The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Active Segmentation for Robotics

Ajay Mishra Yiannis Aloimonos Cornelia Fermuller
Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742
{mishraka@umiacs,yiannis@cs,fer@umiacs}.umd.edu

Abstract— **The semantic robots of the immediate future are robots that will be able to find and recognize objects in any environment. They need the capability of segmenting objects in their visual field. In this paper, we propose a novel approach to segmentation based on the operation of fixation by an active observer. Our approach is different from current approaches: while existing works attempt to segment the whole scene at once into many areas, we segment only one image region, specifically the one containing the fixation point. Furthermore, our solution integrates monocular cues (color, texture) with binocular cues (stereo disparities and optical flow). Experiments with real imagery collected by our active robot and from the known databases [1] demonstrate the promise of the approach.**

I. INTRODUCTION

Twenty years ago, researchers asked a number of basic questions about vision systems, which led to the Active Vision Revolution in the late 80s. A flurry of activity followed in several areas. 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. The success of SLAM (simultaneous localization and mapping) which grew out of Active Perception and the existence of gargantuan programs such as the one on Future Combat Systems funded by the DoD, are a testimony to the success of that early work in the field [2], [3], [4].

Now the field has developed numerous techniques for successfully dealing with large spaces (e.g. SLAM) 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 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).

This paper is devoted to one of the basic visual competences – automatic segmentation – that robots need to function in a realistic environment. The robot should possess

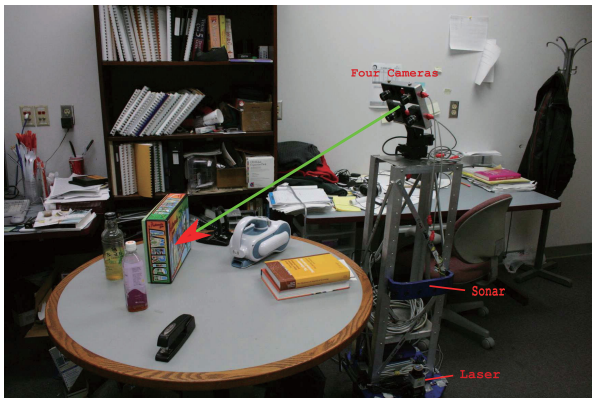


Fig. 1. Our robot with a Quad camera vision system mounted on top. The green arrow indicates the line of sight as it fixates on an object in the scene.

capability to segment a part of the image that it sees and then recognize it as some kind of object. More precisely, there should be an attention mechanism which makes the robot fixate at certain location in the scene and the segmentation method then carves out the boundary of the region/object containing that fixation location. This way the robot can keep looking in the scene till it finds the object of interest. The problem of segmentation however is an open question and constitutes a core challenge addressed in this paper. We are interested in the solutions that are generic and can be used by a variety of robots, since the problem of visually segmenting objects is universal.

II. ACTIVE SEGMENTATION

The problem of image segmentation has occupied scientists, philosophers and engineers for many years, with very interesting results. But what does it really mean to segment an image or part of it? Is segmentation a bottom-up process resulting from analysis of the perceptual input? Or is it perhaps a top-down process where we recognize the object and thus we segment it? Or could it be a synergistic process where bottom-up processes interact with top down attention mechanisms? Questions like these remain a challenge even today.

The most prominent definition of segmentation in the literature is dividing the image into regions with some homogeneous property. This is a very general definition and it clearly includes the case where the system processes only a single image. Thus, a large part of today's literature is devoted to segmenting single images, which is not surprising as many

applications in today are driven by image databases requiring single image segmentation and related visual computing. It is nevertheless noteworthy that the ability to segment a single image is solely a human capability only humans segment and understand pictures (because only when we see a photograph we see only one image.) Our semantic robot however, like biological systems, does not look at single images it receives binocular video and fixates at different parts of the scene. It is an active system. Because of that it is able to understand the spatial layout of the scene, it can understand boundaries and occlusions. This gives us the freedom to revise the definition of segmentation, to a more concrete and physical one.

We define segmentation as the division of the image (or view) into regions that correspond to different surfaces. Thus, the boundaries of the segmentation will be depth boundaries. Note that this definition is sound because every object occupies some volume of space and has boundaries. You may of course find a picture on the wall where the boundaries of the segment (the picture) are not depth boundaries, but for the objects our semantic robot will deal with, it is certainly true that segmentation boundaries are depth boundaries .

Is our definition a good one? We know that humans when look at a scene do not segment the whole scene at once. They only segment the region on which they fixate. In fact, the structure of the human retina is such that only the small neighborhood around fixation points is captured in high resolution by the fovea, while the rest of the scene is captured in lower resolution by the sensors on the periphery of the retina. This is the well known figure vs ground problem [5], [6]. Why then would we want our segmentation algorithms to segment the whole scene? Is this not too ambitious?

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 interest in the scene which can either be an object or just a part of it. Our semantic robot can make fixations as well. 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 [2], [3], [4], [7], [8]. 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. It is only natural to make fixation part and parcel of any visual processing. We will not segment the whole scene at once. Instead, our semantic robot, shown in Fig. 1, will fixate at points in the scene and segment the surface (object) containing the fixation point.

III. Our approach: polar space is the key!

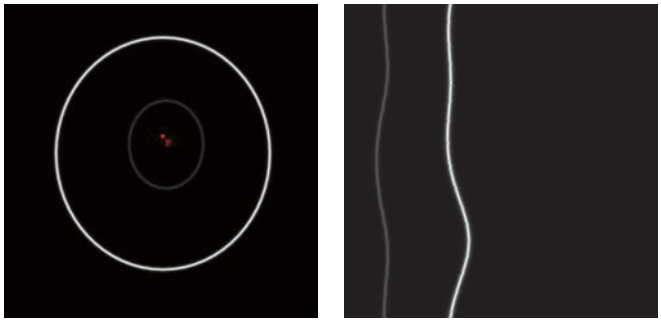
The goal to find the region containing the fixation is equivalent to finding the enclosing closed contour around the fixation. Fig. 3(a) shows an image that our semantic robot sees in the room (fixation is shown by the green cross). Fig. 3(b) shows the boundary edge map of the image given by the semantic edge detector [9] which has learned from an annotated database how the edge at depth boundary looks like. Thus, in Fig. 3(b), edges along the depth boundaries are bright, whereas the texture (internal) edges are dim. The brightness of an pixel of this map is proportional to its probability of being at a depth boundary. We must emphasize that this is not all we have: we also have a disparity stereo map and an optical flow field at each time stamp to further improve this boundary edge map as discussed later in the paper in section V. However, for now let us concentrate on a single image.

But as we can see in Fig. 3(b), there are many ways to enclose the fixation point due to the presence of internal contours. Thus, we define the optimal contour as the one with minimum cost where the cost of tracing an enclosing contour is an accumulated cost of adding all the edge pixels along the contour. Such a cost however depends on the number of edge pixels in the contour or the length of the contour in the Cartesian space, making tracing small contours inherently preferable over long ones. This means it is important to normalize the lengths of all possible contours before trying to find the optimal one.

To explain it further, let us consider two enclosing contours around the fixation F, as shown in Fig. 2(a), of length 40 and 100 pixels and of constant brightness 150 and 200 respectively, where 255 is the maximum brightness. The accumulated cost of tracing the small and the long contour is $4200(=(255-150)\times 40)$ and $5500(=(255-200)\times 100)$ respectively. The small contour costs less and hence will be declared optimal. However, the long contour is brighter than the short contour and should actually be the optimal boundary around the fixation point F. But, for that to happen, lengths of the possible closed paths around the fixation should be normalized such that their accumulated cost does not depend on their length.

To achieve this, the boundary edge map, given in Fig. 2(a), is transformed from the Cartesian co-ordinate system to the polar co-ordinate system with fixation as the pole. (Our convention is that the angle $(\theta \in [0^\circ, 360^\circ])$ is represented along the vertical axis and increases from top to bottom and the radial distance (r) is represented along horizontal axis and increases from left to right.) Now, in the polar space (Fig. 2(b)), both enclosing contours become the open curves of normalized length (360 pixels). The brightness of the edge pixels in the polar space remain unchanged. The cost of tracing the dimmer contour and the brighter contour is $3600(=360\times(255-150))$ and $1800(=360\times(255-200))$ respectively. The brighter contour now costs less and becomes the optimal closing contour around the fixation point when transformed back to the Cartesian co-ordinate system.

In this paper, we propose a two step process to segment the region for a given fixation: first, the boundary edge map of the image/scene is generated by using all available low level cues such that the edges at the depth boundaries in the image are brighter than the other internal edges. This process is described in section V; second, the boundary edge



(a) (b)

Fig. 2. (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.

map is transformed to the polar space with the fixation as the pole. The optimal path passing through the polar edge map is found using a modified version of the shortest path algorithm described in section VI. We mention that a graph cut based alternate implementation to find the optimal path has also been proposed in [10]. The interested reader can refer to it for details.

The rest of the paper is organized as follows: A short review of the different segmentation algorithms is given in section IV. Section V and section VI describe the two major steps of our algorithm in detail. The results obtained using our algorithm are shown in section VII.

IV. Related Work

Segmentation algorithms always depend upon user inputs to define the optimal segmentation of a scene (or image). The definition of the optimal segmentation is invariably linked with the object of interest which should ideally be represented by one segment. But it is hard to determine the appropriate parameter as the object can appear at different scales in the scene. In fact, it is inappropriate to define the optimal segmentation of a scene if it has two or more objects present at significantly different scales (see Fig. 4). The best segmentation with respect to one object will result in either over or under segmentation of the other objects in the scene. Such segmentation algorithms [11], [12], [13] are unusable for our semantic robot as they need the global parameters such as number of regions, clustering thresholds and segment the entire scene. The interactive segmentation algorithms [14], [15], [13], however, take a different approach. They always segment the image into only two regions: foreground and background. Though this approach is more similar in spirit to our work, it cannot be automated the user must supply the critical parameters to the algorithm like a rectangle around the object in the scene or seed points from the desired foreground and the background. Besides all this, these algorithms can only handle monocular cues whereas we intend to segment the objects by combing monocular with motion/stereo cues. [16] combines color, texture and

(a) (b) (c)

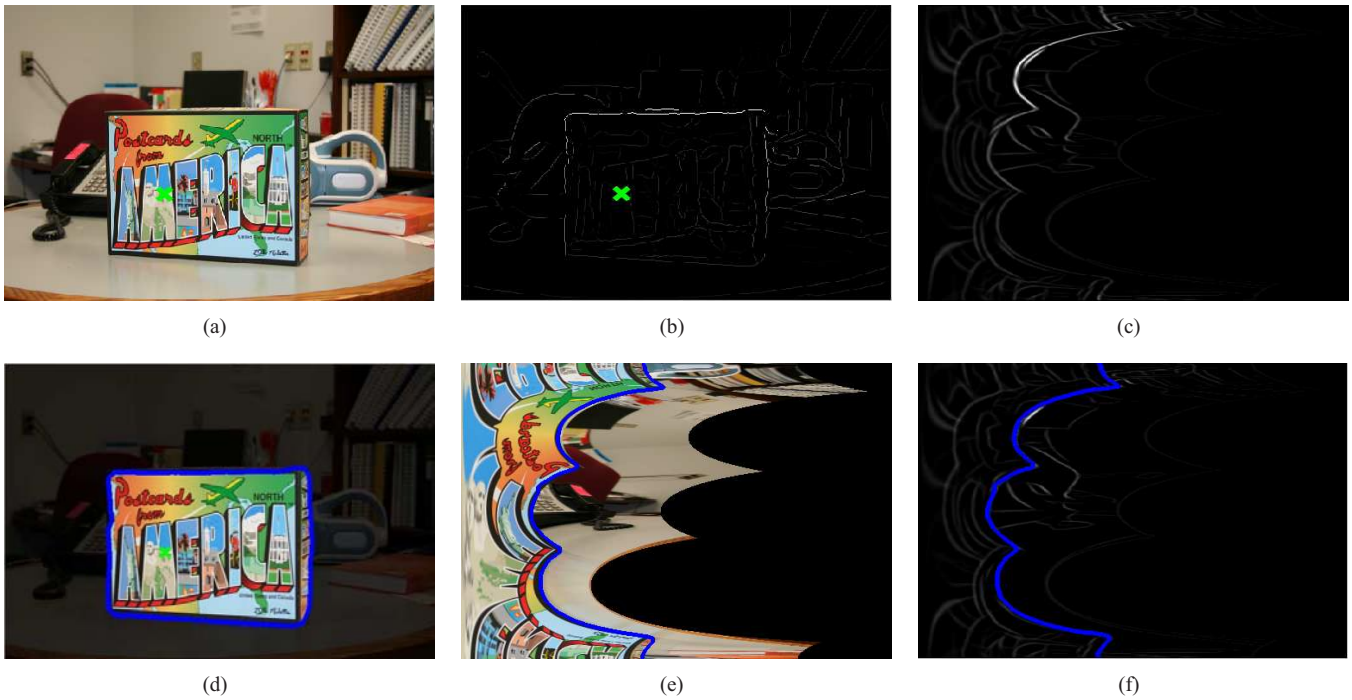
Fig. 4. The segmentation of the image (a) using the normalized cut algorithm [11] are shown in (b) and (c) when its parameter (estimated number of segments) is set to 10 and 60 respectively. (b) is the desired segmentation if the trees are of interest whereas if the user is interested in the horse (c) is the better segmentation.

stereo cues to segment a binocular video into foreground and background regions, but their videos are captured with a static camera.

V. Improving Boundary Edge Map Using Binocular Cues

To be able to correctly trace the depth boundary around the fixation point, the edge pixels along the depth boundary in the edge map should be stronger (brighter) than the rest of the edge pixels. But, the boundary edge maps given by Martin et al. [9], though significantly better than a plain gradient map, have some strong (or brighter) internal edges and weak (or lighter) boundary edges. As we know that there is a bigger step change in the disparity or flow values at the boundary edges than at the internal edges, we use this fact to modify the boundary edge map further such that the boundary edges in the modified edge map are almost always brighter than the internal edges. To do so, the disparity or the flow map of the image is first calculated depending upon whether stereo or motion is being used. (The flow map for the image sequence used in our experiment is shown in Fig. 5(b).)

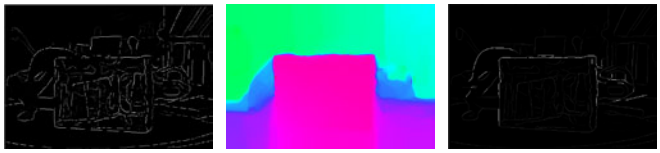
Now, the boundary edge map is broken into straight line segments. On the two sides of each line segment, rectangular regions of fixed width (10 pixels for our experiments) parallel to the segment are selected. We calculate the average disparity and/or average flow inside the two regions. The difference between the average disparity and the magnitude of average flow in the two regions is represented by $\triangle d$ and $\triangle f$ respectively. Now, the brightness of an edge pixel on the line segment is changed as $I(x,y) = \alpha_b I(x,y) + (1-\alpha_b)(\Delta f / max(\Delta f))$ or $I(x,y) = \alpha_b I(x,y) + (1-\alpha_b)(\Delta d / max(\Delta d))$ in the case of motion or stereo cues respectively. α_b is the weight associated with the relative importance of the monocular cue based boundary estimate and $I(x,y)$ is the probability of the edge pixel (x,y) to be at the boundary. For our experiments, we chose α_b to be 0.2. It is important to mention that the rectangular regions are not immediately close to the edge owing to the corrupted flow/disparity values at the depth boundary in the image. The regions are at a distance (5 pixels for our experiments) away from the boundary. Fig. 5(c) shows the improved boundary edge map as a result of this process. With the improved boundary edge map, our algorithm traces the real boundary of the region as shown in the third column of Fig. 7.



(a) (b) (c)

(d) (e) (f)

Fig. 3. (a) The first frame of the image sequence captured with a moving camera. The fixation point is shown by a green "X". (b) The final probabilistic boundary edge map as obtained in section V. (c)The polar image of the the boundary edge map for the fixation. (d) The polar edge map with the optimal path (as calculated in section VI) shown by the blue curve. (e) The color image after the polar transformation with the optimal path (the blue curve) superimposed on it. (f)The region segmented by our algorithm containing the fixation point.



(a) (b) (c)

Fig. 5. (a) The boundary edge map as given by [9] for the image in Fig. 3(a). (b) the color coded map of the magnitude of optical flow [17]. (c) the improved boundary edge map using motion information. Note how the bright internal edges on the box in (a) become dim whereas the depth boundaries become brighter.

VI. Segmentation for a Fixation

A. Cartesian to polar edge map

Let us assume $G^c(\cdot)$ and $\Theta^c(\cdot)$ are the gradient and the orientation map of the edge pixels in the image respectively in the Cartesian space. $G^p(\cdot)$ is the polar transformation of $G^c(\cdot)$ with (x_o, y_o) as the pole. The Cartesian to polar transformation is usually achieved by simple bi-linear interpolation. A pixel location (r, θ) in the polar co-ordinate system corresponds to a sub-pixel location (x, y) : $x = rcos\theta + x_o, y = rsin\theta + y_o$ in the Cartesian space. So, the gradient value $G^p(r, \theta)$ is the same as the value calculated by bi-linear interpolation in the Cartesian gradient edge map $G^c(\cdot)$ at the subpixel location (x, y), which considers only the four immediate neighbors.

We propose an alternate way to generate the polar edge map $G^p(\cdot)$. Let E be the set of all edge pixels. The gradient value $G^p(r, \theta)$ is calculated by sampling a continuous 2D function at the subpixel location (x, y) as given here:

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

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

where $\sigma_{x_e}^2 = \frac{K_1}{\sqrt{(x_e - x_o)^2 + (y_e - y_o)^2}}$, $\sigma_{y_e}^2 = K_2, \theta_e = \Theta^c(x_e, y_e)$, $K_1 = 900$ and $K_2 = 4$ are constants.

This function is generated by placing 2D Gaussian kernel functions on every edge pixel. It connects the broken edge fragments in the process. The Gaussian kernel functions at an edge pixel is placed such that its major axis aligns with the orientation at the edge location. The variance of the kernel along the same major axis will be inversely proportional to the distance between the edge pixel and the pole (x_o, y_o). The reason for setting the square of variance along 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 same as the corresponding edge pixel in the Cartesian edge map. The intuition behind using variable width kernel functions is: Imagine an edge pixel is 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. In the end, G^p is scaled such that the gradient values lie between 0 and 1.

B. Finding the optimal boundary in the polar edge map

Definition: *Given the polar edge map G^p for a fixation, the closed boundary around that fixation in the Cartesian co-ordinate system corresponds to a connected set of pixels, $O = \{p_1, p_2, ..., p_n\}$, in the polar co-ordinate system, where $p_i = (r_i, \theta_i), r_i = r_n, \theta_n = \theta_1 + 360°$, and p_j is 4 neighbor of p_{j-1}.*

The optimal path O'is the one corresponding to the minimum cost, where the cost for path O is defined as:

$$C(O) = \sum_{i=1}^{n} h(p_i)$$

$$h(p_i) = \begin{cases} G^p_{max} - G^p(r_i, \theta_i) & \text{if } G^p(r_i, \theta_i) \neq 0 \\ k.G^p_{max} & \text{otherwise} \end{cases}$$

where G^p_{max} is the maximum intensity of the polar edge map G^p and $k = 50$ to penalize heavily for including any non edge pixel into the path. The path should stay along the bright edge pixels in the polar edge map. Consider every pixel as a node in the graph and is connected to their 4 neighbors. The first row $(\theta = 0°)$ is connected to the last row $(\theta = 360°)$ because the closed boundary can cut any ray multiple times. To find the optimal path through the polar map, we can use a modified version of Dijkstra's algorithm to find the shortest path between two nodes in a graph as proposed in [18].

As the shortest path is found between two nodes, we create a source node that is connected to all nodes in the first row of the graph, and the target node is connected to the last row of the graph. We also define $h(s) = h(t) = 0$, $N(s) = \{p = (r, 0°) : 0 < r < r_{max}\}$ and $N(t) = \{p = (r, 360°) : 0 < r < r_{max}\}$. See Fig. 6. The shortest path between the source and the target node found using our algorithm is $\{s, p_1, p_2, ..., p_n, t\}$ (see Fig. 3(d). This optimal path is then mapped back to the Cartesian co-ordinate system providing the closed boundary around that fixation point (see Fig. 3(f).

It is possible that sometimes the path found in the first step is not closed, meaning $r_1 \neq r_n$. In that case, to close the loop, the first and the last points should be same. We assume that one of the two points $\{p_1, p_n\}$ is a correct end point. We pick p_1 first and disconnect all the links between the source and the nodes $(r, 0°)$ in the first row of the graph except to $p_1 = (r_1, 0°)$. The connections to the target from all the nodes along the last row except $(r_1, 360°)$ are disconnected. The new shortest path between source and target nodes is calculated with its cost. The same process is repeated for the case when the source is connected to a single node $(r_n, 0°)$ in the first row and the target to a single node $(r_n, 360°)$ in the last row. At the end of it, the best of these two paths is considered the optimal path through the polar edge map.

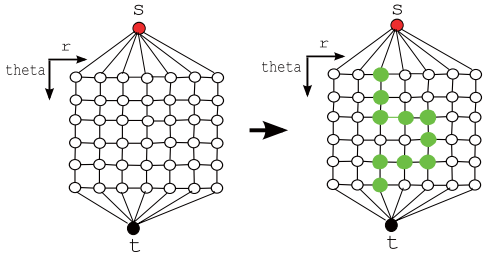


Fig. 6. Lattice representing the polar edge map with circles as the pixels(or nodes). Source node 's' and target node 't' are shown by red and black circle. The lattice on the right shows a possible shortest path from 's' to 't'. A path from 's' to 't' through the lattice is highlighted by green circles.

Algorithm 1 Optimal Path Search

Input:
s (source), t (target)
Data Structures:
A (list of active pixels sorted by cost)
N(p) (4 neighbors of pixel p)
e(p) (Boolean function indicating if p is expanded)
T(p) (total cost of tracing path from source to point p)
Output:
B (pointers from each pixel for minimal path to source)
Algorithm:
$T(s) \leftarrow 0$**;**
$A \leftarrow s$**;**
$P \leftarrow min(A)$**;**
while A $\neq \phi$ **and** $p \neq t$ **do**
  e(p) $\leftarrow$ **TRUE**
  for q $\in$ **N(p) and e(q) is TRUE do**
    tmp $\leftarrow T(p) + h(q)$**; (cost of adding pixel q to the path)**
    if q $\in A$ **and** $tmp < T(q)$ **then**
      q $\leftarrow A$**; remove q from the active list**
    else if $q \notin A$ **then**
      T(q) $\leftarrow tmp$**;**
      B(q) $\leftarrow p$**;**
      A $\leftarrow q$**;**
    end if
  end for
  $P \leftarrow min(A)$
end while

VII. Results

We evaluated the performance of our algorithm on 20 videos (of average length seven frames) and 50 stereo pairs with respect to the ground-truth segmentation of the data. The most prominent object in every frame is segmented manually to create the ground-truth. The fixation is chosen randomly anywhere on this object of interest. These videos have been captured with a moving camera and also have multiple independently moving objects in them.

The segmentation results given by our algorithm are compared with the ground truth segmentations in terms of the F-measure defined as $2PR/(P+R)$ where P stands for the precision which calculates the fraction of our segmentation overlapping with the ground truth, and R stands for recall which measure the fraction of the ground-truth segmentation overlapping with our segmentation. The results are given in the Table I. The source code of our implementation is available at *http://www.umiacs.umd.edu/∼mishraka/activeSeg.html*. In

For Videos	F-measure	For Stereo Pairs	F-measure
With Motion	0.95 ± 0.01	With stereo	0.96 ± 0.02
Without Motion	0.72 ± 0.02	Without stereo	0.68 ± 0.02

TABLE I

SEGMENTATION PERFORMANCE.

Fig. 7. Left: The frame of an image sequence captured with a moving camera. The fixation is shown by a green X. Middle: the segmentation using the boundary edge map as returned by martin et al.[9] using only monocular cues. Right: the segmentation using modified boundary edge map obtained by using motion cues to differentiate between internal and boundary edges.

Fig. 8, we show that even with the boundary edge map [9] based on color and texture cues only, we are able to segment the fixated regions successfully. We have also attached with the paper the segmentation output for a challenging test video.

VIII. 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 [19], [20], [21] primarily to find the salient locations in the scene where the human eye may fixate. For our segmentation framework, 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 [22], [23]. 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. Fig. 8 shows multiple segmentation using this technique.

IX. CONCLUSION

We proposed a novel formulation of segmentation in conjunction with fixation. The framework combined monocular cues with motion and/or stereo to identify the boundary edges in the scene which helps the algorithm trace the depth boundaries around the fixation point. Our contribution

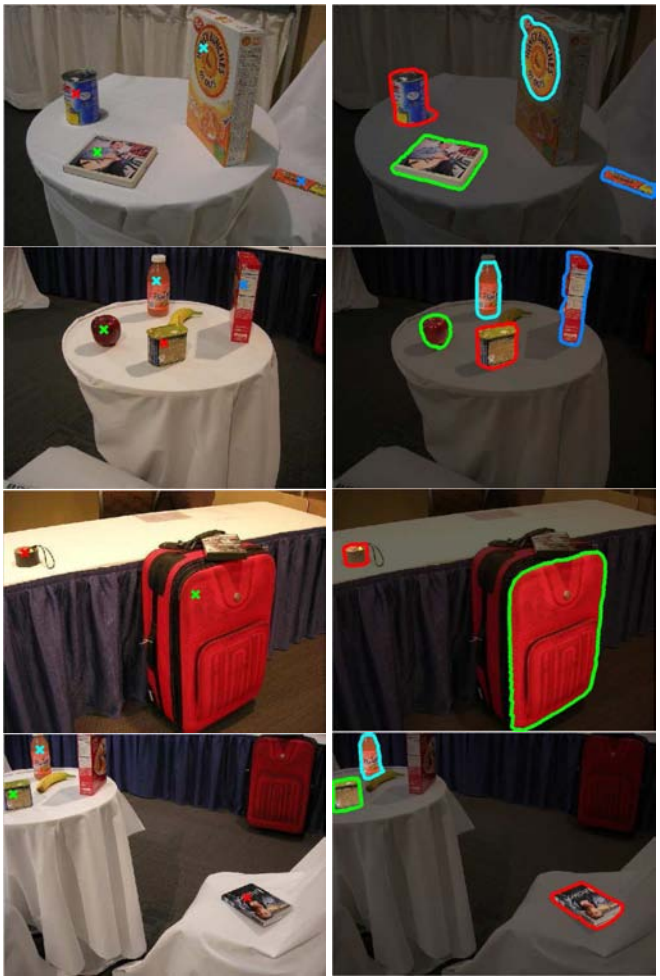


Fig. 8. Left column: fixations shown by cross signs of different colors. Right column: Segmentation corresponding to those fixations. These images are from the semantic robot vision challenge.

was to formulate an old problem – segmentation – in a new way and show that existing computational mechanisms in the state of the art computer vision were sufficient to give us a promising automatic solution to the segmentation problem. Our approach, which is still in its beginnings, can be complemented in a variety of ways, for example by introducing other high level cues. An interesting avenue is to use shape models of objects from the world to successfully segment any instance of these objects in the scene. For example, if we had a model of a ”suitcase”, we could segment the entire suitcase correctly in Fig. 8, instead of just a part of it. This interaction between low level bottom up processing and high level top down attentional processing, is a fruitful research question.

X. ACKNOWLEDGMENTS

The support of the European Union (Cognitive Systems, project POETICON) is gratefully acknowledged.

REFERENCES

[1] “Semantic robot vision challenge. http://www.semantic-robot-vision-challenge.org/.”

[2] J. Aloimonos, I.Weiss, and A. Bandyopadhyay, "Active vision," *IJCV*, vol. 1, no. 4, pp. 333–356, January 1988.
[3] D. Ballard, "Animate vision," *Artificial Intelligence Journal*, vol. 48, no. 8, pp. 57–86, August 1991.
[4] R. Bajcsy, "Active perception," *Proc. of the IEEE special issue on Computer Vision*, vol. 76, no. 8, pp. 966–1005, August 1988.
[5] J.-O. Eklundh, "Vision in robotics: How a robot can segment figure from ground," in *ECAI*, 1998, pp. 689–693.
[6] P. Nordlund and J.-O. Eklundh, "Real-time maintenance of figure-ground segmentation," in *ICVS '99: Proceedings of the First International Conference on Computer Vision Systems*. London, UK: Springer-Verlag, 1999, pp. 115–134.
[7] K. Pahlavan, T. Uhlin, and J.-O. Eklundh, "Dynamic fixation and active perception," *IJCV*, vol. 17, no. 2, pp. 113–135, 1996.
[8] K. Daniilidis, "Fixation simplifies 3d motion estimation," *CVIU*, vol. 68, no. 2, pp. 158–169, 1997.
[9] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color and texture cues," *T-PAMI*, vol. 26, no. 5, pp. 530–549, May 2004.
[10] A. Mishra and Y. Aloimonos, "Active segmentation with fixation," in *ICCV*, 2009.
[11] J. Shi and J. Malik, "Normalized cuts and image segmentation," *T-PAMI*, vol. 22, no. 8, pp. 888–905, 2000.
[12] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, pp. 167–181, 2004.
[13] Z. Tu and S. Zhu, "Mean shift: a robust approach toward feature space analysis," *T-PAMI*, vol. 24, no. 5, pp. 603–619, May 2002.
[14] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *ICCV*, 2001, pp. I: 105–112.
[15] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.
[16] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother, "Bi-layer segmentation of binocular stereo video," in *CVPR*, 2005, pp. II: 407–414.
[17] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping." Springer, 2004, pp. 25–36.
[18] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," in *SIGGRAPH*, 1995, pp. I: 191–198.
[19] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 4, pp. 1395–1407, April 2006.
[20] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *T-PAMI*, vol. 20, no. 11, pp. 1254–1259, 1998.
[21] J. T. Serences and S. Yantis, "Selective visual attention and perceptual coherence," *Trends in Cognitive Sciences*, vol. 10, no. 1, pp. 38 – 45, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/B6VH9-4HNSPSP-3/2/7578445d13145acb3c7352786e770868
[22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
[23] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *ECCV*. Springer-Verlag, 2002.