International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:11, 2009

Probabilistic Center Voting Method for Subsequent Object Tracking and Segmentation

Suryanto, Hyo-Kak Kim, Sang-Hee Park, Dae-Hwan Kim, and Sung-Jea Ko, *Senior Member, IEEE*

Abstract—In this paper, we introduce a novel algorithm for object tracking in video sequence. In order to represent the object to be tracked, we propose a spatial color histogram model which encodes both the color distribution and spatial information. The object tracking from frame to frame is accomplished via center voting and back projection method. The center voting method has every pixel in the new frame to cast a vote on whereabouts the object center is. The back projection method segments the object from the background. The segmented foreground provides information on object size and orientation, omitting the need to estimate them separately. We do not put any assumption on camera motion; the proposed algorithm works equally well for object tracking in both static and moving camera videos.

Keywords—center voting, back projection, object tracking, size adaptation, non-stationary camera tracking.

I. Introduction

THE tracking of moving objects from frame to frame in a real time video surveillance is a highly challenging task. This is true especially in Pan-Tilt-Zoom (PTZ) camera surveillance system where the camera has to constantly pan, tilt, and occasionally zoom in and out in order to keep the object under surveillance inside the camera's range of view. In such system, the background scene is constantly changing as the camera moves, thus the widely used background subtraction technique cannot be employed to help locating the object. Additional challenges come from complex object motion, non-rigid object tracking, partial occlusion, illumination change, and real time processing requirement. Despite all these difficulties, many successful algorithms have been reported over past decades. For a comprehensive review of various tracking algorithms, the readers can refer to [1].

Two important aspects that determine the performance of the tracking algorithms are target representation and target localization. Target representation refers to how the object to be tracked is modeled and target localization refers to how the search of the corresponding object in the following frame is accomplished. Popular models used for target representation are object contour [2], [3], feature point [4], [5], [6], [7], and color histogram [8], [9], [10], [11]. Depend on the chosen target representation model, various target localization techniques can be employed.

Tracking with object contour works well even when tracking non-rigid object which shares similar color information with the background. The Condensation algorithm proposed in [2] parameterizes the contour using B-Spline control points. The target localization is performed by comparing the contour model at the previous frame with the local edge map at the following frame. The edges which are parallel and proximately close to the contour model are considered as good candidates for the new object contour. Even though the Condensation algorithm demonstrates impressive tracking performance, it is ill suited for real time tracking application due to its high computation complexity.

Another algorithm for object contour tracking was proposed in [3]. In their algorithm, the object contour is represented by the two linked lists and a level set array. Contour adaptation is realized by performing switching on elements of the linked list. At each frame, the elements of linked list are adjusted to fit the object contour.

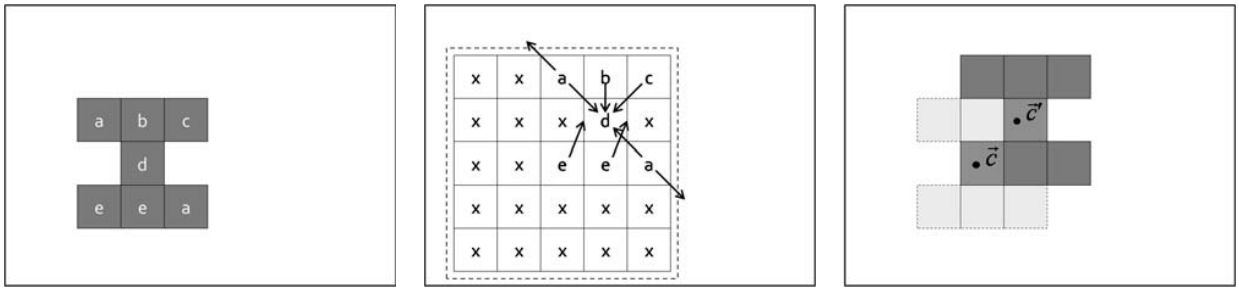
Tracking using feature points produces good results when the object has rich texture. In [4], a point is considered as a good feature if it is unique when compared to its local neighborhood. To find the corresponding point in the next frame, the iterative Newton Raphson minimization algorithm is employed [5], [12]. Tracking with feature points is fast and reliable. However, when the object turn around or partially occluded, the tracking algorithm fails miserably.

Using the color histogram for target representation has been increasingly popular due to its robustness against object pose changes and partial occlusion. Bradski developed an algorithm called CAMSHIFT [8] which tracks the face in video sequence using the color histogram of the skin. In order to locate the face from frame to frame, an iterative procedure based on mean shift is applied to center the object rectangle in the face region. At each iteration, the rectangle position is moved to a new position until convergence. Even though the Bradski's algorithm was developed for face tracking, it can be used to track any object of interest.

Comaniciu et. al. proposed the Kernel Based Tracking (KBT) algorithm [9] which uses the kernel weighted color histogram to represent the color distribution of the object. In the kernel weighted color histogram representation, pixels at the object peripheral are given smaller weights while the pixels at the center of the object are given larger weights. The localization of the object is performed iteratively through a mean shift method similar to CAMSHIFT.

The introduction of the kernel weighted histogram to the mean shift based algorithms greatly improves the tracking performance. However, the algorithm does not work well when tracking the object that changes in orientation and size. In order to address these problems, tracking using color correlogram was proposed in [11] to subsequently track both

Authors are with the School of Electronics Engineering, Korea University, South Korea.
E-mail: suryanto, hkkim, jerry, dhkim@dali.korea.ac.kr; sjko@korea.ac.kr



(a) (b) (c)

Fig. 1. Illustration of object tracking using the proposed algorithm. (a) The target object. (b) Center voting procedure. (c) Tracking result.

the object location and orientation. Collins proposed a mean shift method in scale space [13] to obtain the object location and scale simultaneously.

In this paper, we introduce a new object representation model and localization method. In order to represent the object to be tracked, we proposed a spatial color histogram model. Each bin in the spatial color histogram contains the information about the number of pixels belong to the color bin and the mean vectors of relative location of the color bin from object center. To find the new object location as object moves, we employ the center voting procedure to have each pixel in the new frame cast its votes about the new object center. After we obtain the new object center, we use the back projection method to segment the object from the background. Provided with the segmented object, there is no need to estimate the object size and orientation separately.

In the next section, we present our algorithm in detail. In section 3, we present the experiment results of our tracking algorithm and compare them with the KBT and level set based tracking algorithm results. We conclude our work in section 4.

II. Proposed Algorithm

Given an object centered at location $\vec{c} = [c_x, c_y]$ in the current frame, the tracking objective is to find the new object center $\vec{c}'$ at the next and the following frames. We will use the term *target object* to refer to the object to be tracked.

Following the convention in the field, we assume that at the initial frame, the object has been segmented out from the background. This object segmentation can be done at the initial frame by performing background subtraction algorithms [16], [14], [15] or even by manual selection by human operator. This initial object segmentation is only performed once and subsequent object localization at the following frame has to be accomplished without it.

A. Target Representation

Let $\{\vec{x}_i\}_{i=1\ldots n}$ be the location of the pixels belong to the target object and $\vec{c}$ be the location of the object center. To represent the target object, we use the spatial color histogram model $h_T(b) = \langle n_b, \vec{\mu}_{b,k} \rangle_{k=1\ldots K}$, where n_b is the number of pixels whose quantized values fall into b-th bin of the color histogram and $\vec{\mu}_{b,k}$ is the mean vectors of the position of those pixels relative to the object center. More formally,

$$n_b = \sum_{i=1}^{n} \delta\left[I(\vec{x}_i) - b\right], \tag{1}$$

$$\vec{\mu}_{b,k} = \frac{\sum_{i=1}^{n} (\vec{x}_i - \vec{c})\, \delta\left[I(\vec{x}_i) - b\right]}{\sum_{i=1}^{n} \delta\left[I(\vec{x}_i) - b\right]}, \text{if } \|(\vec{x}_i - \vec{c}) - \vec{\mu}_{b,k}\| < \varepsilon, \tag{2}$$

where δ is the Kronecker delta function, $I(\vec{x}_i)$ maps the location vector x_i to the color histogram bin, and ε is a decision threshold to decide whether the pixel should be grouped into k-th cluster or not. This target representation is similar to the spatiogram model proposed in [10], but instead of forcing a single mean and covariance to represent the spatial distribution of the pixels belong to the bin, we allow each bin to have more than one mean values, forming a codebook of mean vectors.[1]

As an illustration, consider the target object in Fig. 1(a). The alphabets indicate which color histogram bin the pixels belong to. The spatial color histogram for this object is given in Table I. Note here that there are two pixels belong to the same color histogram bin *a*, but they are not clustered together due to their spatial distance. Thus, instead of a single mean for the histogram bin, it has two mean vectors. On the other hand, the two pixels *e* are proximately close to each other and has been grouped into the same cluster. It has only a single mean. At the next section, we will see how we can use this model to locate the object in the image.

TABLE I
Spatial Color Histogram for the Target Object in Fig. 1(a)

bin index	n_b	$\vec{\mu}_{b,k}$
a	2	(-1,-1), (1,1)
b	1	(0,-1)
c	1	(1,-1)
d	1	(0, 0)
e	2	(-0.5, 1)

[1]However, we use a very different approach in target localization as we will show later in the next subsection.

B. Target Localization

The localization of the target object in the following frame is accomplished in two steps: center voting and back projection.

1) Center Voting Procedure: As the name implies, the center voting procedure asks every candidate pixels near the previous center to cast votes about the location of the object center. Several rules about the center voting procedure:

1) Only the pixels whose color exist in the target model may cast a vote.
2) The pixels cast their votes based on the codebook of mean vectors $\vec{\mu}_{b,k}$.
3) More reliable pixels cast votes of higher weights than the less reliable pixels.

In order to see how these rules apply, see Fig. 1(b). The pixels labeled *x* are the pixels whose colors do not fall into any of the bins of the target model histogram. Thus, these pixels may not cast a vote on whereabouts the object center is. The bin *a* contains two mean vectors, thus pixels whose colors fall into this bin may cast two votes.

In this example, we have made a very naive assumption that the background does not share the same color with the object. This assumption, of course, does not hold in most cases and will cause the algorithm to fail when trying to estimate the correct center. The addition of the third rule, that the reliable pixels should cast votes of higher weight, solves this problem. We consider a pixel as a reliable pixel if it satisfies the following criteria:

1) The pixel does not share the same color with the background.
2) The pixel's color is dominant color of the object.

To quantify this criteria, we define the voting weight w_b for pixels whose color fall into the b-th bin as follows:

$$w_b = \frac{n_b}{m_b}, \tag{3}$$

where n_b is as given in (1) and m_b is the total number of pixels (both object and background pixels) in the neighborhood of previous center $\vec{c}$ whose quantized values fall into b-th bin. If there are no pixels in the background who share the same color with the object, the voting weight will be at its maximum, one. On the other hand, the voting weight will be small when there are a lot of background pixels who share the same color with the target object, indicating that the particular color bin is not reliable for localizing the correct object center.

Finally, the center of the votes can be calculated by

$$\vec{c}\,' = \frac{\sum_{\vec{x}_i \in \mathrm{Rect}(\vec{c})} w_b \left(\vec{x}_i - \vec{\mu}_{b,k}\right)}{\sum_{\vec{x}_i \in \mathrm{Rect}(\vec{c})} w_b}, \tag{4}$$

where $\vec{x}_i \in \mathrm{Rect}(\vec{c})$ indicates the search range, is the set of pixels inside a rectangle centered at previous object center $\vec{c}$. This search rectangle is shown in Fig.1(b) as a region enclosed by dash-line border. We limit the calculation of the votes to the pixels inside the search rectangle (instead of calculating the votes of all pixels in the frame) as we expect that the new center will be located near the previous center. The center of the votes calculated by this equation is the new object center $\vec{c}\,'$ as shown in Fig.1(c).

2) Back Projection: After the center voting procedure is completed, we obtain the new object center $\vec{c}\,'$. Based on this new object center, we re-scan the pixels in the neighborhood to see which pixels have casted the correct votes. The pixels who have casted the correct votes about the object center are marked as object pixels. For the robustness in tracking non-rigid object and also in order to deal with object posture change, we allow pixels who voted close enough to the new object center $\vec{c}\,'$ to be categorized as object pixels as well. This back projection method is best presented with the pseudo code followed.

Algorithm 1 Back Projection

```
for all x⃗ ∈ Rect(c⃗) do
  distance ← ‖vote(x⃗) − c⃗′‖
  if distance < ε then
    x⃗ is foreground pixel
  else
    x⃗ is background pixel
  end if
end for
```

III. Experiment Result

In this section, we provide the result of tracking using our algorithm. At the initial frame (and only at the initial frame), we manually select the target object and model it using the proposed spatial color histogram model presented in section II-A. For all experiments, we use 30x30x30-bins RGB color histogram and set the spatial clustering threshold ε to 5.

Fig. 2(a) and 2(b) show the initial frame with the target object marked in green and the tracking result for frame 100 with the predicted object marked by rectangle, respectively. In order to give the reader a visualization of how the center voting procedure work, we present the vote-map for the particular frame in Fig. 2(c). The intensity indicates the weight of the votes w_b; higher intensity denotes higher weight, thus higher probability of being the object center $\vec{c}\,'$ we are trying to estimate. After we successfully predict the object center by employing the center voting procedure, we perform the back projection and subsequently segments the object from the background. We show this segmentation result in Fig. 2(d). Figure 2(e), 2(f), 2(g), and 2(h) show more tracking results along with the segmented object results.

In order to assess its performance, we compare our algorithm with kernel based tracking algorithm [9] and the level set algorithm [3]. The results of KBT algorithm, level set algorithm, and our proposed algorithm are shown in Fig. 3(a), 3(b), and 3(c), respectively. The performance differences among these algorithms become apparent when the target object become smaller and smaller as it moves away from the camera. As shown in the second column of the figure, the object has become smaller than when it was in the initial frame, thus, without a good size adaptation technique, the algorithm like KBT algorithm will fail to localize the correct object center.

The third and fourth column of the figure compare the robustness of the algorithm when dealing with background

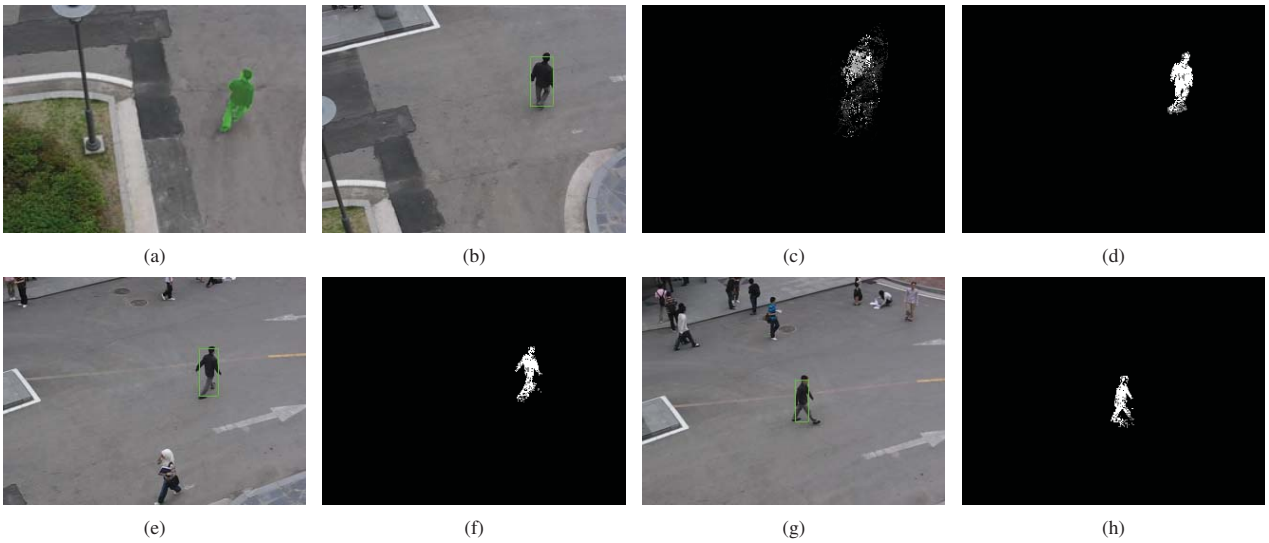


Fig. 2. Tracking result using proposed algorithm. (a) Target object at initial frame. (b) Tracking result at frame 100. (c) Vote map for frame 100. (d) Segmented object for frame 100. (e) Tracking result at frame 130. (f) Segmented object for frame 130. (g) Tracking result at frame 283. (h) Segmented object for frame 283.

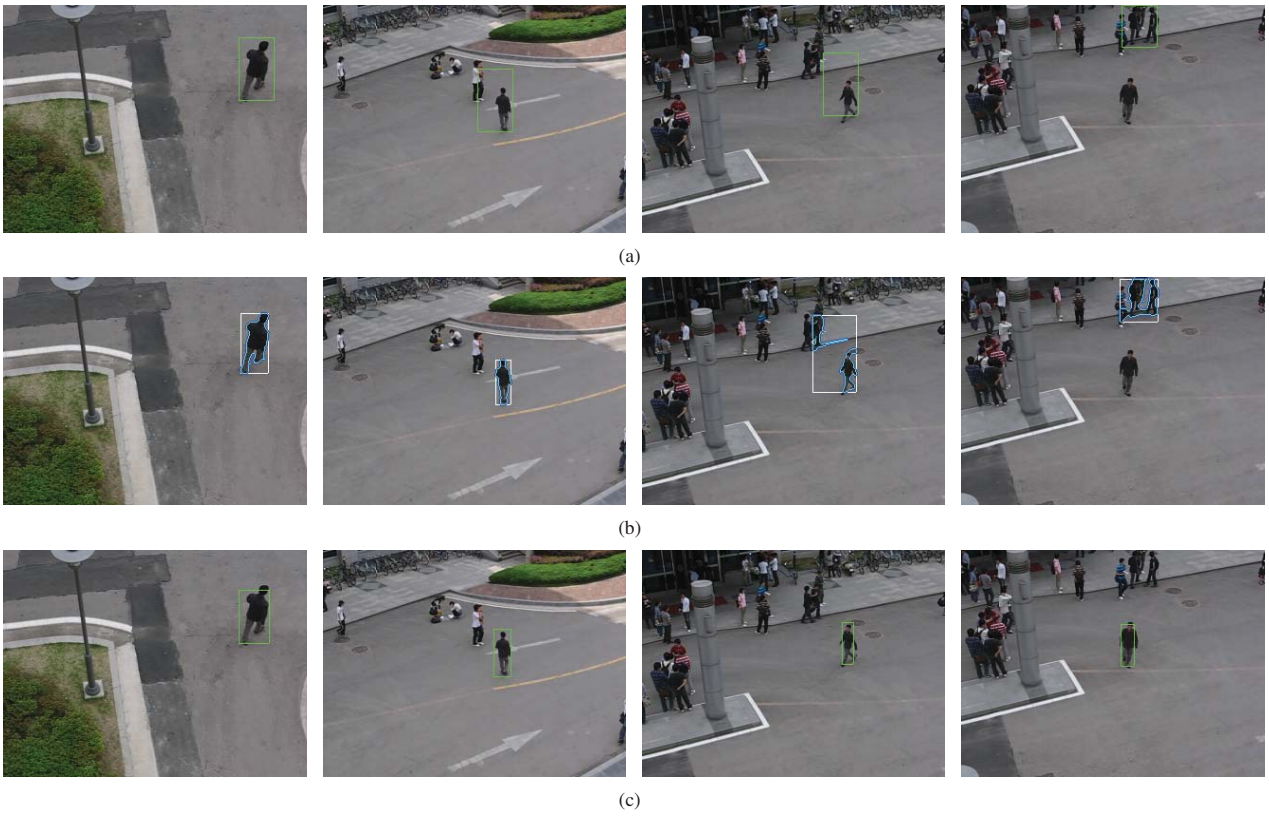


Fig. 3. Comparison of tracking results using: (a) KBT algorithm, (b) Level set based tracking algorithm, and (c) The proposed algorithm. From left to right: frame 79, 168, 235, and 258.

with similar color. Our algorithm successfully tracks the target object despite of the presence of background with color similar to the target object. This robustness against the background with similar color is due to the use of adaptive voting weights.

IV. Conclusion

In this paper, we have proposed a new approach to object tracking. Our tracking method is based on a special class of color histogram called spatial color histogram. This spatial

color histogram allows a loose representation of both the object's color and spatial configuration, make it a suitable model for representing both rigid and non-rigid object.

As we show in the experiment results, the algorithm tracks the object successfully through frames and correctly adapt the tracking rectangle as the object change in size. Tracking object changing in size is challenging as many algorithms has failed to cope with. The algorithms also shows a good robustness when tracking against the background with similar color, which attribute to its use of adaptive voting weights.

In addition to the object location, our algorithm also categorizes each pixel to either foreground or background, effectively segments the object from the background. This segmented object, to our knowledge, was not available in the existing color histogram based tracking algorithms. Thus, this work is a good addition to the tracking algorithm in this class.

ACKNOWLEDGMENT

This research was supported by Seoul Future Contents Convergence (SFCC) Cluster established by Seoul R&BD Program and was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MEST) (No. 2009-0080547).

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object Tracking: A survey," ACM Computing Surveys, vol. 38, no. 4, pp. 13, 2006.
[2] M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking," International Journal of Computer Vision, vol. 29, no. 1, pp. 5-28, August 1998.
[3] Y. Shi and W. C. Karl, "Real-Time Tracking Using Level Sets," Proc. IEEE Computer Vision and Pattern Recognition, vol. 2, pp. 34-41, June 2005.
[4] J. Shi and C. Tomasi, "Good Features to Track," Proc. IEEE Computer Vision and Pattern Recognition, pp. 593-600, 1994.
[5] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," Technical Report CMU-CS-91132, Pittsburgh:Carnegie Mellon University School of Computer Science, April 1991.
[6] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, vol. 60, 1999, pp. 91-110, November 2004.
[7] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346-359, June 2008.
[8] G.R. Bradski, "Real Time Face and Object Tracking as A Component of A Perceptual User Interface," Applications of Computer Vision, pp. 214 - 219, 1998.
[9] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based Object Tracking," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, May 2003.
[10] S. T. Birchfield and S. Rangarajan, "Spatiograms versus Histograms for Region-Based Tracking," Proc. IEEE Computer Vision and Pattern Recognition, vol. 2, pp. 1158-1163, June 2005.
[11] Q. Zhao and H. Tao, "Object Tracking Using Color Correlogram," Proc. IEEE Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 263-270, October 2005.
[12] B.D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," Proc. of the 7th International Joint Conference on Artificial Intelligence, Vancouver, pp. 674–679, 1981.
[13] R. T. Collins, "Mean-shift Blob Tracking through Scale Space," Proc. IEEE Computer Vision and Pattern Recognition, vol. 2, pp. 234-240, 2003.
[14] C. Stauffer and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," Proc. IEEE Computer Vision and Pattern Recognition, vol. 2, pp. 246-252, June 1999.
[15] A. Elgammal, D. Harwood, and L.S. Davis, "Non-parametric Model for Background Subtraction," European Conference on Computer Vision, vol. 2, pp. 751-767, 2000.
[16] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-Time Foreground Background Segmentation Using Codebook Model," Real-Time Imaging, vol. 11, no. 3, pp. 172-185, June 2005.