Int J Comput Vis (2009) 82: 325–357
DOI 10.1007/s11263-008-0203-z

Occlusion Boundaries from Motion: Low-Level Detection and Mid-Level Reasoning

Andrew N. Stein · Martial Hebert

Received: 9 June 2008 / Accepted: 23 December 2008 / Published online: 3 February 2009

Abstract The boundaries of objects in an image are often considered a nuisance to be "handled" due to the occlusion they exhibit. Since most, if not all, computer vision techniques aggregate information spatially within a scene, information spanning these boundaries, and therefore from different physical surfaces, is invariably and erroneously considered together. In addition, these boundaries convey important perceptual information about 3D scene structure and shape. Consequently, their identification can benefit many different computer vision pursuits, from low-level processing techniques to high-level reasoning tasks.

While much focus in computer vision is placed on the processing of individual, static images, many applications actually offer video, or *sequences* of images, as input. The extra temporal dimension of the data allows the motion of the camera or the scene to be used in processing. In this paper, we focus on the exploitation of subtle relative-motion cues present at occlusion boundaries. When combined with more standard appearance information, we demonstrate these cues' utility in detecting occlusion boundaries locally. We also present a novel, mid-level model for reasoning more globally about object boundaries and propagating such local information to extract improved, extended boundaries.

Keywords Occlusion detection · Occlusion boundaries · Occluding contours · Edge detection · Motion

A.N. Stein (✉) · M. Hebert
The Robotics Institute, Carnegie Mellon University, Pittsburgh, USA
e-mail: stein@ri.cmu.edu

M. Hebert
e-mail: hebert@ri.cmu.edu

1 Introduction

Consider the scene depicted in Fig. 1, taken from the LabelMe database (Russell et al. 2005). There are many overlapping objects and surfaces in this scene. Indeed, almost *everything* in the scene is occluded by, and/or occludes, another object or surface! A user has begun to label a few foreground objects thus far (indicated by the blue and red dots), but the process of painstakingly labeling each and every inter-related boundary is daunting.

In natural images, however, this type of complexity is common. Objects are generally not conveniently laid out in well-separated poses or in front of uniform backgrounds. As a consequence, occlusion and dis-occlusion at objects' boundaries frustrate many processing techniques frequently used in computer vision. Due to imaging noise and the inherent lack of information (and resulting ambiguity) when considering an individual pixel, most processing techniques in our field aggregate information spatially in images. This aggregation may be the result of simple smoothing (*e.g.* , Gaussian blurring for the purpose of down-sampling and multi-scale processing), the consideration of discrete patches of pixels (*e.g.* , for feature-based object recognition or face/pedestrian detection), or the use of graphical models connecting neighborhoods of pixels (*e.g.* , Markov, Conditional, or Discriminative Random Fields (Geman and Geman 1984; Kumar and Hebert 2006; Lafferty et al. 2001)).

Each of these methods implicitly assumes that all the nearby or connected pixels "belong together" (*e.g.* are from the same object, motion layer, *etc.*). But this assumption is violated at object boundaries, where information from two different physical surfaces is smoothed/transmitted across the boundary or collected within a single patch (as shown in Fig. 2), and thus subsequent results will be muddied or

Fig. 1 Example scene exhibiting extensive occlusion (from Russell et al. 2005). Almost every object or surface is occluding and/or occluded by another object or surface. Any computer vision method which spatially aggregates information in this scene will almost certainly simultaneously consider data from two different objects

Fig. 2 In cluttered scenes, any technique that aggregates information spatially, such as a patch-based method, will erroneously combine information from physically different objects/surfaces, leading to poor results

completely incorrect. For this reason, pixels near boundaries are often treated as outliers to be handled by robust methods, or multiple/adaptive-windowing techniques are employed (Fusiello et al. 1997; Hirschmüller et al. 2002; Kanade and Okutomi 1994). These sorts of approaches are inherently focused on reasoning based on information contained *within* the regions enclosed by the boundaries. By contrast, this paper will focus on the boundaries themselves and will attempt to detect and reason about them directly. Note that these two sources of information, regions and their boundaries, are complementary and are each useful in their own way (Fowlkes et al. 2003; Heitz and Bouthemy 1993; Smith 2001); we are not advocating the exclusive use of one over the other.

The boundaries present in a scene are not only a nuisance for many processing techniques, they are also a valuable source of perceptual information important for understanding a scene's overall structure and content (Black and Fleet 2000; Stein 2008). Since occlusion boundaries exist at locations in an image where one physical surface is closer to the camera than another, they correspond to the physical extent of objects and structures in a scene, providing strong 3D shape cues without explicit 3D reconstruction.

Consider the methodology employed by feature-based object recognition relying on appearance, *e.g.* the popular Scale Invariant Feature Transform (SIFT) approach due to Lowe (2004). There are generally two stages: feature detection and feature description. Image information is spatially aggregated during both. During detection, filtering processes such as Gaussian smoothing, which are used to create scale-space Difference-of-Gaussian pyramids, will smear information across boundaries. Then, patches of image data which may also cross object boundaries are used to create descriptor vectors for matching. In particular, for scale-invariant methods, larger and larger neighborhoods are considered as the scale of detection and description increases, resulting in many unusable large-scale features which contain information from (multiple) objects and background. Equivalently, as an object appears smaller and smaller within a scene, we must rely more heavily on its larger scale features (relative to observed size of the object) for matching. Knowledge of the location of object boundaries in a scene could be used to combat these problems and improve recognition in cluttered scenes (Stein and Hebert 2005).

Furthermore, while there has been impressive progress in the last few years in *recognizing* specific objects in images such as cars, bicycles, people, *etc.* , using feature-based methods, for example, the problem of *detecting* generic, never-before-seen objects—*i.e.* without a given library of

knowns—remains a difficult challenge. For instance, how may we determine a telephone sitting on our desk is an object separate from its surroundings, without already knowing what a telephone is? Or as Adelson and Bergen put it in 1991, how do we distinguish the "things" from the "stuff?" This goal is variously known as object segmentation, *pop-out*, or figure-ground labeling, and we believe that detected occlusion boundaries, which themselves delineate the physical extents of objects, can provide a strong cue for tackling it (Stein et al. 2008).

With these high-level tasks as motivation, we propose to revisit the use of motion cues in extracting occluding contours as a step toward identifying object boundaries in a scene. We use subtle motion cues, such as parallax induced by a moving camera, in reasoning about these crucial boundaries which separate "things" in a scene.

After discussing related work and our dataset, we will begin in Sect. 4 with an over-segmentation of the image, with the assumption that the true object/occlusion boundaries of interest are a subset of the fragmented boundaries formed by the regions (or *segments*) in that over-segmentation. Next we will estimate the motion of those segments and fragments in Sect. 5. Combined with appearance cues (Stein and Hebert 2007), this motion information will be used to generate features for a classifier trained to distinguish fragments that are merely surface markings from those that are object/occlusion boundaries. Finally, by learning a notion of fragment connectivity and constructing a factor graph to model fragment and junction interdependencies (Sect. 6), we will perform global inference to estimate the optimal labeling of the fragments jointly. Using this approach, we will demonstrate in Sect. 8 improved object boundary labeling when (a) using motion information and (b) additionally using global inference.

2 Related Work

Prior attempts to use motion cues to extract object contours (or object segmentations implying the contours) can be divided roughly into two groups: those that segment regions directly from the motion input, and those that detect contours via some local computation on the motion data. The first category includes approaches that attempt to infer segmentation or scene structure directly from reasoning about large-scale occlusions observed through dynamic object motion (Brostow and Essa 1999; Ogale et al. 2005) and/or the use of multiple, calibrated cameras for obtaining silhouettes (Guan et al. 2007).

Also in this category is layered motion segmentation (Darrell and Pentland 1995; Wang and Adelson 1994), in which regions are segmented from an input image sequence based on the consistency of motion within each region, *e.g.* (Irani and Peleg 1993; Jepson et al. 2002; Jojic and Frey 2001; Ke and Kanade 2002; Kumar et al. 2005; Ogale et al. 2005; Shi and Malik 1998; Smith et al. 2004; Xiao and Shah 2005). Most of these methods use a parametric motion model for each layer, and employ various techniques, such as Expectation Maximization, for estimating those models and for assigning pixels to the correct layer or model. Typical models are restricted to near-planar, rigidly-moving regions. In addition, many approaches assume a known, fixed number of layers in the scene or do not scale well as that number increases. We argue that attempting to explain the scene generatively in terms of a specific number of motion-consistent connected regions may not be necessary, and instead we propose to detect a large fraction of the objects' boundaries by estimating local motion cues and using them in a discriminative statistical classifier combined with a mechanism to enforce global consistency. Quite recently, a method for *binary* segmentation of video was presented which combats some of the difficulties of layered motion segmentation methods by combining clustered motion features (akin to the textons popularized by recognition research) with a boosted tree-based classifier (Yin et al. 2007).

Usually, the erratic results near boundaries are treated by methods such as those above as outliers to an underlying smooth process. The subsequent delineation of precise motion boundaries, if performed at all, is generally of secondary importance. A notable exception, however, is found in Heitz and Bouthemy (1993), where vertical and horizontal between-pixel motion boundaries plus their interactions with nearby dense optical flow vectors are considered in an MRF framework. Similarly, dual graphs on the pixels and the edges between them (sometimes referred to as "cracks") are considered in a normalized cuts framework in Yu and Shi (2001). Stereo or structure from motion techniques also have trouble near occlusion boundaries and usually focus on the interiors of regions while handling data near occlusions as complex special cases (Fusiello et al. 1997; Hirschmüller et al. 2002; Kanade and Okutomi 1994). Our work, on the other hand, is not focused on the precise, dense motion estimates themselves, nor on full 3D scene reconstruction; we first seek only to *identify* boundary locations that correspond to visible occlusion, with the eventual goal of employing those boundaries in higher-level reasoning (Stein 2008).

In the second category, techniques have been developed based on the observation that occluding contours can be defined as extremal boundaries, where the viewing ray is tangent to the object's surface. This led to the development of algorithms that rely on an explicit geometric model of the motion of occluding contours (Lazebnik and Ponce 2005; Sato and Cipolla 1999; Sethi et al. 2004; Vaillant and Faugeras 1992). These approaches are appealing because they rely on well-defined, mathematically correct, geometric models. However, one drawback is their sensitivity to deviations of the actual data from the model. An alternative is

to use an implicit model, either learned from local motion cues estimated from training data or based on some fixed model of the distribution of motion cues in the vicinity of occluding boundaries (Black and Fleet 2000; Fleet et al. 2002; Nestares and Fleet 2001; Stein and Hebert 2006a, 2007; Stein et al. 2007). Our work falls in this general category in that we do not attempt to precisely *model* the motion of occlusion boundaries directly. Instead we rely on the statistical discrimination of *relative* local motion cues at those boundaries.

Finally, although we focus in this paper on the use of motion cues, considerable prior work exists in extracting boundaries from a single image. Two major threads emerge from this line of work. The first one is the idea of combining multiple cues into a single boundary classifier (Konishi et al. 2003; Martin et al. 2004; Dollár et al. 2006). The second key idea, largely due to Ren et al. (2006), is to use the region boundaries of an image's (over-)segmentation as initial candidates to be labeled as occluding/non-occluding, thereby inducing a labeling of the regions as figure/ground. We build upon each of these ideas in our work. We combine many local cues into a single classifier, with the difference that we use motion cues in addition to appearance cues. We also start with an over-segmentation of the image, with the goal of filtering it to retain only those region boundaries that correspond to physical object boundaries. In addition, we use a novel model for inferring a globally consistent labeling of the boundary fragments.

3 Our Dataset

We first need appropriate data for testing our methods as well as ground truth labelings of that data in order to train those techniques relying on learned classifiers. In addition, using such a dataset with a significant number and variety of scenes offers a more complete, quantitative analysis as compared to typical anecdotal examples often provided in research using motion data. While segmentation datasets exist, most notably the Berkeley Segmentation Data Set (BSDS) (Martin et al. 2001), they are not appropriate for our task for two reasons. First, our use of a motion cue requires that we have at least one additional image of each scene in order to observe the effects of camera/scene motion. The BSDS consists only of single isolated images, as do all other segmentation datasets of which we are aware. Second, the BSDS edge labels provided by the human subjects do not necessarily correspond to physical occlusion boundaries: any edge which a subject found *semantically* salient may be marked.

For these reasons, we have created a new publicly available dataset[1] which addresses both of these issues (Stein and Hebert 2007; Stein et al. 2007): 30 short image *sequences*, approximately 8–20 frames in length, are provided to allow motion estimation, and only the occlusion boundaries are labeled as ground truth in the reference (*i.e.* middle) frame of each sequence.[2] Note that all our processing and detection occurs in this reference frame; the remaining frames are used solely as additional information for motion estimation.

Admittedly, 30 sequences may be considered a relatively small set, particularly as compared to the size of datasets commonly used in object recognition, for example. Still, we believe this to be a significant and reasonable number of examples, particularly for work in motion analysis, where data collection and labeling of video can be prohibitive. Furthermore, we are not making global image- or video-wide classifications in this work and therefore have far more than one example or data point per sequence (though all examples in a single sequence also may not be independent, see Stein and Hebert 2007). Finally, we have made some effort to include a wide variety of conditions, as described below, in order to provide a degree of hope for the method's application on a larger data set.

Some example scenes are depicted in Fig. 3 along with their ground truth occlusion/object boundary labels. 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. All sequences were collected with the same video camera at 15 frames per second (thus they are approximately one to two seconds in duration). For those scenes in which the camera is moving, that motion is generally horizontal (panning) with an effective total baseline on the order of ten centimeters. The objects of interest (not considered as "background") are roughly one to five meters from the camera, with between-object distances ranging from a few centimeters to a few meters. The amount of motion within the image plane is generally 2–20 pixels per frame.

4 Initial Segments and Fragments

We will begin by over-segmenting the image (Comaniciu and Meer 2002; Felzenszwalb and Huttenlocher 2004; Hoiem et al. 2007a; Malisiewicz and Efros 2007; Mori 2005; Ren and Malik 2003; Tao et al. 2001) in order to generate candidate boundary elements, or *fragments*, and associated patches for cue extraction, or *segments* (also known as "super-pixels" (Ren and Malik 2003)). In particular, the boundaries of these segments will serve as our set of hypothesized boundary fragments, and the segments on either side

[1] http://www.cs.cmu.edu/~stein/occlusion_data/

[2] Admittedly, some subjectivity in labeling is unavoidable, but we believe the boundaries we seek are defined more clearly than typical edges.

Fig. 3 Ground truth occlusion boundaries labeled for 12 of the scenes from our dataset. Each example is the reference (*middle*) frame of a short sequence, usually 8–20 frames

of each fragment will provide data-driven regions of support for our motion and appearance cues.

Another option could be to start from (pixel-level) edge detections and then perform an edge chaining procedure, as in Liu et al. (2006) or Smith et al. (2004), for example. But edge chaining is inherently brittle in natural cluttered scenes, and perhaps more importantly, the over-segmentation approach offers two distinct advantages for our model. First, by construction, fragments meet at intersections of regions, and thus closed contours are immediately available. This produces a natural graph structure suitable for global inference without the need to impose artificially such structure later (*e.g.* with Constrained Delaunay Triangulation (Ren et al. 2005)). Second, a direct link is established between fragments and segments. It is clear that a set of segments in an image imply a set of boundaries, but working in the opposite direction to obtain a segmentation from a set of disconnected boundary fragments is non-trivial.

We use a watershed-based over-segmentation driven by the output of a learned, statistical, multi-cue edge detector, after non-local maxima suppression. We have chosen the *Pb* detector (Martin et al. 2004), though others exist, *e.g.* (Konishi et al. 2003). The use of such a detector allows the over-segmentation simultaneously to consider color, brightness, and texture in choosing where to create initial segments. We chose the watershed approach for its more regularly-shaped segments as compared to other methods (Comaniciu and Meer 2002; Felzenszwalb and Huttenlocher 2004), and its simplicity and speed compared to methods relying on normalized cuts (Mori 2005; Mori et al. 2004; Ren and Malik 2003). We first use non-local maxima suppression on the raw *Pb* responses. Next we introduce random "seeds" into large empty regions in the resulting edge map. This helps break apart potentially large segments and increases regularity. The watershed segmentation algorithm is then applied to the distance transform computed from this edge map. See Fig. 4 for an example of this process. A similar technique, which uses the *Pb* detector's output directly in a watershed segmentation, is suggested in Arbeláez (2006), but we found that our approach tends to produce segments with more regular shape. This allows for better cue extraction later, particularly in the case of motion.

From the over-segmentation, we produce a set of fragments along the boundaries of each segment, starting and stopping at junctions with other segments. Rather than operating at the level of pixels when chaining, however, we operate instead on the "cracks" between the pixels. These cracks naturally form a graph and offer a simple, efficient domain on which to chain. For details on our crack-chaining procedure, see Appendix A. Besides efficiency, the fact that cracks have no width in the image offers important advantages. There is no need to decide which segment's pixels are boundary pixels (and thereby effectively assign boundary ownership prematurely). In addition, a maximum of four fragments can meet at any junction, limiting the number of junction labeling cases we must consider when performing the global inference described in Sect. 6 below.

5 Fragment and Segment Motion

As discussed in the introduction, we will employ motion cues to help determine which of our hypothesized fragments correspond to occlusion boundaries. At occlusion boundaries, there may exist an inconsistency in motion due to parallax induced by the observer's motion, dynamic objects in

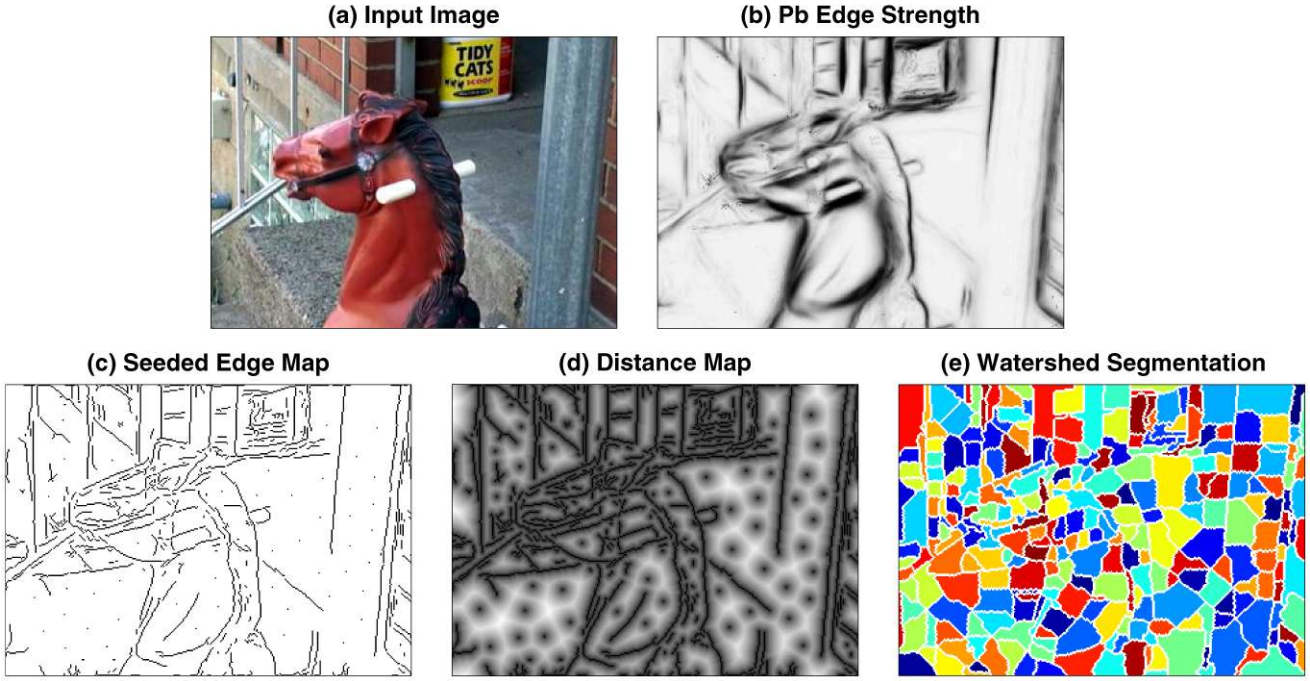


Fig. 4 For an input image (**a**), we use the *Pb* edge detector to get edge strength at each pixel (**b**). After suppressing non-local maxima, we introduce random "seeds" to help break apart large uniform regions and increase segment regularity (**c**). Then, from a distance transform (**d**) of the resulting edge map, we use a standard watershed algorithm to produce our over-segmentation (**e**)

the scene, or both. In our approach, unlike many which rely on a static camera—*e.g.* , for background subtraction (Ross and Kaelbling 2005; Veit et al. 2006)—we do not distinguish between these various cases: moving cameras and/or dynamic scenes are handled equivalently.

The first type of motion inconsistency at a particular fragment is based on the relative motions of the segments on either side. The second type is based on the observation that we can compare the segment motions not only to each other but *to the motion of the fragments which they neighbor*. Consider the common case in which the foreground side of a boundary is nearly textureless and is moving against a cluttered background. The foreground patch motion may be difficult to estimate accurately due to the lack of texture, but we can still use the fact that the occlusion boundary is "owned" by the foreground surface and should move consistently with it (Heitz and Bouthemy 1993). By recognizing this discrepancy, we can still detect the occlusion.

In this work, we are interested in the estimation and analysis of the *instantaneous* motions of fragments and segments. We do not explicitly track either over long periods of time. Instead we examine only a few nearby frames in a short temporal window around the reference frame under consideration.

In the following sections we will discuss in more detail the estimation of motion for our segments and fragments.

5.1 Segment Motion Estimation

A motion estimate for a patch of intensity data may be computed from the patch's spatio-temporal derivatives (Lucas and Kanade 1981; Shechtman and Irani 2005; Tomasi and Kanade 1991). This idea is based on the brightness constancy assumption and forms the fundamental building block for many optical flow, tracking, and registration methods. Such estimation of image motion is a classical problem in computer vision (see Fleet and Weiss 2005 for a recent tutorial on optical flow). In general, the goal is to determine a motion vector (u, v) for each pixel or patch which indicates its displacement from one frame to the next. Here, we will consider several consecutive frames of video and compute *multi-frame* motion estimates. As compared to using only two frames, we find that the increased temporal integration when using multiple frames produces substantially more robust estimates that are more discriminative for our classification task in Sect. 7.

The segments from Sect. 4 naturally specify the spatial support for estimating left- and right-side motions for each boundary fragment. And since each segment is bounded by multiple fragments, we need only compute the motion for each segment in the scene once.

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 segment S in the central reference image, $I^{(0)}$,

with its corresponding pixels in each of the other images, $\{I^{(n)}\}_{n\neq 0}$:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \arg\min \sum_{n=-N}^{N} h(n) \sum_{(x,y)\in S} w(x,y) \times \Big(\underbrace{I^{(n)}(x,y) - I^{(0)}(x-nu, y-nv)}_{r} \Big)^2. \quad (1)$$

Note that this implicitly assumes constant translation for the duration of the set of frames, which is most reasonable over brief time periods. First performing a global, translational stabilization of each frame in the clip to the reference frame can also help make the data better adhere to this constant-translation assumption in addition to removing large motions and focusing subsequent processing on the more subtle relative motions that are most important for our task (Stein 2008). We employ Gaussian-shaped weighting functions, $w(x, y)$ and $h(n)$, with associated bandwidths σ_h and σ_w, to decrease the contribution of pixels spatially close to the segment's borders and temporally distant from the reference frame.

Aggregating patches of pixels 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 (Black and Fleet 2000; Fleet and Weiss 2005). Recently, impressive results for computing dense flow fields in spite of significant occlusion boundaries by using a variational approach and bilateral filtering were demonstrated in Xiao et al. (2006).

For our work, we iteratively estimate u and v using a multi-frame, Lucas-Kanade style differential approach to find the minimum of (1) (Lucas and Kanade 1981). Iteration is employed because the use of finite-sized patches may prevent us from finding the full translation vector (u, v) in one application of least squares. Thus, to solve for the update, (u', v') to the current estimate (u_k, v_k) at iteration k, we replace the residual (or error) term, r, in our objective function (1) by

$$r = I^{(n)}(x + nu_k, y + nv_k) - I^{(0)}(x - nu', y - nv'). \quad (2)$$

Here, the position of the patch in frame n has been adjusted by the previous translation estimate, which can be initialized to zero for the first iteration, when $k = 0$. According to the classical brightness constancy assumption and a first-order approximation we can approximate the second term as

$$I^{(0)}(x - nu', y - nv') \approx I^{(0)}(x, y) - nu' I_x^{(0)}(x, y) - nv' I_y^{(0)}(x, y), \quad (3)$$

where $I_x^{(0)}(x, y)$ and $I_y^{(0)}(x, y)$ represent the spatial derivatives of the reference frame, estimated by finite central differences. Substituting into (2) yields

$$r = \underbrace{I^{(n)}(x + nu_k, y + nv_k) - I^{(0)}(x, y)}_{I_t(u_k, v_k, n)} + nu' I_x^{(0)}(x, y) + nv' I_y^{(0)}(x, y). \quad (4)$$

Finally, the objective function at each iteration becomes

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \arg\min \sum_{n=-N}^{N} h(n) \sum_{(x,y)\in S} w(x,y) \big(I_t(u_k, v_k, n) + nu' I_x^{(0)}(x, y) + nv' I_y^{(0)}(x, y) \big)^2. \quad (5)$$

The corresponding linear least squares formulation is as follows (where $I_x = I_x^{(0)}$ and $I_y = I_y^{(0)}$ for clarity):

$$\underbrace{\begin{bmatrix} n_1 I_{x_1} & n_1 I_{y_1} \\ n_2 I_{x_2} & n_2 I_{y_2} \\ \vdots & \vdots \\ n_M I_{x_M} & n_M I_{y_M} \end{bmatrix}}_{A} \begin{bmatrix} u' \\ v' \end{bmatrix} = - \underbrace{\begin{bmatrix} I_{t_1}(u_k, v_k, n) \\ I_{t_2}(u_k, v_k, n) \\ \vdots \\ I_{t_M}(u_k, v_k, n) \end{bmatrix}}_{b} \quad (6)$$

$$A^T A \begin{bmatrix} u' \\ v' \end{bmatrix} = -A^T b \quad (7)$$

$$\underbrace{\begin{bmatrix} \sum n^2 I_x^2 & \sum n^2 I_x I_y \\ \sum n^2 I_y I_x & \sum n^2 I_y^2 \end{bmatrix}}_{G} \begin{bmatrix} u' \\ v' \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t(u_k, v_k, n) \\ \sum I_y I_t(u_k, v_k, n) \end{bmatrix}, \quad (8)$$

where the sums are taken over all M pixels within the patch, across all frames. (For reduced clutter, we have omitted the weights, $w(x, y)$ and $h(n)$, in this formulation.) The next translation estimate, (u_{k+1}, v_{k+1}), is computed from the previous estimate, combined with the current update:

$$(u_{k+1}, v_{k+1}) = (u_k, v_k) + (u', v'). \quad (9)$$

As is well known, motion estimates near occlusion boundaries are prone to error (Heitz and Bouthemy 1993), but accurate estimates near such boundaries are a crucial component of our approach, rather than outliers to be ignored or filtered out. Therefore, in addition to using the spatial and temporal weighting functions $w(x, y)$ and $h(n)$, we also use iteratively reweighted least squares to solve (8)—further details may be found in Stein (2008).

Furthermore, we initially consider only frame $I^{(0)}$ 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. Since the true corresponding pixels in frames temporally distant from the reference frame could be quite far spatially from their initialized locations, especially when there is significant motion, this process prevents frames at extremes of the temporal window from pulling the solution to poor local minima of (1). Note that the same effect could also be achieved by gradually increasing the bandwidth of $h(n)$. This approach alleviates the need to initially align the patches spatio-temporally to the moving edge, as was suggested by Stein and Hebert (2006a).

Finally, we place a prior on small motions, since the relative motions we seek are quite subtle. In practice, this amounts to adding a small value, inversely proportional to the expected variance of the motion components, to the diagonal of G in (8):

$$\begin{bmatrix} \sum I_x^2 + \frac{1}{2\sigma_u^2} & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 + \frac{1}{2\sigma_v^2} \end{bmatrix} \begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} = -\begin{bmatrix} \sum I_x I_t(u_k, v_k, n) \\ \sum I_y I_t(u_k, v_k, n) \end{bmatrix}, \tag{10}$$

where we use $\sigma_u = \sigma_v = 1$ in this work.

Computing the necessary derivatives within each window (via finite central differences), we can now estimate the motions of the segments on either side of each fragment, $\mathbf{u}_L = [\, u_L \; v_L \,]^T$ and $\mathbf{u}_R = [\, u_R \; v_R \,]^T$, using the least squares approach outlined above.

While one can always find a solution to the least squares formulation for motion estimation in (8) or (10), our "confidence" in the resulting estimated vector, (u, v), depends on the presence of strong spatial gradients within the patch. Loosely speaking, in a patch taken from a nearly-uniform region of the image, the uncertainty on the estimated motion will be much higher than the uncertainty for a patch containing strong gradient information (see Fig. 5).

It is important that we take this uncertainty into account when comparing motion estimates from different patches (Barron et al. 1994; Simoncelli et al. 1991), especially since it is common in practice that one side of an occlusion boundary is nearly uniform, particularly for indoor scenes. In solving for (u, v) according to (8), the Hessian matrix G defines the shape of the solution surface around the particular local minimum of our least squares formulation in (1). Considered in a probabilistic sense, this local minimum also corresponds to the Maximum Likelihood solution for (u, v) under an assumption that the data likelihood, $\Pr(I_t|u, v)$, can be approximated by a Gaussian. When a prior is added, as in (10), we find the Maximum a Posteriori (MAP) solution. In either case, G specifies the covariance of that Gaussian distribution. Thus we can use this matrix to capture the uncertainty in our solution: when G indicates a flat solution surface around our estimated minimum or, equivalently, a Gaussian with large covariance, then we are less certain about the precise location of that minimum than if the surface or Gaussian is strongly peaked. It is worth nothing that the Gaussian assumption does not always hold (Simoncelli et al. 1991). In our experience, though, it is a good enough approximation to be of practical utility in a large number of cases. In addition, resorting to local sampling techniques in order to obtain better estimates of the true underlying distribution of $\Pr(I_t|u, v)$ can be computationally prohibitive (Black and Fleet 2000).

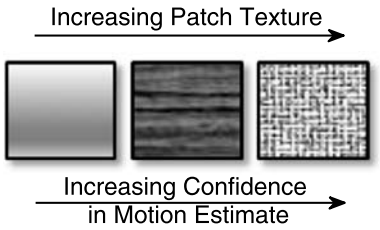


Fig. 5 As horizontal and vertical texture within a patch increases, so too does our confidence in the estimated motion

5.2 Fragment Motion Estimation

A second type of motion cue available related to occlusions is the motion of the boundaries themselves. One approach for estimating this motion would be to attack the problem from a frame-to-frame matching or tracking standpoint (Smith et al. 2004). Perhaps fragments in one frame could be matched to those in the next frame by solving a correspondence problem. However, over-segmentation is an inherently unstable process, so one cannot assume that the fragments found in a subsequent frame will even have a true match in the reference frame. And without any appearance data, short fragments are likely not distinctive enough to be matched reliably without resorting to a more global approach such as Leordeanu and Hebert (2005).

Using tracking approaches like those used in feature tracking (Tomasi and Kanade 1991) is also problematic since, by definition, the pixels on either side of an occlusion boundary belong to different surfaces. Since these two surfaces can move independently, they can confound an approach which assumes all the local data moves contiguously. Thus it would be dangerous to employ local patches of appearance data to estimate motion as described above in order to estimate the boundary's motion by simple tracking. Instead, we need fragment motion estimates derived independently from the neighboring segments.

Therefore, consider treating the video data as a spatio-temporal volume, rather than processing individual frames separately. A moving edge traces out an oriented path along the temporal dimension of the data (Adelson and Bergen

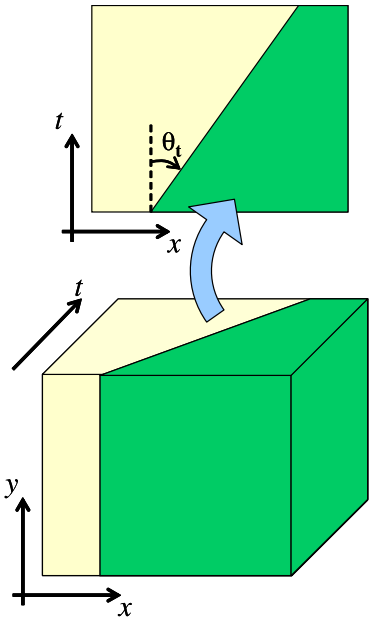


Fig. 6 A moving edge sweeps out an oriented path in a temporal slice of the video volume. The speed of the edge is given by $\tan(\theta_t)$

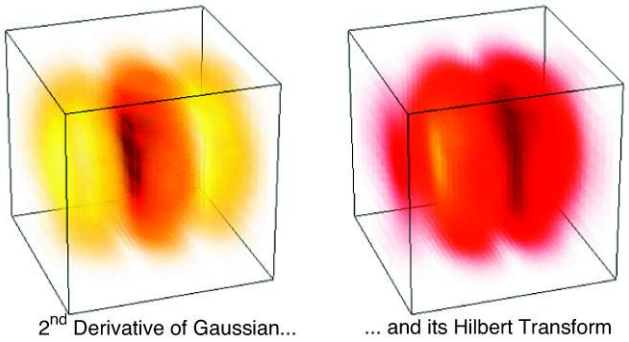


Fig. 7 A quadrature pair of spatio-temporal filters

1985; Bouthemy 1989; Heeger 1988), as depicted for a simple vertical edge in Fig. 6. The tangent of the angle of this path corresponds to its speed. So applying an oriented edge detector to a *temporal* slice of data would detect edges not at some spatial orientation in the image, but at an orientation corresponding to the speed of the edge's motion. The speed estimated by such a detector will correspond to the orientation of a spatio-temporal *plane* in the video volume, rather than an oriented line in a single frame or temporal slice. Note also that this reasoning only requires a consistently different *appearance* on either side of the edge as it moves, but it does not make any assumptions about the consistency of *motion* on either side of that edge (which still may or may not be an occlusion boundary).

Having cast the problem of motion estimation at boundaries into one of spatio-temporal edge detection, we can rely on extensions of existing edge-detection techniques, of which there are two main types: filtering-based and patch-based, with the former being the most common.

For video data, 3D spatio-temporal filters, based on Gaussian derivatives like their 2D counterparts, were designed in the classical work of Adelson and Bergen (1985), Heeger (1988). An example quadrature pair of spatio-temporal filters can be found in Fig. 7. Thus, as described, the combined *spatio-temporal* orientation of the maximum response of such a filter indicates an edge's orientation and speed. Their potential for efficient implementation via separability was recently reported by Derpanis and Gryn (2005).

Unfortunately, filtering performance is often poor at edges which do not conform to the particular intensity profiles for which the filters were designed (Stein and Hebert 2006b). In practice, this is a problem at boundaries bordering textured or cluttered regions, which are also the ones providing the most evidence of occlusion since the texture and clutter can offer the key indicative motion information. Thus we need an edge detector which will perform well in such cases.

Instead of using classical filtering, then, one can also take a more general, non-model-based view of edges. In this case, the profile of the edge is not assumed to have any particular form. Locally, an edge is instead defined to be a line segment which divides a patch of data into halves which contain significantly different *distributions* of some property (*e.g.* brightness, color, texture, *etc.*). By comparing histograms computed in each half of a circular patch, such approaches have been shown to produce good results even on textured/cluttered data (Martin et al. 2004; Maxwell and Brubaker 2003; Ruzon and Tomasi 1999; Stein and Hebert 2006b; Wolf et al. 2006).

Now, to determine the normal speed at a particular edge pixel, we can use a *cylindrical* detector, analogous to the standard spatial, circular one described above, but rotated into the temporal dimension and aligned to the edge's orientation. By dividing the cylinder into halves with a plane, we can detect the speed of the edge's motion in the direction normal to its orientation. In Fig. 8 we see a standard patch-based, oriented spatial edge detector for use in a single frame to the left and the analogous cylindrical, edge *speed* detector in the middle. As described in Stein and Hebert (2006b), it is also possible to design a *spherical* detector capable of simultaneous orientation and speed estimation (shown at the right of Fig. 8), but here the local orientation is provided by the fragments' normals. So a speed-only, cylindrical detector is sufficient for our purposes.

Using such a detector, we can combine local normal speed estimates along the fragment to get a full 2D (u, v) motion estimate for the whole fragment (Drummond and Cipolla 2000; Weiss 1997), as shown in Fig. 9. We use a linear system of equations based on comparisons of the individual 1D speed estimates to the fragment's overall motion

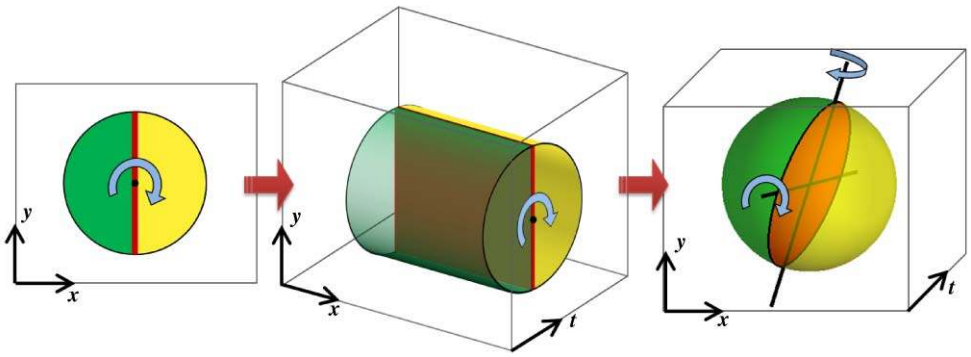


Fig. 8 Three patch-based edge detectors. From *left* to *right*, we have a simple oriented edge detector for a single frame, a cylindrical edge-speed detector for video data, and a spherical detector capable of detecting spatial orientation and normal speed simultaneously

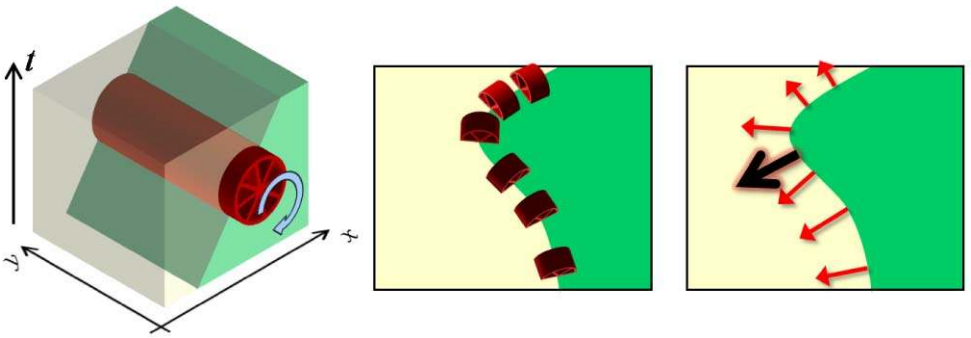


Fig. 9 (Color online) We can use a cylindrical, spatio-temporal edge detector (*left*), at each point along a boundary fragment (*middle*), aligned to the fragment's local orientation. Each detector will yield a normal speed estimate (*right*, *red arrows*), which can be combined into a single motion vector for the fragment (*larger black arrow*) using (12)

vector projected onto the local normal vectors:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \arg\min \sum_{i \in F} w(i) \left(n_{x,i} u + n_{y,i} v - s_i \right)^2, \tag{11}$$

where $n_{x,i}$ and $n_{y,i}$ are the components of the normal at point i on the fragment F, and s_i is the corresponding speed from the spatio-temporal detector. Each point contributes to the solution with a weight, $w(i)$, corresponding to the underlying edge strength reported by the local detector. Here again, we use iteratively reweighted least squares to solve this system:

$$\underbrace{\begin{bmatrix} \sum w n_x^2 & \sum w n_x n_y \\ \sum w n_y n_x & \sum w n_y^2 \end{bmatrix}}_{H} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum w n_x s \\ \sum w n_y s \end{bmatrix}. \tag{12}$$

As with the patch or segment motions, we would like to have a notion of confidence associated with these fragment motion estimates. For example, we are more sure of the motion vector computed for a corner fragment, which constrains the fragment's motion in two dimensions, than for a straight fragment, which is unconstrained *along* the fragment and thus effectively continues to suffer from the aperture problem despite combining multiple estimates via (12). Thus the confidence is tied to the variation of the local normals along the fragment (see Fig. 10). By employing a least squares formulation, our motion estimate for the fragment is the maximum likelihood solution for (u, v) under a Gaussian assumption. The matrix H from (12)—which is constructed from the local normal vectors along the fragment—defines the covariance of that Gaussian, and thus captures our confidence in the estimated motion components. See Sect. 5.1 for further details, where we used G from (8) to model patch motion uncertainty in an analogous setting.

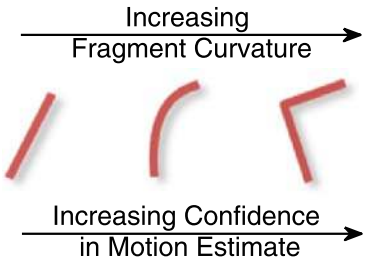


Fig. 10 Our confidence in the estimate of a fragment's motion increases the more corner-like that fragment is

6 A Global Boundary Model

Given segments and fragments from the over-segmentation, the next step is to compute a set of cues associated with each fragment. As before, these cues will be used in estimating the likelihood that each fragment is part of an occluding contour. We will incorporate two general types of cues: motion-based and appearance-based (Stein and Hebert

2007). The specific form of these cues will be explained in Sect. 7.1 below. A similar model and learning approach to the one described here is presented in Hoiem et al. (2007b), where boundary detection is formulated without using motion by instead employing geometric context cues (Hoiem et al. 2005).

For now, assume we have extracted a set of appearance and motion cues for each fragment, using the information found along the fragment and within the segments it separates. Our goal, then, is to classify which fragments are occlusion boundaries. While we hope our local appearance and motion cues will provide strong evidence for this classification, it is unreasonable to hope that a purely local solution will suffice. To facilitate global reasoning and propagation of local estimates, we define here a probabilistic graphical model to capture the structure of our problem.

Our objective is to find an interpretation of the scene's boundaries that is most probable given the observed appearance and motion cues. The toy example in Fig. 11 depicts the type of labeling we will use. Each fragment can take three possible labels. First, the fragment can be labeled "off", indicated by a dashed line, meaning it is believed *not* to lie on an occlusion boundary, *i.e.* it is a surface marking, lighting edge, *etc*. Otherwise it can be labeled "on" (as an occlusion boundary) with the "foreground"—or closer surface—on a specified side indicated by $\texttt{on}_L$ or $\texttt{on}_R$ when appropriate. (We will let the label $\texttt{on}$ represent either, with the appropriate foreground side implied by context.) In the figure, this is indicated by a solid line with an arrow, where the left side of the arrow is the closer side. Where three or four fragments meet, we have a junction, indicated by a solid dot.

Note that fragments along the borders of the image itself are also included in our graph to allow for consistent reasoning about occlusion and closure throughout the image, *i.e.* without requiring special cases at the image borders. There is a special case, however, when a set of segments is fully enclosed within another segment. In this case, two disconnected graphs are formed. Though information will not flow between the two (in this work we have not explicitly avoided this situation), inference as described below can be performed on the two graphs simultaneously. In the extreme case, a single segment could be enclosed within another, and its enclosing fragment would thus be considered and classified completely independently (and therefore entirely based on local information).

To find the most likely labeling of a scene, we need to define the probability of a particular choice of labels for the fragments f given the cues x. Our goal is to maximize this probability, $\Pr(f|x)$, by selecting the best set of *consistent* labels. In our model, we will consider fragments labeled as boundaries and fragments labeled as non-boundaries to be independent. Furthermore, we will assume (1) that each complete boundary, B, is independent of the other boundaries and (2) that each non-boundary fragment is independent of the other fragments. Thus we can approximate the desired probability as

$$\Pr(f|x) \approx \prod_k \Pr(\{f \in B_k\} = \texttt{on}|x) \prod_j \Pr(f_j = \texttt{off}|x). \tag{13}$$

For a particular boundary, which is made up of a set of directed fragments (where the direction indicates the foreground side of the boundary), we assume that each fragment along the boundary is conditionally independent of the other fragments in that boundary, given the preceding fragment. Thus,

$$\Pr(\{f \in B_k\} = \texttt{on}|x) \approx \prod_{\{i \to j\} \in B_k} \Pr(f_j = \texttt{on}|f_i = \texttt{on}, x), \tag{14}$$

where $\{i \to j\} \in B_k$ represents the set of pairs of fragments in boundary B_k such that f_i precedes f_j when traversing that boundary according to its directed labeling. Of course, the "on" labels must also specify the same foreground side in order to be consistent.

If B_k is an open boundary, *i.e.* the boundary of an object which is occluded by another object in the scene (or by the border of the image), the starting fragment of that boundary does not have a preceding fragment on which to condition its probability. So we must consider it separately:

$$\Pr(\{f \in B_k\} = \texttt{on}|x) \approx \Pr(f_{0_k} = \texttt{on}|x) \prod_{\{i \to j\} \in B_k} \Pr(f_j = \texttt{on}|f_i = \texttt{on}, x), \tag{15}$$

where f_{0_k} represents the initial fragment of boundary k. For closed fragments, this term is ignored, as in (14), since the set of fragments forms a loop where each fragment *can* be conditioned on its predecessor.

Now we can compute the probability of a given labeling of all fragments f using (13), (14), and (15). Direct inference using this model, which would require explicit identification and chaining together of open and closed boundaries as well as a varying graph structure depending on the current labeling, would be quite cumbersome. In practice, we can instead simplify matters by focusing on the labeling of fragments and *junctions of those fragments*. Thus we can write the desired probability as a product of fragment potentials and junction potentials,

$$\Pr(f|x) \propto \prod_i \psi(f_i) \prod_k \phi_k, \tag{16}$$

where $\psi(f_i)$ represents the potential function for an individual fragment f_i and ϕ_k represents the potential function for

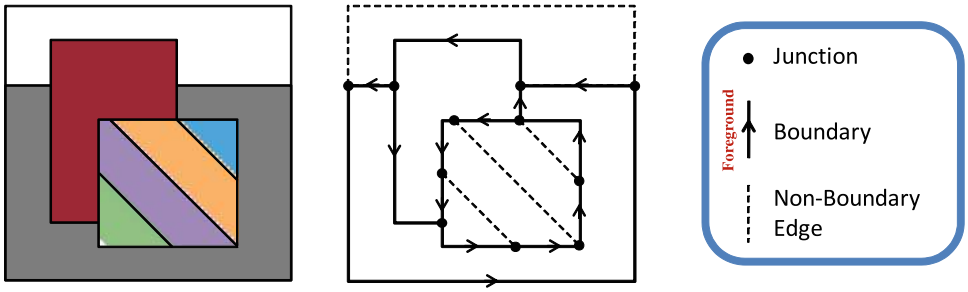


Fig. 11 A toy labeling example showing the fragments of a simple scene labeled with one of three possible labels (non-occlusion or occlusion with specified foreground side). Note that the borders of the image itself are also considered fragments to be labeled

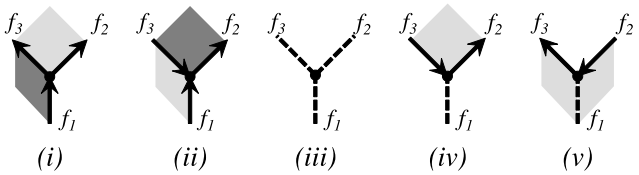


Fig. 12 The five types of *valid* three-fragment junctions. The *shaded regions* are the foreground regions, with *darker* being closer. Similar types exist for four-fragment junctions

the set of fragments meeting at junction k, $\{f_j\}_{j=1}^{N_k}$, where $N_k \in \{3, 4\}$. In practice, note that it is more convenient (and numerically feasible) to take the negative *log* of this probability and work in terms of minimizing an energy function,

$$E(f) = -\log(\Pr(f|x)) = \sum_i \log(\psi(f_i)) + \sum_k \log(\phi_k). \quad (17)$$

Defined carefully, these potentials can capture exactly the same model as in (13), (14), and (15). First, we define the unary fragment potential, based on its fragment's label, as

$$\psi(f_i) = \begin{cases} 1 & f_i = \texttt{on} \\ \Pr(f_i = \texttt{off}|x) & f_i = \texttt{off}. \end{cases} \quad (18)$$

The junction potential ϕ is defined and evaluated for all junctions depending on the labelings of their constituent fragments. A junction of three fragments, each with three labels, would imply a set of 27 possible label combinations for that junction. However, only 11 of those are unique under circular permutations. Of those 11, we consider five to be physically valid and consistent with our model. These are shown in Fig. 12, with the shaded regions indicating foreground (*i.e.* the left side of the fragment, when moving in the direction of the arrow), and the darkest region being the closest one. These junctions consist of a single continuous boundary occluding another boundary (i)–(ii), the meeting of three non-boundary edges (iii), or a single continuous boundary passing through the junction with an adjacent non-boundary edge (iv)–(v).

The other six types of three-fragment junctions are shown in Fig. 13. These junctions are considered "invalid" because the foreground/ background labels of their constituent fragments are inconsistent or physically impossible, as in (vi)–(ix), or because we have chosen to discourage the abrupt starting or stopping of a boundary, as in (x)–(xi), in favor of closed boundaries.

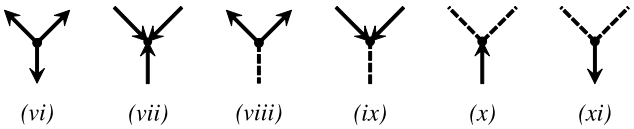


Fig. 13 The six types of *invalid* three-fragment junctions. Similar types exist for four-fragment junctions

Table 1 Junction potentials corresponding to junction types in Fig. 12

Type	Potential (ϕ)
(i)	$\Pr(f_3 = \text{on}\|f_1 = \text{on}, x)\Pr(f_2 = \text{on}\|x)$
(ii)	$\Pr(f_2 = \text{on}\|f_3 = \text{on}, x)\Pr(f_1 = \text{on}\|x)$
(iii)	1
(iv)	$\Pr(f_2 = \text{on}\|f_3 = \text{on}, x)$
(v)	$\Pr(f_3 = \text{on}\|f_2 = \text{on}, x)$

Analogous reasoning is used for the four-fragment junctions as well, though we will not explicitly list all possible labelings here for the sake of brevity.

Now we can define the potentials for each type of junction implied by a particular labeling. First, we strongly discourage the invalid labelings shown in Fig. 13 by setting the potentials of such junctions to a very small value (10^{-4} in our experiments). The five valid junctions in Fig. 12 have potentials shown in Table 1. The combination of these junction potentials with the fragment potential defined in (18) results in the same set of terms in the original model from (13), (14), and (15); *i.e.* we consider "off" fragments independently and effectively walk along the "on" fragments, which make up boundaries, *without double-counting any fragments*. In particular, the definition of certain terms in some potentials as having a value of one—such that they have no effect on the

products in (16)—helps make this possible, at the expense of a somewhat obfuscated model at first glance.

The overall labeling probability $\Pr(f|x)$ can now be computed for any assignment of labels to fragments: for each junction we find the type that is induced by its fragments' labels, and we use the corresponding expression of ϕ to compute the contribution of that junction. The resulting inference problem (finding the assignment of labels f that maximizes $\Pr(f|x)$) is intractable in its exact form on our loopy graph. However, an approximation of the MAP solution can be found by loopy belief propagation (LBP) (Frey 1998; MacKay 2003; Pearl 1982, 1988; Weiss 2000; Yedidia et al. 2005), where the messages passed are based on the potentials defined above for each fragment or junction. Specifically, we use an implementation of the sum-product message-passing algorithm suggested in Heskes et al. (2003), and also exponentiate each potential factor in (16) by $1/T$ (with $T = 0.5$ in our experiments) to provide "soft-max" estimates, according to the mean field approximation suggested by Yuille (2002). Though LBP lacks formal convergence guarantees, we experienced no difficulties in that regard, and always observed convergence to a reasonable minimum-energy solution without restarting or tweaking. (The results provided in Sect. 8 are not specially chosen: they are each the result of a single run of LBP.) Nor was any careful initialization required; we simply initialized all beliefs to be uniform across the possible labels.

Note that this approach bears some similarity to classical approaches to line drawing interpretation (Guzman 1968; Waltz 1975): we have essentially reduced the problem to labeling the junctions using a dictionary of five junction types. A key difference here is that we use soft potentials in a machine learning framework for our reasoning, instead of relying on hard constraint propagation as in the more classical setting.

7 Learned Potentials

Despite the various complex junction cases described in the previous section, note that all the potentials in our model are defined in terms of just two probabilities: a three-class unary probability that a fragment is "on" (and which side is foreground) or "off", $\Pr(f_i = \{\texttt{on}_L, \texttt{on}_R, \texttt{off}\}|x)$, and a pairwise probability that a fragment is "on" given that the preceding fragment is also "on", $\Pr(f_j = \texttt{on}_s | f_i = \texttt{on}_s, x)$, such that the foreground side, s, is consistent. (Though we have not listed them here, one can also define the four-fragment junctions in terms of these same probabilities by following analogous reasoning.) We will now describe a classifier used to estimate these probabilities, as functions of features (x) extracted from labeled training data. For simplicity, we will return to omitting the foreground-side indicator for "on" labels in the following discussion.

We employ Adaboost (Collins et al. 2002) to learn these classifiers, where the weak learners used by the boosting process are decision trees (Friedman et al. 2000; Hoiem et al. 2007b; Stein et al. 2007). We limit each tree to no more than ten nodes and limit the boosting process to no more than ten trees to prevent over-fitting. Passing the results through a logistic function and normalizing such that they sum to one over our classes yields the desired probabilities for each class. Since boosted decision trees are well-suited to feature selection, we can provide a variety of potentially over-complete, or redundant, appearance and motion features, as described below, and allow the classifier to choose the most discriminating ones based on the training data. Such an approach has also been shown to be successful for reasoning about the 3D structure and layout of single images (Hoiem et al. 2007a).

7.1 Motion and Appearance Cues

All of the probabilities used to define the potentials in our factor graph model are conditioned on features or cues, x. In this section, we will describe the corresponding set of appearance and motion cues extracted from the vicinity of the fragments and their associated segments.

For our *appearance* features, which also include some "geometric" information, we provide the following to the unary classifier, $\Pr(f_i = \{\texttt{on}, \texttt{off}\}|x)$:

- Average *Pb*-strength of the pixels along the fragment (Martin et al. 2004).
- Fragment length.
- Ratio of fragment length to the perimeter of the larger of the two neighboring segments. (Intuitively, we may want the classifier to be reluctant to turn "off" a small fragment which would effectively merge one large segment into another.)
- Difference in area between neighboring segments. (This captures some of the same intuition as the previous cue.)
- Euclidean distance between the average colors (in L*a*b* space) of the neighboring segments.
- The χ^2-distance between L*a*b* color distributions provided by kernel density estimates within the neighboring segments. (This is akin to the patch-based edge detection techniques discussed earlier, such as *Pb* strength, but with color only and with regions of support given by the over-segmentation instead of a standard circular patch. Texture/brightness could be added as well.)

Given the fragment and segment motion estimates from Sect. 5, we can compute a set of motion features suitable for our classifier. We compare the fragment motion estimate with each of its neighboring segment motion estimates in addition to comparing the segment motions to each other. For each of these comparisons, we compute the following features to add to the appearance features listed above:

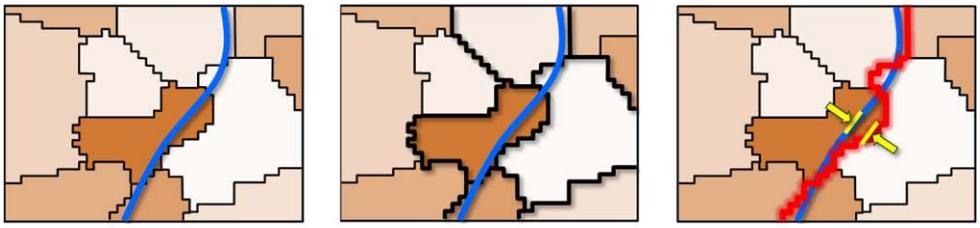


Fig. 14 (Color online) For the over-segmentation and overlaid human-drawn boundary in **a**, we find the segments through which that boundary passes (**b**). In **c**, the fragments associated with those segments (shown in *red*) are then used to approximate the human-specified boundary (shown in *blue*). The error between the two, indicated by the *yellow arrows*, has a mean of 2.6 pixels and a median of 0.8 pixels across our dataset. Substantial errors (deviations greater than ten pixels, *e.g.* where the over-segmentation completely misses a true boundary) occur in only 4.9% of the boundaries, measured pixel-wise

– Absolute difference between individual u and v motion components.
– Simple Euclidean distance between motion vectors.
– Confidences of motion estimates.
– The Mahalanobis-like motion inconsistency score used in Stein and Hebert (2006a).

Once again, these cues are somewhat redundant, deferring the selection of the "right" set of features to the boosted classifier.

For the pairwise classifier, $\Pr(f_j = \texttt{on} | f_i = \texttt{on}, x)$, there are a few additional pair-specific features we can use. Thus the feature set for this term includes the unary cues above for each fragment of the pair, augmented by the following:

– The relative angle between the two fragments (to capture a notion of continuity).
– The difference between the motions of the two fragments.
– The motion and color differences between the two fragments' foreground-side segments.

7.2 Training the Classifiers

Given a set of training data, we apply the over-segmentation and fragment-chaining approaches described in Sect. 4 and compute each cue defined above. We then train the *unary* classifier, $\Pr(f_i = \{\texttt{on}, \texttt{off}\} | x)$, directly from the individual ground truth labels using the extracted set of cues.

For the *pairwise* classifier, we first extract pairs of fragments in the ground truth for which f_j follows f_i and both are labeled "on". These pairs are our positive examples while negative examples are collected from those pairs for which f_j is "off" but is connected via the graph to an f_i that is "on". From these examples, we learn the second classifier, $\Pr(f_j = \{\texttt{on}, \texttt{off}\} | f_i = \texttt{on}, x)$. For all our experiments, we allow ten iterations of boosting with ten-node decision trees.

For training each of these classifiers, we must match the fragments created by our over-segmentation with ground truth boundaries drawn by hand. This is accomplished by first determining through which segments the hand-drawn boundaries pass. Then, the fragments corresponding to those segments are used to get the closest approximation to the ground truth boundary. This process is depicted in Fig. 14.

In Figs. 15 and 16, we plot histograms indicating the frequency with which each of our features was chosen by the boosted decision tree classifiers learned in the experiments below. Since features are chosen for use by a decision tree according to their discriminative ability in classifying available examples, these histograms provide a notion of which features are most useful across the dataset for discriminating the classes present in our problem. Anecdotally, these feature usage frequencies were found to be quite stable across multiple runs with varying splits of the data. Note that each classifier frequently uses motion *and* appearance features. In fact, the two most-used features employed by the unary classifier (Fig. 15) are the Euclidean distance between segment motion vectors and the difference in color distributions on either side of the fragment. Furthermore, the classifier regularly uses different forms of the confidence information derived from computing segment and fragment motion estimates.

Somewhat surprisingly, the relative angle feature used for capturing the continuity between a pair of fragments is not often utilized by the pairwise classifier, despite being a common Gestalt cue for reasoning about saliency and continuation in perceptual organization, *e.g.* (Leung and Malik 1998; Mahamud et al. 2003; Ren et al. 2005) and references therein. Its limited utility here may be tied to the fact that the graph implied by our over-segmentation procedure, which seeks regularly-shaped segments, tends to have plausible relative angles for almost *any* pair of fragments meeting at a given junction. Thus, the relative angle may not be a very *discriminating* feature in our construction.

Finally, the addition of features based on *fragment* motion, rather than only patch or segment motion as employed by Stein and Hebert (2006a, 2007), does appear to be helpful. Fragment motion cues are used regularly, especially those estimated for fragment j in the pairwise classifier.

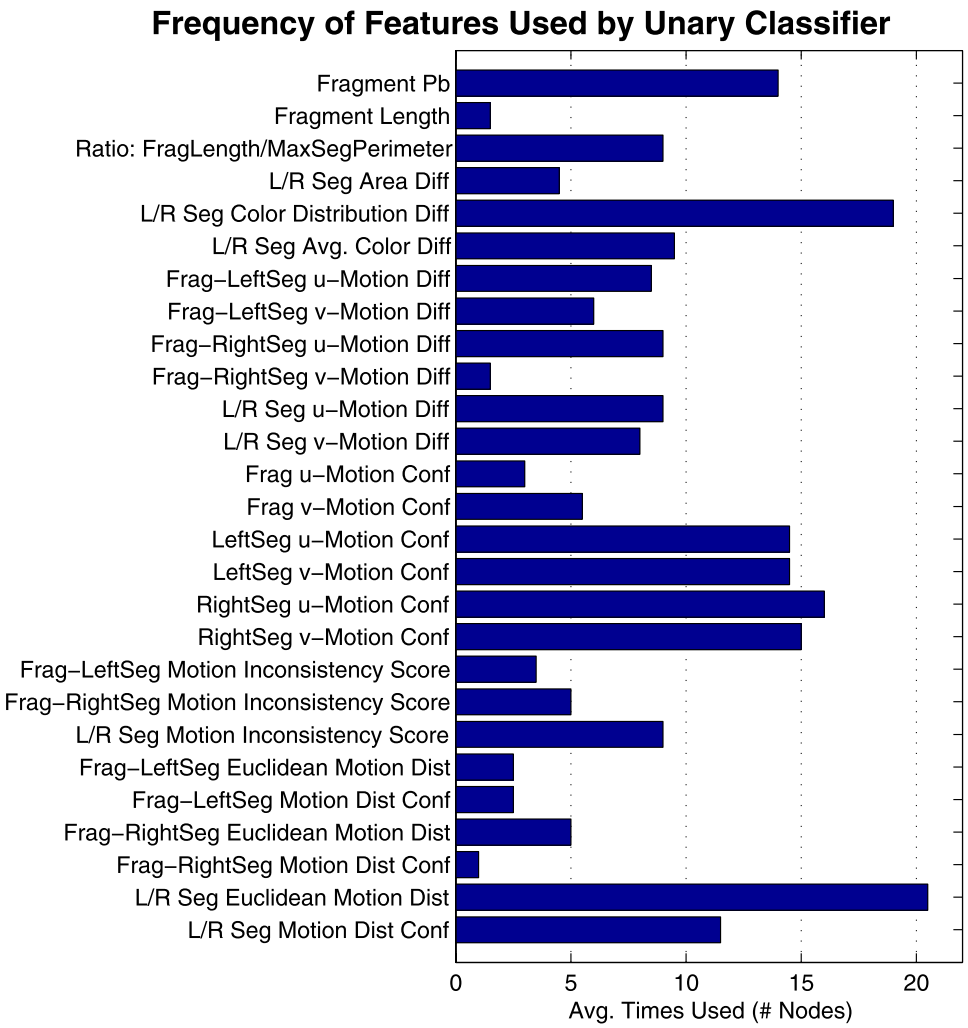


Fig. 15 Frequency with which each computed feature is used by the learned unary fragment classifier

8 Experiments

We first over-segment each scene's reference frame and extract fragments as described. There are approximately 20,000 total fragments available in our dataset for testing and training. For each fragment and segment, we compute the set of appearance and motion cues listed in the previous section. From a training set constructed using half the scenes in the database, we then learn the two classifiers which define the potentials used by our model. Using these classifiers for the other half of the scenes, we can estimate the probability of labeling each fragment as an occlusion boundary, first by using the learned unary classifier in isolation and then within the global inference model. We repeat this procedure, swapping the test and training sets in order to obtain results for all scenes in the database.

We would like to verify that both global inference and motion information result in improved performance overall. We see in Fig. 17 that this is indeed the case in a plot of precision versus recall for final fragment labelings of the entire dataset in aggregate. The parameter varied in creating each plot is the threshold on the likelihood of each fragment being "on". We see that using appearance cues alone results in the worst performance on the graph.[3] In fact, note that including motion cues but only reasoning on individual fragments, *without* global inference, offers equivalent or superior results to using global inference with appearance cues alone. Finally, global inference with combined motion and appearance information consistently yields the highest precision. Also note that the low precision point at 100% recall (corresponding to the trivial solution of labeling *all* fragments as occlusion boundaries) provides some indication of the difficulty of our task and our dataset.

While aggregate statistics captured by precision-recall plots are useful for understanding performance in general, they do hide important semantic measures of quality which can only be understood by looking at individual results. In

[3] We do not show results using motion cues alone, but as with the simpler classifier used in Stein and Hebert (2007), this yields similar or worse performance than using appearance alone

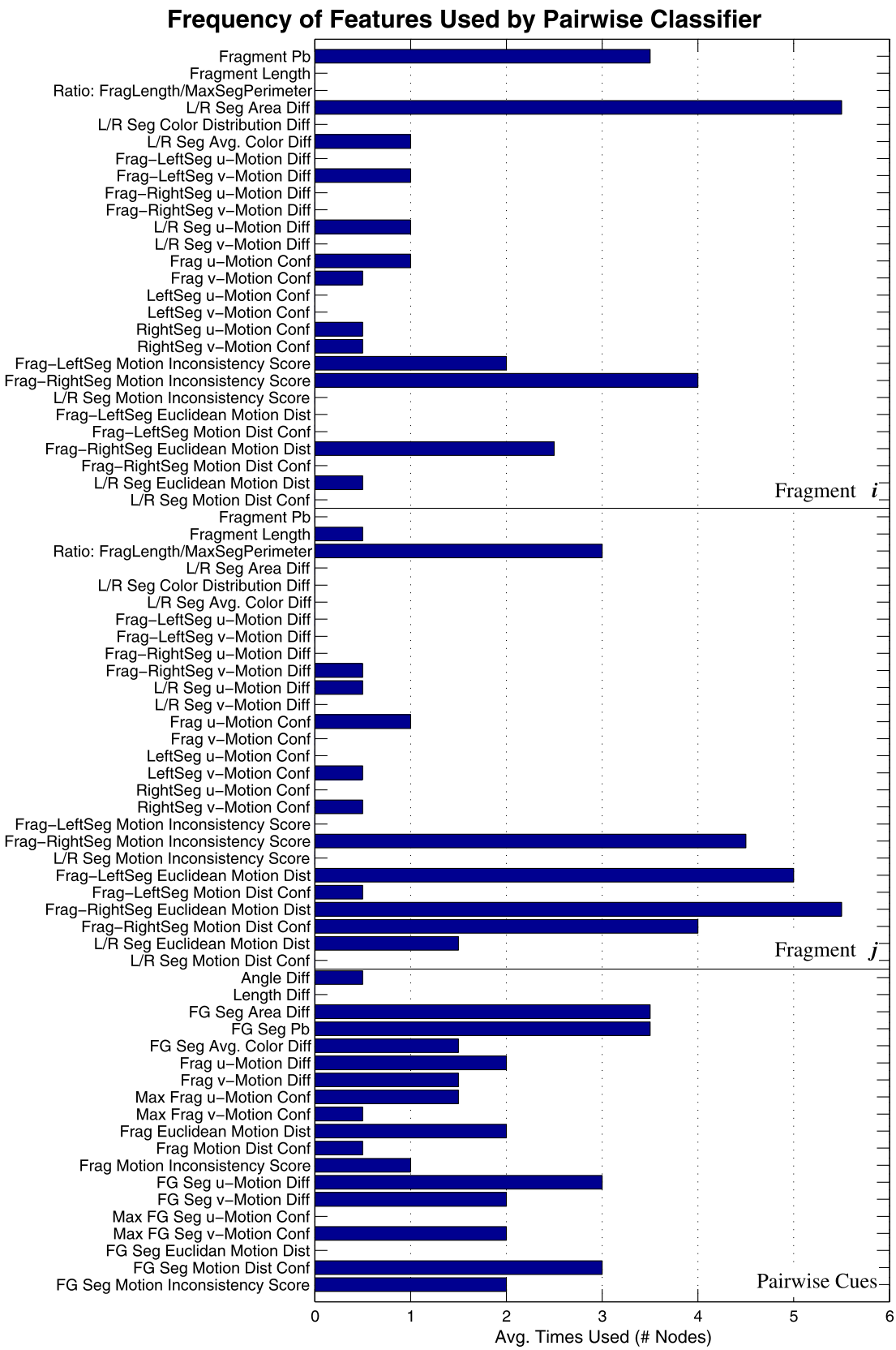


Fig. 16 Frequency with which each computed feature is used by the learned pairwise fragment classifier

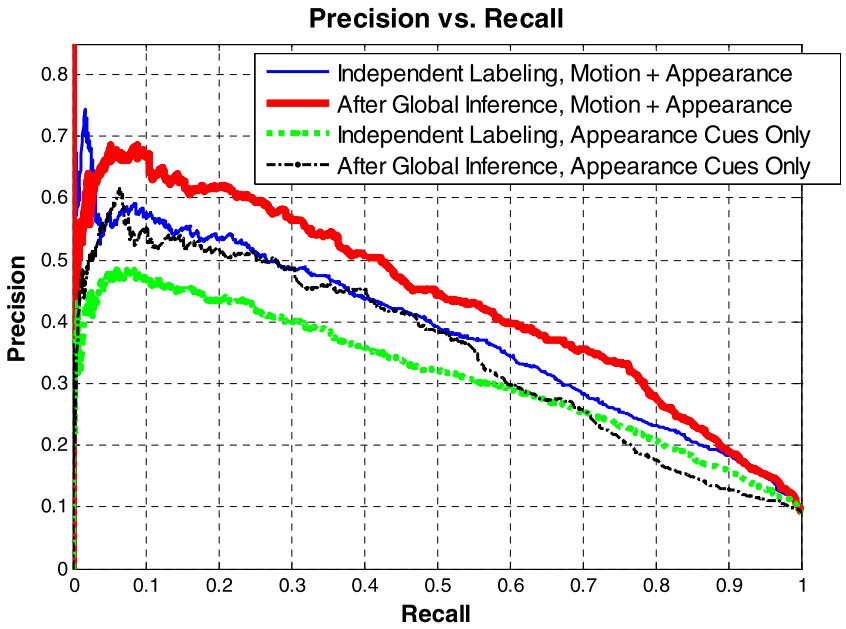


Fig. 17 Precision vs. Recall for the entire dataset, showing that using motion and global inference results in the most accurate identification of those edge fragments which are occlusion/object boundaries

Figs. 18–21, we provide a few such examples out of the 30 in our database. (Additional examples can be found in Appendix B.) In each, the reference image of the sequence and the ground truth labeling are provided in the top row. In the remaining rows we compare the use of appearance only (left column) to that of appearance and motion combined (right column). The second row displays fragments overlaid on the image with brightness and line width proportional to the confidence that they are occlusion boundaries according to their *independent* classification results (*i.e.* before global inference). Thus, the brighter red and thicker a fragment is, the more the system believes it to be an occluding boundary. The next row displays the same type of result but *after* performing global inference on the initial classifications. The final two rows show these global inference results thresholded at equal recall rates for fair comparison. Note how the motion-plus-appearance approach sustains higher precision (*i.e.* fewer false positives) even as the recall is increased. This indicates that the motion adds significant confidence to the classifier's decision.

In Fig. 22, we provide an example of a scene on which performance is consistently poor using our approach. This stems largely from two problems. First, the extremely harsh lighting creates high-contrast edges which result in an oversegmentation whose fragments do not correspond to the true occlusion boundaries in the scene. In such cases (which are rare, see Fig. 14), there is no hope of the subsequent classifier succeeding, since the true boundaries are not even in the set of fragments to be classified. Secondly, the lack of texture (in combination with the lighting) results in several false positives since it prevents confident motion estimation which could otherwise be used to disambiguate and correctly classify lighting-based edges as *non*-boundaries.

Due to the variety of scenes in our dataset, it is not surprising that there exist cases for which motion does not help. Some object boundaries simply *are* easily identified by basic appearance cues, such as color, and the scenes may lack enough texture or depth variation to provide the necessary relative motion cues. However, it is also very rare that using motion *hurts* performance, and in those cases where appearance information simply does not capture the properties of occlusion boundaries well, motion cues often provide substantial improvement. This improvement is realized in reduced false positives since, in many cases, only motion information may allow the system to recognize and filter out high-contrast surface markings which confuse an appearance-only approach.

9 Conclusion and Future Directions

In this work, we presented a mid-level framework capable of reasoning more globally about the existence of occlusion boundaries. In so doing, we alleviated deficiencies in earlier purely-local approaches, such as Stein and Hebert (2006a, 2007), by improving spatial support for feature extraction and motion estimation by using an initial oversegmentation of the scene, by incorporating boundary motion as well as motion confidence information, and by introducing a graphical model with learned classifiers as potentials for propagating local information derived from a combination of appearance and motion cues.

As discussed in the introduction, we have focused mainly on reasoning *at* boundaries themselves. The regions those boundaries enclose could be further integrated into a system capable of utilizing both the techniques presented here and

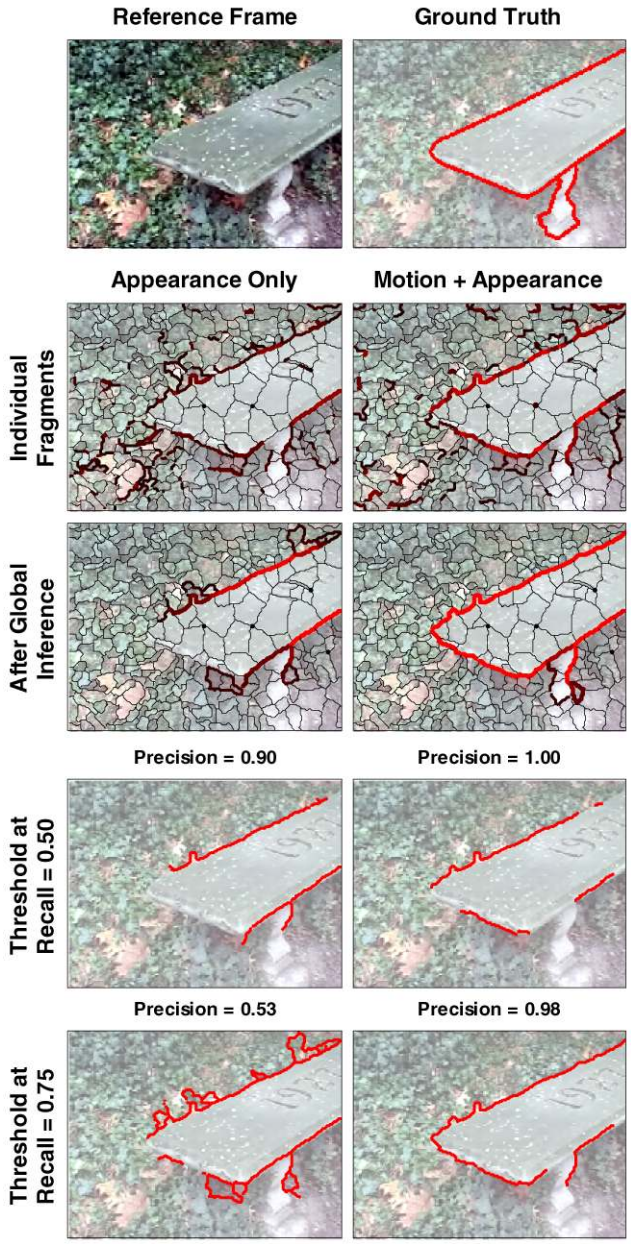


Fig. 18 Example result: The appearance-only classifier's lack of confidence becomes obvious when we use a higher-recall operating point. With the addition of motion, very high precision is maintained

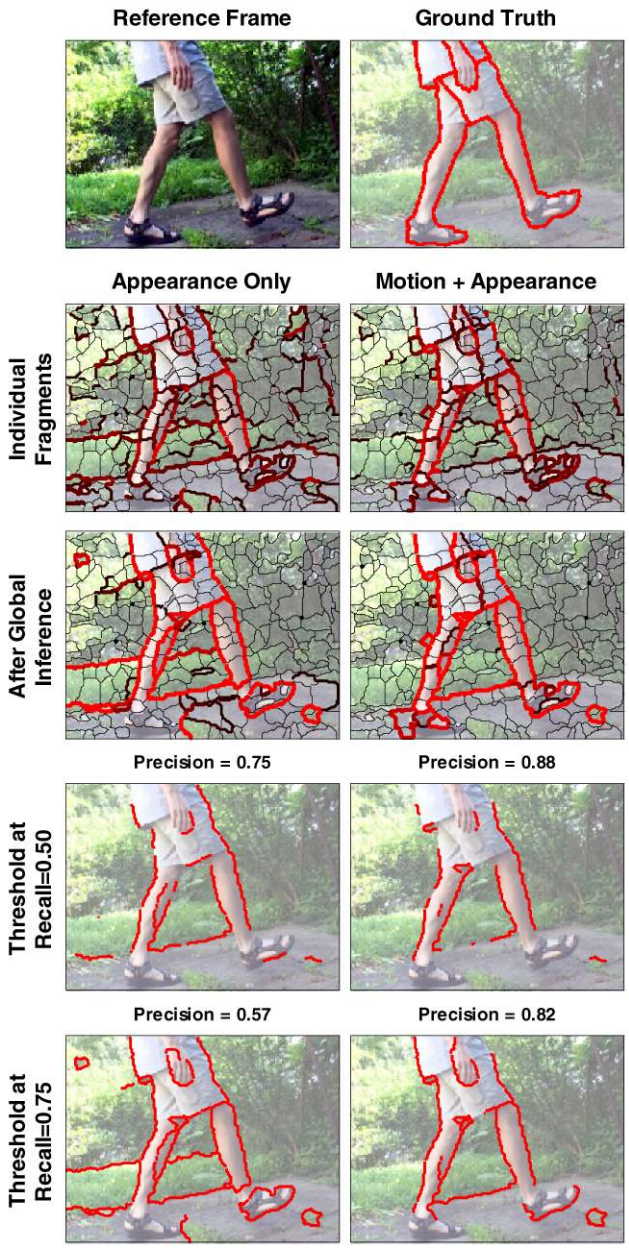


Fig. 19 Example result: Using motion allows for sustained high precision, even at higher recall

existing region-based methods. Moreover, one avenue of research here might be the feedback of our detected boundaries into the initial boundary hypothesis procedure. Confident boundary estimates could be used to merge initial segments, thereby producing longer fragments and larger regions of support for features (Hoiem et al. 2007b).

We have not performed specific analysis and evaluation of figure/ground assignment to our boundaries in this work. While a notion of the foreground side of a boundary was an important part of label propagation in our model in Sect. 6, our focus has mainly been on the *detection* of boundaries. In our experience, our method generally achieves qualitatively *consistent* foreground assignment along boundaries, but sometimes the entire boundary's foreground/background labeling is reversed. We believe this to be partly an artifact of impoverished local cues combined with the limited ability of existing inference techniques to propagate local information effectively over large distances on dense, loopy graphs. Further incorporation of ideas from Ren's work (Ren 2005, 2006), which focuses on the figure-ground assignment problem *given* the object boundaries, may prove beneficial here.

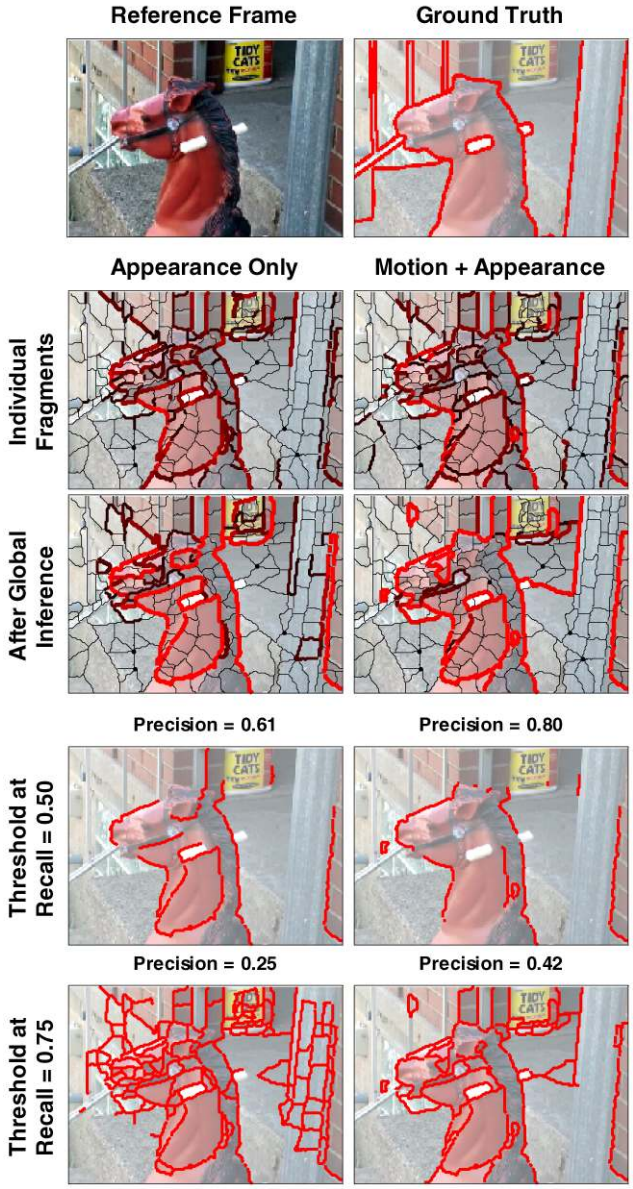


Fig. 20 Example result: On this more difficult example, the motion+appearance classifier still performs best

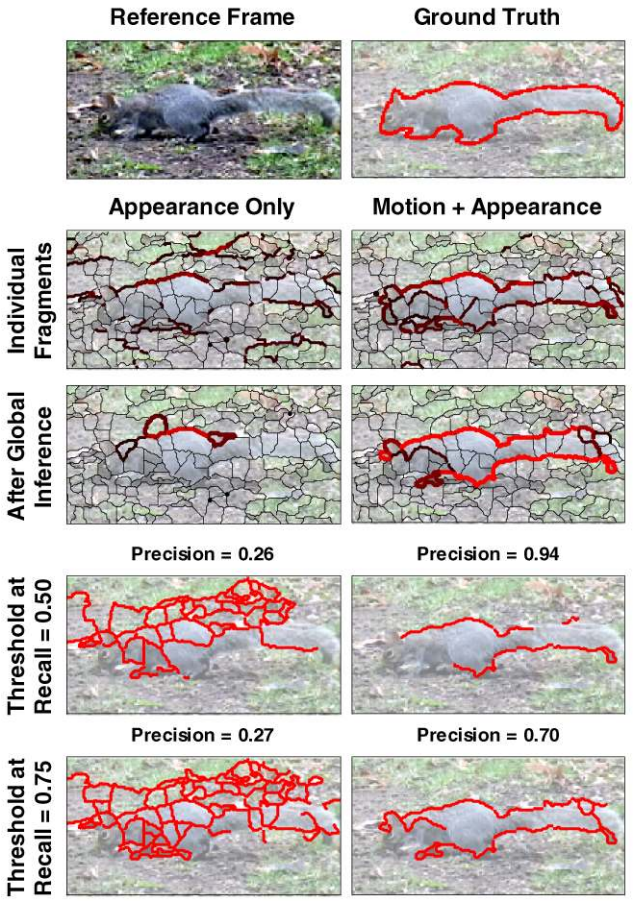


Fig. 21 Example result: The squirrel in this scene is nearly invisible to the appearance-based classifier, but its movement makes its boundaries much more readily detectable when also using motion cues

The classifiers needed to compute the potentials for our factor graph model are currently trained independently (from each other and from the inference procedure itself). It could be beneficial—though also quite expensive—to train them together and within an inference loop. This is akin to the training of parameters in Discriminative Random Fields (Kumar and Hebert 2006) and could help avoid the over-confident fragment classifications and the over-eager closure of small boundary loops we have sometimes observed in our experiments.

Finally, thus far we have sought to discover the occlusion boundaries present in a single reference frame of a short video clip. A next step could be the application of our techniques on a per-frame basis (or every nth frame), with additional temporal reasoning/filtering, in order to utilize more fully the temporal aspect of video data. One difficulty in this domain may be the labeling of several (or all) frames of video for quantitative analysis or training.

Appendix A: Crack Chaining

In this appendix we present in more detail the crack chaining approach used in extracting potential boundary fragments from an over-segmentation, as discussed in Sect. 4.

Consider the example segmentation of a simple image consisting of 4×4 pixels at the left of Fig. 23. There are four segments, each indicated by a different color, with thicker lines at their borders which lie along the cracks of the image's pixels. In the center of the figure, the four-way intersections of the pixels, where four cracks also meet, are labeled with circles. At the right, the corresponding "crack graph" is shown, whose vertexes and edges correspond to the pixel intersections and the cracks, respectively.

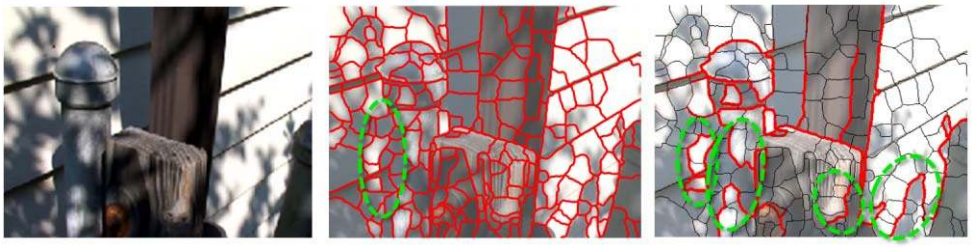


Fig. 22 A harshly-lit, low-texture scene (*left*) causes difficulties for our approach. The *middle* example shows the over-segmentation failing to offer the true occlusion boundary as a hypothesized fragment for classification. At the *right*, we see false positives resulting from a combination of harsh lighting and lack of texture which would provide good motion estimates

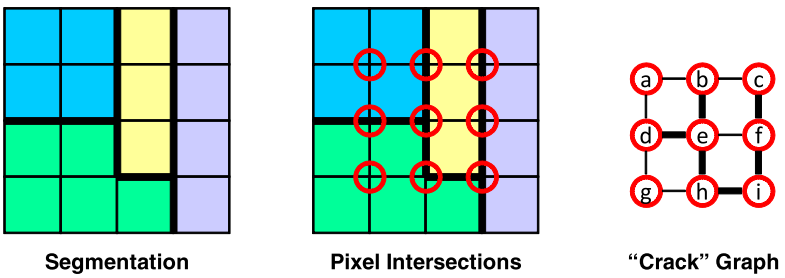


Fig. 23 The cracks between the pixels in a segmentation form a graph we can use for chaining

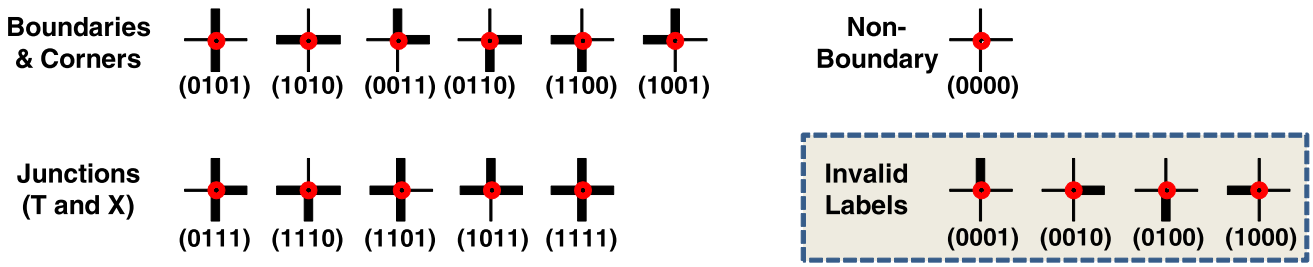


Fig. 24 The possible labels for each type of pixel intersection, based on the binary labelings of their four associated cracks. Some labels are invalid in our approach, which seeks closed boundaries

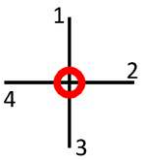


Fig. 25 Bits corresponding to each crack at an intersection

Each crack in this graph can take on a binary label indicating whether or not it corresponds to a segment border in the original segmentation, again indicated by thicker lines. Furthermore, each vertex (a)–(i) can take on one of exactly twelve possible labels, depending on the labeling of its four cracks. These labels, representing the presence of local junctions, (non-)boundaries, and corners, are shown in Fig. 24. Note that the four cases corresponding to a single crack being labeled as a segment border are not possible when working with the closed borders of an over-segmentation. (Though not indicated in Fig. 23, we consider the borders of the image to be segment borders in practice.)

If we associate each crack at a particular vertex with a bit in that vertex's four-bit label, as indicated in Fig. 25 and below each intersection shown in Fig. 24, we can efficiently chain together intersections of the graph by simple logical bit checks. For example, we know vertex d in the graph from Fig. 23 is chained to vertex e by simply checking that the second bit of d's label and the fourth bit of e's label are both set. Vertex d is *not* connected to g, however, since neither d's third bit nor g's first bit are set.

Based on this reasoning, we start at those intersections labeled as junctions (those with three or more bits set) and chain along the crack graph until reaching another junction. Note that it is also necessary to consider the special case of a segment completely enclosed by another, larger segment. In this case, a single fragment encircles the whole inner segment, and does not begin or end at a junction.

The cracks linked together in this way form the graph of "fragments" used in our work. Note that there are no thresholds (*e.g.* on edge orientation variation, edge strength, *etc.*)

Fig. 26 Chair

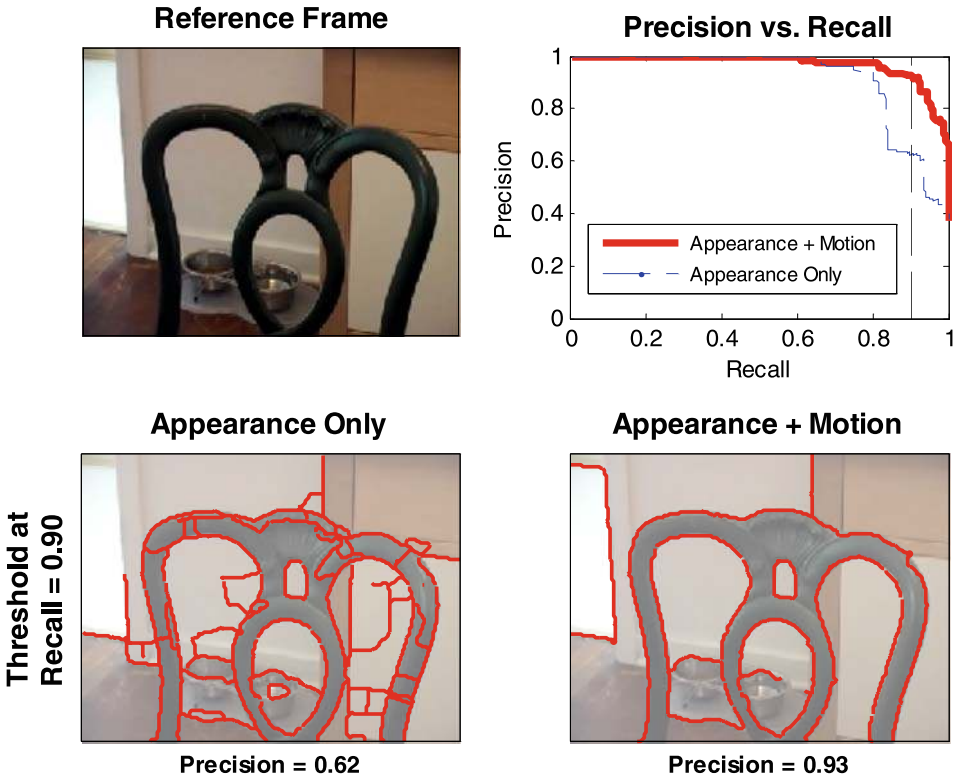
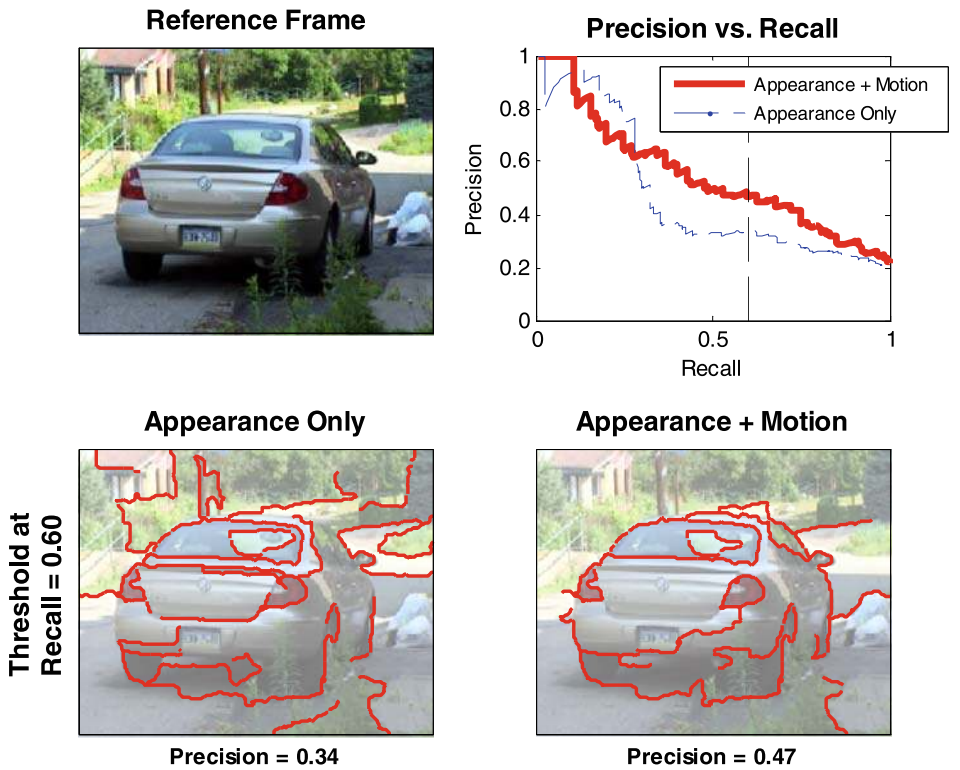


Fig. 27 Car



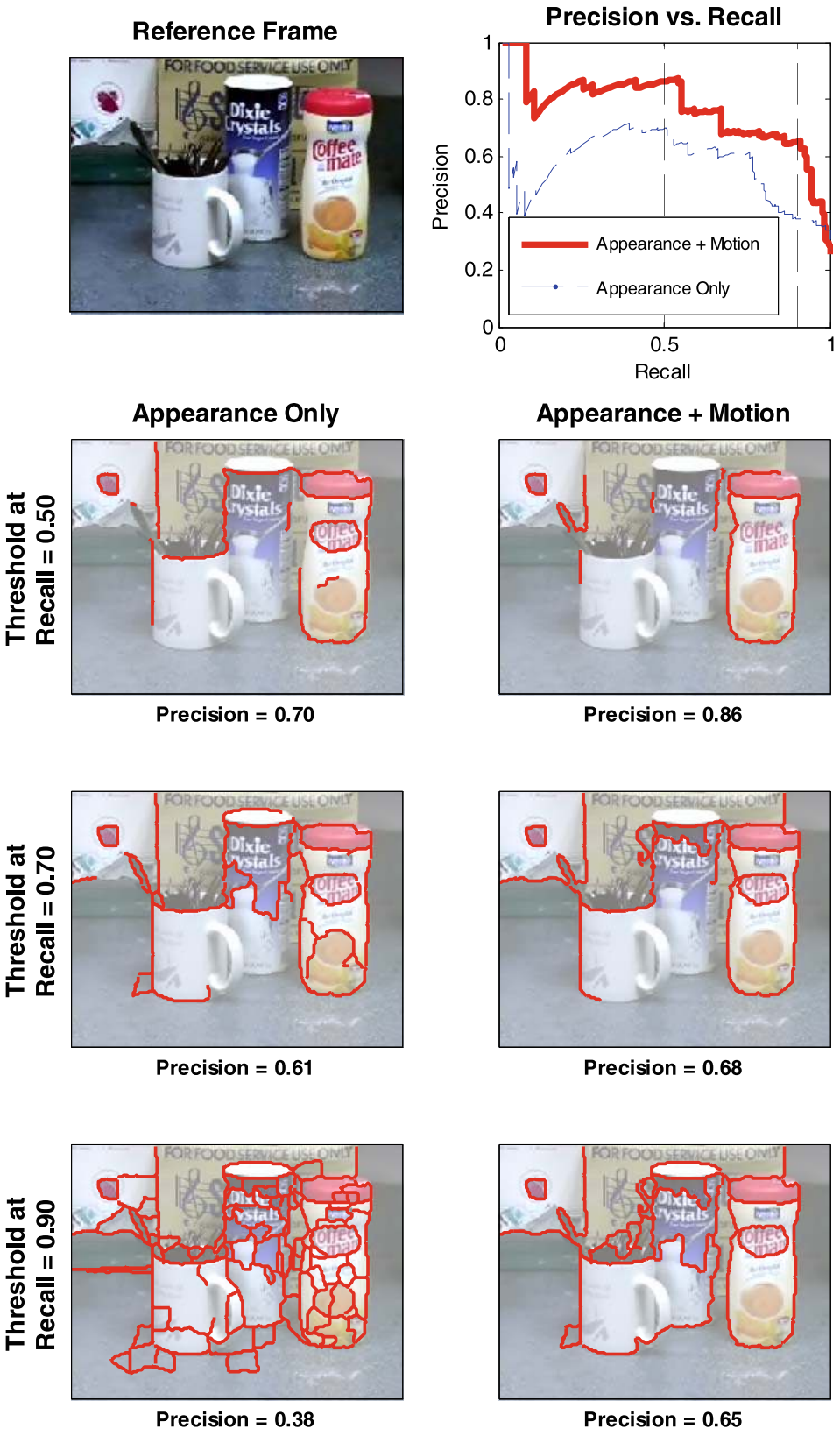


Fig. 28 Coffee stuff

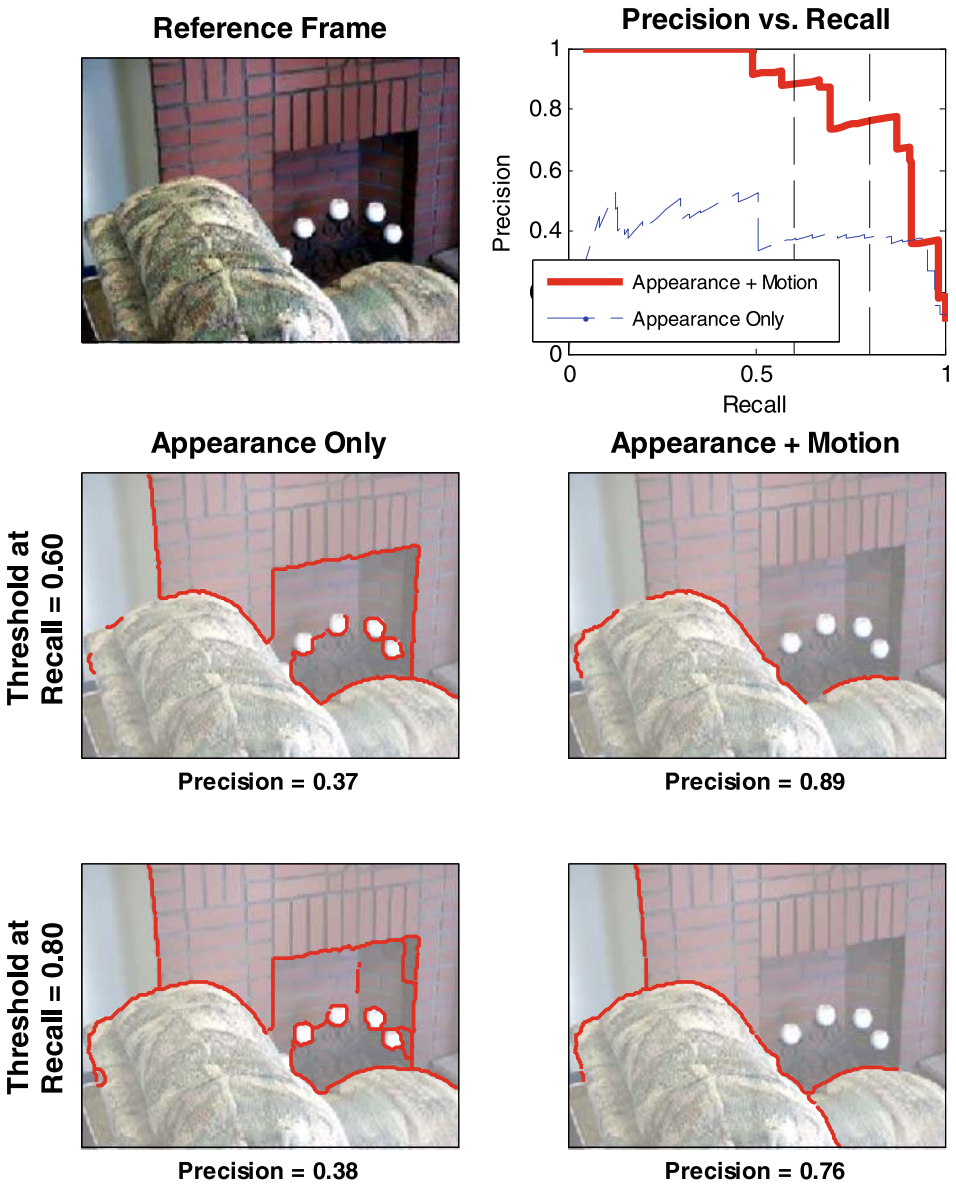


Fig. 29 Couch corner

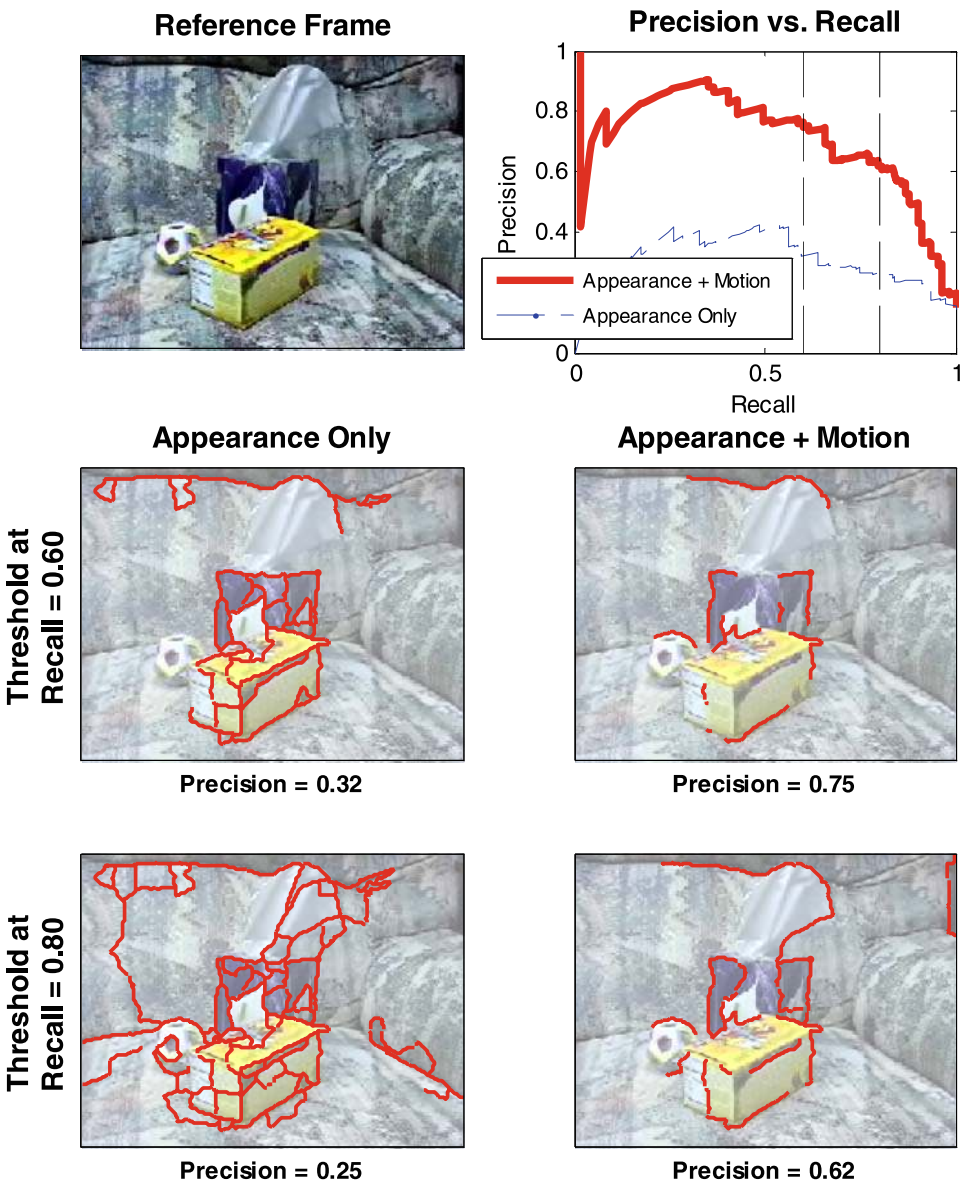


Fig. 30 Couch objects

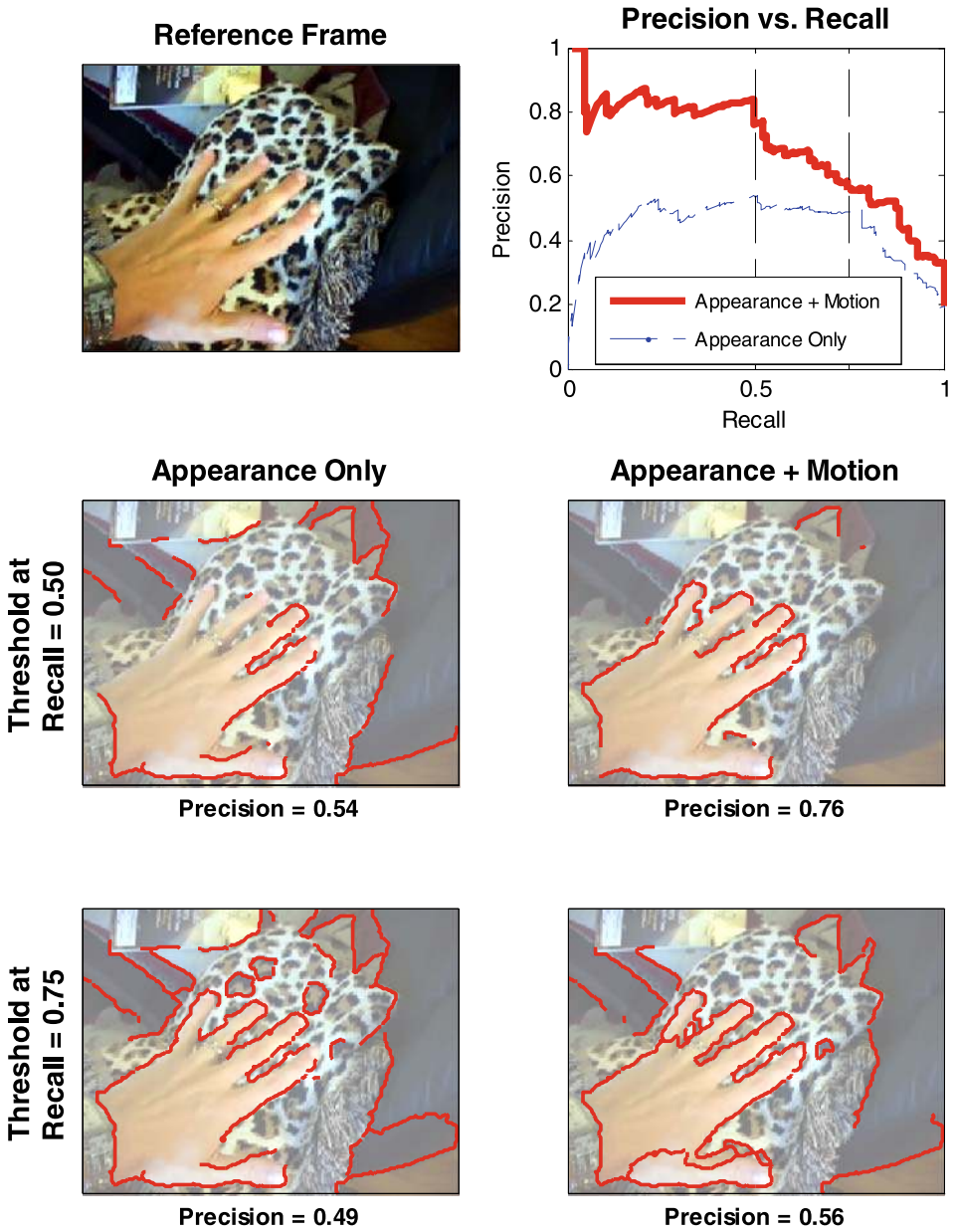

Reference Frame
Precision vs. Recall
Precision
Recall
Appearance + Motion
Appearance Only
Appearance Only
Appearance + Motion
Threshold at Recall = 0.50
Precision = 0.54
Precision = 0.76
Threshold at Recall = 0.75
Precision = 0.49
Precision = 0.56

Fig. 31 Hand

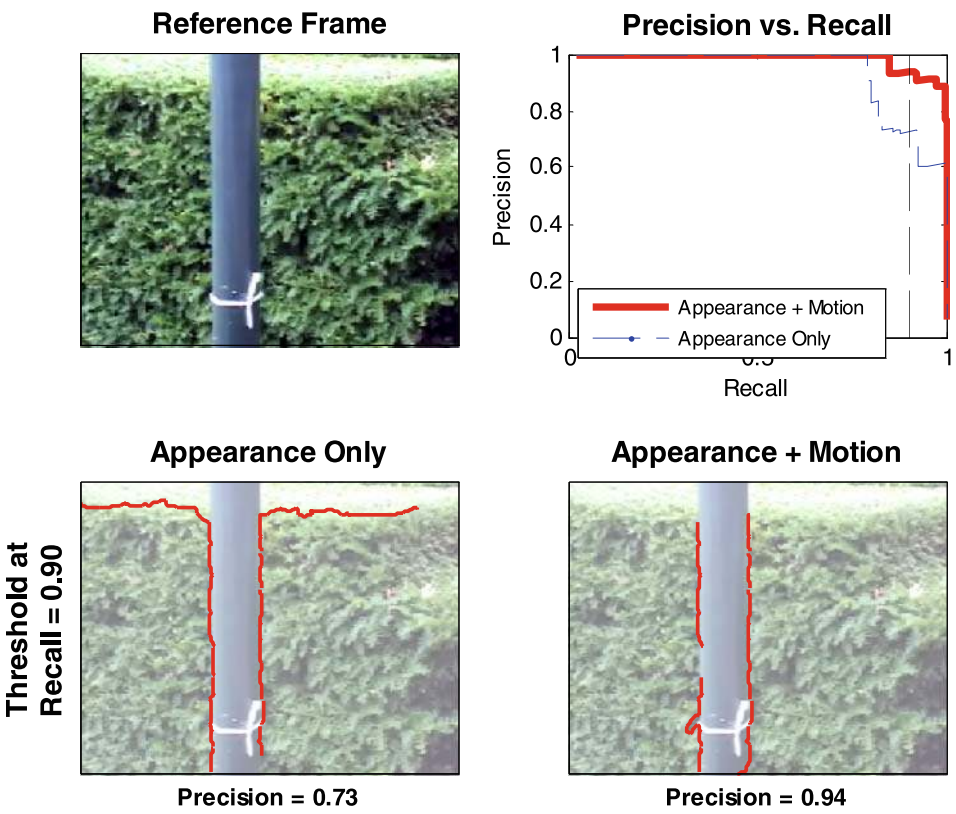

Reference Frame
Precision vs. Recall
Precision
Recall
Appearance + Motion
Appearance Only
Threshold at Recall = 0.90
Appearance Only
Appearance + Motion
Precision = 0.73
Precision = 0.94

Fig. 32 Post

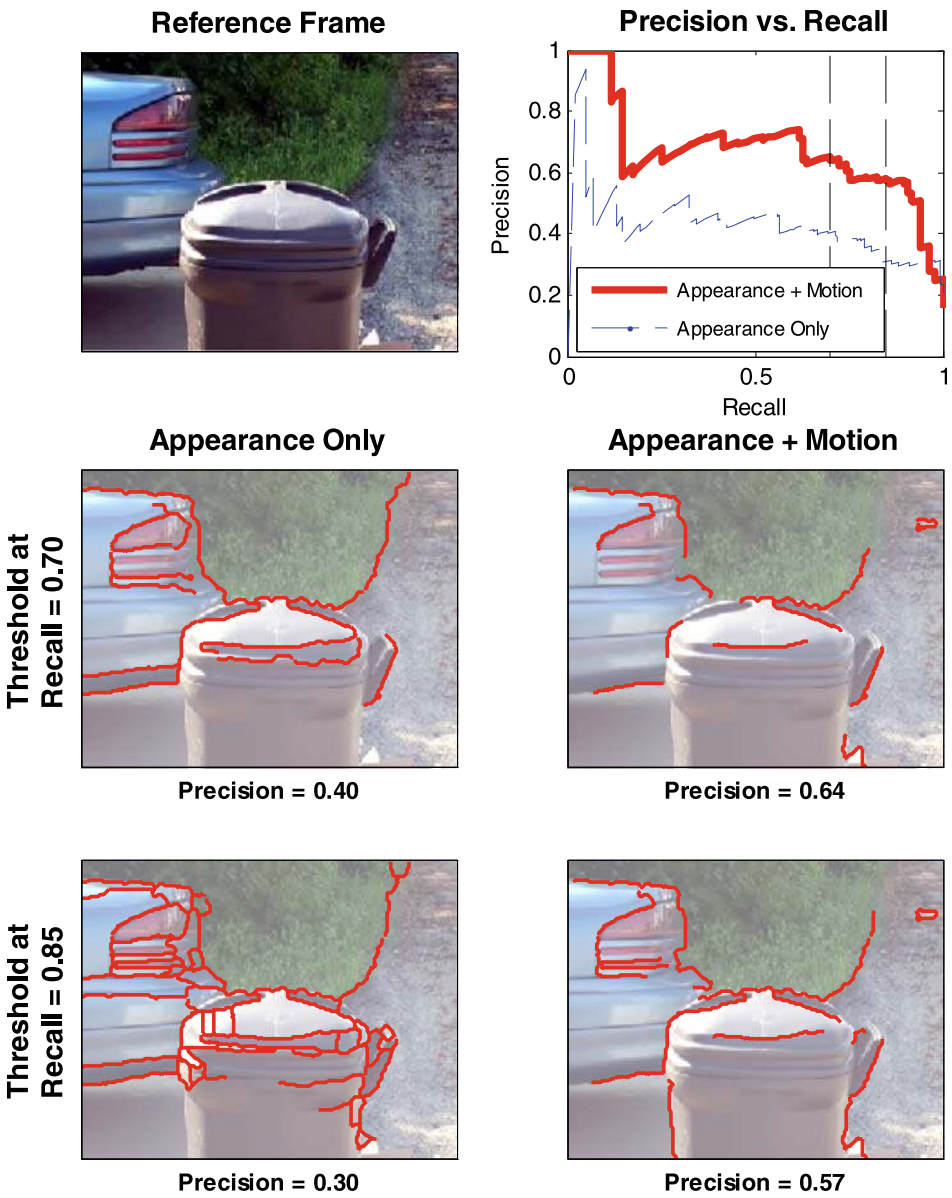
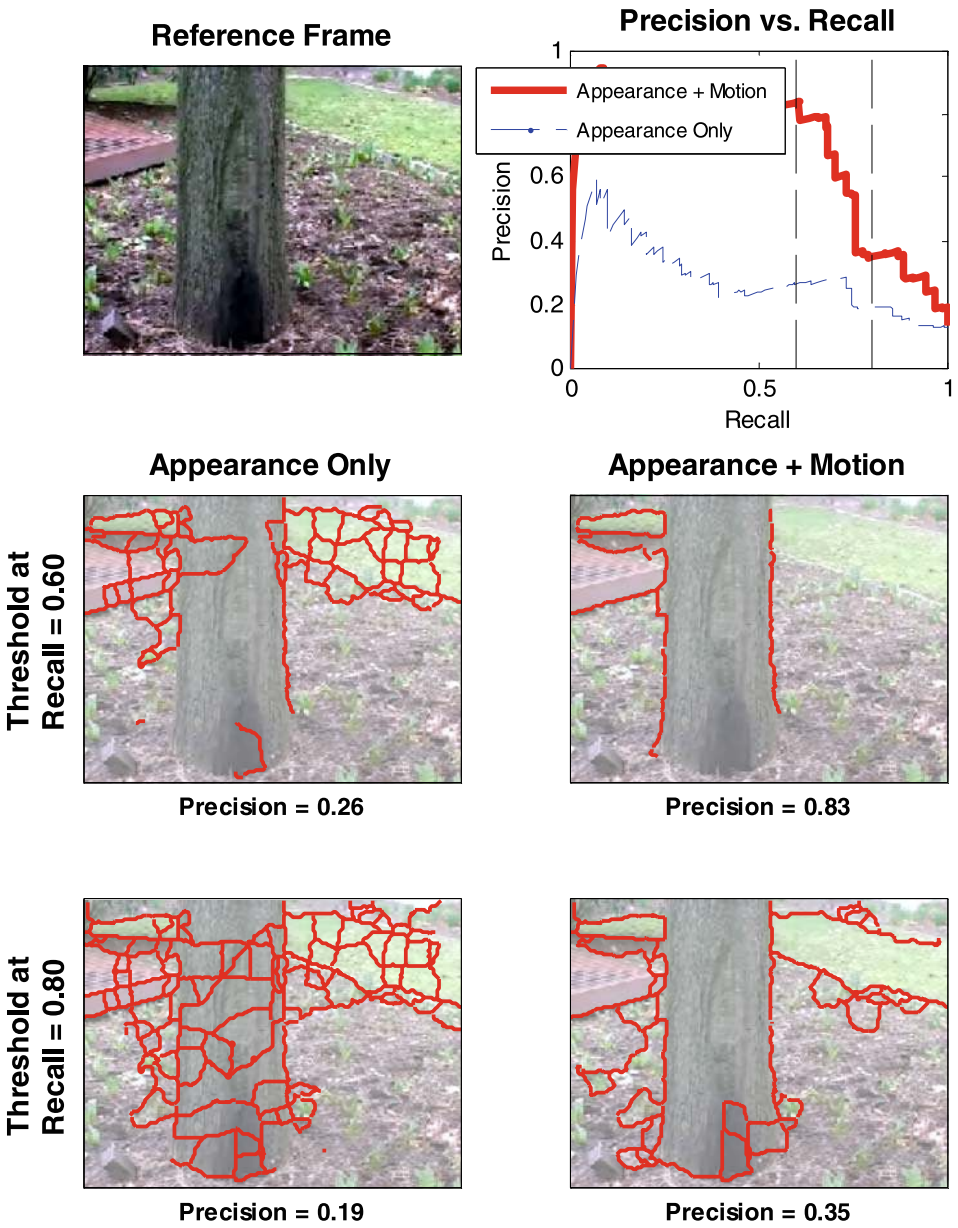


Fig. 33 Trash can

Fig. 34 Tree


Reference Frame
Precision vs. Recall
Appearance + Motion
Appearance Only
Precision
Recall
1
0.6
0.4
0.2
0
0.5
Appearance Only
Appearance + Motion
Threshold at Recall = 0.60
Precision = 0.26
Precision = 0.83
Threshold at Recall = 0.80
Precision = 0.19
Precision = 0.35

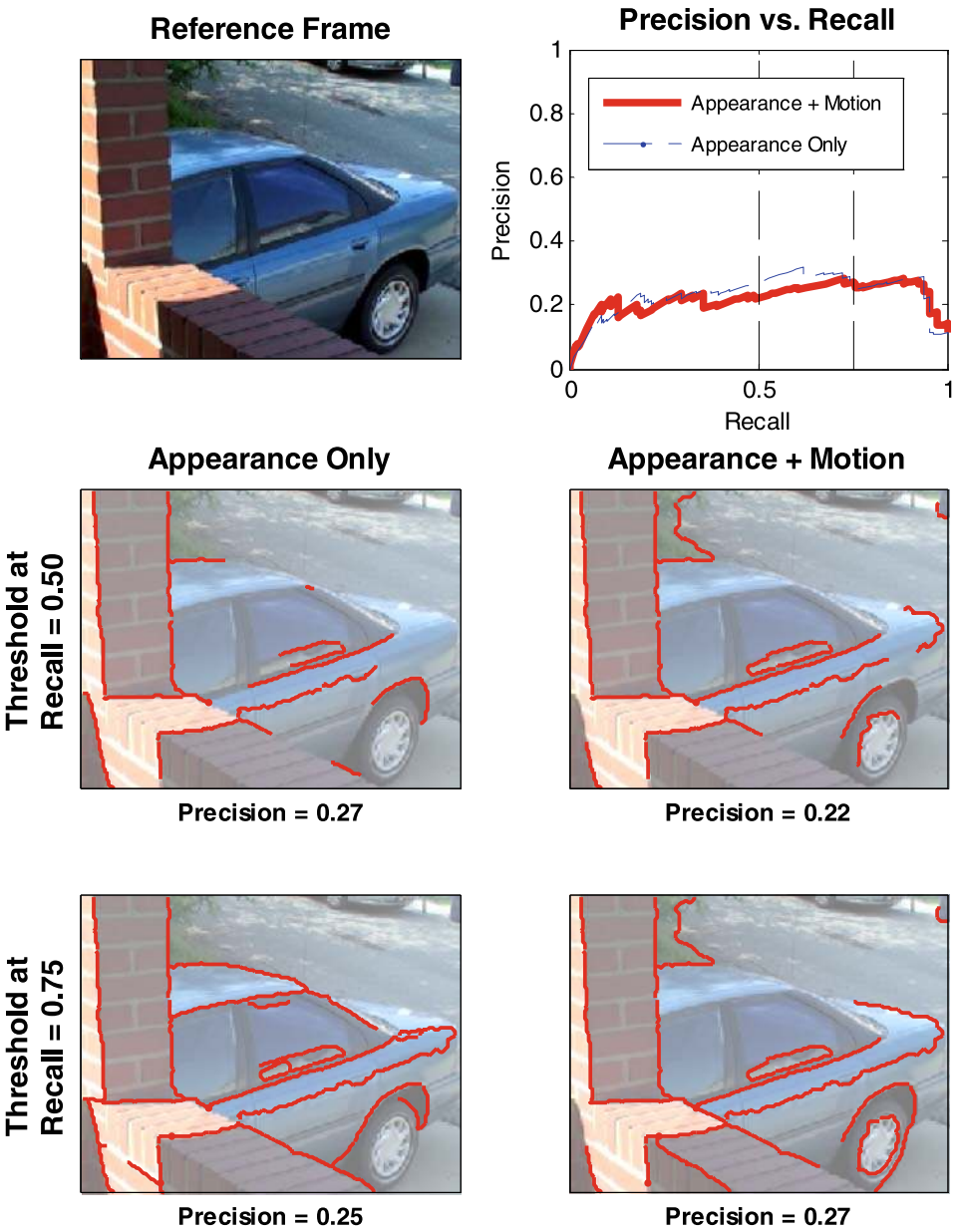


Fig. 35 Car (difficult case)

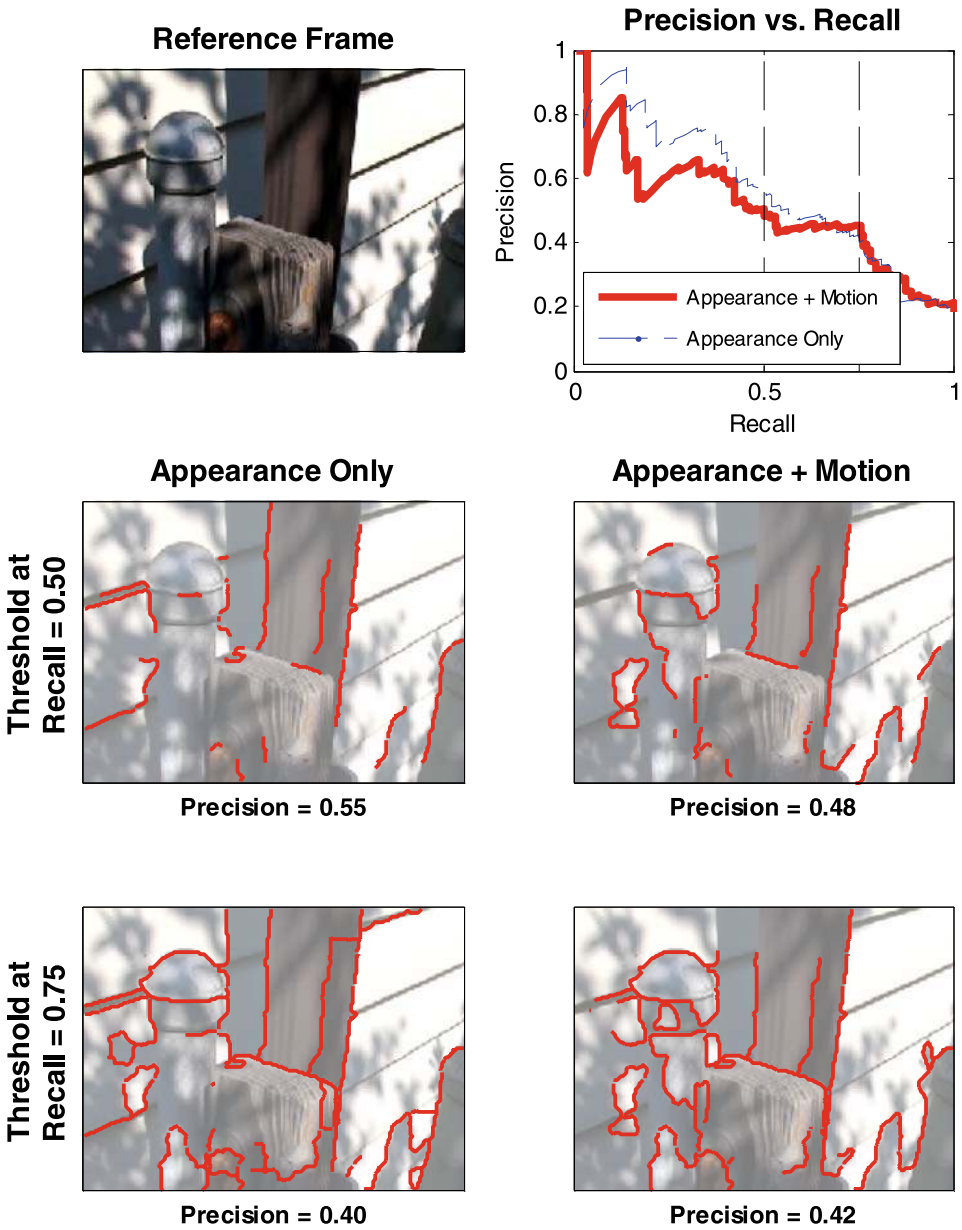


Fig. 36 Fence (difficult case)

required and that the resulting fragments will naturally form closed graph structures since they are based on an underlying over-segmentation of the image.

Appendix B: Additional Boundary Detection Examples

We include here additional examples of boundary detection on our dataset using the approach described in this paper. For each example below, we provide the middle (reference) frame from the video clip, the precision versus recall curve, and boundary detections at various recall operating points (which are indicated on the precision vs. recall plots by vertical dashed lines). We show thresholded boundary detections after global inference when using appearance information alone (left column) and when appearance and motion information are combined (right column). We have chosen operating points for each example to highlight the performance increase when using motion information (*i.e*. when using motion, the number of false positives is generally lower—and thus the precision is higher—for a given recall point, as compared to using appearance information alone). Note that the selected recall point is always the *same* between the columns of a given example, to facilitate fair comparison.

The last two figures (Figs. 35 and 36), show difficult examples. In these cases, very harsh lighting conditions create high contrast shadow edges that confuse our system. In addition, large un-textured regions combined with reflections and specularities visible on the car make motion estimation difficult.

Acknowledgements Partial support provided in part by National Science Foundation Graduate Fellowship and Grant IIS-0713406, and by the 21st Century Frontier R & D Program of the Korean Ministry of Commerce, Industry, and Energy. Any opinions, findings, and conclusions or recommendations expressed in this material are solely those of the authors.

References

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

Adelson, E. H., & Bergen, J. R. (1991). The plenoptic function and the elements of early vision. In M. Landy & J. A. Movshon (Eds.), *Computational models of visual processing* (pp. 3–20). Cambridge: MIT Press. Chap. 1.

Arbeláez, P. (2006). Boundary extraction in natural images using ultrametric contour maps. In *IEEE computer society workshop on perceptual organization in computer vision (POCV)*.

Barron, J. L., Fleet, D. J., & Beauchemin, S. S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision (IJCV)*, *12*(1), 47–77.

Black, M. J., & Fleet, D. J. (2000). Probabilistic detection and tracking of motion discontinuities. *International Journal of Computer Vision (IJCV)*, *38*(3), 231–245.

Bouthemy, P. (1989). A maximum likelihood framework for determining moving edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *11*(5), 499–511.

Brostow, G., & Essa, I. (1999). Motion based decompositing of video. In *IEEE international conference on computer vision (ICCV)* (Vol. 1. pp. 8–13).

Collins, M., Schapire, R., & Singer, Y. (2002). Logistic regression, Adaboost and Bregman distances. *Machine Learning*, *48*(1–3), 253–285.

Comaniciu, D., & Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *24*(5), 603–614.

Darrell, T., & Pentland, A. P. (1995). Cooperative robust estimation using layers of support. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *17*(5), 474–487.

Derpanis, K. G., & Gryn, J. M. (2005). Three-dimensional *N*th derivative of Gaussian separable steerable filters. In *IEEE international conference on image processing (ICIP)* (Vol. III. pp. 553–556).

Dollár, P., Tu, Z., & Belongie, S. (2006). Supervised learning of edges and objects boundaries. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Drummond, T., & Cipolla, R. (2000). Application of Lie algebras to visual servoing. *International Journal of Computer Vision (IJCV)*, *37*(1), 21–41.

Felzenszwalb, P., & Huttenlocher, D. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, *59*(2), 167–181.

Fleet, D. J., & Weiss, Y. (2005). Optical flow estimation. In N. Paragios, Y. Chen, & O. Faugeras (Eds.), *Mathematical models for computer vision: The handbook*. Berlin: Springer.

Fleet, D. J., Black, M. J., & Nestares, O. (2002). Bayesian inference of visual motion boundaries. In G. Lakemeyer & B. Nebel (Eds.), *Exploring artificial intelligence in the new millenium* (pp. 139–173). San Mateo: Morgan Kaufmann.

Fowlkes, C., Martin, D., & Malik, J. (2003). Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Frey, B. J. (1998). *Graphical Models for Machine Learning and Digital Communication*. Cambridge: MIT Press.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, *28*(2), 377–407.

Fusiello, A., Roberto, V., & Trucco, E. (1997). Efficient stereo with multiple windowing. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 858–863).

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *6*(6), 721–741.

Guan, L., Franco, J.-S., & Pollefeys, M. (2007). 3D occlusion inference from Silhouette cues. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Guzman, A. (1968). Decomposition of a visual scene into three dimensional bodies. In *AFIPS fall joint conference* (Vol. 33. pp. 291–304).

Heeger, D. J. (1988). Optical flow using spatiotemporal filters. *International Journal of Computer Vision (IJCV)*, *1*, 270–302.

Heitz, F., & Bouthemy, P. (1993). Multimodal estimation of discontinuous optical flow using Markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *15*(12), 1217–1232.

Heskes, T., Albers, K., & Kappen, B. (2003). Approximate inference and constrained optimization. In *Uncertainty in artificial intelligence (UAI)* (pp. 313–320).

Hirschmüller, H., Innocent, P. R., & Garibaldi, J. (2002). Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision (IJCV)*, *47*(1–3), 229–246.

Hoiem, D., Efros, A. A., & Hebert, M. (2005). Automatic photo pop-up. *ACM Transactions on Graphics (SIGGRAPH)*, *24*(3), 577–584.

Hoiem, D., Efros, A. A., & Hebert, M. (2007a). Recovering surface layout from an image. *International Journal of Computer Vision (IJCV)*, *75*(1), 151–172.

Hoiem, D., Stein, A. N., Efros, A. A., & Hebert, M. (2007b). Recovering occlusion boundaries from a single image. In *IEEE international conference on computer vision (ICCV)*.

Irani, M., & Peleg, S. (1993). Motion analysis for image enhancement: resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, *4*(4), 324–335.

Jepson, A. D., Fleet, D. J., & Black, M. J. (2002). A layered motion representation with occlusion and compact spatial support. In *European conference on computer vision (ECCV)* (Vol. 1. pp. 692–706).

Jojic, N., & Frey, B. J. (2001). Learning flexible sprites in video layers. In *IEEE conference on computer vision and pattern recognition (CVPR)* (Vol. 1. pp. 196–206).

Kanade, T., & Okutomi, M. (1994). A stereo matching algorithm with an adaptive window: theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *16*(9), 920–932.

Ke, Q., & Kanade, T. (2002). A robust subspace approach to layer extraction. In *IEEE workshop on motion and video computing (MOTION)* (pp. 37–43).

Konishi, S., Yuille, A. L., Coughlan, J. M., & Zhu, S. C. (2003). Statistical edge detection: learning and evaluating edge cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *25*(1), 57–74.

Kumar, M. P., Torr, P., & Zisserman, A. (2005). Learning layered motion segmentations of video. In *IEEE international conference on computer vision (ICCV)* (Vol. 1. pp. 33–40).

Kumar, S., & Hebert, M. (2006). Discriminative random fields. *International Journal of Computer Vision (IJCV)*, *68*(2), 179–202.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *International conference on machine learning (ICML)*.

Lazebnik, S., & Ponce, J. (2005). The local projective shape of smooth surfaces and their outlines. *International Journal of Computer Vision (IJCV)*, *63*(1), 65–83.

Leordeanu, M., & Hebert, M. (2005). A spectral technique for correspondence problems using pairwise constraints. In *IEEE International conference on computer vision (ICCV)*.

Leung, T., & Malik, J. (1998). Contour continuity in region based image segmentation. In *European conference on computer vision (ECCV)*.

Liu, C., Freeman, W. T., & Adelson, E. H. (2006). Analysis of contour motions. In *Advances in neural information processing systems (NIPS)*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, *60*(2), 91–110.

Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *International joint conferences on artificial intelligence (IJCAI)* (pp. 674–679).

MacKay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge: Cambridge University Press.

Mahamud, S., Williams, L. R., Thornber, K. K., & Xu, K. (2003). Segmentation of multiple salient closed contours from real images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *25*(4), 433–444.

Malisiewicz, T., & Efros, A. A. (2007). Improving spatial support for objects via multiple segmentations. In *British machine vision conference (BMVC)*.

Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE international conference on computer vision (ICCV)* (Vol. 2. pp. 416–423).

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

Maxwell, B. A., & Brubaker, S. J. (2003). Texture edge detection using the compass operator. In *British machine vision conference (BMVC)* (Vol. II. pp. 549–558).

Mori, G. (2005). Guiding model search using segmentation. In *IEEE international conference on computer vision (ICCV)*.

Mori, G., Ren, X., Efros, A., & Malik, J. (2004). Recovering human body configurations: combining segmentation and recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)* (Vol. 2. pp. 3226–3333).

Nestares, O., & Fleet, D. J. (2001). Probabilistic tracking of motion boundaries with spatiotemporal predictions. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 358–365).

Ogale, A. S., Fermüller, C., & Aloimonos, Y. (2005). Motion segmentation using occlusions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *27*(6), 988–992.

Pearl, J. (1982). Reverend Bayes on inference engines: A distributed hierarchical approach. In *Association for the advancement of artificial intelligence (AAAI)* (pp. 133–136).

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Mateo: Morgan Kaufmann.

Ren, X., & Malik, J. (2003). Learning a classification model for segmentation. In *IEEE international conference on computer vision (ICCV)* (Vol. 1. pp. 10–17).

Ren, X., Fowlkes, C. C., & Malik, J. (2005). Cue integration for figure/ground labeling. In *Advances in neural information processing systems (NIPS)*.

Ren, X., Fowlkes, C. C., & Malik, J. (2006). Figure/ground assignment in natural images. In *European conference on computer vision (ECCV)*.

Ross, M. G., & Kaelbling, L. P. (2005). Learning static object segmentation from motion segmentation. In *Association for the advancement of artificial intelligence (AAAI)*.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2005). *LabelMe: a database and web-based tool for image annotation* (Memo AIM-2005-025). MIT AI Lab, http://labelme.csail.mit.edu/.

Ruzon, M., & Tomasi, C. (1999). Color edge detection with the compass operator. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 160–166).

Sato, J., & Cipolla, R. (1999). Affine reconstruction of curved surfaces from uncalibrated views of apparent contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *21*(11), 1188–1197.

Sethi, A., Renaudie, D., Kriegman, D., & Ponce, J. (2004). Curve and surface duals and the recognition of curved 3d objects from their silhouettes. *International Journal of Computer Vision (IJCV)*, *58*(1), 73–86.

Shechtman, E., & Irani, M. (2005). Space-time behavior based correlation. In *IEEE conference on computer vision and pattern recognition (CVPR)* (Vol. 1. pp. 405–412).

Shi, J., & Malik, J. (1998). Motion segmentation and tracking using normalized cuts. In *IEEE international conference on computer vision (ICCV)* (pp. 1154–1160).

Simoncelli, E., Adelson, E. H., & Heeger, D. J. (1991). Probability distributions of optical flow. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Smith, P., Drummond, T., & Cipolla, R. (2004). Layered motion segmentation and depth ordering by tracking edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *26*(4), 479–494.

Smith, P. A. (2001). *Edge-based motion segmentation*. Ph.D. thesis, Jesus College, University of Cambridge.

Stein, A., & Hebert, M. (2005). Incorporating background invariance into feature-based object recognition. In *IEEE workshop on applications of computer vision (WACV)* (pp. 37–44).

Stein, A., & Hebert, M. (2007). Combining local appearance and motion cues for occlusion boundary detection. In *British machine vision conference (BMVC)*.

Stein, A., Hoiem, D., & Hebert, M. (2007). Learning to find object boundaries using motion cues. In *IEEE international conference on computer vision (ICCV)*.

Stein, A. N. (2008). Occlusion boundaries: low-level processing to high-level reasoning. Doctoral Dissertation, The Robotics Institute, Carnegie Mellon University.

Stein, A. N., & Hebert, M. (2006a). Local detection of occlusion boundaries in video. In *British machine vision conference (BMVC)* (pp. 407–416).

Stein, A. N., & Hebert, M. (2006b). Using spatio-temporal patches for simultaneous estimation of edge strength, orientation, and motion. In *Beyond patches workshop at IEEE conference on computer vision and pattern recognition (CVPR)* (p. 19).

Stein, A. N., Stepleton, T. S., & Hebert, M. (2008). Towards unsupervised whole-object segmentation: combining automated matting with boundary detection. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Tao, H., Sawhney, H. S., & Kumar, R. (2001). A global matching framework for stereo computation. In *IEEE international conference on computer vision (ICCV)* (Vol. 1. pp. 532–539).

Tomasi, C., & Kanade, T. (1991). *Detection and tracking of point features* (Technical Report CMU-CS-91-132). Carnegie Mellon University.

Vaillant, R., & Faugeras, O. D. (1992). Using extremal boundaries for 3-D object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *14*(2), 157–173.

Veit, T., Cao, F., & Bouthemy, P. (2006). An a contrario decision framework for region-based motion detection. *International Journal of Computer Vision (IJCV)*, *68*(2), 163–178.

Waltz, D. A. (1975). Understanding line drawings of scenes with shadows. In *The psychology of computer vision* (pp. 19–91). New York: McGraw-Hill.

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

Weiss, Y. (1997). Interpreting images by propagating Bayesian beliefs. In *Advances in neural information processing systems* (Vol. 9, p. 908).

Weiss, Y. (2000). Correctness of local probability propagation in graphical models with loops. *Neural Computation*, *12*(1), 1–41.

Wolf, L., Huang, X., Martin, I., & Metaxas, D. (2006). Patch-based texture edges and segmentation. In *European conference on computer vision (ECCV)* (pp. 481–493).

Xiao, J., & Shah, M. (2005). Accurate motion layer segmentation and matting. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Xiao, J., Cheng, H., Sawhney, H., Rao, C., & Isnardi, M. (2006). Bilateral filtering-based optical flow estimation with occlusion detection. In *European conference on computer vision (ECCV)* (Vol. I, pp. 211–224).

Yedidia, J. S., Freeman, W. T., & Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, *51*(7), 2282–2312.

Yin, P., Criminisi, A., Winn, J., & Essa, I. (2007). Tree-based classifiers for bilayer video segmentation. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Yu, S. X., & Shi, J. (2001). *Perceiving shapes through region and boundary interaction* (Technical Report CMU-RI-TR-01-21). Robotics Institute, Carnegie Mellon University.

Yuille, A. L. (2002). CCCP algorithms to minimize the Bethe and Kikuchi free energies: convergent alternatives to belief propagation. *Neural Computation*, *14*(7), 1691–1722.