Segmentation of Moving Objects using Cue Integration

Ajay Mishra, Yiannis Aloimonos
University of Maryland, College Park

Abstract

The ability to extract independently moving objects in a video is an essential requirement for a vision system to be able to learn about objects in an unsupervised manner. But the focus of most motion segmentation algorithms has been to group pixels with similar motion characteristics into regions, and usually ignore static cues. In this paper we propose a segmentation process that will extract, using both dynamic and static visual cues, only the regions corresponding to the moving objects. Instead of grouping pixels into regions, the proposed process groups edge pixels into the closed boundaries of the moving objects in the video.

The grouping of the edge pixels is accomplished using [16] that finds the closed boundary around a given point in an image without being affected by the size of the boundary. We propose a strategy to select the points inside the moving objects and use [16] to generate closed boundaries for the selected points. Following segmentation, a novel process selects a subset of the closed boundaries corresponding to the moving objects. We evaluate quantitatively the performance of our proposed system in extracting moving objects from videos and the results show its usefulness in generating the closed boundaries (or regions) for high-level visual processing such as object recognition.

1. Introduction

Motion information is critical for visual perception. It can help a vision system learn about objects in an unsupervised manner. But the focus of most motion segmentation algorithms has been to group pixels with similar motion characteristics into regions. While having regions instead of pixels for high level visual processing reduces computational complexity, not knowing which subset of regions actually correspond to the moving objects in the scene makes it hard to learn new objects in a truly bottom-up fashion. The problem is especially hard when both the objects and the camera are moving in a scene. We propose a segmentation process that will extract only the regions corresponding to the moving objects using both dynamic as well as static visual cues.

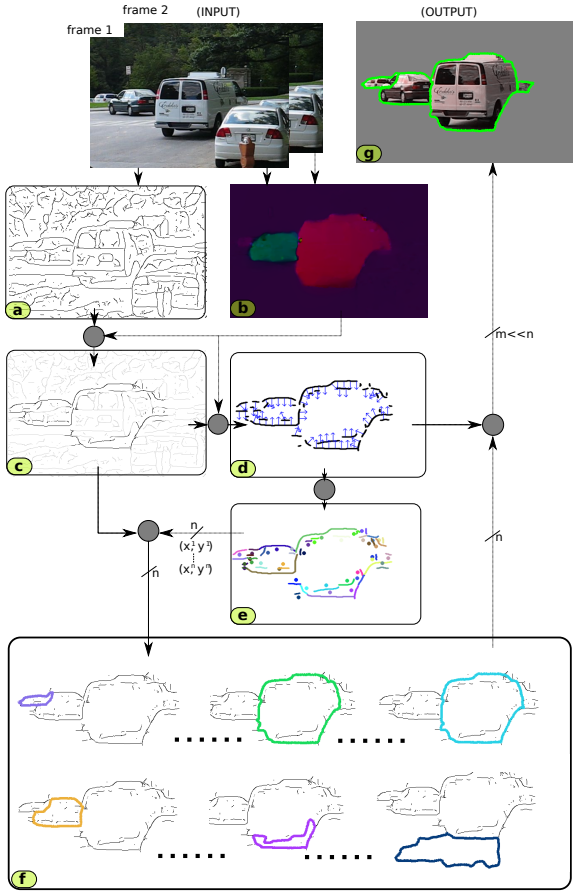


Figure 1. Segmentation Process. (a) The color and texture based binary edge map. (b) The optical flow map. (c) The probabilistic boundary edge map obtained by combining (a) and (b) - the edges with a motion discontinuity across them. (d) The predicted boundaries of moving objects obtained from (c), also the arrows indicating the object side of the predicted boundary pixels are computed using (b). (e) Candidate fixation points inside the moving objects. (f) The closed boundaries for all fixation points. For each fixation point, the fixation-based segmentation method**(author?)** [16] finds the optimal closed boundary by combining the edge pixels in (c) such that it encloses the fixation point. (g) The final segmentation which is effectively the minimum number of closed boundaries in (f) that traces almost all of the predicted boundary pixels in (d).

Recently, Mishra et al.[16] proposed a strategy that, instead of segmenting an entire scene all at once, segments

one region at a time, which is identified by a single point (called fixation point in an anthropomorphic sense) inside that region provided as input and the segmentation of the region is not affected by its scale. With this fixation-based approach, segmenting multiple moving objects will entail selecting points inside each of the objects and carrying out the segmentation process repeatedly. But[16] did not propose a strategy to select those points. So, in this paper, we propose a strategy to select the points inside the moving objects without knowing either their actual number or their sizes.

With this fixation strategy, we build on the above mentioned fixation based segmentation approach. Our segmentation process has four major components: first, color, texture and motion cues are used to find the motion boundaries, represented by a probabilistic boundary edge map (section 3); second, the points inside the moving objects are selected (section 4); third, for every point, an optimal closed boundary enclosing that point is found (section 5); fourth, a subset of the closed contours, which correspond to the moving objects, is selected (section 6.2). The diagram of the proposed segmentation process is shown in Figure 1. We conduct our experiments to quantify the performance of our system using a recent dataset proposed by Brox and Malik in[7] in section 7.

Our two main contributions are a strategy to select points inside moving objects and a method to identify only the regions corresponding to the moving objects. To achieve the latter objective, we have used the concept of assigning "object" side information to the boundary pixels which amounts to what is known as border-ownership in the neuroscience literature[9, 30].

2. Related Work

The problem of finding independently moving objects in a video, in the general case when the camera is itself moving, is a difficult problem. This is because the image motion is generated by the combined effects of camera motion, structure of the scene and the motion of independently moving objects. Isolating these three factors proves to be a difficult task.

Prior research can mostly be classified in the approaches relying, prior to 3D motion estimation, on 2D motion measurements only[4, 8, 18, 26], and the 3D approaches which identify clusters with consistent 3D motion [1, 17, 23, 21, 25, 28] using a variety of techniques. The limitations of the techniques of the first kind are well understood. 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. Some techniques, such as [25] are based on alternate models of image formation like weak perspective. These additional constraints can be justified for domains

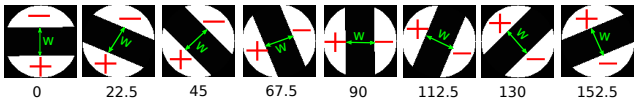


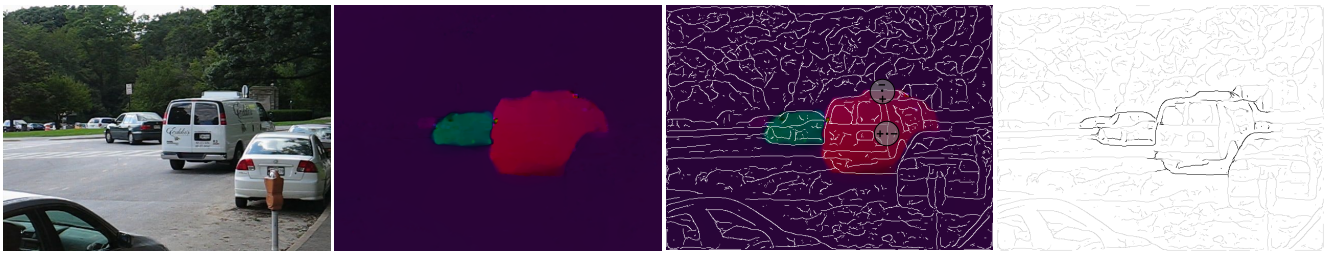
Figure 2. The oriented disc filters with opposite polarity. The corresponding orientation values (in degree) are shown at the bottom. The middle zone (width "w") in the filters are suppressed to tolerate the misalignment between the optical flow boundary, and the actual boundary of the moving objects. The disc radius is 0.015 times the image diagonal and "w" is 0.2 times the radius.

such as aerial imagery. In this case, the planarity of the scene allows a registration process [3, 24, 27, 29], and uncompensated regions correspond to independent movement. This idea has been extended to cope with general scenes by selecting models depending on the scene complexity [22], or by fitting multiple planes using the plane plus parallax constraint [13, 20].

Improvements can be obtained through integration of the motion measurements over time. In a recent breakthrough paper, Brox and Malik introduced a segmentation technique [7] doing exactly this with excellent results. They renewed interest in the problem that has been nascent for a while. This is because the 3D approaches that dominated this field in the past, have been challenged by the difficulty of estimating 3D motion and the inherent ambiguities associated with the problem [2, 5, 15, 12, 10]. The work of Brox and Malik mentioned above, with its high quality results, demonstrated that we can achieve a solution to this problem without a detailed 3D analysis. Along this line of thought, we propose here object segmentation in video using cue integration. We test our proposed technique on the dataset and improve on the results by Brox and Malik7.

3. Probabilistic Boundary Edge Map

In a probabilistic boundary edge map of an image, the intensity of a pixel is set to be the probability that it is on the boundary of a moving object in the scene. We can determine this probability by checking for a discontinuity in the optical flow map at the corresponding pixel location because the boundary of a moving object is also a motion boundary. But the exact location of discontinuities in optical flow maps often do not match with the true object boundaries, a well-known issue for optical flow algorithms. To account for this, we use static cues such as color and texture to first find all possible boundary locations in the image which are the edge pixels with positive color and/or texture gradient (See Figure 1(a)). The probability of these edge pixels to be on the motion boundary is then determined as a function of the optical flow gradient across them. Unlike[16], color and texture gradient values do not participate in determination of the boundary probability.



(a) (b) (c) (d)

Figure 3. (a) The first frame of the "cars3" sequence. (b) 2D optic flow map. (c) All the edge pixels with non-zero color and texture gradient overlaid on the flow map. (d) The final probabilistic boundary edge map.

3.1. Boundary Localization using Static Cues

Using color and texture cues, all locations in frame 1 with positive color and/or texture gradient are detected and stored in a binary edge map[14]. At all the edge pixels, the dominant tangential direction, which can be one of eight quantized values between 0 and π, is calculated as well.

The binary edge map, by selecting only a subset of all pixels, effectively assigns zero probability to the pixels from inside smooth areas in the scene to be on the boundary of a moving object. However, the boundary probability of the edge pixels can not be estimated using the color or texture gradient at their locations. An edge pixel with a high color or texture gradient value can be both inside and on the boundary of a moving object. We need motion cues to determine the boundary probability of the edge pixels in the binary edge map.

3.2. Boundary Probability Estimation using Motion Cues

We start by computing the optical flow using two consecutive frames [6]. See Figure 3(b) for an example. We also define eight disc filters for the eight possible orientations of the edge pixels in the binary edge map computed above (Figure 2 shows all the disc filters). By placing the appropriately oriented disc filter in the optical flow map at the location corresponding to an edge pixel, we compute the optical flow gradient as the magnitude of the difference in the mean optical flow vectors $||\vec{V_+} - \vec{V_-}||$, where $\vec{V_+}$ and $\vec{V_-}$ are the mean optical flow vectors in the two halves of opposite polarity indicated by "+" and "−" in Figure 2. Figure 3(c) shows the binary edge pixels overlaid on the flow map. While the gradient for an edge pixel inside a moving object is less than that for an edge pixel on the boundary, we still have to convert the gradient value into the boundary probability such that $0 \leq P_B(x, y) \leq 1$ where P_B is the final probabilistic boundary edge map. Figure 3(d) shows the inverted probabilistic boundary edge map of the image in Figure 3(a).

The relationship between the optical flow gradient at an edge pixel and its boundary probability is not linear because a large flow discontinuity implies a faster moving object but does not imply a proportionally high probability of the boundary pixels to be on the motion boundary. In fact, the relationship is close to being a step function as the optical flow gradient for all the pixels at the boundary of a moving object is usually higher than a threshold, and the opposite is true for the edge pixels from inside a moving object. Thus, we use a logistic function, $L : \mathbb{R} \mapsto [0, 1]$, to convert the optical flow gradient into the probability value, which is given by:

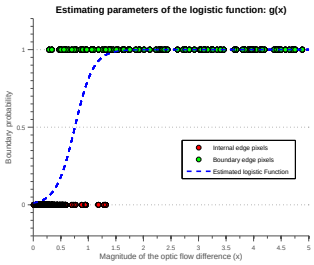


Figure 4. Learning the parameters of a logistic function converting the optical flow gradient into a probability.

$$L(g) = \frac{1}{1 + e^{-\beta_1(g-\beta_2)}} \quad (1)$$

where g is the optical flow gradient, and β_1 and β_2 are the parameters of the logistic function.

To determine β_1 and β_2, we take any consecutive pair of frames in a video sequence and calculate their optical flow map. In the first frame, we manually segment the moving objects. We then select 200 edge pixels at regular intervals on the boundary and inside the moving objects and calculate the optical flow gradients for these edge pixels. This will form the training set, where the gradient values for the boundary pixels should be mapped to 1 and those for the internal edge pixels to 0. Using logistic regression, we fit the logistic function to this data as shown in Figure 4. β_1 and β_2 are found to be 5.952 and 0.756 respectively. All our experiments were performed with these values.

4. Fixation Strategy

The goal of the fixation strategy is to select points inside objects so that the fixation-based segmentation method[16] takes those points along with the probabilistic boundary edge map as its input and returns the closed boundaries enclosing those points as its output. But, selecting the points inside objects even before segmenting them appears as a chicken and egg problem to which the solution in the logic used to generate the probabilistic boundary edge P_B map above.

To generate P_B , only the local processing of the optical flow vectors was done to assign the probability to an edge pixel to be the motion boundary. An edge pixel more likely to be on the motion boundary has a significant difference in the mean optical flow vectors on its opposite sides. We can identify the "object" side of such a boundary pixel as the side with larger mean optic flow. In the presence of static camera, it is easy to see that the logic that the "object" side has larger flow at the boundary holds. What happens when the camera is moving?

If we subtract the effect of the camera motion from the optical flow map, the moving camera case essentially becomes the static camera case. We estimate the ego-motion of the camera in terms of four parameters, (x and y translations, scale and rotation). Phase correlation on two consecutive frames in the Cartesian representation give the 2D translations (tx and ty) and in the log-polar representation gives scale and z-rotation[19]. Phase correlation can be thought of as a voting approach [11], and hence we find empirically that these four parameters depend primarily on the background motion even in the presence of moving objects. This assumption is true as long as the background edges dominate the edges on the moving objects. This four-parameter transform predicts a flow direction at every point in the image, which is subtracted from the computed optic flow map to remove the effect of camera motion.

From the probabilistic boundary edge map, we pick the edge pixels with the boundary probability greater than 0.5 and assume that they lie on the boundary of some object in the image. We represent this subset of boundary edge pixels by I_B given as:

$$I_B(x,y) = \begin{cases} 1 & \text{if } P_B(x,y) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

For the edge pixels in I_B, we identify the "object" side by comparing the mean optic flow vectors $\vec{V_+}$ and $\vec{V_-}$ on the opposite sides and store the information in a 2D matrix, O_B, defined as:

$$O_B(x,y) = \begin{cases} +1 & \text{if } I_B(x,y) \neq 0 \,\&\, ||\vec{V_+}|| > ||\vec{V_-}|| \\ -1 & \text{if } I_B(x,y) \neq 0 \,\&\, ||\vec{V_-}|| > ||\vec{V_+}|| \\ 0 & \text{if } I_B(x,y) = 0 \end{cases}$$

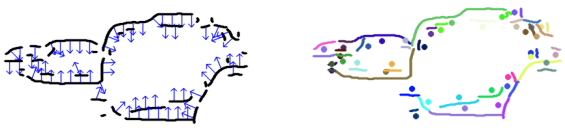


Figure 5. Left: (a) Boundary pixels (shown in black) with "object" side information (shown by blue arrows). Right: (b) Boundary fragments and corresponding fixation points selected on their "object" side.

See Figure 5(a) for an example of I_B and O_B for the first frame of the "cars3" sequence shown in Figure 3(a).

Using O_B, we could select the points inside objects by moving a fixed distance towards the "object" side in the normal direction at the boundary edge pixels in I_B. But this would give us as many points as the edge pixels causing redundancy because many fixations inside the same object will result in the same segmentation. To reduce the redundancy, we break the contours of I_B into fragments interrupted by either an end point or a junction. Now, instead of an edge pixel, we select a point for each edge fragment by moving towards the "object side" in the normal direction of the edge pixel in the middle of the fragment. The point is selected at a fixed distance of 20 px from the edge pixel. For instance, see Figure 5(b) for the edge fragments and the corresponding points displayed in same color.

5. Fixation Based Segmentation

For each selected point, the fixation-based segmentation approach [16] finds the closed boundary around the given point by combining the edge pixels in the probabilistic boundary edge. The segmentation for each point has two intermediate steps: first, the probabilistic boundary edge map P_B is converted from the Cartesian to the Polar space using the given point as the pole for the conversion, in order to achieve scale invariance. Following this, a binary segmentation of the polar edge map generates the optimal path through the polar edge map such that when the curve is mapped back to the original Cartesian space, it encloses the point. The two-step segmentation process is repeated for all fixation points found in section 4 using the same P_B. The source code for this segmentation algorithm was obtained from the author's website.

6. Selecting Regions Corresponding to Moving Objects

We have as many closed contours as the number points selected by the fixation strategy. As one can see in Figure 5, since the selection of points depends on the contour fragments in I_B, the multiple fragments which are part of the same closed boundary generate the points that lie inside the

same object; these points lead to duplications of the closed contours. Also, sometimes due to error in "object" side estimation for an edge fragment, the corresponding point lies outside of any object in the image which will lead to a closed contour that does not correspond to an object. See Figure 1(f) for an example of both types of contours. So, we need a process that sifts through all the closed contours to pick only the ones that uniquely correspond to the moving objects in the scene. This means that we require a method to differentiate between any two closed contours which will be described in section 6.1. Following this, we describe our minimum set cover formulation to select the subset of closed contours that correspond to the moving objects in section 6.2.

Notation: I_C^i is the binary mask representing the i^{th} closed contour whose interior is represented by another binary mask I_R^i. If I is a 2D matrix with binary entries, X_I is the set of 2D coordinates of the non-zero pixels in I.

6.1. Coverage of a Closed Contour

We define the coverage of a closed contour so that a high coverage is more likely to correspond to the boundary of a moving object. A closed boundary of a moving object is composed of the edge pixels in I_B such that the "object" side of these edge pixels is the interior of the contour (see Figure 6(a) for an example). The opposite of this is the closed boundary of a hole wherein the object side of all the boundary edge pixels lie outside of the hole (see Figure.6(b) for an example of a hole). So, the coverage of a closed contour is defined as:

$$
\begin{aligned}
Coverage(I_C, I_R) &= \frac{1}{|X_{I_C}|} \sum_{x \in X_{I_C}} I_B(x)\Pi(x) \qquad (2)\\
\Pi(x) &= \begin{cases} +1 & \text{if } I_R(x + \lambda O_B(x) u_\perp) \neq 0 \\ -1 & \text{otherwise} \end{cases} \\
u_\perp &= \begin{bmatrix} cos(\theta - \frac{\pi}{2}) \\ sin(\theta - \frac{\pi}{2}) \end{bmatrix}
\end{aligned}
$$

where x is a 2D coordinate, θ is the orientation of the tangential direction at the edge pixel x and λ is the distance from the selected point on the "object" side to the edge pixel. In our experiment, we keep λ to be 5 px.

6.2. Selecting Closed Boundaries of Moving Objects

With the definition of the coverage of a closed contour given above, simply looking at the sign of the coverage value differentiates between the boundary of a moving object and to a non-object. A closed contour outside of any moving object traces the edge pixels in I_B with "object" side lying outside of the contour and thus has a coverage that is negative or close to zero if positive. To handle the duplicate closed contours of the same moving object, we can pick the one resulting in the highest coverage with the object boundary. We formulate the process of selecting the closed boundaries of moving objects from the set of all closed contours as a type of a min-cover problem, whose standard definition is:

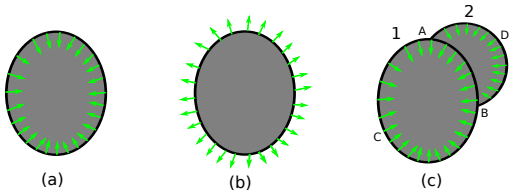


Figure 6. (a) and (b) are the closed contours likely to a moving object and a hole respectively. (b) A closed contour hole. (c) Two occluding closed contours. Note that the arrows indicate the "object" side of the boundary pixels.

Given an Universal set U and a set of subsets S, find $S' \subseteq S$ such that S' contains all the elements of U and $|S'|$ is minimum.

In this case, X_{I_B} is the universal set U. $\{X_{I_C^i}\}_{i=1}^n$ is the set of subsets since $X_{I_C^i}$, the pixels along each closed contour, contains a subset of pixels in X_{I_B}. n is the total number of closed contours. Our objective is to find the minimum number of closed contours that together trace almost all the pixels in I_B. Since the minimum set cover problem is NP-Complete, we propose a greedy solution. The pseudocode is given in Algorithm 1. The selected closed contours at the end of the process are the boundaries of the different moving objects in the scene.

The greedy solution works iteratively. It starts with computing coverage of all closed contours and then selects the best contour in each iteration. At the end of the iteration, the universal set is updated by eliminating all the pixels traced by the current best contour. After updating, the coverage of the remaining contours are recomputed for the next iteration. The selection process repeats until the "best" closed contour in the current iteration has a coverage below a certain threshold.

An important step in the proposed greedy solution is the the update of the remaining closed contours at the end of each iteration. To handle the case of occluding contours, when one of them is selected as the best contour in an iteration, the remaining closed contours are updated such that the pixels on the shared boundary do not affect their coverage as those pixels have already been traced by the current best contour.

Consider, for instance, the two occluding closed contours $ABCA$ and $ADBA$ in Figure 6(c). They share the pixels along the segment AB and the "object" side indicates it will contribute positively to the coverage of $ABCA$ and negatively to that of contour $ADBA$. After the selection of $ABCA$, in the next iteration, we would like to make $ADBA$ prominent because the boundary portion AB that

Algorithm 1 Selecting "good" regions, $t_{coverage} = 0.3$, $t_o = 0.05$

Input:
I_B ▷ edge pixels predicted to be on object boundaries
O_B ▷ object side information
$S_{in} = \{I_R^i, I_C^i\}_{i=1}^n$ ▷ closed contours for all n fixations
Output:
$S_{out} = \{I_R^j, I_C^j\}_{j=1}^m$ ▷ final closed contours
Intermediate variables:
I_T^k ▷ all closed contours traced until iteration (k-1)
I_M^k ▷ accumulated region mask until iteration (k-1)

begin
Initialize $k \leftarrow 0$; $I_B^0 \leftarrow 0$; $I_A^0 \leftarrow 0$
while $|S_{in}| > 0$ **do**
  Compute coverage of all closed contours $\in S_{in}$;
  Let b be the index of the closed contour with maximum coverage;
  if $Coverage(I_R^b, I_C^b) < t_{coverage}$ **then**
    Exit the loop;
  else
    Move the closed contour from S_{in} to S_{out};
    $I_A^k \leftarrow I_A^k + I_R^b$;
    $X_{I_B} \leftarrow X_{I_B} - (X_{I_B} \cap X_{I_C^b})$;
    $X_{I_T^k} \leftarrow (X_{I_T^k} \cup X_{I_C^b})$;
    for $I_C^i \in S_{in}$ **do**
      overlap = $|X_{I_R^i} \cap X_{I_A}|/|X_{I_R^i}|$;
      if overlap $< t_o$ **then**
        $X_{I_C^i} \leftarrow X_{I_C^i} - (X_{I_C^i} \cap X_{I_T^k})$ ▷ to handle occlusion
      end if
    end for
    Increment k;
  end if
end while

was contributing negatively has already been traced by region $ABCA$. So, the new coverage of $ADBA$ will include the contribution from the remaining pixels in the segment ADB. But, this could have unintended consequence for overlapping duplicate contours where once the best contour for the moving object is selected, all the duplicate contours with a slight change will become important just as the occluding contour case. To avoid this situation, we check for the overlap of the inside of the remaining closed contours with that of already selected closed contours. Only the closed contours with no overlap are updated.

7. Experiments

7.1. Data and Evaluation Criteria

For quantitative analysis, we have selected the dataset created in [7] containing 26 video sequences with ground truth segmentation of the moving objects. Unlike the feature-based sparse segmentation algorithms that track features over a large number of frames for better results, our method requires only two consecutive frames to segment the moving objects in them. So, we use the first 10 frames of each sequence to do our analysis. We also split the dataset into two groups: ("cars1" to "cars10" sequences) containing rigid objects and the rest containing non-rigid objects. Although we compare our dense segmentation results with the sparse segmentation of Brox and Malik using the pixel-wise metrics defined in [7], we define the our evaluation metrics that are more appropriate for a dense edge based segmentation like ours. These metrics evaluate both the accuracy and stability of the proposed technique.

We compute four different accuracy metrics for each moving object and report the average score over all the objects in the dataset. Using ground truth segmentation G of each object, the segmented closed contours lying inside the ground-truth mask are identified and the pixels inside these closed contours are combined to form a segmented object mask M. The number of the identified contours is the **over-segmentation** error and the overlap between the ground-truth and the segmented object mask $\frac{|G \cap M|}{|G|}$ is the **overlap score** for that object. The fraction of boundary pixels of M within 5 px distance of any pixel along the true object boundary is the **boundary accuracy** of the object. Finally, the **detection rate** is the fraction of all objects detected where an object is considered detected if its overlap score is more than 0.8.

Stability of the system is evaluated by repeating the segmentation for the next pairs of consecutive frames. After all, the system is robust only if the output is consistent. We compute the accuracy metrics using ground-truth that we generated for all 10 frames of the sequence. The variation in the accuracy metrics is the measure of the stability of the proposed method.

	over-segmentation	boundary accuracy (%)	detection rate (%)	overlap score (%)
Rigid	1.3 ± 0.2	88.61 ± 0.96	70.67 ± 5.12	88.52 ± 4.23
Non-rigid	1.57 ± 0.4	83.30 ± 7.43	52.94 ± 7.12	89.11 ± 3.61

Table 1. Segmentation accuracy results of the proposed method.

7.2. Results and Analysis

We show the segmentation results for a number of sequences from the video dataset[7]. A quick qualitative analysis reveals that the proposed method handles occlusion, results in good boundary accuracy and is scale-invariant. For quantitative analysis, we first compare with [7]. We create dense tracks of the pixels inside each closed contour found using just two consecutive frames and assign them an unique label. We also generate tracks for the background pixels. Table 2 contains the values of the pixel-wise evaluation metrics defined in [7]. Although we use only two consecutive frames compared with 10 frames used in [7], our algorithm performs better. Also, we provide dense segmentation compared with sparse segmentation of [7]. Due to that very difference, we evaluate our segmentation using the metrics defined in the previous section. The results are shown in Table. 1.

	Density	overall error	average error	over-segmentation
our segmentation	100%	3.71%	16.55%	2.25
Brox&Malik(**author?**) [7]	3.34%	7.75%	25.01%	0.54

Table 2. Evaluation Results using the metrics defined in **(author?)** [7].

The proposed segmentation process mostly finds the moving objects as one region as indicated by the over-segmentation error in Table 1, which is 1.3 and 1.57 for rigid and non-rigid objects respectively. The low variation in the values show the stability of the method in consistently producing the same decomposition across time. Note that 80% of the boundary of the detected moving object is traced faithfully and almost 90% of the object interior is covered by the segmented regions. While the detected moving objects are segmented well by our method, the detection rates are 70.67% and 52.94% of the rigid and non-rigid objects. This means the system is not segmenting 3 out of 10 rigid objects. The two likely causes are: first, no point was selected inside the undetected moving objects and hence no segmentation for those objects; second, a point was selected inside the undetected moving objects but the corresponding contours could not make it through the selection process.

We found that, on average, 8 points were selected inside each moving object and at least one point was selected inside all the moving objects. But we also found that only 75% of all selected points landed inside the object masks. The reason for this is also the reason for why the closed contours corresponding to some undetected objects are not extracted even after having a point selected inside them. And that reason is the wrong "object" side information for the boundary pixels of those objects.

Consider, for instance, the first frame of the "cars10" sequence (see Figure 7: Last Row, Column 1) for which the segmentation results are shown in Figure 7: Last row, Column 3. Although the "object" side estimation for most of the boundary pixels of the bus behind the car are wrong, some points were selected inside the bus due to the strong internal edges (see Figure 8). The closed contours corresponding to the points on the bus will have a negative coverage, and will be rejected as a possible hole in the selection process described in section 6. We found that 87.12% of all predicted boundary pixels in I_B lying on the true object boundaries have the correctly estimated "object" side information. The estimation accuracy can be improved with a better 3D ego-motion estimation.

The boundary accuracy of the detected object depends primarily on the ability of motion cues to assign high probability to the actual motion boundary. But sometimes due to error in the optical flow a significant portion of the object boundary is not identified, resulting in lower accuracy. For instance, in Figure 9, the actual boundary edge on top of the hat of the lady is not assigned an appropriate probability causing [16] to deviate from the desired closed contour.

Figure 7. Segmented Moving Objects. Left column: the first frame of the video sequence. Middle column: the predicted boundary pixels of the moving objects, I_B. Right column: The final set of closed contours corresponding to the moving objects in the scene. Row 1 and Row 2 show that the proposed approach can handle occluding objects and demonstrate the scale invariance as the size of segmented objects are very different. The last row contains an example of the effect of erroneous visual cues on segmentation.

Finally, due to noise in the edge and the optical flow maps, the final set of contours for each sequence has, on average, $2.8 \approx 3$ closed contours that do not belong to any object in the scene. However, these spurious closed contours can be identified easily as they do not correspond across

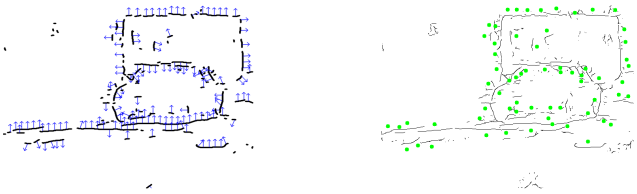


Figure 8. Left: (a) The "object" side information. Right: (b) the points selected by the fixation strategy.

Figure 9. From left to right: (a) Frame 344 in "marple4" sequence. (b) The optic flow map (c) the predicted boundary pixels. (d) Final segmentation results.

time.

8. Conclusion

We have presented a motion segmentation process that, instead of grouping pixels into regions, groups edge pixels into the closed boundaries of the moving objects in the scene. These contours can be used for unsupervised learning of objects from videos. The reported improvements over the state-of-the-art is primarily due to the integration of static cues into motion processing. Also, the modular nature of the proposed process makes it possible to translate any improvement in edge detection or optical flow estimation directly into better segmentation results. The actual segmentation step is separated from the cues that are used only to create the boundary edge map. The separation makes it a generic segmentation process that can be run depending on the available visual cues.

References

[1] G. Adiv. Determining 3d motion and structure from optical flow generated by several moving objects. *T-PAMI*, 7:384–401, 1985. 2

[2] G. Adiv. Inherent ambiguities in recovering 3-d motion and structure from a noisy flow field. *T-PAMI*, 11(5):477–489, 1989. 2

[3] S. Ayer, P. Schroeter, and J. Bigün. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *ECCV*, pages 316–327, London, UK, 1994. Springer-Verlag. 2

[4] M. Bober and J. Kittler. Robust motion analysis. In *CVPR*, pages II: 947–952, 1994. 2

[5] T. Brodsky, C. Fermuller, and Y. Aloimonos. Structure from motion: beyond the epipolar constraint. *IJCV*, 37:231–258, 2000. 2

[6] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. pages 25–36. Springer, 2004. 3

[7] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 2, 6, 7

[8] P. Burt, R. Bergen, J.R.and Hingorani, R. Kolczynski, W. Lee, A. Leung, J. Lubin, and H. Shvayster. Object tracking with a moving camera. In *Proc. IEEE Workshop on Visual Motion*, pages II: 2–12, 1989. 2

[9] E. Craft, H. Schütze, E. Niebur, and R. von der Heydt. A neural model of figure-ground organization. *Journal of Neurophysiology*, 6(97):4310–4326, 2007. 2

[10] C. Fermuller and Y. Aloimonos. Observability of 3d motion. *IJCV*, 37:231–258, 2000. 2

[11] D. J. Fleet. Disparity from local weighted phase-correlation. In *IEEE International Conference on Systems, Man, and Cybernetics*, 1994. 4

[12] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid moton i: Algorithm and implementation. *IJCV*, 7:95–117, 1992. 2

[13] M. Irani and P. Anandan. A unified approach to moving object detection in 2d and 3d scenes. *T-PAMI*, 20(6):577–589, 1998. 2

[14] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *T-PAMI*, 26(5):530–549, May 2004. 3

[15] S. Maybank. *A theoretical study of optical flow*. PhD thesis, University of London, 1987. 2

[16] A. Mishra, Y. Aloimonos, and L. F. Cheong. Active segmentation with fixation. In *ICCV*, 2009. 1, 2, 4, 7

[17] R. Nelson. Qualitative detection of motion by a moving observer. *IJCV*, 7:33–46, 1991. 2

[18] J.-M. Odobez and P. Bouthemy. Mrf-based motion segmentation exploiting a 2d motion model robust estimation. In *ICIP*, page 3628, Washington, DC, USA, 1995. IEEE Computer Society. 2

[19] B. S. Reddy and B. N. Chatterji. An fft-based technique for translation, rotation and scale-invariant image registration. *IEEE Trans. Image Processing*, 5:1266–1271, 1996. 4

[20] H. S. Sawhney, Y. Guo, and R. Kumar. Independent motion detection in 3d scenes. *T-PAMI*, 22(10):1191–1199, 2000. 2

[21] W. Thompson and T. Pong. Detecting moving objects. *IJCV*, 4:39–57, 1990. 2

[22] 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):1321–1340, 1998. 2

[23] P. H. S. Torr and D. W. Murray. Stochastic motion clustering. In *ECCV*, pages 328–337, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc. 2

[24] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372, London, UK, 2000. Springer-Verlag. 2

[25] J. Weber and J. Malik. Rigid body segmentation and shape description from dense optical flow under weak perspective. *T-PAMI*, 19(2):139–143, 1997. 2

[26] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *CVPR*, page 520, Washington, DC, USA, 1997. IEEE Computer Society. 2

[27] C. S. Wiles and M. Brady. Closing the loop on multiple motions. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 308, Washington, DC, USA, 1995. IEEE Computer Society. 2

[28] Z. Zhang, O. Faugeras, and N. Ayache. Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints. In *ICCV*, pages 177–186, 1988. 2

[29] Q. Zheng and R. Chellapa. Motion detection in image sequences acquired from a moving platform. In *ICASSP*, pages 201–204, 1993. 2

[30] H. Zhou, H. Friedman, and R. von der Heydt. Coding of border ownership in monkey visual cortex. *The Journal of Neuroscience*, 20:6594–6611, September, 2000. 2