

# Video Object Boundary Reconstruction by 2-Pass Voting

Like Zhang<sup>1</sup>, Qi Tian<sup>1</sup>, Nicu Sebe<sup>2</sup>, Jingsheng Ma<sup>3</sup>

<sup>1</sup>Department of Computer Science, University of Texas at San Antonio, TX 78249

*{lzhang, qitian}@cs.utsa.edu*

<sup>2</sup> Faculty of Science, University of Amsterdam, The Netherlands, nicu@science.uva.nl

<sup>3</sup> Institute of Petroleum Engineering, Heriot-Watt University, Edinburgh, UK, Jingsheng.ma@pet.hw.ac.uk

## Abstract

In this paper we propose a voting-based object boundary reconstruction approach. Tensor voting has been studied by many people recently, and it can be used for boundary estimation on curves or irregular trajectories. However, the complexity of saliency tensor creation limits its applications in real-time systems. In order to have an efficient solution, we introduce an alternative voting approach. Rather than creating saliency tensors, we use a “2-pass” method for orientation estimation. For the first pass, Sobel detector is applied on a coarse boundary image to get the gradient map, then the orientation information is updated by accumulating votes on the corresponding direction. In the second pass, edge linking is performed based on the pixels orientation map, and extra lines are eliminated by detecting intersections. The approach has been applied to various video clips under different conditions, and the experimental results demonstrate significant improvement on the final extracted objects accuracy.

## 1. Introduction

Object boundary reconstruction is an important step in the post processing of video object extraction, which is the fundamental element for intelligent surveillance systems, video content retrieval, object tracking and classification. Previous approaches for object extraction are mostly based on the statistical models of pixel color or illumination changes, as in [1][2]. Recently, methods based on object shape have attracted people’s attention for more accurate results and video content understanding. The silhouette-based method proposed in [3] tried to detect people carrying objects by shape changes. In [4] and [5], edge information was used instead of pixel color to extract moving objects, and the extracted boundary information was used for object tracking in [4].

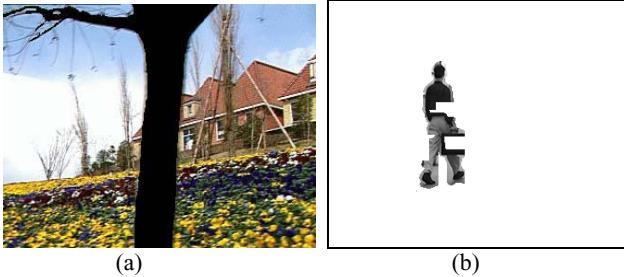
Tracking or identifying objects based on their shapes requires accurate boundary information. However, due to the extreme environment uncertainty, such as illumination changes or shadow effects, the extracted boundaries usually contain a lot of gaps or missing areas. To solve this problem, people use edge-linking techniques in the post processing step. Many approaches have been developed for edge linking. Canny edge detector was widely used to extract object boundary. Because the Canny detector has simple

edge-linking ability, the discontinuity of extracted boundaries could be reduced. Morphological processing was adopted in [4] to close the gaps on boundaries, but the unknown size of the gaps made it difficult to select an optimized threshold. Other approaches include using graph-based edge linking, shortest path algorithm, or stick algorithm as in [5][6]. However, these approaches could not bring satisfying results due to the difficulty to estimate the correct boundary trajectory without any pre-known knowledge, especially when there are curves or irregular shapes in the missing areas.

Methods based on non-linear voting algorithms attracted people’s attention in the recent years. Tensor voting was introduced for structure inference from sparse data by Guy and Medioni in [7]. The strength of the method is its ability to detect discontinuity and gaps by collecting local neighborhood orientation information as well as saliency measurement. In their approach, a non-iterative technique called “vector voting” was introduced to estimate orientation in neighbor pixels. This approach has been applied to many areas as in [8][9][10]. However, due to the complexity of 2D tensor estimation, the original algorithm is usually modified or simplified in real applications [11][12].

In [8], tensor voting was applied for repairing corrupted images with empty holes. Missing curve elements were first interpolated by gathering tensor support from the neighborhood. Then, using and adaptive ND tensor, a voting process infers the optimal values for defective pixels. Dealing with images with empty space is also a common problem when extracting video object boundaries. The difference between image repairing and object extraction is the size of the missing areas, as can be seen in Fig. 1. In image repairing, the missing area size is relatively big and there is very little neighbor pixel information for estimation. On the other hand, for video object extraction, the missing areas are much smaller, and there are enough neighbor pixels for orientation estimation. The complicated 2D tensor voting proves useful in image repairing because it could construct accurate saliency model with few pixels around the missed point, but it is only redundant for real-time applications such as video surveillance system and highway monitoring.

In the following sections, we are going to introduce an efficient approach based on tensor voting for object boundary reconstruction.

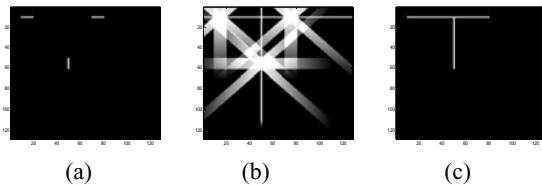


**Figure 1.** Image repairing and object reconstruction. (a) An image containing a black hole (the missed tree). (b) An extracted object from a video sequence

## 2. Proposed Approach

The original tensor voting contains two steps. The first step is saliency tensor creation, which encodes the saliency of features such as points, curves and surface path elements in saliency tensor. The created tensor can be classified into a stick tensor, which describes accurate orientation estimation, a plate tensor, which describes orientation uncertainty except in one direction, and a ball tensor, which describes total orientation uncertainty [6]. In the second step, tensors communicate with each other to infer the most possible trajectory.

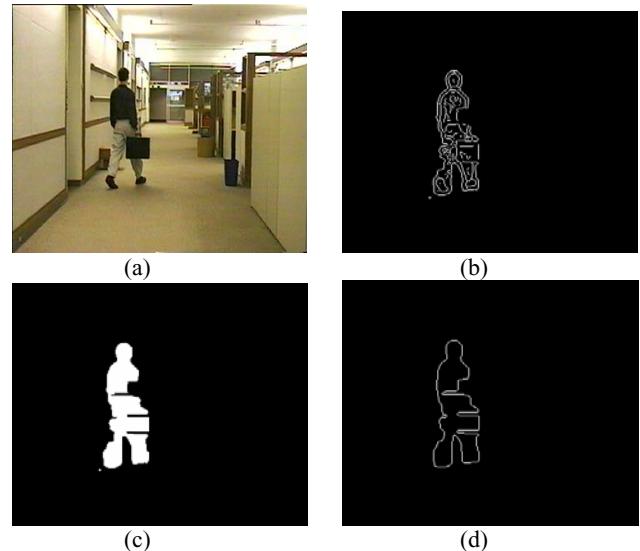
Here we propose a more efficient approach. First, the complicated saliency tensor is removed and a “2-pass” method is brought in for orientation estimation. For the first pass, Sobel detector was applied on the coarse boundary image to get the gradient map. In the second pass, decreasing weights based on the corresponding gradient information are set by each pixel, and the direction with maximum weights sum is selected as the correct orientation of the pixel. After the orientation map is obtained, the pixels start linking the edges or intersections along their direction. If no edges or intersections are found on the corresponding orientation, the pixel is identified as “not linked” and all weight values along its direction will be put to zero. Eventually, only linked edges will be left on the voting map. Fig. 2 is a simple example illustrating how the algorithm works.



**Figure 2.** Voting based edge linking approach. (a) Original picture with three isolated lines. (b) The result of assigning weights to the 8-direction map of each pixel (Sobel detector was not applied in this sample). (c) The voting result by selecting the direction with highest weight sum.

Comparing with the original tensor voting algorithm, this approach simplifies the step of orientation estimation. The tensor voting approach needs to create three different tensors (ball, plate, stick) for pixels under various conditions based on the eigenvector calculation, then collect neighborhood information according to the tensor type. While in our algorithm, tensor creation is replaced by a simple Sobel detector and a second-pass orientation map updating.

The initial object boundary is obtained by comparing color difference on edge points of current video frame and the initial background frame, which is similar to [4][5]. Due to the color similarity between the moving object and the background, there are some missed edge points (e.g., the person’s right hand and the case, as in Fig.3 (b)). For object reconstruction, only the outer boundary is necessary, and we apply the object filling technique from [4] to get the video object plane (VOP) (see Fig.3(c)). Finally, the edge detector is applied again on the VOP to get the object outer boundary image (Fig.3(d)).

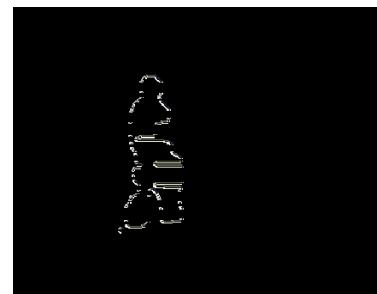


**Figure 3.** Original object boundary from frame 46 in “hall” video sequence. (a) The original frame; (b) The extracted boundary by Canny edge detector; (c) The result after applying object-filling technique; (d) The extracted object out boundary.

The following steps illustrate how we reconstruct the object boundary by the proposed approach. Suppose the boundary image is  $E$ .

**Step 1:** Let  $S_x$  and  $S_y$  be the Sobel detectors for horizontal and vertical direction of the image, and  $G$  be the gradient map which contains the gradient information of all pixels (Fig. 4)). In our implementation, the orientation is quantized to 8 directions.

$$G = \arctan((E \otimes S_y) ./ (E \otimes S_x))$$



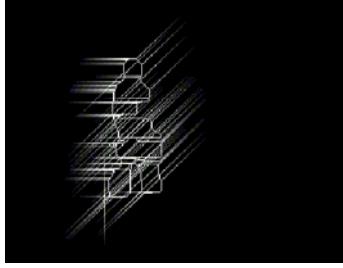
**Figure 4.** Gradient map after applying Sobel detector

**Step 2:** Suppose we have a voting length  $L$ . The largest weight, which corresponds to an existing edge point, is 255 (the largest gray level value). The weight decreases along the pixel's orientation by  $255/L$ . Then we have a tensor map  $T$  (Fig. 5):

```

if E(x, y) > 0
  T(x, y) = 255 ;
  for i = 1 to L
    T(x + i, y + i * tan( G(x, y))) += ((L - i + 1) * 255 / L);
  end
end

```



**Figure 5.** Created 1-pass tensor map. Here we use the name “tensor”, but it is totally different from the original tensor definition.

**Step 3:** Choose the direction with maximum weights as the correct direction of a pixel, and update the voting map.  $S$  stands for the sum of weights along directions from  $1^\circ$  to  $360^\circ$ ,  $V$  is the voting map.

```

for theta = 1 : 360
  S(theta) += sum(T(x, y + tan(theta)));
end
(max_value, index) = max(S);
V(x, y) = index;

```

**Step 4:** Recreate the tensor map  $T$  based on the voting map  $V$ . First we reset  $T$  to all zeros. Then we go through a process similar to step 2. However, this time, we put the same weights along the direction in voting map rather than putting decreasing weights along gradient orientation.

```

theta = V(x, y);
for i = 1 : L
  T(x + i, y + i * tan(theta)) += 255 / L;
end

```

**Step 5:** Find intersections along pixel's direction (Fig. 6). The intersection usually consists of overlapped tensor points which have higher weight value. An intersection is defined as a point with weight larger or equal to a threshold, which is usually 255, the existing pixel value.

```

theta = V(x, y);
for i = 1 : L
  if (T(x + i, y + i * tan(theta)) >= threshold)
    T(x + 1 : x + i; y + tan(theta) : y + i * tan(theta)) = threshold ;
  end
end

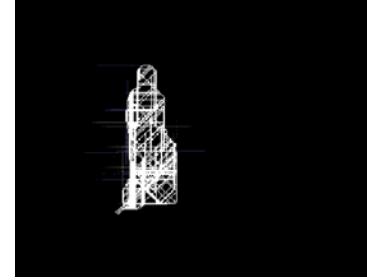
```

**Step 6:** Original edge points have been linked with the tensor intersection points together. Now we need to remove the unnecessary tensor pixels which have no intersection. This is simply done by removing all pixels with weight smaller than the threshold. The threshold is defined as 255 (the highest pixel value) in our experiment.

```

if (T(x, y) < threshold)
  T(x, y) = 0
end

```



**Figure 6.** Voting result after step 5 (the result of 2-pass)

**Step 7:** Use the object-filling technique again and map the VOP to the original frame (Fig. 7).



**Figure 7.** Final Result

### 3. Experimental Result

We tested the proposed approach on two different video clips in various conditions. The performance is evaluated by false negative (FN), false positive (FP), and matching error. The FN stands for the number of pixels which exist in the object but not in the final result. The FP stands for the number of extracted pixels not belonging to the ground-truth object. The matching error is used in [4] to evaluate the accuracy of extracted object. However, the original matching error is the ratio between number of mismatching pixels and the image size. This could vary a lot according to the object size, and cannot be taken as a stable evaluation measure. Instead of comparing with the frame size, we redefine the matching error as the ratio between the number of mismatching pixels and the object size, which is:

$$ME = \frac{\sum_{i=1}^h \sum_{j=1}^w |VO(i, j) - Mask(i, j)|}{\sum_{i=1}^h \sum_{j=1}^w Mask(i, j)}$$



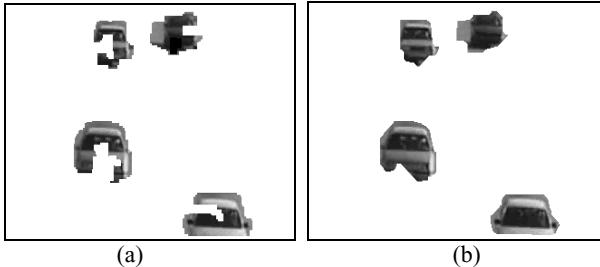
**Figure 8.** Comparison of results from frame 46 in the “hall” video sequence. (a) The extracted object before the voting process; (b) The result after using the proposed reconstruction approach.

**Table 1.** Results obtained for the “hall” video sequence.

	FN	FP	Matching Error
Original	716	1114	0.3007
After reconstruction	237	1023	0.4368

Fig. 8 compares the results before the voting process and after the reconstruction. In the indoor environment, the weak illumination could bring trouble for edge detection. After applying the proposed reconstruction approach, the improvement is significant (see also Table 1). With our method all the missing areas have been filled in.

Next we tested the algorithm on multiple objects. The testing video clip is from a highway monitoring sequence which contains fast moving vehicles. The empty holes are arbitrarily made to test the ability of our approach with relatively big missing areas.



**Figure 9.** Highway monitoring with shadow removal. (a) The original extraction result with arbitrarily made empty holes. (b) The result after applying the proposed reconstruction method.

**Table 2.** Results obtained for the “highway” sequence

	FN	FP	Matching Error
Original	3106	1673	0.5174
After reconstruction	2026	1155	0.3444

Fig. 9 proves the voting-based algorithm works well even if there are multiple moving objects. Traditional edge-linking algorithms usually only work for single object

situation and do not performance very well in congested scenes. Without any change for the threshold, our approach works well for the multi-object situations.

#### 4. Conclusion

We proposed an efficient voting-based approach for object boundary reconstruction in this paper. Different from the previous tensor voting method, an efficient 2-pass orientation estimation solution is brought out to replace the complicated saliency tensor creation. The experimental results demonstrated that the method has significant improvement on the extraction result significantly. Besides, the only parameter that needs to be pre-selected is the voting length, and no other human interaction is needed, which helps the algorithm able to be applied for applications in different situations. This is a great benefit for intelligent surveillance systems.

#### Reference:

1. C. Stauff and W. Grimson, “Adaptive background mixture models for real-time tracking”, CVPR, pp. 246-552, 1999
2. N. Ohta, “A statistical approach to background subtraction for surveillance systems”, ICCV, vol.2, pp. 481-486, 2001
3. I. Haritaoglu, D. Harwood, and L. S. Davis, “W4: Real-time surveillance of people and their activities”, IEEE Trans. PAMI, vol. 25(8), pp. 809-830, 2000.
4. C. Kim and J. Hwang, “Fast and automatic video object segmentation and tracking for content-based applications”, IEEE Trans. on Circuits and Systems for Video Technology, vol. 12(2), pp. 122-129, 2002
5. E. P. Ong, B. J. Tye, W. S. Lin, M. Etoh, “An efficient video object segmentation scheme”, ICASSP, pp. 3361-3364, 2002
6. R. N. Czerwinski, D. L. Jones, and W. D. O’Brien, “Detection of lines and boundaries in speckle images – Application to medical ultrasound”, IEEE Trans. Med. Image, vol. 18, pp. 126-136, Feb. 1999
7. M.S. Lee and G. Medioni, “Grouping ., -, ->, O-, into Regions, Curves, and Junctions”, CVIU, vol. 76, no. 1, pp. 54-69, 1999
8. J. Jia, C. Tang, “Image repairing: Robust image synthesis by adaptive ND tensor voting”, CVPR, vol. 1, pp.18-20 June 2003
9. M. Niculescu and G. Medioni, “Motion segmentation with accurate boundaries – A tensor voting approach”, CVPR, vol. 1, pp. 382-389, 2003
10. J. Jia and C. Tang, “Inference of segmented color and texture description by tensor voting”, IEEE Trans. PAMI, vol. 26(6), 2004
11. J. Jia, T. Wu, Y. Tai, and C. Tang, “Video repairing: Inference of foreground and background under severe occlusion”, CVPR, vol. 1, pp. 364-371, 2004
12. J. Jia and C. Tang, “Image registration with global and local luminance alignment”, ICCV, pp. 156-163, 2003