International Journal of Humanoid Robotics
Vol. 6, No. 3 (2009) 361–386
© World Scientific Publishing Company

ACTIVE SEGMENTATION

AJAY K. MISHRA* and YIANNIS ALOIMONOS†

Computer Vision Laboratory,
Institute of Advanced Computer Studies,
University of Maryland, College Park
**mishraka@umiacs.umd.edu*
†*yiannis@cs.umd.edu*

Received 2 December 2008
Accepted 8 May 2009

The human visual system observes and understands a scene/image by making a series of fixations. Every fixation point lies inside a particular region of arbitrary shape and size in the scene which can either be an object or just a part of it. We define as a basic segmentation problem the task of segmenting that region containing the fixation point. Segmenting the region containing the fixation is equivalent to finding the enclosing contour — a connected set of boundary edge fragments in the edge map of the scene — around the fixation. This enclosing contour should be a depth boundary.

We present here a novel algorithm that finds this bounding contour and achieves the segmentation of one object, given the fixation. The proposed segmentation framework combines monocular cues (color/intensity/texture) with stereo and/or motion, in a cue independent manner. The semantic robots of the immediate future will be able to use this algorithm to automatically find objects in any environment. The capability of automatically segmenting objects in their visual field can bring the visual processing to the next level. Our approach is different from current approaches. While existing work attempts to segment the whole scene at once into many areas, we segment only one image region, specifically the one containing the fixation point. Experiments with real imagery collected by our active robot and from the known databases[1] demonstrate the promise of the approach.

Keywords: Fixation; active vision; region segmentation; cue integration.

1. Introduction

Active Vision has had tremendous successes in the past twenty years. Head/eye active binocular systems appeared in Universities and the Industry, research on visual motion, navigation and 3D recovery achieved new heights, a series of sophisticated tracking systems made its appearance,computational work on attention and work on navigation made significant advances.[4,9,10]

Now the field has developed numerous techniques for successfully dealing with large spaces (going from one place to another) and researchers are turning their attention to small spaces (objects). Indeed, a pressing need for a large number of applications is to develop semantic robots — the robots that are equipped with sensors and effectors capable of finding and fetching (picking up, carrying) objects in a room, while possibly communicating with a human through speech. We borrow the term, "semantic robots", from the synonymous Challenge sponsored by the National Science Foundation: The Semantic Robot Vision Challenge (SRVC).[1] (In this challenge, robots (possessing sensors) were given names of twenty objects. The robots were then supposed to find those objects in a simplified room-like setting. Before entering the rooms, the robots were connected to the Internet to obtain images and build visual representations of the objects under consideration).

Imagine the following scenario in an elderly care facility: An elderly person is not able to perform a regular task due to a temporary loss of memory, fatigue, or similar causes and he/she requests assistance. Such assistance may be in the form of a robotic device roaming the hallways which is summoned to attend the person in the particular room. It will have to perform variety of tasks. While some of these tasks may be simple, others may require significant effort (both mental and physical) on the part of the person needing the assistance. Some common (and simpler) tasks may, for example, include finding the pill box and giving it to the person. In general, a complex task may involve fetching an object that could be in plain view, or it could be partially seen or it could be hidden inside a drawer. A complex task such as the one described above will require both "gentle" dextrous manipulation as well as vision. Both of these sub-problems (manipulation and vision) should be triggered by speech or sound, whereby the person may instruct the robot to carry out the task either autonomously or through cooperative search strategies (i.e., the person could instruct the robot to look in specific locations and in real-time guide the robot to accomplish the task by perhaps voice commands to your left, behind you, on the shelf above the stove).

This paper is devoted to basic visual competences needed by the robot to function inside the room of the hypothetical scenario described before. When the robot is inside the room where it is supposed to assist someone, it will have to visually search the area and find an object. For this to happen, the robot must possess capabilities to segment the part of the image it sees and recognize the segment as some kind of object that it knows about.

This problem of segmentation is an open question and constitutes a core challenge addressed in this paper. We are interested in solutions that are generic and can be used by a variety of robots, since the problem of visually segmenting objects is universal.

2. Active Segmentation

The problem of segmentation has occupied scientists, philosophers and engineers for many years, with very interesting results. But what does it really mean to segment a scene? The most prominent definition of segmentation in the literature is dividing the scene (or image) into regions with some homogeneous property. This is done by grouping pixels together depending upon their properties such as their color,

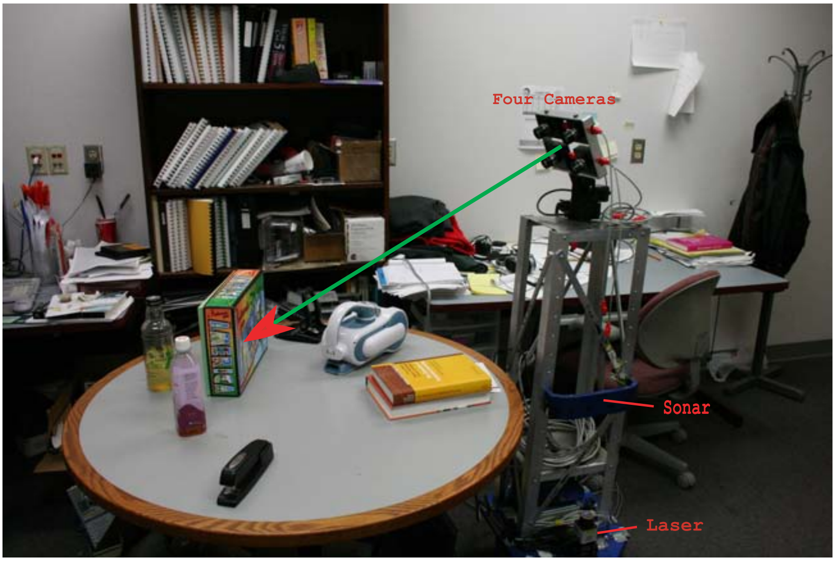
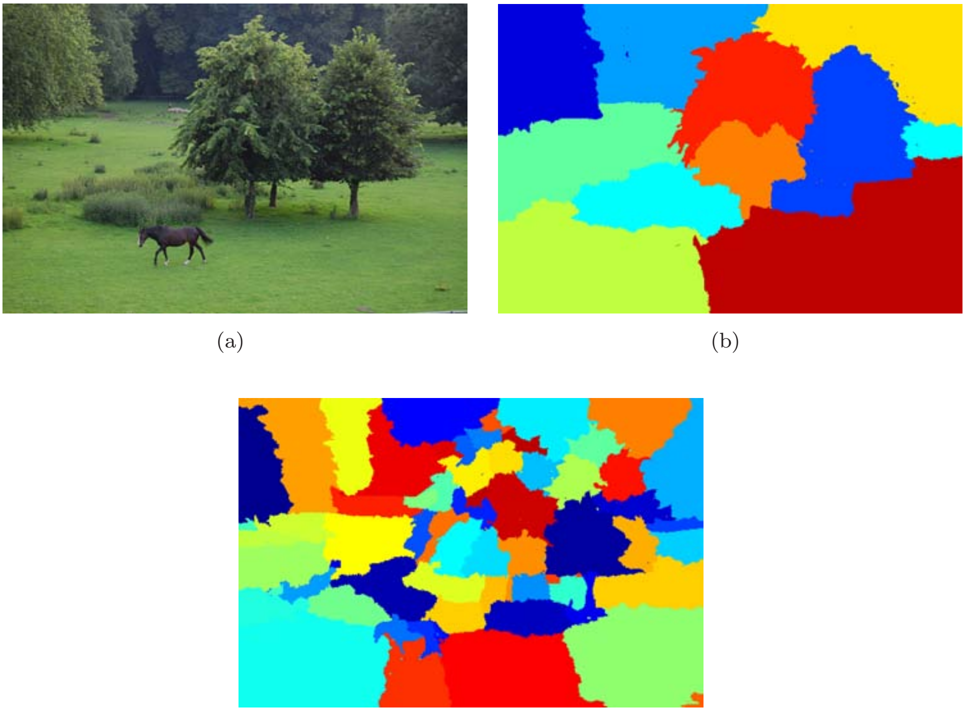


Fig. 1. Our robot with a quad-camera vision system mounted on top. The green arrow shows its line of sight as it fixates on an object on the table. For every such fixation, our algorithm returns the region containing the object. The robot makes multiple fixations at other locations and use the associated regions returned by our algorithm to understand the scene.

brightness, and texture, and merging these groups in a hierarchical fashion. The clustering will eventually put all the pixels together in one big group, the entire image. So, it is important to stop the process of clustering at an appropriate level to obtain the desired segmentation. The current segmentation algorithms[23,40,47] take the user inputs such as the expected number of regions,[40] thresholds[23] to stop the process of segmentation and output the results.

The problem with depending on the user specified parameters to stop the segmentation is that these parameters cannot be calculated automatically for a new test image. Inappropriate parameters result in over-segmentation or under-segmentation of the image. In the former case, the region of interest is broken into many small regions, whereas in the latter case, the region of interest get merged with other regions to form a bigger region. So, to have a segmentation as an essential first step of visual processing, it should be fully automatic and not depend upon any user input.

Furthermore, the definition of the "desired" segmentation of a scene (or image) depends on the object of interest. For example, in Fig. 2(a), the tiny horse and the big tree are two possible objects of interest. Now, if the tiny horse is of interest, Fig. 2(c) is the "desired" segmentation output. However, if the big tree is of interest, Fig. 2(b) is the "desired" segmentation. Note that in Fig. 2(b), there is in fact no region corresponding to the horse. So, if the object of interest had been the horse, the segmentation in Fig. 2(b) would be a case of under-segmentation. This clearly



(a)

(b)

(c)

Fig. 2. Segmentation results for the image shown in (a) by Normalized Cut[40] algorithm with its parameter (number of regions) set to 10 and 60 are shown in (b) and (c), respectively.

illustrates that having global parameters to segment an image is not a meaningful exercise. The reason that the practice of choosing a single global parameter for an image has prevailed is the images in the segmentation databases usually have only a single object in prominence or in case of multiple objects, they all exist at comparable scale. Thus, the "desired" segmentation of the image is decided by looking at the number of regions spanning the prominent object(s) in the image.

We need a segmentation algorithm that segments the object of interest rather than the entire image at once. This object of interest is what human eyes fixate on. But, before we explain our segmentation algorithm which is fully automatic and segments the region of interest in the scene, we explain our motivation to design a fixation based segmentation algorithm. Our inspiration comes from analyzing how the human visual system works. One of the fundamental steps is that the human visual system observes and makes sense of a dynamic scene (video) or static scene (image) by making a series of fixations at various salient locations in the scene. These salient locations are in fact the objects or the parts of the objects in the scene. Researchers have studied in great length about where human eye fixates,[19,50] but little is known about the operations carried out in the human visual system during a fixation. We argue that during a fixation, the human visual system segments the region of interest which contains the fixation point. As it moves to the new fixation location, it segments another region of interest. In fact, instead of segmenting the entire image at once (what is done conventionally in the segmentation literature), the scene is segmented in terms of a series of individual regions associated with the fixations in the scene. This is also likely because of the structure of the human retina which captures only the small neighborhood around the fixation in high resolution by the fovea, and the rest of the scene in lower resolution by the sensors on the periphery of retina.

In this paper, we define segmenting the region containing the fixation point as a basic segmentation problem. Since the early attempts on Active Vision, there has been a lot of work on problems surrounding fixation, both from a computational and psychological perspective.[4,9,10,22,35] Despite all this development however, the operation of fixation never really made it into the foundations of computational vision. Specifically, the fixation point has not become a parameter in the multitude of low and middle level operations that constitute a big part of the visual perception process. This is the avenue we pursue in this paper. It is only natural to make fixation part and parcel of any visual processing — First fixate, then segment the surface containing the fixation point.

For instance, for the image (see Fig. 2(a)) discussed above, Figs. 3(a) and 3(c) show the two different cases with fixations on the tree and the horse respectively (the fixation point is shown by the green "X"). Similarly, our segmentation algorithm returns the regions corresponding to these fixations as shown in Fig. 3(b), and Fig. 3(d) respectively. Similarly, our semantic robot (see Fig. 1) fixates at different salient locations in the scene and segment the surface (object) containing those fixation points.

The rest of the paper is organized as follows: In Sec. 3, the existing segmentation algorithms are discussed in detail. In Sec. 4, we describe our algorithm to segment the region being fixated on. The experimental results with quantitative analysis is presented in Sec. 5. In Sec. 6, the fixation strategy and how stable the segmentation results are as the location of fixation change inside the region of interest is described. Finally, we conclude our paper with some suggestions for future research in this area.

3. Related Work

Without prior knowledge or context and only on the basis of signal processing, whatever the segmentation algorithm may be, somehow it needs to decide when to stop growing the segments and stop the process of segmentation. And that input comes from the user. Without any user input, segmenting an image into regions is an ill-posed problem because segmentation can be fine or coarse depending on when the process is stopped. Most popular algorithms amount to such global methods. It is also widely known that it is difficult to estimate the input parameters automatically for any given image.

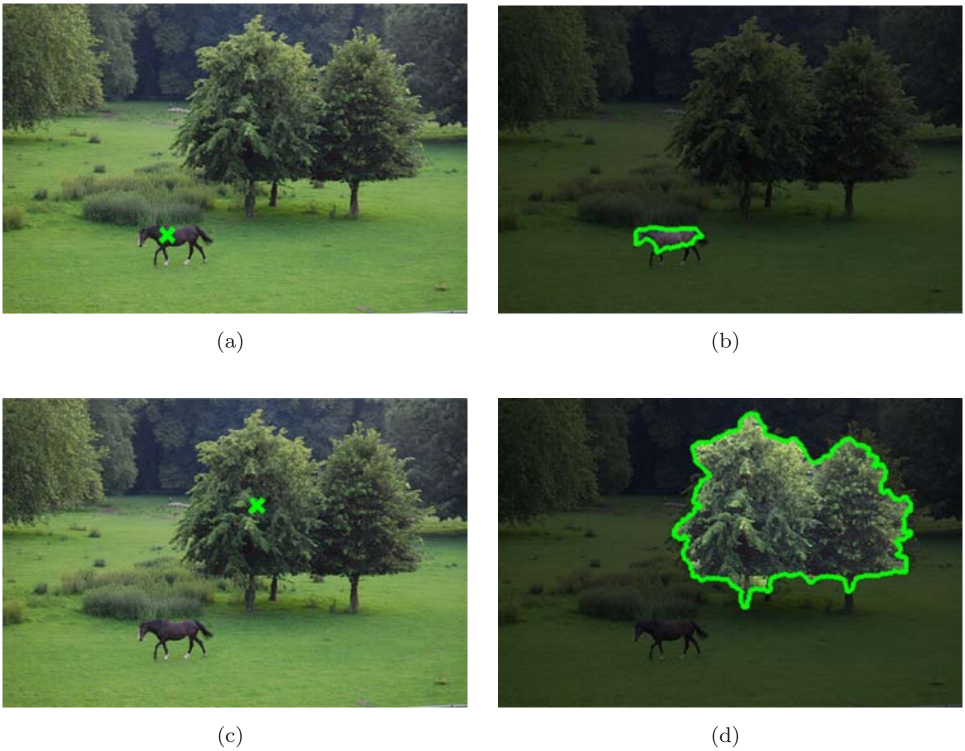


Fig. 3. In (a) and (c), the fixation points are shown by a green "X". The regions segmented by our algorithm for these fixation points are shown in (b) and (d), respectively. Please note that only the region corresponding to the fixation is segmented, and the scale of the region being fixated does not influence the outcome.

So, several interactive algorithms have been proposed where the objective is to always segment the entire image into two regions: foreground and background. There are different types of these algorithms and they take inputs from the user differently. These algorithms are not automatic and can not be used to build an autonomous visual system. They are used in interactive applications, such as image/video editing, image databases, etc.

Segmentation approaches can be broadly classified into two main categories: image segmentation where monocular cues are used to segment the image, and motion segmentation where motion cues are used to segment the image. Here, we provide a brief overview of both types of segmentation algorithms.

3.1. *Image segmentation*

An image is a two dimensional array of pixels where every pixel has a color, intensity and texture information. A region is a connected set of pixels in the image which have similar color, intensity and texture information. These regions are either obtained by clustering the pixels into coherent groups (such methods are called region based methods) or by identifying the closed boundaries along the edges in the image formed by the gradients in color, intensity and texture values. The closed boundaries are the closed paths through the gradient map of the monocular cues in the image. Each of these closed contours corresponds to a region.

3.1.1. *Region based methods*

An image is considered to be a graph with each pixel represented by a node in the graph which is connected to the neighboring pixels. The edge connecting two pixels i and j is weighted according to the features of the pixels. In Ref. 29, the edge weight is computed based on the texture cues and the intervening contour between the pixels. The graph is divided into clusters using eigenvectors of the similarity matrix formed by collecting all the edge weights. In Ref. 23, the dissimilarity of the color information of the pixels are used to assign the weights to the edges and clusters are formed in an hierarchical clustering fashion. The criterion to group the nodes as it moves up the hierarchy is adapted to the degree of variability in the neighboring regions. Both Refs. 23 and 24 and all other region based segmentation algorithms need a user input to stop the process of grouping the pixels. Reference 29 needs the expected number of regions as input whereas Ref. 23 takes the threshold to stop the clustering process. In fact, without any user input, it is impossible to define the optimal segmentation. There are many other segmentation algorithms[48,57] which are based on global user parameters like the number of regions or threshold.

Unlike the global parameter based segmentation algorithms, the interactive segmentation algorithms[8,14,37,56] always segment the entire image into only two regions: foreground and background. Reference 14 poses the problem of foreground/background segmentation as a binary labeling problem which is solved exactly using the maxflow algorithm.[15] It, however, requires users to label some pixels as foreground or background to build their color models. Reference 12 improved upon Ref. 14 by using a Gaussian mixture Markov random field to better learn the foreground and background models. Reference 37 requires users to specify a bounding box containing the foreground object. Reference 6 requires a seed point for every region in the image. For foreground/background segmentation, at least two seed points are needed. Although these approaches give impressive results, they can not be used as an automatic segmentation algorithm as they critically depend upon the user inputs. Reference 56 tries to automatically select the seed points by using spatial attention based methods and then use these seed points to introduce extra constraints into their normalized cut based formulation.

References 49 and 8 need only a single seed point from the user. Reference 49 imposes a constraint on the shape of the object to be a star, meaning the algorithm prefers to segment the convex objects. Also, the user input for this algorithm is critical as it requires the user to specify the center of the star shape exactly in the image. Reference 8 needs only one seed point to be specified on the region of interest and segment the foreground region using a compositional framework. But the algorithm is computationally intensive. It runs multiple iterations to arrive at the final segmentation.

3.1.2. *Contour based methods*

Contour based segmentation methods start with finding edge fragments in the image first, and then joining the edge fragments to form closed contours. The regions are enclosed by each of these closed contours. Due to the presence of textures and low contrast regions in the image, detecting edge fragments is a hard problem. The second step of joining the edge fragments is done in probabilistic fashion using image statistics. In Ref. 54, first order Markov model is used for contour shape and the contours were completed using random walk. In Ref. 36, multiple scales are used to join the contours using orientation and texture cues. References 36 and 54 are edge based segmentation methods.

Similar to the global region based segmentation methods, edge based segmentation algorithms suffer from ambiguity of choosing the appropriate closed loops which are actually the boundaries of the regions in the image. References 32 and 11 need the user to specify the seed points along the contour to be traced. References 27 and 55 need the user to initialize a closed contour which then evolves to adjust the actual boundary in the image.

3.2. *Motion segmentation*

Prior research in motion segmentation can broadly be classified into two groups:

(a) The approaches relying on 2D motion measurements only.[13,18,34,52] There are many limitations in these techniques. Depth discontinuities and independently moving objects both cause discontinuities in the 2D optical flow, and it is not possible to separate these factors without 3D motion and structure estimation. Generally, dense optical flow is calculated at every point in the image and like in the image segmentation, the flow value of each pixel is used to decide similarity between the pixels which is used to cluster them into regions with consistent motion. The main problem with this approach is that the optical flow is inaccurate at the boundaries and hence the region obtained by this approach has generally poor boundaries.

To overcome this problem, many algorithms first segment the frames into regions and then merge the regions by comparing the overall flow of the two regions. The accuracy of this method is dependent on the accuracy of the image segmentation step. If a region is produced by the image segmentation step which include parts from different objects in the scene, it can not be corrected by the later processing of combining regions into bigger regions. To avoid that problem, some techniques over-segment the image into small regions to reduce the chances of having overlapping regions. But discriminating small regions on the basis of their overall flow is difficult.

(b) 3D approaches which identify clusters with consistent 3D motion[2,20,33,41 43,45,58]using a variety of techniques. Some techniques, such as Ref. 51, are based on alternate models of image formation. These additional constraints can be justified for domains such as aerial imagery. In this case, the planarity of the scene allows a registration process,[7,46,53,59] and un-compensated regions correspond to independent movement.

This idea has been extended to cope with general scenes by selecting models depending on the scene complexity,[44] or by fitting multiple planes using the plane plus parallax constraint.[25,38] Most techniques detect independently moving objects based on the 3D motion estimates, either explicitly or implicitly. Some utilize inconsistencies between ego-motion estimates and the observed flow field, while some utilize additional information such as depth from stereo, or partial ego-motion from other sensors. The central problem faced by all motion based techniques is that, in general, it is extremely difficult to uniquely estimate 3D motion from flow. Several studies have addressed the issue of noise sensitivity in structure from motion. In particular, it is known that for a moving camera with a small field of view observing a scene with insufficient depth variation, translation and rotation are easily confused.[3]

4. Our Approach

Our segmentation strategy involves two consecutive steps: first, all available visual cues are used to generate a probabilistic boundary edge map. The gray scale value of an edge pixel in the map is proportional to the probability of that pixel to be at a region boundary. The method to obtain the map is described in detail in Sec. 1.3.1. Second, the fixation point is selected in the scene either by a visual attention module or by any other meaningful strategy. The probabilistic edge map from the previous step is then transferred from the Cartesian space to the polar space with this fixation point as its pole. In the polar image of the edge map, the closed boundary of the region containing the fixation point from the Cartesian space becomes the path that optimally cuts the edge map into two halves as described in Sec. 1.3.2. The left half of the polar edge map corresponding to the pixels from insides the region is transferred back to the Cartesian space resulting in the segmentation for the selected fixation.

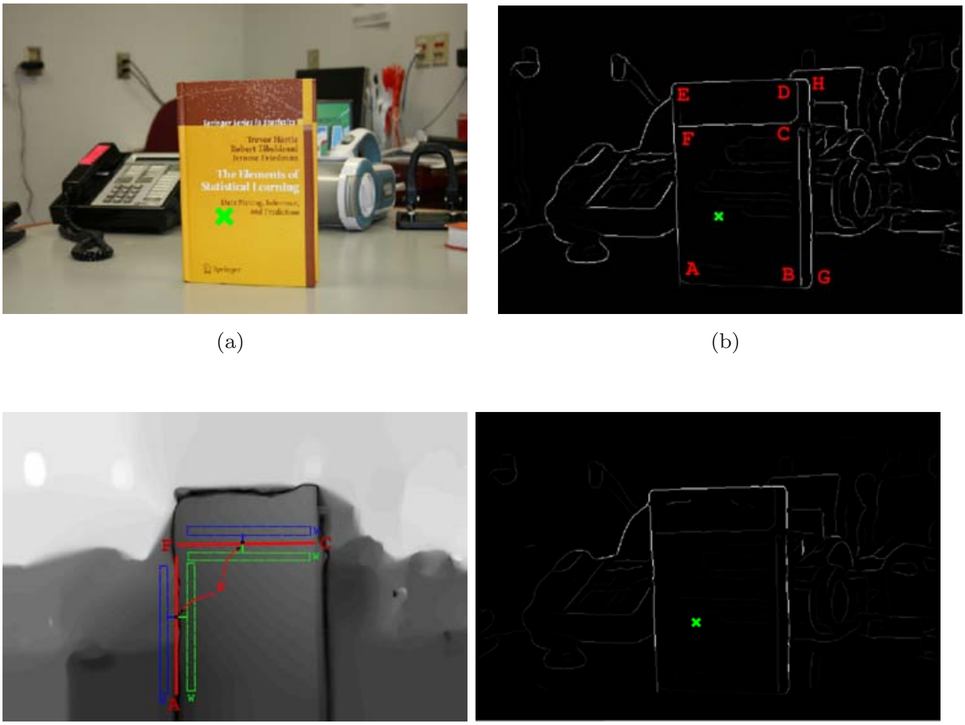
The reason for splitting the segmentation process into two steps is that once all the visual cues are used to obtain the probabilistic boundary edge map, the segmentation is defined optimally for every fixation selected in the scene (or image).

4.1. *Computing probabilistic boundary edge maps*

As explained before, the probabilistic boundary edge map encodes the probability of the edge pixels to be at the region boundary as their gray value. This means the edge pixels along the boundary will be brighter than the internal/texture edges. So, ideally, we would want to have a probabilistic boundary edge map wherein all bright edge pixels are the points along the region boundary (depth boundary) in the image. We are going to explore how to generate such a probabilistic boundary edge map.

Our initial probabilistic boundary edge map is the output of the Berkeley edge detector.[30] Martin *et al.* learned the color and texture properties of the boundary pixels from the labeled data (∼300 images) and use that information to differentiate the boundary edges from the internal edges. See Fig. 4(b) (the edge map of Fig. 4(a)) as a typical output of the edge detector. Unlike binary edge detectors like canny, it successfully removes the spurious texture edges and highlights the boundary edges, but it still has some strong internal edges (BC, CD, CF) which are not the depth boundaries.

Now, to suppress these strong internal edge segments and reinforce the boundary edges (AG, GH, HE, EA), we can use motion and(or) stereo cues. We know that the change in disparity or flow across the internal edges is less than that across the



(a)

(b)

(c)

(d)

Fig. 4. (a) The first frame of one of the motion sequences used in our experiments. The scene is static and the camera is moving. (b) The probabilistic boundary edge map as given by Martin *et al.*[30] (c) The magnitude of the optical flow vectors calculated by Brox *et al.*[17] The gray value of an image pixel represent the magnitude. (d) The modified boundary edge map as a result of using motion cues to reinforce the boundary edges and suppress the internal edges.

boundary edges. So, we can look into both sides of the edges to find the change in flow and disparity and modify its probability (or gray value) accordingly.

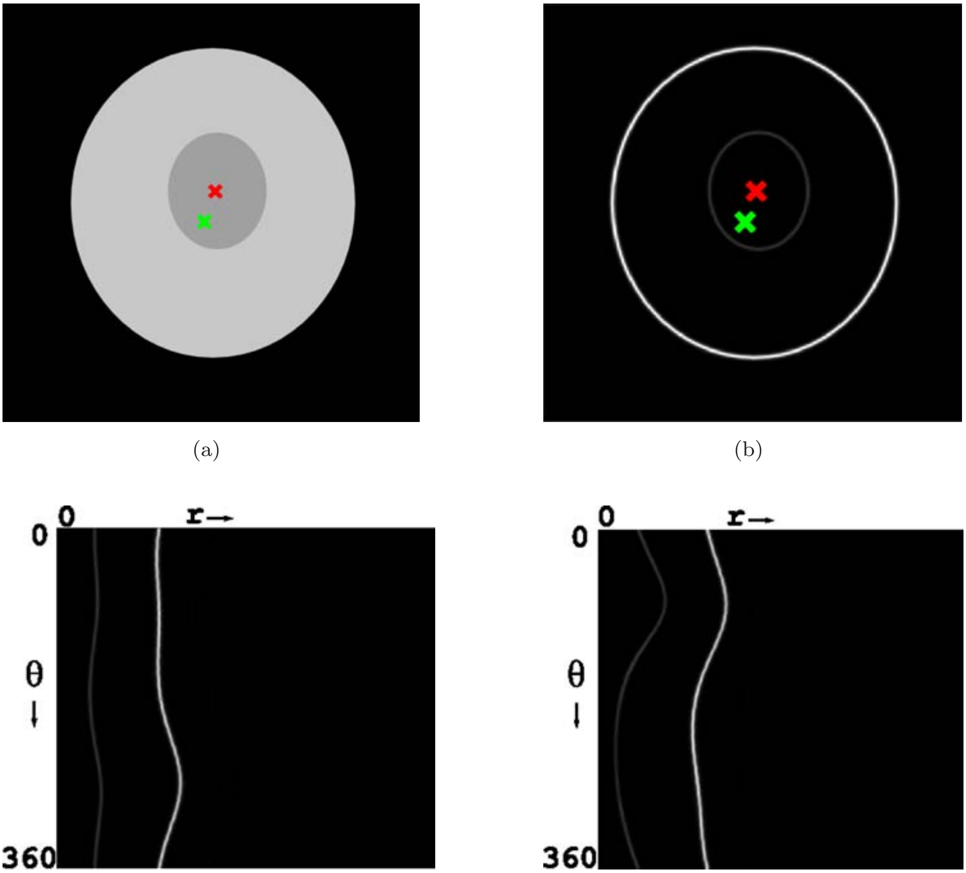
We break the edge map into straight line segments (such as AB, BC, CD, etc. shown in Fig. 4(b)) and select rectangular regions of width w at a distance r on its both sides. See Fig. 4(c). We then calculate the average disparity and/or average flow inside these rectangles. The absolute difference in the average disparity, $\triangle d$, and the magnitude of the average flow, $\triangle f$, is a measure of how likely a segment is to be at the depth boundary. The greater the difference, higher is the likelihood of the edge segment to be at the boundary. The rectangular regions are selected at a equal distance r on both sides from the edge segment, because, at the boundary, the flow or disparity is more corrupted than inside the object. We chose r and w to be 5 and 10 pixels for our experiments.

Now, the brightness of an edge pixel on the edge segment is changed as $I'(x,y) = \alpha_b I(x,y) + (1-\alpha_b)(\Delta f/\max(\Delta f))$ or $I'(x,y) = \alpha_b I(x,y) + (1-\alpha_b)(\Delta d/\max(\Delta d))$ for motion and stereo cues respectively where $I(\cdot)$ and $I'(\cdot)$ are the original and the improved edge maps respectively, α_b is the weight associated with the relative importance of the monocular cue based boundary estimate. For our experiments, we chose α_b to be 0.2. The improved probabilistic boundary edge map is shown in Fig. 4(d) wherein the internal edge are dim and the boundary edges are bright.

4.2. *Polar space for scale normalization*

Before we explain the method to find the optimal closed boundary around the fixation point, it is important to first explain why we choose to do so in the polar co-ordinate system. Let us consider finding the optimal contour for the red fixation on the disc shown in Fig. 5(a). The gradient edge map (Fig. 5(b)) of the disc has two concentric circles. The big circle is the actual boundary of the disc whereas the small circle is just the internal edge on the disc. Say, the edge map correctly assigns the boundary contour intensity 0.78 and the internal contour 0.39 (the intensity ranges from 0 to 1). The lengths of the two circles are 400 and 100 pixels. Now, the cost of tracing the boundary and the internal contour in the Cartesian space will be $88 = 400 \times (1 - 0.78)$ and $61 = 100 \times (1 - 0.39)$. Clearly, the internal contour costs less and hence will be considered optimal even though the boundary contour is the brightest and should actually be the optimal contour. In fact, this problem of inherently preferring short contours over long contours has already been identified in the graph cut based approaches where the minimum cut usually prefers to take "short cut" in the image.[42]

To fix this "short cut" problem, we have to transfer these contours to a space where their lengths no longer depend upon the area they enclose in the Cartesian space. And, the cost of tracing these contours in this space will now be independent of their scales in the Cartesian space. The polar space has this property and we use it to solve the scale problem. The contours are transformed from the Cartesian co-ordinate system to the polar co-ordinate system with the red fixation in Fig. 5(b) as



(a)

(b)

(c)

(d)

Fig. 5. (a) An image of a disc. (b) The gradient edge map. (c) and (d) are the polar images of the gradient edge map with pole being the red and green fixation respectively. In our polar representation, the radial distance increases along the horizontal axis and the angular distance increases along the vertical axis from top to bottom.

the pole. See Fig. 5(c). In the polar space now, both contours become open curves (0°–360°). Thus, the costs of tracing the inner contour and the outer contour become $80.3 = 365 \times (1 - 0.78)$ and $220.21 = 361 \times (1 - 0.39)$ respectively. As expected, the outer contour (the actual boundary contour) costs the least in the polar space and hence becomes the optimal enclosing contour around the fixation.

4.3. *Segmenting the polar edge map*

Now, after explaining the rationale for using the polar co-ordinate system, we present our method to convert the probabilistic boundary map from the Cartesian to polar co-ordinate system. After that, our algorithm to obtain the optimal contour which is essentially an optimal path through the resulting polar probabilistic boundary edge map starting from its top row to its bottom row.

4.3.1. *Cartesian to polar conversion*

Let's say, $I_E^{cart}(\cdot)$ is an edge map in Cartesian coordinate, $I_E^{pol}(\cdot)$ is its corresponding polar plot and $F(x_o, y_o)$ is chosen as a pole. Now, a pixel $I_E^{pol}(r, \theta)$ in the polar coordinate system corresponds to a sub-pixel location $(x, y), x = r\cos\theta + x_o, y = r\sin\theta + y_o$ in the Cartesian coordinate system. $I_E^{cart}(x, y)$ is typically calculated by bi-linear interpolation which only considers four immediate neighbors.

We propose to generate a continuous 2D function $W(\cdot)$ by placing 2D Gaussian kernel functions on every edge pixel. The major axis of these Gaussian kernel functions is aligned with the orientation of the edge pixel. The variance along the major axis is inversely proportional to the distance between the edge pixel and the pole O. Let E be the set of all edge pixels. The intensity at any sub-pixel location (x, y) in Cartesian coordinates is

$$W(x, y) = \sum_{e \in E} \exp\left(-\frac{x_e^t}{\sigma_{x_e}^2} - \frac{y_e^t}{\sigma_{y_e}^2}\right) \times I^{cart}(x_e, y_e),$$

$$\begin{bmatrix} x_e^t \\ y_e^t \end{bmatrix} = \begin{bmatrix} \cos\theta_e & \sin\theta_e \\ -\sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} x_e - x \\ y_e - y \end{bmatrix},$$

where $\sigma_{x_e}^2 = \frac{K_1}{\sqrt{(x_e - x_o)^2 + (y_e - y_o)^2}}$, $\sigma_{y_e}^2 = K_2$, θ_e is the orientation at the edge pixel e, $K_1 = 900$ and $K_2 = 4$ are constants. The reason for setting the square of variance along the major axis, $\sigma_{x_e}^2$, to be inversely proportional to the distance of the edge pixel from the pole is to keep the gray values of the edge pixels in the polar edge map, the same as the corresponding edge pixel in the Cartesian edge map. The intuition behind using variable width kernel functions for different edge pixels is as follows: Imagine an edge pixel being a finite sized elliptical bean aligned with its orientation, and you look at it from the location chosen as pole. The edge pixels closer to the pole (or center) will appear bigger and those farther away from the pole will appear smaller.

The polar edge map $I_E^{pol}(r, \theta)$ is calculated by sampling $W(x, y)$. The intensity values of I_E^{pol} scaled to lie between 0 and 1. An example of this polar edge map is shown in Fig. 6(c). Our convention is that the angle $\theta \in [0°, 360°]$ varies along the vertical axis of the graph and increases from the top to the bottom whereas the radius $0 \leq r \leq r_{\max}$ is represented along the horizontal axis increasing from left to the right. $r_{\max}$ is the maximum Euclidean distance between the fixation point and any other location on the image.

4.3.2. *Finding the optimal cut through the polar edge map*

Let us consider every pixel $p \in P$ of I_E^{pol}as a node in the graph and is connected to their 4 immediate neighbors (Fig. 8). As the rows of the graph represent rays

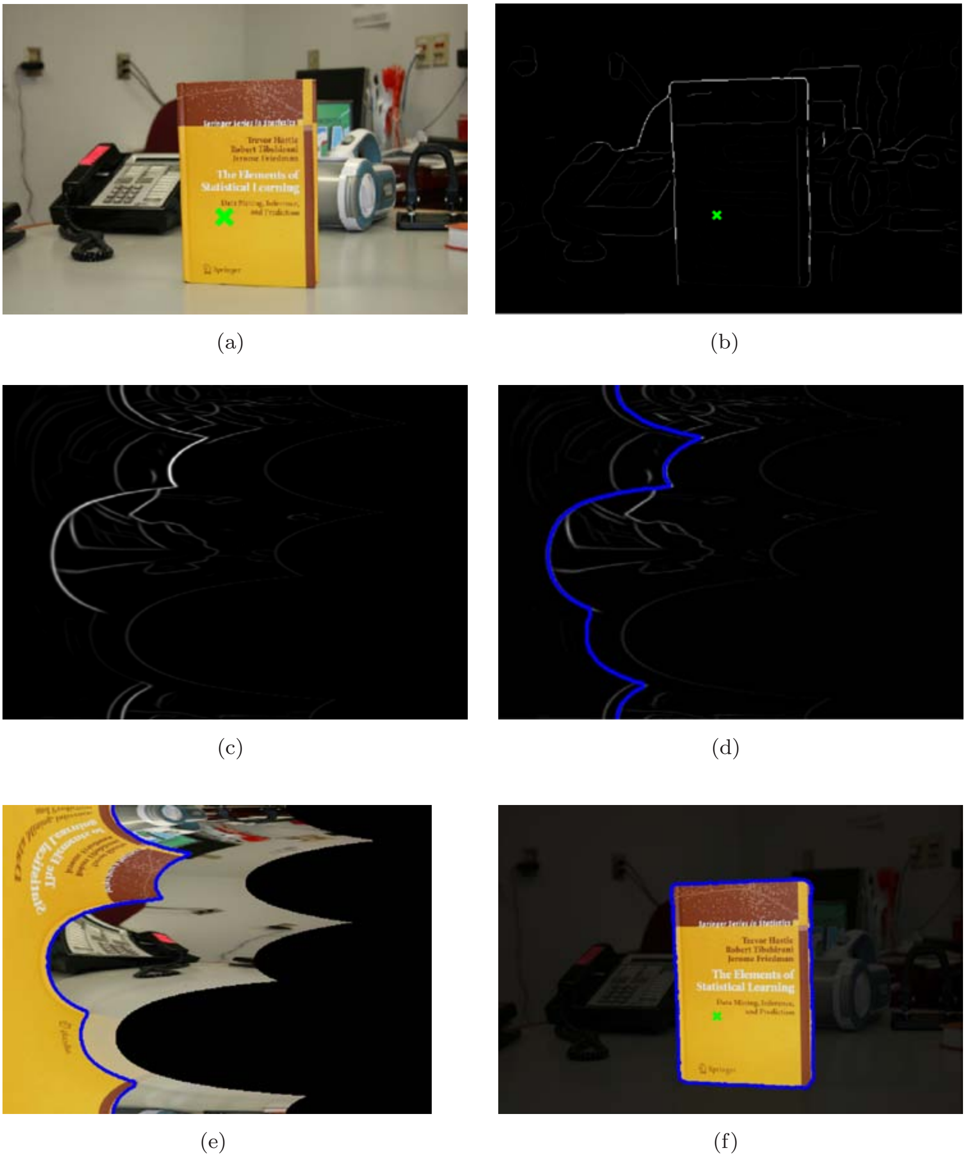


Fig. 6. (a) The first frame of the image sequence captured with a moving camera, also shown in Fig. 4(a). The fixation point is shown by a "X". (b) The final probabilistic boundary edge map as obtained in Sec. 4.1. (c) The polar image of the boundary edge map for the fixation. (d) The polar edge map with the optimal path (as calculated in Sec. 4.3.2) shown by the gray curve. (e) The color image after the polar transformation with the optimal path (the gray curve) superimposed on it. (f) The region segmented by our algorithm containing the fixation point.

originating from the fixation point, the first and the last rows of this graph are connected. They represent the points along the rays $\theta = 0°$ and $\theta = 360°$ which are the same ray in the polar representation. Thus, every pair of ($\theta = 0°$, r) and ($\theta = 360°$, r) should be connected by an edge in the graph. The set of all the edges between nodes in the graph is denoted by Ω. Let us assume $l = \{0, 1\}$ are the two possible labels for each pixel where $l_p = 0$ indicates 'inside' and $l_p = 1$ denotes 'outside'. The goal is to find a labeling $f(P) \mapsto l$ that corresponds to the minimum energy where the energy function is defined as:

$$Q(f) = \sum_{p \in P} U_p(l_p) + \lambda \sum_{(p,q)\in\Omega} V_{p,q}.\delta(l_p, l_q),$$

$$V_{p,q} = \begin{cases} \eta \exp(-I_{E,pq}^{pol}) & \text{if } I_{E,pq}^{pol} \neq 0 \\ k & \text{otherwise} \end{cases},$$

$$\delta(l_p, l_q) = \begin{cases} 1 & \text{if } l_p \neq l_q \\ 0 & \text{otherwise} \end{cases},$$

where $\lambda = 50$, $k = 20$, $\eta = 5$, $I_{E,pq}^{pol} = (I_E^{pol}(r_p, \theta_p) + I_E^{pol}(r_q, \theta_q))/2$.

At the start, there is no information about how the inside and outside of the region containing the fixation looks. So, the data term for all the nodes in the graph except the ones in the first column and the last column is zero ($U_p(l_p) = 0, \forall p \in (r, \theta), 0 < r < r_{\max}, 0° \leq \theta \leq 360°$). The nodes in the first column correspond to the fixation point in the Cartesian space and hence must be labeled $l_p = 0$: $U(l_p = 1) = D$ and $U(l_p = 0) = 0$ for $p \in (0, \theta), 0° \leq \theta \leq 360°$. The nodes in the last column must lie outside the region and are initialized to the $l_p = 1$: $U(l_p = 0) = D$ and $U(l_p = 1) = 0$ for $p \in (r_{\max}, \theta), 0° \leq \theta \leq 360°$. See Fig. 8. For our experiments, we chose D to be 100; the high value is to make sure the initial labels do not change as a result of minimization. We use the graph cut algorithm[16] to minimize the energy function, $Q(f)$. The resulting binary segmentation is transferred back to the Cartesian space to get the desired segmentation. Fig. 6(f) shows the segmentation for the fixation (the "X") in the image Fig. 6(a).

The binary segmentation as a result of the minimization step explained above splits the polar edge map into two parts: left side (inside) and right side (outside). See Figs. 6(e) and 6(f). The color information on the left (inside) and the right (outside) can now be used to modify the data term, $U_p(\cdot)$, in the energy function $Q(f)$. The RGB value at any pixel in the polar image ($I_{rgb}^{pol}(r, \theta)$) is obtained by interpolating the RGB value at the corresponding sub-pixel location in the Cartesian space. See Fig. 6(e) for an example of such a $I_{rgb}^{pol}(\cdot)$. Let us say, $F_{in}(r, g, b)$ and $F_{out}(r, g, b)$ are the color distributions of the inside and outside, respectively. These distributions are represented by a normalized three dimensional histogram with 10 bins along each color channel. The new data term for all the nodes except the first

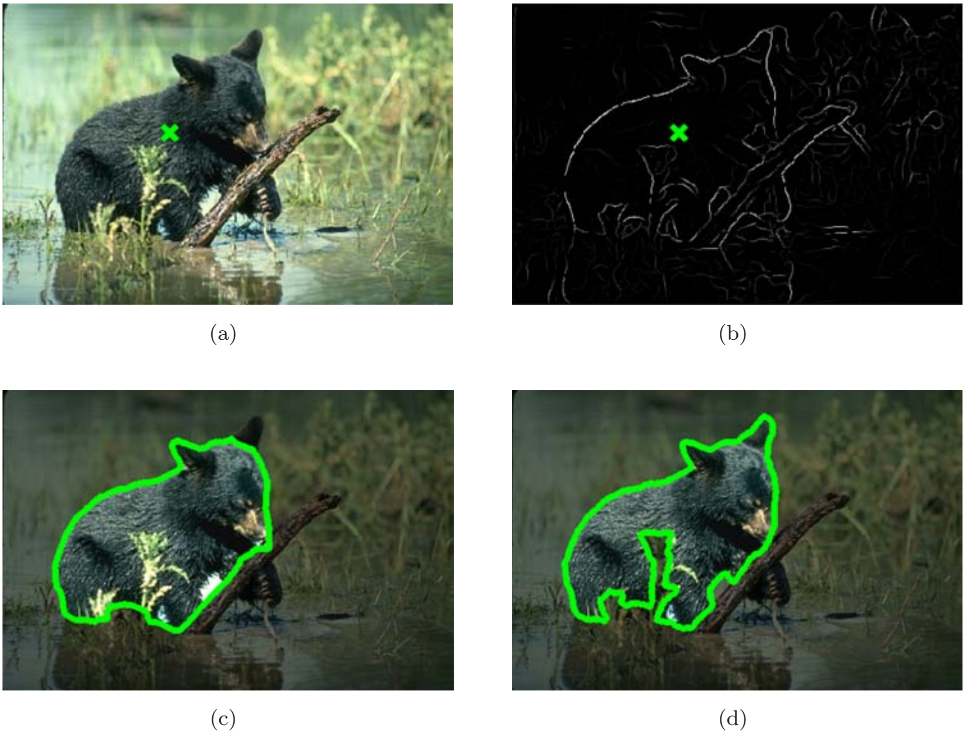


Fig. 7. (a) An image of a bear in natural setting. The location of the selected fixation is indicated by the "X". (b) The probabilistic boundary edge map. (c) The segmentation based on the edge information alone. (d) The segmentation result after combining the edge information with the color information.

and the last column nodes is:

$$U_p(l_p) = \begin{cases} -\dfrac{\ln(F_{in}(I^{pol}_{rgb}(r_p,\theta_p)))}{Z_p} & \text{if } l_p = 0 \\ -\dfrac{\ln(F_{out}(I^{pol}_{rgb}(r_p,\theta_p)))}{Z_p} & \text{if } l_p = 1 \end{cases},$$

where $Z_p = \ln(F_{in}(I^p(r_p,\theta_p)) + \ln(F_{out}(I^p(r_p,\theta_p))$. We again use the graph cut algorithm to minimize the energy function, $Q(f)$ with new data term. The segmentation result improves after introducing the color information in the energy formulation. See Fig. 7. The boundary between the left (label 0) and the right (label 1) regions in the polar space will correspond to a closed contour in the Cartesian space.

5. Results

We evaluated the performance of the proposed algorithm on 20 videos with average length of seven frames and 50 stereo pairs with respect to their ground-truth segmentation. For each sequence and stereo pair, only the most prominent object of

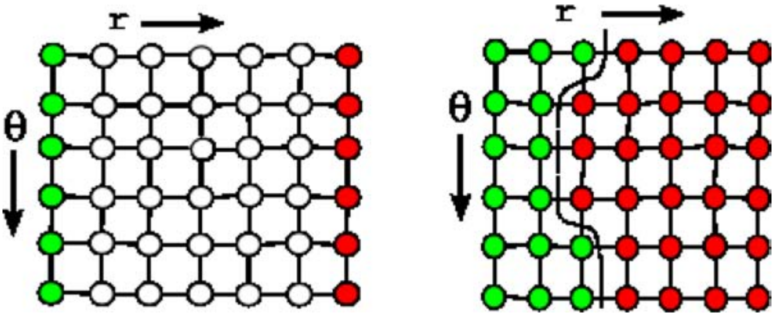


Fig. 8. Left: initialization of the first and last column of the polar image to be inside and outside the region of interest. Right: the final binary labeling as a result of minimizing the energy function using graph cut.

interest is identified and segmented manually to create the ground-truth foreground and background masks. The fixation is chosen randomly anywhere on this object of interest. The videos used for the experiment are of all types: stationary scenes captured with a moving camera, dynamic scenes captured with a moving camera, and dynamic scenes captured with a stationary camera.

The segmentation output of our algorithm is compared with the ground truth segmentation in terms of the F-measure defined as $2.P.R/(P+R)$ where P stands for the precision which calculates the percentage of our segmentation overlapping with the ground truth, and R stands for recall which measure the percentage of the ground-truth segmentation overlapping with our segmentation.

Table 1 shows that after adding motion or stereo cues with color and texture cues, the performance of the proposed method improves significantly. With color and texture cues only, the strong internal edges prevent the method from tracing the actual depth boundary. See Fig. 9(Row 2). However, the motion or stereo cues clean the internal edges as described in Sec. 3 and the proposed method finds the correct segmentation (Fig. 9, Row 3).

To also evaluate the performance of the proposed algorithm in the presence of the monocular cues only, the images from the Alpert image database[5] has been used. The Berkeley edge detector[30] provides the probabilistic boundary maps of

Table 1. The performance of our segmentation for the videos and the stereo pairs. See Fig. 9.

	F-measure
For videos	
With Motion	0.95 ± 0.01
Without Motion	0.62 ± 0.02
For stereo pairs	
With Stereo	0.96 ± 0.02
Without Stereo	0.65 ± 0.02

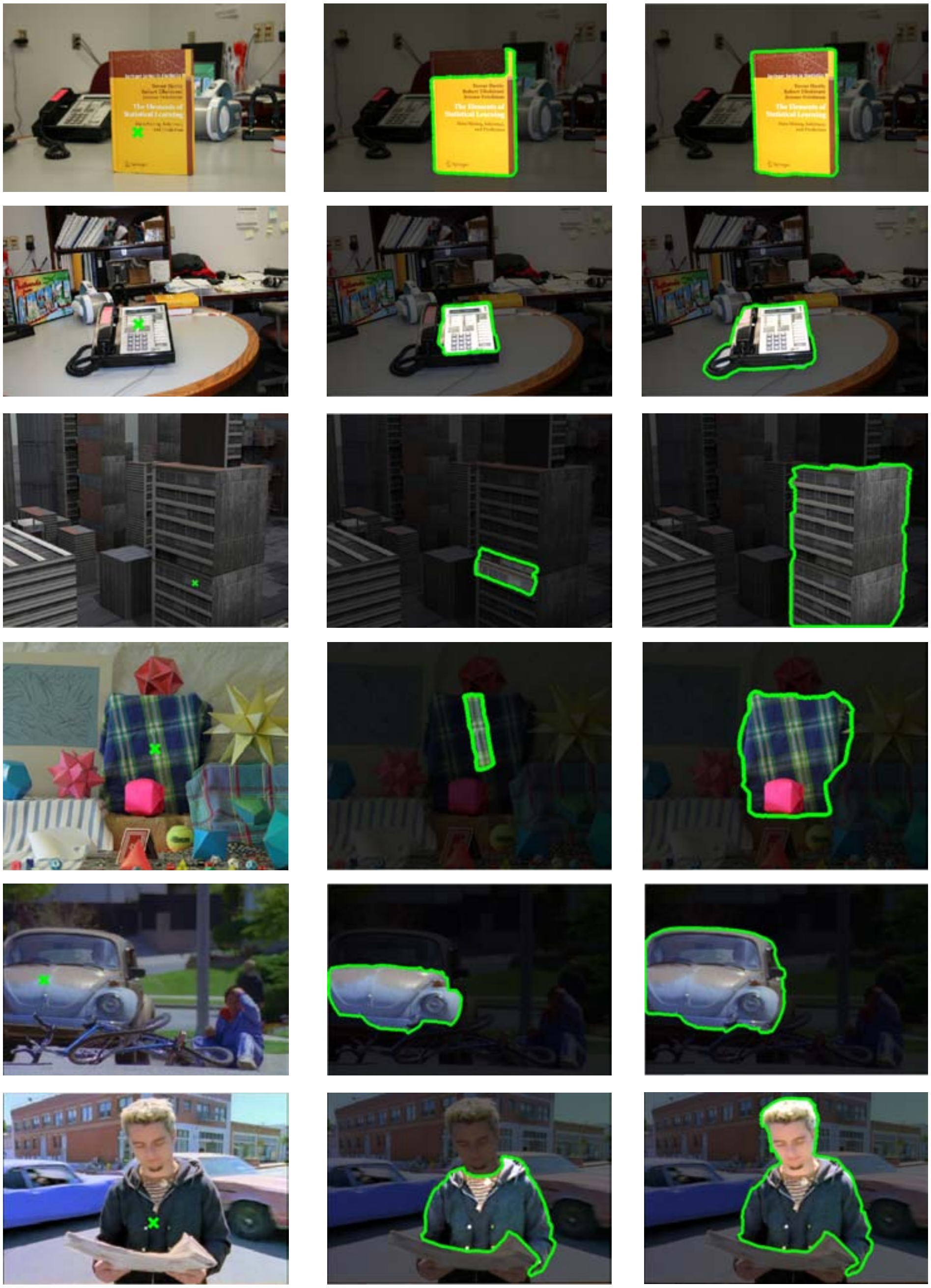


Fig. 9. Row 1–3: a moving camera and stationary objects. Row 4: an image from a stereo pair. Row 5: a moving object (car) and a stationary camera. Row 6: moving objects (human, cars) and a moving camera. Column 1: the original images with fixations (the "X"). Column 2: Our segmentation results for the fixation using monocular cues only. Column 3: Our segmentation results for the same fixation after combining motion or stereo cues with monocular cues.

Table 2. One single segment coverage results.

Algorithm	F-measure score
Bagon *et al.*[8]	0.87 ± 0.010
Alpert *et al.*[5]	0.86 ± 0.012
Our Method	0.83 ± 0.019
NCut[40]	0.72 ± 0.012
MeanShift[47]	0.57 ± 0.023

these images. The fixation on the image is chosen at the center of the bounding box around the foreground. Our definition of the segmentation for a fixation is the region enclosed by the depth boundary which is difficult to find with the monocular cues only. Table 2 shows that we perform better than that of Refs. 40 and 47 and close to Refs. 5 and 8.

6. Fixation Strategy

The proposed method clearly depends on the fixation point and thus it is important to select the fixations automatically. Fixation selection is a mechanism that depends on the underlying task as well as other senses (like sound). In the absence of these cues, one has to concentrate on generic visual solutions. There is a significant amount of research done on the topic of visual attention[26,39,50] primarily to find the salient locations in the scene where the human eye may fixate. For our segmentation framework as the next section shows, the fixation just needs to be inside the objects in the scene. As long as this is true, the correct segmentation will be obtained. Fixation points amount to features in the scene and the recent literature on features comes in handy.[28,31] Although we do not yet have a definite way to automatically select fixations, we can easily generate the potential fixations that lie inside most of the objects in a scene. Figure 11 shows multiple segmentation using this technique.

6.1. *Stability analysis*

Here, we verify our claim that the optimal closed boundary for any fixation inside a region remains same. The possible variation in the segmentation will occur due to the presence of bright internal edges in the probabilistic boundary edge map. To evaluate the stability of segmentation with respect to the location of fixation inside the object, we devise the following procedure: Choose a fixation roughly at the center of the object and calculate the optimal closed boundary enclosing the segmented region. Calculate the average scale, S_{avg}, of the segmented region as $\sqrt{Area/\pi}$. Now, the new fixation is chosen by moving away from the original fixation in the random direction by $n \cdot S_{avg}$ where $n = \{0.1, 0.2, 0.3, \ldots, 1\}$. If the new fixation lies outside the original segmentation, a new direction is chosen for the same radial shift until the new fixation lies inside the original segmentation. The overlap between the segmentation with respect to the new fixation, R_n, and the original segmentation, R_o, is given by $\frac{|R_o \cap R_n|}{|R_o \cup R_n|}$. We calculated the overlap values for 100 textured regions and 100 smooth regions from the BSD and Alpert Segmentation Database. It is clear from the graph Fig. 12(a) that the overlap values are better for the smooth regions than for the textured regions. Textured regions might have strong internal edges making it possible for the original optimal path to modify as the fixation

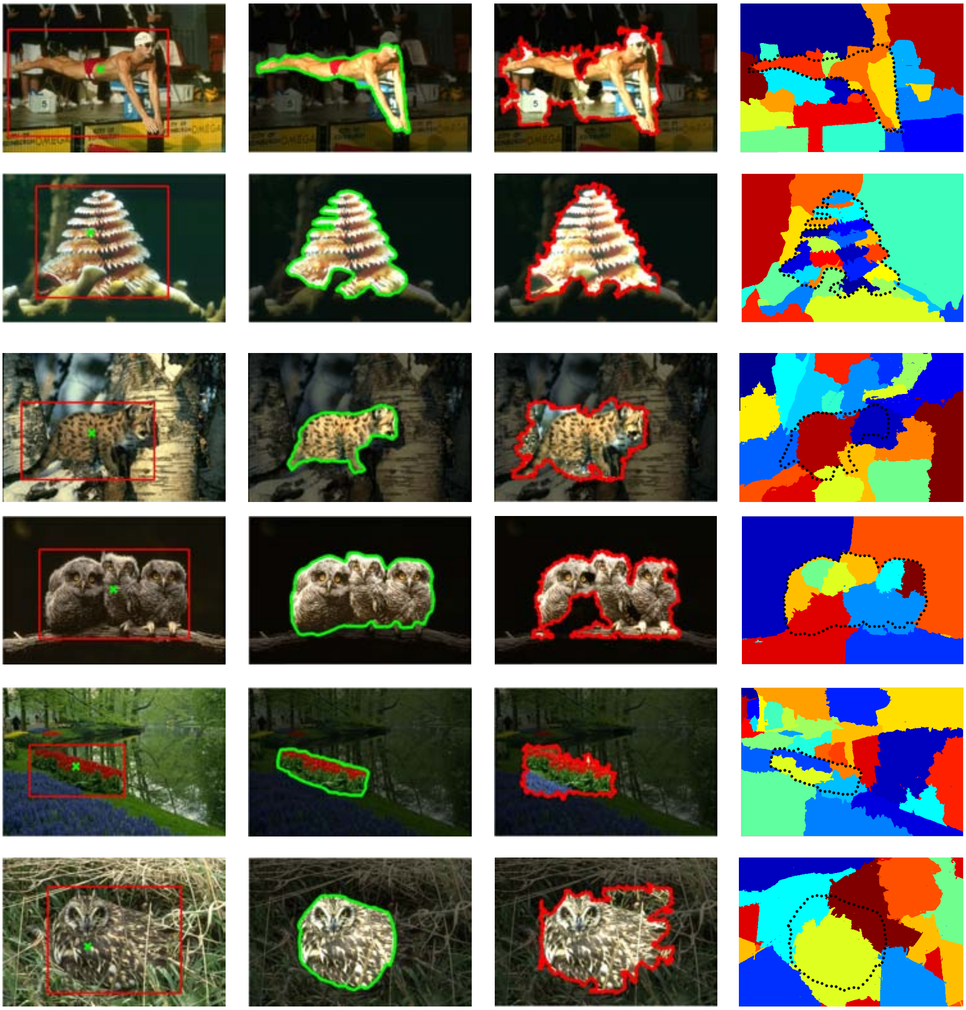


Fig. 10. The first column contains images with the fixation shown by a green "X". Our segmentation for these fixations is shown in the second column. The red rectangle around the object in the first column is the user input for the GrabCut algorithm.[37] The segmentation output of the iterative GrabCut algorithm (implementation provided by www.cs.cmu.edu/˜mohitg/segmentation.htm) is shown in the third column. The last column contains the output of normalized cut algorithm with the region boundary of our segmentation overlayed on it.

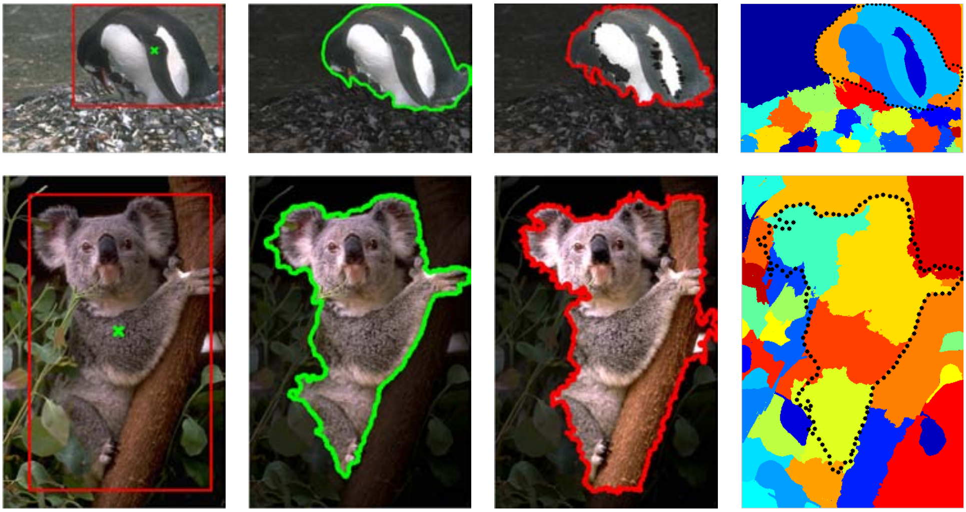
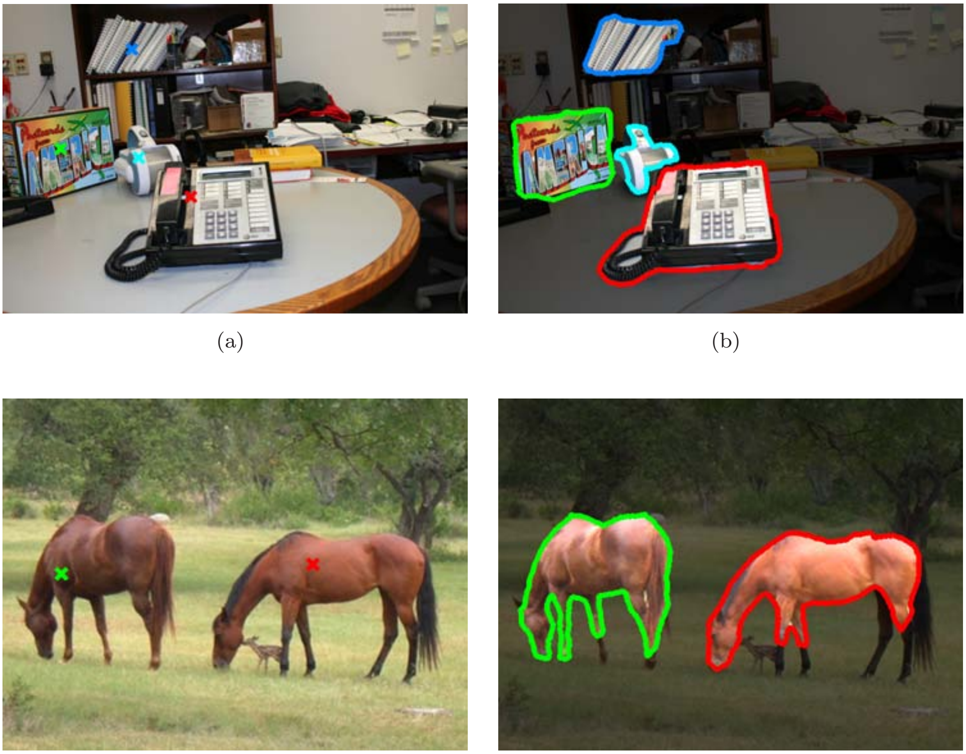


Fig. 10. (*Continued*)

moves to a new location. However, for the smooth regions, there is a stable optimal path around the fixation, it does not change dramatically as the fixation moves to a new location. We also calculate the overlap values for the 100 frames from video sequences; first with their boundary edge map given by Ref. 30, and then using the enhanced boundary edge map after combining motion cues. The results is shown in Fig. 12(b). We can see that the segmentation becomes stable as motion cues suppress the internal edges and reinforce the boundary edge pixels in the boundary edge map.[30]

7. Conclusion

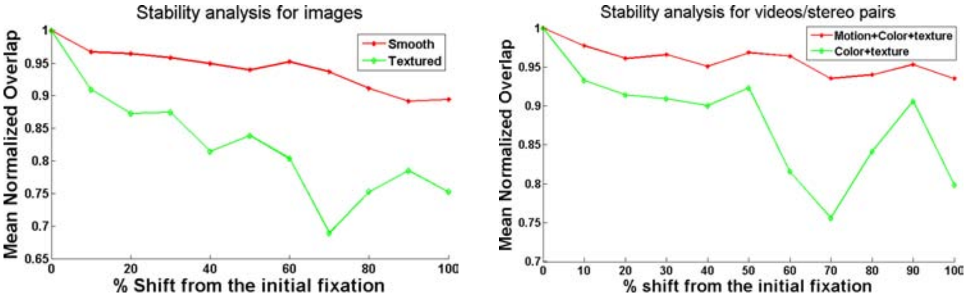
We proposed here a novel formulation of segmentation in conjunction with fixation. The framework combines monocular cues with motion and/or stereo to disambiguate the internal edges from boundary edges. The approach is motivated by biological vision and it may have connections to neural models developed for the problem of border ownership in segmentation.[21] Although the framework was developed for an active observer, it applies to image databases as well, where the notion of fixation amounts to selecting an image point which becomes the center of the polar transformation. One of the reasons for getting good segmentation with only monocular cues is the better probabilistic boundary edge map given by Ref. 30. Our contribution here was to formulate an old problem — segmentation — in a different way and show that existing computational mechanisms in the state of the art computer vision are sufficient to lead us to promising automatic solutions. Our approach can be complemented in a variety of ways, for example by introducing a multitude of cues. An interesting avenue has to do with learning models of the



(a) (b)

(c) (d)

Fig. 11. (a) and (c) are the images with multiple fixations. (b) and (c) are the regions segmented by our algorithm for those fixations. The color of the region boundary is same as the color of the corresponding fixation.



(a) (b)

Fig. 12. Stability analysis of the segmentation with respect to the locations of fixations inside the regions. (a) For images only. (b) For videos and stereo image pairs.

world. For example, if we had a model of a horse, we could segment the entire body of the horse in Fig. 3(b).

Acknowledgment

The support of (a) NSF under a collaborative project with Yale University (NETS/NOSS), (b) the European Union's 7th Framework program on Cognitive Systems under the project POETICON, and (c) NIH under grant R21 DA024322 is gratefully acknowledged.

References

1. Semantic robot vision challenge, http://www.semantic-robot-vision-challenge.org/.
2. G. Adiv, Determining 3D motion and structure from optical flow generated by several moving objects, *T-PAMI* **7** (1985) 384–401.
3. G. Adiv, Inherent ambiguities in recovering 3D motion and structure from a noisy flow field, *IEEE Trans. Patt. Anal. Mach. Intell.* **11**(5) (1989) 477–489.
4. J. Aloimonos, I. Weiss and A. Bandyopadhyay, Active vision, *IJCV* **1**(4) (January 1988) 333–356.
5. S. Alpert, M. Galun, R. Basri and A. Brandt, Image segmentation by probabilistic bottom-up aggregation and cue integration, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (June 2007).
6. P. Arbelaez and L. Cohen, Constrained image segmentation from hierarchical boundaries, *CVPR* (2008) 454–467.
7. S. Ayer, P. Schroeter and J. Bigün, Segmentation of moving objects by robust motion parameter estimation over multiple frames, *ECCV*, London, UK (Springer-Verlag, 1994), pp. 316–327.
8. S. Bagon, O. Boiman and M. Irani, What is a good image segment? A unified approach to segment extraction, D. Forsyth, P. Torr and A. Zisserman (eds), *Computer Vision — ECCV 2008*, Vol. 5305 of *LNCS* (Springer, 2008), pp. 30–44.
9. R. Bajcsy, Active perception, *Proc. of the IEEE Special Issue on Computer Vision*, **76**(8) (August 1988) 966–1005.
10. D. Ballard, Animate vision, *Artif. Intell. J.* **48**(8) (August 1991) 57–86.
11. W. A. Barrett and E. N. Mortensen, Interactive live-wire boundary extraction, *Medical Image Analysis* **1** (1997) 331–341.
12. A. Blake, C. Rother, M. Brown, P. Perez and P. Torr, Interactive image segmentation using an adaptive gmmrf model, *ECCV* (2004) 428–441.
13. M. Bober and J. Kittler, Robust motion analysis, *CVPR* **II** (1994) 947–952.
14. Y. Boykov and M. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images, *ICCV* **I** (2001) 105–112.
15. Y. Boykov and V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, *IEEE Trans. Patt. Anal. Mach. Intell.* **26** (2004) 359–374.
16. Y. Boykov and V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, *PAMI* **26**(9) (September 2004) 1124–1137.
17. T. Brox, A. Bruhn, N. Papenberg and J. Weickert, High accuracy optical flow estimation based on a theory for warping (Springer, 2004) 25–36.

18. P. Burt, R. Bergen, J. R. Hingorani, R. Kolczynski, W. Lee, A. Leung, J. Lubin and H. Shvayster, Object tracking with a moving camera, *Visual Motion 1989 Proceedings Workshop on* **II** (1989) 2–12.
19. M. Cerf, J. Harel, W. Einhäuser and C. Koch, Predicting human gaze using low-level saliency combined with face detection, *Advances in Neural Information Processing Systems (NIPS) 20* (MIT Press, 2008).
20. J. Costeira and T. Kanade, A multi-body factorization method for motion analysis, *ICCV*, p. 1071, Washington, DC, USA (1995). IEEE Computer Society.
21. E. Craft, H. Schtze, E. Niebur and R. von der Heydt, A neural model of figure-ground organization, *J. Neurophysio.* **6**(97) (2007) 4310–4326.
22. K. Daniilidis, Fixation simplifies 3D motion estimation, *Comput. Vis. Image Underst.* **68**(2) (1997) 158–169.
23. P. F. Felzenszwalb and D. P. Huttenlocher, Efficient graph-based image segmentation, *IJCV* **59**(2) (2004) 167–181.
24. C. Fowlkes, D. R. M. and J. Malik, Local figure/ground cues are valid for natural images, *JV* **7**(8) (2007) 1–9.
25. M. Irani and P. Anandan, A unified approach to moving object detection in 2D and 3D scenes, *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(6) (1998) 577–589.
26. L. Itti, C. Koch and E. Niebur, A model of saliency-based visual attention for rapid scene analysis, *T-PAMI* **20**(11) (1998) 1254–1259.
27. M. Kass, A. Witkin and D. Terzopoulos, Snakes: Active contour models, *Int. J. Comput. Vision* **1** (1988) 321–331.
28. D. G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vision* **60**(2) (2004) 91–110.
29. J. Malik, S. Belongie, T. Leung and J. Shi, Contour and texture analysis for image segmentation, *IJCV* **43**(1) (June 2001) 7–27.
30. D. Martin, C. Fowlkes and J. Malik, Learning to detect natural image boundaries using local brightness, color and texture cues, *T-PAMI* **26**(5) (May 2004) 530–549.
31. K. Mikolajczyk and C. Schmid, An affine invariant interest point detector, *Proc. Eur. Conf. on Computer Vision (ECCV)* (Springer-Verlag, 2002).
32. E. N. Mortensen and W. A. Barrett, Intelligent scissors for image composition, *SIGGRAPH* **I** (1995) 191–198.
33. R. Nelson, Qualitative detection of motion by a moving observer, *IJCV* **7** (1991) 33–46.
34. J.-M. Odobez and P. Bouthemy, MRF-based motion segmentation exploiting a 2D motion model robust estimation, *ICIP*, p. 3628, Washington, DC, USA, (1995), IEEE Computer Society.
35. K. Pahlavan, T. Uhlin and J.-O. Eklundh, Dynamic fixation and active perception, *Int. J. Comput. Vision* **17**(2) (1996) 113–135.
36. X. Ren and J. Malik, A probabilistic multi-scale model for contour completion based on image statistics, *ECCV '02: Proc. 7th Eur. Conf. on Computer Vision-Part I*, pp. 312–327, London, UK (Springer-Verlag, 2002).
37. C. Rother, V. Kolmogorov and A. Blake, "grabcut": Interactive foreground extraction using iterated graph cuts, *ACM Trans. Graph.* **23**(3) (2004) 309–314.
38. H. S. Sawhney, Y. Guo and R. Kumar, Independent motion detection in 3D scenes, *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(10) (2000) 1191–1199.
39. J. T. Serences and S. Yantis, Selective visual attention and perceptual coherence, *Trends in Cognitive Sciences*, **10**(1) (2006) 38–45.
40. J. Shi and J. Malik, Normalized cuts and image segmentation, *PAMI* **22**(8) (2000) 888–905.

41. D. Sinclair, Motion segmentation and local structure, *ICCV* **93** 366–373.
42. A. K. Sinop and L. Grady, A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm, *ICCV* (2007) 1–8.
43. W. Thompson and T. Pong, Detecting moving objects, *IJCV* **4** (1990) 39–57.
44. P. H. S. Torr, O. Faugeras, T. Kanade, N. Hollinghurst, J. Lasenby, M. Sabin and A. Fitzgibbon, Geometric motion segmentation and model selection [and discussion], *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, **356**(1740) (1998) 1321–1340.
45. P. H. S. Torr and D. W. Murray, Stochastic motion clustering, *ECCV*, pp. 328–337, Secaucus, NJ, USA, (Springer-Verlag New York, Inc, 1994).
46. B. Triggs, P. F. McLauchlan, R. I. Hartley and A. W. Fitzgibbon, Bundle adjustment — A modern synthesis, *ICCV '99: Proc. Int. Workshop on Vision Algorithms*, pp. 298–372, London, UK (Springer-Verlag, 2000).
47. Z. Tu and S. Zhu, Mean shift: A robust approach toward feature space analysis, *T-PAMI* **24**(5) (May 2002) 603–619.
48. Z. Tu and S.-C. Zhu, Image segmentation by data-driven Markov chain Monte Carlo, *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5) (2002) 657–673.
49. O. Veksler, Star shape prior for graph-cut image segmentation, *ECCV (3)* (2008) 454–467.
50. D. Walther and C. Koch, Modeling attention to salient proto-objects, *Neural Networks* **19**(4) (April 2006) 1395–1407.
51. J. Weber and J. Malik, Rigid body segmentation and shape description from dense optical flow under weak perspective, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(2) (1997) 139–143.
52. Y. Weiss, Smoothness in layers: Motion segmentation using nonparametric mixture estimation, *CVPR*, p. 520, Washington, DC, USA, IEEE Computer Society, 1997.
53. C. S. Wiles and M. Brady, Closing the loop on multiple motions, *ICCV '95: Proc. 5th Int. Conf. on Computer Vision*, p. 308, Washington, DC, USA (IEEE Computer Society, 1995).
54. L. R. Williams and D. W. Jacobs, Stochastic completion fields: A neural model of illusory contour shape and salience, *Neural Comput.* **9**(4) (1997) 837–858.
55. C. Xu and J. L. Prince, Snakes, shapes, and gradient vector flow, *IEEE Trans. on Image Processing* **7**(3) (March 1998) 359–369.
56. S. X. Yu and J. Shi, Grouping with bias, *NIPS* (2001).
57. C. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, *IEEE Trans. on Computers* **20**(1) (1971) 68–86.
58. Z. Zhang, O. Faugeras and N. Ayache, Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints, *ICCV* (1988) 177–186.
59. Q. Zheng and R. Chellapa, Motion detection in image sequences acquired from a moving platform, *ICASSP* (1993) 201–204.

Ajay Mishra received his B.Tech degree from the Indian Institute of Technology, Kanpur in 2003 and currently pursuing the Doctoral Degree at the National University of Singapore, Singapore. Since 2007, he has been a visiting Researcher in the computer vision lab at the Institute of Advanced Computer Studies, University of Maryland, College Park.

Yiannis Aloimonos (PhD 1987, University of Rochester) is a Professor of Computational Vision and Intelligence in the Department of Computer Science at the University of Maryland, College Park and the Director of the Computer Vision Laboratory at the Institute for Advanced Computer Studies. He is also affiliated with the Cognitive Science Program. He is known for his work on Active Vision and his study of vision as a dynamic process. He has contributed to the theory of Computational Vision in various ways, including the discovery of the trilinear constraints (with M. Spetsakis), and the mathematics of stability in motion analysis as a function of the field of view (with C. Fermuller), which led to the development of omni directional sensors. He has received several awards for his work (including the Marr Prize for his work on Active Vision, the Presidential Young Investigator Award from President Bush (1990) and the Bodossaki Prize in Artificial Intelligence). He has coauthored four books, including Active Perception and Visual Navigation. He is interested in cognitive systems, specifically the integration of visual cues and the integration of vision, action and language.