Video object segmentation and tracking using region-based statistics

Çiğdem Eроğlu Erdem*

Momentum Digital Media Technologies Inc., TÜBİTAK - MAM - TEKSEB, A-205, Gebze, 41470 Kocaeli, Turkey

Received 26 March 2007; received in revised form 29 August 2007; accepted 3 September 2007

Abstract

Two new region-based methods for video object tracking using active contours are presented. The first method is based on the assumption that the color histogram of the tracked object is nearly stationary from frame to frame. The proposed method is based on minimizing the color histogram difference between the estimated objects at a reference frame and the current frame using a dynamic programming framework. The second method is defined for scenes where there is an out-of-focus blur difference between the object of interest and the background. In such scenes, the proposed “defocus energy” can be utilized for automatic segmentation of the object boundary, and it can be combined with the histogram method to track the object more efficiently. Experiments demonstrate that the proposed methods are successful in difficult scenes with significant background clutter.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Object tracking; Active contours; Histogram matching; Curve evolution; Defocus; Selective focus

1. Introduction

Segmentation and tracking of objects in a 2D video is an important and challenging research area, which has many important applications including object-based video coding (MPEG-4), video post-production, content-based indexing and retrieval, surveillance, and 3D scene reconstruction for 3D TV [9]. These broad applications have different constraints and requirements in terms of tracking accuracy, processing speed, allowed user interaction, and available prior information about the object and the background.

An interesting application of video object segmentation is creating 3D TV content from 2D image sequences recorded by a single camera. Since it is quite challenging and perhaps not necessary to form a complete 3D model of the scene for creating a depth sensation in 3D TV applications, a relatively simple approach can be followed, which consists of segmenting the objects in a given 2D video and ordering their video object planes (VOPs) with respect to their relative depths [9]. Then, left and right views of the scene are rendered for stereo viewing, which yields a satisfactory sense of three dimensions. The described application requires the segmentation of objects with minimal or no user interaction and with no a priori knowledge about

---

*This work was partially supported by EC within FP6 under Grant 511568 with the acronym 3DTV NoE.

Tel.: +90 216 3632257.

E-mail addresses: cigdem.erdem@momentum-dmt.com, cigdem@ieee.org (Ç.E. Erdem).
the objects. The object segmentation results are also required to be temporally stable (i.e. the segmentation maps should not change drastically from frame to frame) for 3D viewing comfort [9]. Fully automatic object segmentation and tracking is a very challenging task since real world scenes are very different from each other and it is not fair to expect a single algorithm to segment all of them. Automatic object tracking can be achieved under certain conditions such as when prior information about objects to be tracked is available, or when the objects are very distinct from the background in terms their color, motion or texture which can be learned a priori [5,26].

Active contours have been widely utilized for semi-automatic video object segmentation and tracking [14,13,11]. The active contour model for object segmentation was first introduced by Kass et al. [16], which formulates the contour that will snap to the desired object boundary as a snake which slides down an “energy hill”. The energy minimization problem is handled in a variational framework. Later, a dynamic programming approach has been introduced for solving variational problems in vision [1], which assures global optimality of the solution, is numerically more stable and allows easy incorporation of hard constraints to the solution. This dynamic programming approach for minimizing the energy of an active contour has been utilized in a scalable object-tracking framework [13,11]. In [11], the energy of the active contour is composed of four energy terms which are weighted adaptively and derived from the edges, color and motion segmentation boundaries and the local curvature of the boundary. Isard and Blake [14] have developed the CONDENSATION (conditional density propagation) method for real-time object tracking using active contours. This algorithm needs prior training to learn the shape space for the possible object boundary shapes. The parameters of the motion model for the object is also estimated in the training phase.

Xie et al. [25] have proposed a method for using region segmentation boundaries as a vector flow force in a level set-based partial differential equation formulation. This Region-Aided Geometric Snake (RAGS) method can be used with any region segmentation technique and it provides improvements over the standard geometric snake especially around weak edges and in noisy images. In [17], active shape models method [5] for non-rigid object tracking has been extended for color images and a hierarchical implementation has been proposed. The performance in the RGB color space was found to be better as compared to HSI and YUV color spaces. In [9], the authors present a face segmentation and tracking algorithm. The face is first segmented using color information, which is then used to initialize a snake for face tracking. The face color information is utilized to guide portions of the snake in the correct direction. That is, if a portion of the snake is inside the face region, it is pushed outside, and if it is outside the face region, it is pulled inside. A method for image segmentation that uses the texture information as a region descriptor has been presented in [18]. The method computes a set of derivative features for multiple spatial orientations and scales to capture the texture information. Then, a discriminant snake deforms to achieve the final segmentation. In [3], a region snake-based probabilistic framework has been presented for segmentation of an object with unknown gray levels in front of a random background with pdf from the exponential family. This method is implemented using polygonal snakes and it has been demonstrated that this approach is efficient even the boundary edges do not exist in the image. Another method of implementing region-based approaches to snakes is to use level sets [26], which enables automatic splitting and merging of the snake boundary to delineate disconnected yet similar regions in the image.

Although there are many object tracking algorithms in the literature, there are a only few studies [10,6] to evaluate the quality of the tracking results in an objective and quantitative way in the absence of reference segmentation. It has been shown that [9,10] the difference between color histograms of segmented VOPs in successive frames is a good indicator of the temporal stability of the object segmentation results, which is especially very important for the problem of object-segmentation-based depth extraction in 3D TV applications [9]. Therefore, minimal histogram difference is a desired property of the video object segmentation results under the assumption that the color histogram of objects is nearly stable between successive frames. The idea of enforcing minimal histogram differences for object tracking has been recently explored by Comaniciu et al. [4]. In their method, the boundary of the object is approximated by an ellipse which is drawn by the user in the first frame. Then, the algorithm tracks this elliptic region in subsequent frames by maximizing the Bhattacharyya coefficient.
between the color histograms of the given object and the candidate regions, which are weighted by a kernel. However, using an ellipse to define an object may not be accurate and flexible enough for some object-tracking applications. The histogram matching approach has also been studied recently in a variational framework [15,28] in cooperation with some constraints on the shape of the object.

The technique of defocusing (selective focus) is widely used in commercial videos to attract the attention of the viewer to the object of interest or to hide some unwanted details in the background. Therefore, out-of-focus blur is an important region-based statistical property that can be utilized for automatic segmentation and tracking of the objects in a scene. Defocus property of objects has been used for unsupervised image segmentation in the literature [21,27,22,23]. In [21], a moment-preserving method is used to measure the amount of defocus at an edge pixel. The in-focus foreground object is segmented from the out-of-focus background after an edge-linking process. In [23], the local variance of the image is used to measure the existence of high frequency components at a pixel. Direct thresholding of the local variance image field (LVIF) results in blob-like holes in the in-focus foreground object. In order to eliminate such holes, a block-wise Markov random field approach is followed for maximum a posteriori segmentation of the LVIF. In [22], the image is divided into blocks, which are then labeled as background (out-of-focus) or object-of-interest (in-focus). The average intensity of the image block and the variance of the wavelet coefficients in the high frequency bands are used as the features in the classification process.

In this paper, we first introduce a new object tracking method, which is based on minimizing the histogram differences between successive frames and can be utilized in a dynamic programming-based active contour framework such as [11]. Color histogram is a region-based property of an object, which is different from the boundary-based (e.g. edge-based) energy terms used in conventional snakes. As the boundary of the object is modeled using a polygonal snake, it is more flexible than the ellipse or bounding-box-based tracking approaches. Since the dynamic programming approach is utilized for energy minimization, hard constraints on the solution (e.g. constraints on the distance of two boundary points) can easily be enforced, as compared to the variational approaches. The initialization of the boundary of the object to be tracked is carried out by user interaction.

The second contribution of this paper is to introduce a defocus-based object segmentation method, which is also designed for a dynamic-programming-based active contour framework. Our approach of utilizing the defocus statistic for object tracking is as follows. First, out-of-focus blur parameter for each pixel of the given frame is estimated [12,24,8]. Then, the estimated dense blur parameters are converted into a snake energy term, as will be explained in the following sections. It is important to note that, the proposed defocus statistic-based energy term can easily be combined with the proposed color histogram-based energy term or other snake energy terms.

The organization of the paper is as follows. In Section 2, the background theory of active contours is briefly reviewed. In Section 3, the histogram-based energy term is introduced. In Section 4, the process of dense blur parameter estimation is briefly reviewed and the defocus-based energy term is introduced. In Section 5, experimental results are provided, and finally concluding remarks are given in Section 6.

2. Overview of active contours using dynamic programming

The boundary of the object to be tracked is modeled using a polygonal snake, as shown in Fig. 1. Let \( n_i = (x_i, y_i), \quad i = 1, \ldots, N + 1, \) denote the vertices of the polygon for the estimated initial location of object boundary at frame \( t \), which can be the same as the final location of the boundary at frame \( t - 1 \). In the rest of the paper, the superscript \( t \) will be dropped. The parameter \( N \) above denotes the number of edges in the polygonal snake.

Fig. 1. The object boundary is modeled by a polygonal snake. The object region \( R \) is the shaded area.
The total energy of the object boundary, is defined as the summation of internal and external energies of each polygon edge as follows:

\[ E_{\text{snake}} = \sum_{i=1}^{N} (E_{\text{int},i} + E_{\text{ext},i}), \]  

(1)

\[ E_{\text{int},i} = \beta_{\text{curv},i} E_{\text{curv},i}, \]  

(2)

\[ E_{\text{ext},i} = \beta_{\text{hist},i} E_{\text{hist},i} + \beta_{\text{edge},i} E_{\text{edge},i} + \beta_{\text{defocus},i} E_{\text{defocus},i}, \]  

(3)

where \( E_{\text{hist}}, E_{\text{edge}} \) and \( E_{\text{defocus}} \) are external energy terms, which are calculated from the color histogram of the region enclosed by the polygonal boundary, edge and defocus information, respectively. \( E_{\text{curv}} \) is an internal energy term associated with the curvature of the boundary. The details for the calculation of these energy terms will be provided in the following sections. The parameters \( \beta_{\text{curv}}, \beta_{\text{hist}}, \beta_{\text{edge}} \) and \( \beta_{\text{defocus}} \) are the weighting coefficients of the energy terms.

We want to minimize the total snake energy (1) in order to estimate the optimal object boundary. The snake energy is minimized using the dynamic programming method [1,13,11], where the search locations for each node are selected along the normal lines (angle bisectors) drawn at the vertices \( \{n_i\}, i = 1, \ldots, N + 1 \) as illustrated in Fig. 2. The dynamic programming energy minimization is repeated until most (e.g. 95\%) of the boundary points are stationary between two successive iterations.

For the calculation of the external energy term \( E_{\text{edge},i} \), the procedure in [11] is followed. First, the edges of the current frame are estimated using a Canny edge detector [2], and then the binary edge image is transformed using a distance transformation method (e.g. Chamfer distance transform [11]) in order to obtain a smooth terrain that will guide the snake to the nearest edges. This approach enables the snake to move towards the nearest edges even if the image gradient is zero at its current location. Therefore, the snake can be pulled towards the nearest edges even from large distances.

Let the Chamfer distance transform of the edge map be denoted as \( C(., .) \). The edge energy of a line segment is calculated as follows:

\[ E_{\text{edge},i}(k, l) = \sum_{(x,y) \in L} C(x, y) / \|L\|, \quad k, l \in \{1, \ldots, M\}, \]  

(4)

where \( \|L(k, l)\| \) denotes the length of the line segment \( L(k, l) = n_{i-1}(k)n_i(l) \) that is connecting the current search nodes \( n_{i-1}(k) \) and \( n_i(l) \) (see Fig. 2) during energy minimization using dynamic programming. The indices \( k \) and \( l \) denote the search locations along the normal lines (which are the angle bisectors) drawn at nodes \( i - 1 \) and \( i \), respectively, and \( M \) is the total number of search locations at a given node. The denominator term in the above equation makes the edge energy term independent from the length of the polygon boundary, which enables hierarchical implementation of the algorithm [13].

![Fig. 2. Illustration of search locations at each vertex of the polygonal snake during dynamic programming. The parameters \( j, k \) and \( l \) denote the indices of the candidate locations for vertices \( n_{i-2}, n_{i-1} \) and \( n_i \), which will be tested during dynamic programming.](image-url)
The internal curvature energy term is calculated as follows [11]:

\[ E_{\text{curv},i}(j, k, l) = 2 + 2 \cos(\angle n_{i-1}(k)n_{i-2}(j), n_{i-1}(k)n_{i}(l)), \]

where the indices \( j, k, l \in \{1, \ldots, M\} \) denote the search locations along the normal lines drawn from the nodes \( i-2, i-1 \) and \( i \), respectively, as shown in Fig. 2.

The histogram energy for the object is then defined as follows:

\[ E_{\text{hist},i}(k, l) = \frac{\sum_{m=1}^{2} I_m \sum_{(x,y) \in T_m} \sum_{b=1}^{B} \delta[b(x, y) - s] \| L \|}{\| L \|}, \]

where \((x, y)\) denotes a pixel location inside triangle \( T_1 \) or \( T_2 \) corresponding to \( m = 1 \) and 2, respectively, \( b(x, y) \) denotes the color histogram bin that the pixel at \((x, y)\) belongs to, and \( B \) is the total number of bins in the color histogram. In (8), \( H^o_{i-1}(.) \) denotes the reference (desired) color histogram, which is chosen as the color histogram of the object at frame \( t - 1 \), and \( H^o(.) \) denotes the current color histogram of the object at a certain iteration during dynamic programming.

An illustration of the idea expressed in (8) is given in Fig. 4. Let the solid line denote the reference (desired) histogram of the object that we are tracking and let the dashed line denote the histogram of the currently estimated object region. Consider a pixel at location \((x_1, y_1)\), the color of which belongs to the histogram bin \( b(x_1, y_1) = s_1 \). The decision to add or remove this pixel from the currently estimated object is given according to the difference between the desired and current histogram differences of the bin \( s_1 \). If the current histogram value is less than the desired value, we should try to add that pixel to the object, by contributing a negative value to the histogram energy (8). Similarly, if the current histogram value at bin \( s_1 \) is more than the desired value, we should try to remove that pixel from the object region, by contributing a positive value to the histogram energy (8).

The color histogram statistic estimates the color distribution inside the whole object boundary. Sometimes a portion of the object region may also have the same color histogram as the whole object. For example, if we want to track a red ball in the scene, half of the ball will have the same color histogram of the whole ball. In such cases, the color
histogram energy defined in (8) may not be successful in finding the correct object boundary. Therefore, if we assume that the color histogram of the background around the object is also stationary, we can expect the differences between the background color histograms in two successive frames to be small.

Hence, the above definition of histogram energy (8) can easily be extended for the background:

\[
E_{\text{hist}}^a(k, l) = \frac{\sum_{m=1}^{2} - I_m \sum_{(x,y) \in \mathcal{T}_m} \sum_{s=1}^{B} (H_{i-1}^a(s) - H^a(s)) \delta[b(x,y) - s]}{\| \mathcal{L} \|},
\]

where the superscript \(^a\) has been used to show that the color histograms belong to the background. The "background" can be defined as the pixels outside the object boundary and in a region that is several times the area of the object.

Combining the "object" and "background" color histogram energies given in (8) and (9), the overall definition of the histogram energy is then defined as follows:

\[
E_{\text{hist},i}(k, l) = w^o E_{\text{hist},i}^o(k, l) + w^a E_{\text{hist},i}^a(k, l),
\]

where \(w^o\) and \(w^a\) determine the relative weights of the two terms. If the background is changing rapidly from frame to frame, the weight \(w^a\) should be set to zero.

4. Defocus energy

The technique of defocusing (or selective focusing) is widely used in cinematography to direct the attention of the viewer to the object(s) of interest in the scene or to hide some unwanted details in the background. Selective focusing is carried out by
blurring the background while keeping the foreground objects in focus. Therefore, out-of-focus blur is an important region-based statistical property that can be utilized for automatic segmentation and tracking of the objects in the scene.

In this section, we introduce a new energy definition for segmenting/tracking the in-focus objects in the scene. The new “defocus energy” utilizes the dense blur parameters, which are estimated for all pixels at each frame. The approach that we used for dense blur parameter estimation [24] is briefly reviewed in the next subsection. Then, the formulation of the “defocus energy” term will be presented.

4.1. Estimation of dense out-of-focus blur parameters

In the literature, out-of-focus blur is generally modeled with a Gaussian blurring kernel [8]. When a sharp edge, which is modeled using a step function, is filtered with a Gaussian filter of blur scale $s_{b}$, its profile looks similar to the solid curve given in Fig. 5. When this blurred edge is filtered with a Laplacian of Gaussian (LoG) function with a scale of $s_{f}$, the result is as given by the dashed curve in Fig. 5, where the distance between the two peaks is related to the LoG filter and out-of-focus blur parameters as $d^{2} = s_{f}^{2} + s_{b}^{2}$. Therefore, the out-of-focus blur parameter $s_{b}$ can be estimated by filtering the edges using an LoG filter in the direction of the gradient and measuring the distance between the peaks of the resulting function [8].

The above basic method is adopted to estimate a blur parameter for each pixel of a given frame as follows [24]:

(1) First, color segmentation of the given frame is estimated [12] using a watershed-based method. An example is shown for the iguana image in Fig. 6(a), where the color segmentation boundaries are drawn by white lines in Fig. 6(b).

(2) For each pixel on a color segmentation boundary, a blur parameter is estimated using an LoG filtering approach [12, 24, 8] as described above. For robustness, the median of the $s_{b}$ values estimated using different $s_{f}$ values is chosen.

(3) The blur values estimated for all pixels on a piece of color segmentation boundary (shown by white line segments) are averaged to obtain a single blur parameter for that line. An example is shown in Fig. 6(c), where the red color indicates segmentation boundaries with large blur parameters and the blue indicates boundaries with small blur parameters. The colors are spread to the neighborhood of segmentation boundaries to fill the whole image.

(4) The out-of-focus blur parameters estimated for segmentation boundaries (white lines) are spread to the whole image using averaging as follows. Each color segmentation region is surrounded by a set of line segments. The out-of-focus blur parameters estimated for these line segments are averaged to obtain a single blur value for the color segmentation region.

![Fig. 5. The blurred edge modeled using a Gaussian blur parameter $s_{b}$ (drawn with a solid curve). The Laplacian of Gaussian (second derivative) of the blurred edge is shown by the dashed curve (magnitude is scaled by 1000 for better visualization). The distance between the two peaks of the dashed curve $2d$ is related to the out-of-focus blur parameter $s_{b}$ and the LoG filter parameter $s_{f}$ with the equation $d^{2} = s_{f}^{2} + s_{b}^{2}$. The actual parameters chosen for the above plot are $s_{b} = 10, s_{f} = 15, d \approx 18$.](image)
The blur parameters estimated for each color segmentation region as described above may be noisy. Therefore, a final smoothing procedure is applied to the whole image as follows. The blur parameter estimated for each region is updated using the average value of its neighboring regions via a constrained relaxation procedure [24]. The final result for the iguana image is shown in Fig. 6(d), where red indicates out-of-focus regions and blue indicates in-focus regions.

Another example for dense blur estimation is given for the “Flikken” sequence in Fig. 7. In this sequence, the lady who is closer to the camera is out-of-focus and the background including the walking lady is in focus. In Fig. 7(b), the color segmentation of frame 464 is shown. The image in Fig. 7(c) shows the estimated dense blur parameters, where the brightness of blue or red is proportional to the reliability (variance) of the estimated blur value for that region [24].

### 4.2. The definition of defocus energy

The estimated dense out-of-focus blur parameters can be formulated as a region-based energy term in our snake-based object tracking framework since the object and the background have different blur parameter distributions.

In Fig. 8, the histograms of the estimated blur parameters for the iguana image and a frame of the

---

1A TV series produced by MMGNV for VRT.
The two peaks seen in the plots correspond to the foreground and background regions. The simplest approach to delineate the object is to cluster the pixels of the frame into two classes by thresholding the estimated dense blur parameters. In order to find a suitable threshold, the histogram of the dense blur parameters (Fig. 8) can be modeled as a mixture of two Gaussian distributions. Let the mean and standard deviation of these Gaussians be denoted by $\mu_o, \sigma_o$ for the object and $\mu_b, \sigma_b$ for the background. These means and the standard deviations can be estimated using the object segmentation results of the previous frame at time $t-1$, and can be updated after the segmentation of frame $t$ is completed. This update is necessary because if the object is moving, the relative depths of the objects in the scene and hence blur parameters may change. Once the means and the standard deviations are estimated, they provide us the class conditional distributions of the blur parameters for the object and the background. We can use a logarithmic Bayesian discriminant function [7]:

$$
\begin{align*}
    d(x) &= \frac{1}{2} \left( \frac{1}{\sigma_b^2} - \frac{1}{\sigma_o^2} \right) x^2 + \left( \frac{\mu_b}{\sigma_o^2} - \frac{\mu_o}{\sigma_b^2} \right) x \\
    &+ \frac{1}{2} \left( \frac{\mu_b^2}{\sigma_b^2} - \frac{\mu_o^2}{\sigma_o^2} \right) + \frac{1}{2} \log \frac{\sigma_b^2}{\sigma_o^2} - \log \frac{p(b)}{p(o)},
\end{align*}
$$

(11)

where $p(o)$ and $p(b)$ denote the probabilities of the object and the background, which can be estimated from the relative areas of the object and the background in the previous frame. The discriminant function gives us a suitable threshold $\tau_b$, which is obtained by solving the equation $d(x) = 0$. Other unsupervised non-parametric classifiers could also be used [7].

After thresholding the blur parameters using the estimated threshold $\tau_b$, a defocus image $D(x,y)$ is obtained such that background pixels have the value $-1$ and the object pixels have the value $1$.

Then, the defocus energy is defined as follows:

$$
E_{\text{defocus}}(k,l) = \frac{\sum_{m=1}^{2} I_{m} \sum_{(x,y) \in T_{m}} D(x,y)}{||L||},
$$

(12)

where $I_{m}$ and $T_{m}$ are the indicator functions and the triangles formed during dynamic programming, which were defined in Section 3 and $||L||$ is the length of the line segment connecting the search nodes $n_{i-1}(k)$ and $n_{i}(l)$.

5. Experimental results

The effectiveness of the proposed color histogram and defocus-based methods are demonstrated via the experimental results using three different video sequences: Snooker, Market and Flikken. Snooker sequence (see Fig. 11) contains two balls moving in front of a heavily textured background. Market sequence is a crowded scene (see Fig. 13), where many people are walking and the Flikken sequence contains out-of-focus blur (see Fig. 17(a)).

5.1. Utilization of histogram energy

In the following experiments, the normalized RG color space is used due to its illumination invariance properties, which is defined as $R = r/(r + g + b)$ and $G = g/(r + g + b)$ [20].

5.1.1. Snooker sequence

In the Snooker sequence, we are trying to track the red ball, which moves towards the yellow ball, becomes occluded by it, and then comes back.
The motion of the red ball can be visualized in Fig. 11. In Fig. 9(a), the initial reference segmentation (which is drawn interactively) of the red ball, and the corresponding reference object histogram are shown for frame 20.

First, let us demonstrate that the color histogram energy is able to find the correct object boundary even if we start from a very bad initialization in the next frame. The initial boundary in frame 21 of the Snooker sequence is shown in Fig. 9(b) together with the color histogram of the region inside the boundary. The goal is to find the region in frame 21 that has the closest color histogram to the reference histogram given in Fig. 9(a). Starting from the bad initialization, shown in Fig. 9(b) the snake is able to find the red ball object correctly in 17 iterations as shown in Fig. 9(c)–(e). The weights in (2), (3) were chosen as $b_{\text{curv}} = 1$ and $b_{\text{hist}} = 2$.

Next, we demonstrate that using only the edge and curvature energy terms in (1) as in traditional snakes may result in a tracking failure in cluttered scenes. In Fig. 10(a) the edge map for frame 21 of

Fig. 9. (a) The reference ball object and the corresponding reference color histogram in frame 20. (b) The initial boundary in frame 21 (top) and the corresponding object histogram (bottom). Plots in (c)–(e) show the object boundaries (top) and the corresponding histograms after iterations 7, 11, 17.

Fig. 10. (a) The edge map of frame 21. (b), (c) The tracking results for frames 21 and 30 using only edge and curvature energy terms. The edge energy is not successful in tracking, since it is easily disturbed by the very cluttered background.
the Snooker sequence is shown, which reveals that the edge map of the background is very cluttered. Starting from the initialization in frame 20, given in Fig. 9(a), the edge energy-based tracking results for frames 21 and 30 are shown in Fig. 10(b) and (c), respectively, which shows that edge energy is easily distracted by the cluttered background edges. The weights of the energy terms have been selected as $\beta_{\text{curv}} = 2$, $\beta_{\text{edge}} = 1$ and $\beta_{\text{hist}} = 0$ to obtain the results in Fig. 10.

Next, let us demonstrate the tracking results with the addition of the color histogram energy to (1). In Fig. 11, the tracking results for 80 frames of the Snooker sequence are shown. We can see that the object boundaries are accurately located in all frames. The utilization of the color histogram-based energy term together with the edge energy improves the tracking results greatly. The weights for the energy term used in (1) are $\beta_{\text{curv}} = 2$, $\beta_{\text{edge}} = 1$ and $\beta_{\text{hist}} = 5$.

The weights of the energy terms are selected experimentally; however, a few trials are often enough to select a suitable set of parameters, which specify the relative weights of the energy terms. In order to demonstrate how sensitive the results may be to the selection of weights, we set the parameters as $\beta_{\text{curv}} = 2$, $\beta_{\text{edge}} = 1$ and $\beta_{\text{hist}} = 1$. The tracking results with these parameters are given in Fig. 12 for

![Fig. 11](image1.png)

Fig. 11. The tracking results for the Snooker sequence with the addition of the proposed color histogram energy term for frames 30, 40, 50, 60, 70, 80, 89 and 99, from left to right, top to bottom, respectively. The weights for the energy term are (1) $\beta_{\text{curv}} = 2$, $\beta_{\text{edge}} = 1$, and $\beta_{\text{hist}} = 5$.

![Fig. 12](image2.png)

Fig. 12. Top row: the tracking results for the Snooker sequence demonstrating the sensitivity of the tracking results to the selection of energy weights. Frames 27, 30, 40 and 45 are shown from left to right, respectively. The weights for the energy term are (1) chosen as $\beta_{\text{curv}} = 2$, $\beta_{\text{edge}} = 1$ and $\beta_{\text{hist}} = 1$. Bottom row: the edge maps for frames 27, 30, 40 and 45.
the Snooker sequence. We can observe that the results are much better than shown in Fig. 10 (where $\beta_{\text{hist}} = 0$); however, they are not as good as the tracking results given in Fig. 11 (where $\beta_{\text{hist}} = 5$). Since the relative weight of the histogram energy term with respect to the edge energy term is not high enough, the boundary is pulled towards incorrect edges in the background in frames 27 and 30 as illustrated in Fig. 12. The Canny edge maps of the four frames are shown in the bottom row of Fig. 12. The parameters for the Canny edge detector are chosen as 0.5 for the standard deviation of the Gaussian filter and 0.05 for the percentage parameter during threshold selection. In Fig. 12, we can observe that the histogram energy term helps to correct the errors seen in frames 27 and 30 and the tracking continues with minor deviations as shown for frames 40 and 45. Next, we increased the weight of the histogram energy term to 10, i.e. we set $\beta_{\text{curv}} = 2$, $\beta_{\text{edge}} = 1$ and $\beta_{\text{hist}} = 10$. The tracking results are similar to the results shown in Fig. 11, for this sequence. Hence, a suitable set of weights can be selected easily after a few trials.

5.1.2. Market sequence

First, another demonstration for color histogram matching using snakes is given in Fig. 13 for the Market sequence. In this sequence, we want to track the head and neck of the walking man. The reference segmentation and its corresponding color histogram are shown Fig. 13(a) for frame 35. Starting from a crude rectangular initialization given in figure (b), the histogram energy term manages to pull the object boundary in 30 iterations to the correct location where the object histogram matches the reference histogram, as shown in (c). The weights were chosen as $\beta_{\text{curv}} = 2$, $\beta_{\text{edge}} = 1$ and $\beta_{\text{hist}} = 5$, for this sequence. Next, we demonstrate that using only the edge and curvature energy terms in (1) as in traditional snakes results in a tracking failure in this sequence. The tracking results without the histogram energy term are given in Fig. 14. We can see that there are large errors since the snake snaps to the wrong edges in the cluttered background.

The tracking results with the addition of the histogram energy term are given in Fig. 15. We can observe that the tracking results improve greatly with the utilization of the histogram energy term. The algorithm successfully tracks the head and neck of the walking man, despite the cluttered background. Although some local deviations are observable due to the rapidly changing color composition of the background, the head and neck is correctly tracked for 23 frames. The weights of the energy terms are chosen as $\beta_{\text{curv}} = 2$, $\beta_{\text{edge}} = 0.2$ and $\beta_{\text{hist}} = 5$ for this experiment.

5.2. Utilization of defocus energy

The utilization of the defocus energy will be demonstrated using the Flikken sequence, a sample

![Fig. 13. Market sequence. (a) The user-initialized boundary in frame 35 (top) and the corresponding reference histogram (bottom). (b) The initial location of the boundary in frame 36 (top) and the corresponding histogram. (c) The final boundary after 30 iterations (top) and the object histogram (bottom). $\beta_{\text{edge}} = 0.2$, $\beta_{\text{curv}} = 2$, $\beta_{\text{hist}} = 5$.](image-url)
frame of which was given in Fig. 7. In the Flikken sequence, the background is in focus and the foreground object (the lady closer to the camera) is out-of-focus. Our aim is to segment and track the out-of-focus lady object in the foreground. The gray scale images given in Fig. 16(a), (b) show the estimated dense blur parameters (mapped to the range $\frac{1}{255} > 0$) for frames 464 and 465 of the Flikken sequence.

In Fig. 16(c), (d), $D(x, y)$ images showing the thresholded dense blur parameters are given for frames 464 and 465 of the Flikken sequence. The pixels with the lightest gray value have the value of 1 and others have negative values (e.g. $-1$). The threshold value calculated using (11) was estimated to be $\tau_b = 2.1$, where the parameters $\mu_o = 2.67$, $\sigma_o = 0.41$, $\mu_b = 1.23$ and $\sigma_b = 0.40$ are calculated from the reference initialization at frame 464, shown in Fig. 17(a).

In order to demonstrate the effectiveness of the defocus energy, we start with a challenging

Fig. 14. Tracking results for the Market sequence without histogram energy term. Frames 35, 40, 49, 56, 58 are shown from left to right and top to bottom, respectively. The weights are chosen as $\beta_{\text{edge}} = 0.2, \beta_{\text{curv}} = 2, \beta_{\text{hist}} = 0$.

Fig. 15. Tracking results for the Market sequence with the addition of the histogram energy term. Frames 35, 40, 49, 56, 58 are shown from left to right and top to bottom, respectively. The weights are chosen as $\beta_{\text{edge}} = 0.2, \beta_{\text{curv}} = 2, \beta_{\text{hist}} = 5$.

Fig. 16. (a) and (b) The gray scale images for frames 464 and 465 of Flikken sequence showing the dense blur parameters (mapped to the range $[0, 255]$ for better visualization). (c) and (d) The defocus images $D(x, y)$ obtained by thresholding the dense blur parameters for frames 464 and 465.
rectangular initialization in frame 465, as shown in Fig. 17(b) and let the defocus energy pull the snake towards the correct object boundaries. Using only the defocus and curvature energy terms, we obtain the results given in Fig. 17(c) and (d) after 10 and 27 iterations. We can see that the defocus energy is able to locate the object boundaries correctly.

The tracking results for 20 frames of the Flikken sequence with and without using the defocus energy term are given in Figs. 19 and 18, respectively. Since the color histograms of foreground object and the background are close to each other, the correct object boundary cannot be located using the histogram energy term, as can be observed for frame 483 in Fig. 18. After the addition of the defocus energy term, the object boundary is tracked correctly, up to frame 483 as given in Fig. 19.

6. Discussion and conclusions

We presented two new region-based methods for object tracking using active contours in a dynamic programming framework. The color histogram-based method assumes that the color histogram of the tracked object is stationary from frame to frame. The “defocus energy”-based method is useful when there are out-of-focus regions/objects in the scene. Experimental results on various video sequences demonstrate that the proposed methods are successful in difficult scenes with significant background clutter or scenes containing out-of-focus regions.
Acknowledgments

The author would like to thank Philips Research Laboratories, Eindhoven, The Netherlands, for providing the image sequences and the dense blur estimation results used in this paper.

References


