between our methods and the difference between the cues that are used.

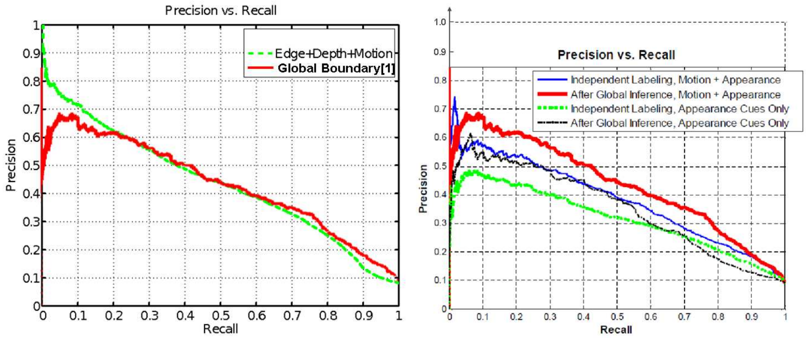

Fig. 7. Left panel: we compare our results with the best result reported by the global boundary model in [1]. Right panel: the precision-recall curve from [1] for comparison. Observe that our performance is better in the high precision regime and that their results rely heavily on their global boundary model which we do not use. Best seen in color.

Figure 9 shows a few examples of dense pseudo-depth fields and occlusion boundary detection results. We can see that our approach handles both static scenes (row 1-4) and dynamic scenes (row 5-7) well. In the static scenes, the pseudo-depth estimation provides a sharp 3D boundary, which makes occlusion boundary detection much easier than using image cues only. The ”pseudo-depth” in image sequences containing moving objects is also very informative for the occlusion detection task because it helps indicate depth boundaries even though the pseudo-depth values within the moving objects are highly inaccurate.

Note that the evaluation of occlusion boundaries are performed only at super-pixel boundaries [1] so there may be some errors introduced. But visual inspection shows that almost all the occlusion boundaries do occur at super-pixel boundaries.

Finally, note that our pseudo-depth smoothing method is successful at filling in small regions (super-pixels) where the motion estimation is noisy and hence the depth estimated by SVD is also noisy. But this smoothing cannot compensate for serious errors in the motion flow estimation. It will obviously not compensate for pseudo-depth estimation errors caused by moving objects, which may require some type of segmentation and separate depth estimation within each segmented region. We investigated whether the presence of moving objects could be detected from the eigenvalues of the SVD, as described in [29], but this was not successful– for almost all image sequences we typically only found two significant eigenvalues independent of whether the sequences contained moving objects. More research is needed here.

Our algorithm runs in approximately 10 seconds with all parts of the code, except the motion estimation, implemented in Matlab. So it should be straightforward to speed this up to real time performance. Stein and Hebert do not report computation time [1].

	Classifier Input	Appearance	Motion	Depth	Global Boundary
Our Method	Pixel level	Edge Map	Motion Field	Pseudo-Depth	No
Stein and Hebert [1]	Super-pixel level	Super-pixel Statistics (edge, color, length and area ratio – 20 features.)	Motion Statistics	No	Boundary Consistency (Corners, T and X junctions – 50 features)

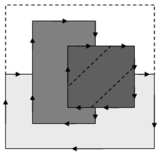

Fig. 8. Left panel: We contrast the cues used in our paper with those reported in [1]. We rely on pseudo-depth, static edges, and motion (but motion adds little). Stein and Hebert use static edges, motion cues, and a global boundary process. We classify individual pixels while they classify super-pixel boundaries directly. Right panel: This illustrates the surface consistency cues used in the global boundary process in [1] which, presumably, would improve our results.

7 Conclusion and Discussion

This paper shows that crude estimation of depth, which we call pseudo-depth, provides useful cues for estimating occlusion boundaries particularly in combination with static edge cues. We show that pseudo-depth can be estimated efficiently from motion sequences and that the discontinuities in pseudo-depth occur at occlusion boundaries. We train a classifier for occlusion boundary detection with input from pseudo-depth, edge cues, and motion cues. We show that pseudo-depth and edge cues give good results comparable with the state of the art [1] when evaluated on benchmarked datasets. But that enhancing the cue set to include the motion separately does not give significant improvements. We note that the methods we use do not exploit global surface consistency constraints which are used extensively in [1] as global boundary models. Hence we conjecture that even better results can be obtained if these surface cues are combined with pseudo-depth and edge cues.

Acknowledgments. We acknowledge the support from the NSF 0736015. We appreciate conversations with Shuang Wu and George Papandreou.

References

1. Stein, A., Hebert, M.: Occlusion boundaries from motion: Low-level detection and mid-level reasoning. International Journal of Computer Vision **82** (2009) 325–357
2. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A database and evaluation methodology for optical flow. In: ICCV. (2007)
3. Anandan, P.: A computational framework and an algorithm for the measurement of visual motion. International Journal of Computer Vision **2** (1989) 283–310
4. Roth, S., Black, M.J.: On the spatial statistics of optical flow. International Journal of Computer Vision **74** (2007) 1
5. Wedel, A., Cremers, D., Pock, T., Bischof, H.: Structure- and motion-adaptive regularization for high accuracy optic flow. In: ICCV. (2009)

6. Geiger, D., Ladendorf, B., Yuille, A.: Occlusions and binocular stereo. International Journal of Computer Vision **14** (1995) 211–226
7. Belhumeur, P., Mumford, D.: A bayesian treatment of the stereo correspondence problem using half-occluded regions. (1992) 506–512
8. Gamble, E., Geiger, D., Poggio, T., Weinshall, D.: Integration of vision modules and labeling of surface discontinuities. IEEE Transactions on Systems, Man and Cybernetics **19** (1989) 1576 –1581
9. Kontsevich, L., Kontsevich, M.L., Shen, A.K.: Two algorithms for reconstructing shapes. Optoelectronics, Instrumentation and Data Processing **5** (1987) 75–81
10. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision **9** (1992) 2
11. Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV. (2003)
12. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE Trans. Pattern Anal. Mach. Intell. **31** (2009) 824–840
13. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans. Pattern Anal. Mach. Intell. **26** (2004) 530–549
14. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence **17** (1981) 185–203
15. Black, M., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. CVIU **63** (1996) 1
16. Fleet, D.J., Black, M.J., Nestares, O.: Bayesian inference of visual motion boundaries. In: Exploring artificial intelligence in the new millenniu. (2003) 139–173
17. Zitnick, C.L., Jojic, N., Kang, S.B.: Consistent segmentation for optical flow estimation. In: ICCV. (2005)
18. Barbu, A., Yuille, A.: Motion estimation by swendsen-wang cuts. In: CVPR. (2004)
19. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
20. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: ICCV. (2009)
21. Xiong, Y., Shafer, S.A.: Dense structure from a dense optical flow sequence. Comput. Vis. Image Underst. **69** (1998) 222–245
22. Ernst, F., Wilinski, P., Overveld, C.W.A.M.v.: Dense structure-from-motion: An approach based on segment matching. In: ECCV. (2002) 217–231
23. Calway, A.: Recursive estimation of 3d motion and surface structure from local affine flow parameters. IEEE Trans. Pattern Anal. Mach. Intell. **27** (2005) 562–574
24. Russell, B.C., Torralba, A.: Building a database of 3d scenes from user annotations. In: CVPR. (2009)
25. Hoiem, D., Stein, A., Efros, A., Hebert, M.: Recovering occlusion boundaries from a single image. In: ICCV. (2007)
26. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
27. Konishi, S., Yuille, A., Coughlan, J., Zhu, S.C.: Statistical edge detection: learning and evaluating edge cues. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 57 – 74
28. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer-Verlag New York, Inc. (2006)
29. Costeira, J., Kanade, T.: A multibody factorization method for independently moving-objects. **29** (1998) 159–179

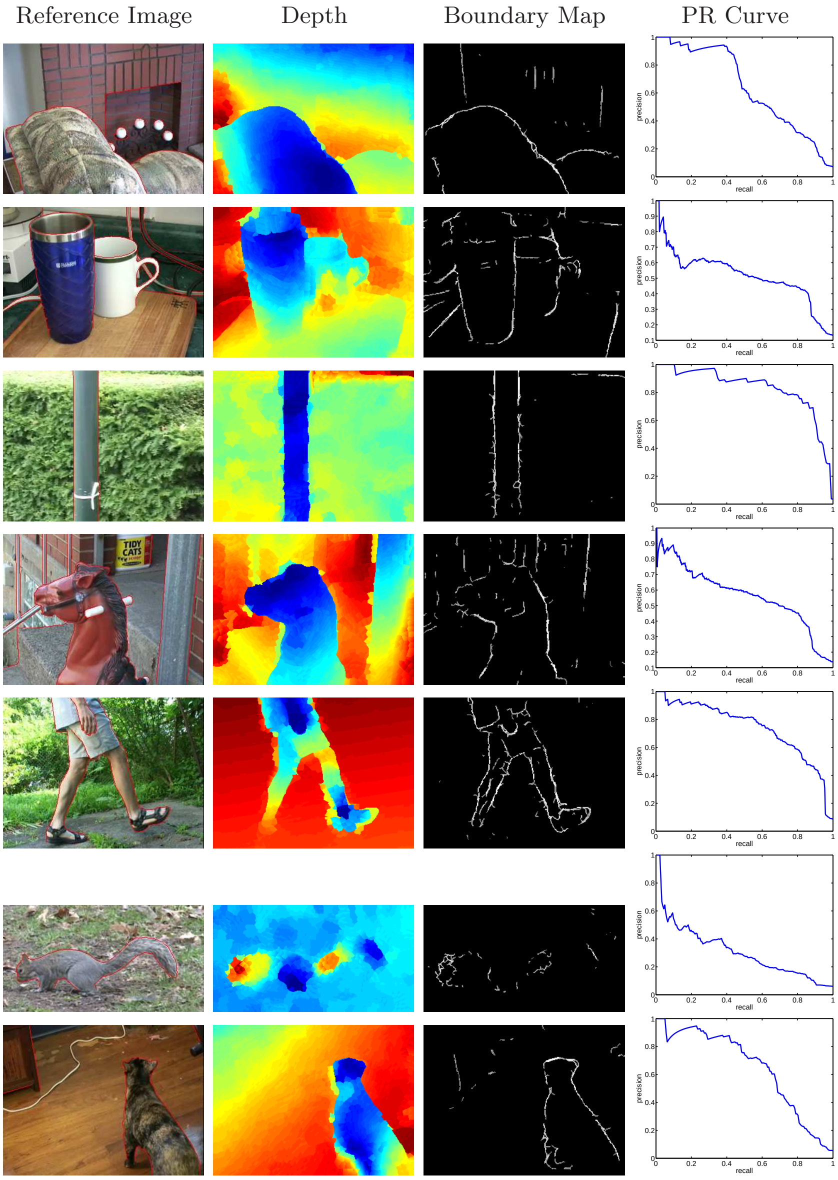

Fig. 9. Example of occlusion boundary detection results on the CMU dataset. First column: the reference frame with the ground truth boundaries overlaid. Second column: the estimated pseudo-depth field. Third column: the confidence map of the detected occlusion boundaries. Fourth column: Precision-Recall curves for the corresponding individual image sequences. Best seen in color.