Visual Tracking Based on Color Kernel Densities of Spatial Awareness

Zhuan Q. Huang, Zhuhan Jiang
School of Computing and Mathematics, University of Western Sydney, Australia
zhuang@scm.uws.edu.au, z.jiang@uws.uws.edu.au

Abstract

We propose a kernel-density based scheme that incorporates the object colors with their spatial relevance to track the object in a video sequence. The object is modeled by the color probability density function across a set of pixel regions on the object, partitioned in terms of the base shapes such as the concentric annuli or polygons at the object centre. The probability density of the object is derived by applying the kernel density estimator region-wise to the pixels within such partitioned areas. This proposed object representation enables the independent processing of the color features while at the same time making the implicit use of location information without having to involve additional model parameters. Weighting factors are also introduced to differentiate the significance of the relative physical locations when measuring the similarity of two probability density functions, and this facilitates the tracking of the more robust object features. The located object is then finalized for its boundary deformation by demanding a neighborhood similarity in colors at the object pixels near the boundary. Our experimental results showed this method is effective at tracking a non-rigid object on a moving background.

1. Introduction

Tracking an object throughout a video sequence often uses the similarity of certain features between the template and candidate target in the subsequent frame. Color features are the most common and important ones for this purpose. Many algorithms, such as template matching, histogram matching, region growing, and covariance preservation, to name a few, have been tried out by the researchers in the field to achieve a good tracking performance through the use of color features. On the other hand, some have utilized the edge, boundary, shape model or point feature to locate the newer object position within the sequence. It is well appreciated that the object is the one that often needs to be modeled *properly* in an algorithm, especially when the video sequence has a non-stationary background.

Though the template match method [1] uses both pixel color and location during the matching process, it is too strict to tie the pixel color to its precise location. This is because the color sampling of the pixels often results in slight difference in a neighborhood location. Moreover, the object appearance in a video would in general undergo some color changes and certain degree of shape deformation locally. Hence this type of template matching methods, even with the deformable template matching [2, 3], are not designed to handle such situations well. Other efforts like in [4, 5] tried to improve the robustness by representing the variability at each pixel in the template, and a learning stage has to be established there to estimate the variance of the brightness at each pixel over the training ensemble. Similarly, the view-based subspace model [6] can model variations in pose and lighting with the principal component analysis, but they also require training prior to tracking. In a different perspective, the histogram method [7] may offer some robustness under image distortions and occlusion using certain statistical properties; yet it does not utilize any potentially valuable spatial information of the pixels at all. Hence the tracking could deviate from its course when there are other regions nearby with similar statistics to the candidate region. Other efforts in this regard include Hager and Belhumeur's work [8] that explicitly modeled the geometry and illumination changes, which was later improved by Sclaroff and Isidoro [9] by using the robust M-estimators. In a further undertaking, Jepson et al [10] modeled the object appearance by a mixture of three components: a stable component that is learned with a long time-course, a two-frame transient component, and an outlier process. By identifying the stable properties of the appearance, they give these properties more significance in tracking. A real-time EM-algorithm is then adopted to adapt the appearance model parameters over time. However this complex scheme has to assume implicitly that non-target constraints should never be accepted as

0-7695-3067-2/07 $25.00 © 2007 IEEE
DOI 10.1109/DICTA.2007.39

consistent with the appearance model. Finally in a similar pursuit of robustness, a dynamic object representation [11] based on the characteristics of the local background has been recently proposed to use dominant feature elements to improve the tracking robustness.

There are other approaches in the field, such as recently the use of kernel estimators. The authors of [12], for instance, modeled the object and the background by a statistic with a kernel density estimator that integrates with the joint domain-range representation, where the spatial address of the image lattice is the domain and the corresponding color space is the range. Though the joint domain-range feature space can explicitly model the spatial dependencies, it results in high dimensions for the kernel that would lead to huge computational complexity. For the reduction on the computational cost, the fast Gauss transform proposed by Elgammal [13] has been used for the kernel density estimator. The kernel based tracker is also proposed by Dorin and Peter [14], where the reference target is initially represented by a histogram. They then regularize the similarity function by masking the object with an isotropic kernel in the spatial domain, and the candidate target is finally located by maximizing the Bhattacharyya coefficient via a gradient optimization method. The mean shift technique [15], most similar to a given metric based on the Bhattacharyya coefficient, has also been employed to search the target candidate. For other varieties of approaches, Shai [16] integrated the support vector machine classifier into an optic-flow-based tracker, and Oliver et al [17] made use of the relevance vector machine, similar to support vector machine but capable of delivering a fully probabilistic output, in their real-time tracking system. Due to the sparsity properties of the relevance vector machine, the tracker only needs to use a fraction of the CPU time.

In this work, we propose a method that represents an object with a kernel based color density function constructed also in the context of physical locations of the object pixels. These spatial locations are derived from certain predefined basic shapes that help reflect the robustness of the object appearance. In the case of having a set of concentric annuli at the centre of the object, this concept of robustness refers to the better preservation of the appearance stability when it is closer to the object centre. The feature of a given region is represented by the color density function which is derived from the object pixels within that region. This way, the difference of the density functions across the selected regions will be able to reflect the more generic difference of the object appearance. Hence it also makes sense for the weighting factors to be introduced to signify the importance of the pixels at different physical locations when measuring the similarity of two probability density functions, without the explicit use of the location parameters. The positioning of the object in a newer frame is based on minimizing the distance between the density function for the object in the previous frame and that for the candidate region. Once the centre of the object in a newer frame is known and the object shape is predicted, or rather projected, there, the object shape verification process is then carried out to refine the final object shape in the frame. This method is designed for tracking a non-rigid object in a moving background, and also for providing a flexible framework to deal with different situations arising from different types of object appearance changes, and from the different cost requirements on the computation.

This paper is organized as follows: Section 2 models the object appearance with a kernel based probability density across the spatial regions partitioned by a set of concentric annuli. Section 3 aims at locating the object in the current frame, using the proposed similarity metric to compare the probability density of the object with that of a candidate region. Section 4 then devises a procedure to verify whether the pixels near the projected object border remain indeed part of the object in the current frame. Some implementation issues and experimental results are shown in Section 5, and finally Section 6 is the conclusion.

2. Object model

An object's appearance may change throughout a video sequence due to the change of object posture, illumination condition, and the camera viewing angles. The pixel colors of the object are obviously correlated to the corresponding pixel positions in general. Hence we propose to model the object by the color of the pixels as well as their spatial locations, sufficient enough to ensure a stable structure of the object appearance.

The appearance change could take place anywhere on the object. We thus propose to model the object presence with its regions of stable appearance, surrounded by those areas of volatile appearance, so that we can track the object by locating the more trustworthy appearance features. For this purpose, we need to make the following regularity assumptions about the object appearance: 1) A moving object retains certain stable appearance within a small period of time, e.g. within a section of a video sequence; 2) The appearance of the central region of the object is more likely to be much more stable than the regions close to the object boundary; 3) The appearance in local object regions does not change too drastically, in other words, the changes or transitions are somewhat smooth; 4) The geometric centre of object remains on the object, similar to a convex shape. We assume that the scale of admissible appearance change is relatively uniform within each region, and in any case the object should be partitioned to fulfill this assumption as much as possible. Since the object may turn towards different directions as it moves across different frames, it would be beneficial to select the base shapes so that it remains invariant when the object rotates around the centre.

For the representation of an object in our model, we define a set of annular regions S_k by the concentric circles at the object centre, as shown in figure 1 (a), where the object template is the area within the dashed contour. The S_k are sequenced in such a way that S_1 is the disk containing the centre, and S_{k+1} is the annulus next to S_k on the outside. The pixels on the object are then grouped for each annulus S_k, and the object is represented by these groups of pixels in the annular regions. More precisely, the object feature is characterized by a probability density of the object pixels falling into each annulus, which can be calculated by a kernel density estimator on the pixel color within the annulus, as shown in figure 1 (b).

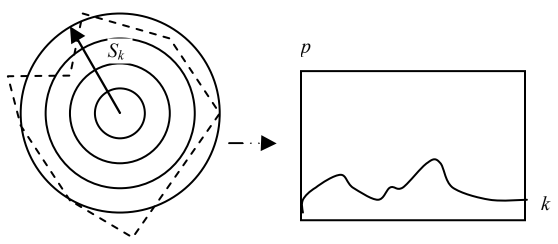


Figure 1. (a) Annular regions S_k. (b) Region-wise color density on k.

Let r_0=0, and r_k for k=1, .., m be the increasing radii for the annuli. An annulus $S_k = \{ x: r_{k-1} \leq \|x-c\| < r_k \}$, where x and c are both two dimensional vectors and c denotes the centre, is thus represented by the positions of all the pixels on the annulus, or disk when k=1. For each pixel position x, we denote by $I(x)$ the colors at the pixel in the current frame, while by $I^*(x)$, the corresponding colors in the previous frame. If we further let R_k be the object region within S_k, i.e. $R_k = \{x \in S_k$: x is on the object$\}$, then the object can be modeled by the density distribution of the object pixels in the annuli, and the nonparametric density function can be expressed as

$$p(k,c) = \frac{1}{D(c)} \sum_{x \in R_k} \psi(I^*(x-c)), \qquad \textbf{(1)}$$
$$D(c) = \sum_{x \in R} \psi(I^*(x-c)),$$

where $R = \cup_k R_k$, D is a normalization factor, and $\psi(\cdot)$ is a kernel function such as the Gaussian, the triangular, or the bi-weight functions. Such a density distribution, as illustrated in figure 1(b), implicitly characterizes the object appearance by mean of the pixels annularly distributed relative to the object centre. For simplicity, we often drop the parameter c in $p(k, c)$ and $D(c)$ when there can be no confusion. Hence, the density $p(k)$ for a smaller k reflects the characteristics of the pixels close to the object centre and the $p(k)$ for a larger k reflects the characteristics of the pixels relatively far away from the object centre. With this representation, we will then be able to analyze and process the pixels with the implicit consideration of their physical locations.

For a candidate region in the current frame of timestamp t, we denote by c^t the derived object centre c at timestamp t, and denote by $I(y)$ the color values at the pixel position y. Then the density for the current frame, similar to (1), is

$$p^t(k,c^t) = \frac{1}{D(c^t)} \sum_{y \in R_k} \psi(I(y-c^t)) \ \cdot \qquad \textbf{(2)}$$

The density $p^t(k)$ is similar to other type of color densities that can statistically represent the object features except that ours here has also incorporated the pixel locations.

There are several advantages of this representation for the object: a) It integrates the pixels location into the color description without the explicit use of any location parameters, thus simplifying the later matching steps. b) The geometric base shape for the regions can vary, and can even be object dependent. One can choose a proper geometric shape that can best reflect the change distribution of the object appearance. c) The statistical representation of pixels in a region reduces the effect of changes on the number of object pixels in that region. It is thus able to better retain the object characteristics. d) The probability density proposed in this method provides a straightforward connection between the pixel locations and the pixel colors. Since the pixels with similar stableness are largely grouped together within a base shape, such importance can be easily strengthened or weakened by selecting proper coupling weights. e) It facilitates the coarse-to-fine searching strategy in the later matching process.

3. Locating the object

If we use the object density $p^{t-1}(k)$ at time t-1 as the template, and compare it with the density $p^t(k)$ of a candidate region in the current frame of time t, then the closeness of these two density functions should be weighted differently for the different elements derived from the different annuli. For this purpose, we introduce a coupling weight function $\varphi(r)>0$ that is monotonically decreasing with $\varphi(r)\to 0$ as $r\to\infty$. For any two distributions $p(k)$ and $q(k)$, we define their inner product as $(p, q)=\sum_{1\le k\le m} \varphi(r_k)p(k)q(k)$. The more these two distributions, or vectors, are considered similar to one another, the more $(p, q)^2/[(p, p)(q, q)]$ is closer to 1. The weight function can give more significance or trust to the region closer to the object centre. This is consistent with the fact that a tracked object is more likely to deform along its boundary than in the centre area. Hence our model of annuli is particularly ideal to accommodate the changes that are somewhat proportional to the distances to the centre.

The object location in the current frame should then be the one that minimizes the distance between the density function for the object template and the density function for the candidate region. The distance metric for the two density functions could simply be a Euclidean distance. However, in order to accommodate the potential illumination changes in the frames, which is a common phenomenon in video sequences, we calculate the c^t as

$$\begin{aligned} &c^t = \min\arg(1-\omega(\cdot, c^{t-1})), \qquad (3)\\ &\omega(c, c^{t-1}) = \sum_{k=1}^{m}\varphi(r_k)\hat{p}^t(k,c)\hat{p}^{t-1}(k)) / \\ &\sqrt{(\sum_{k=1}^{m}\varphi(r_k)\hat{p}^t(k,c)^2)(\sum_{k=1}^{m}\varphi(r_k)\hat{p}^{t-1}(k)^2)}, \\ &\hat{p}^t = p^t - \bar{p}^t, \qquad \hat{p}^{t-1} = p^{t-1} - \bar{p}^{t-1}, \end{aligned}$$

where $\bar{p}^t$ and $\bar{p}^{t-1}$ are the mean value of density for time t and time t-1 respectively. This calculation in (3) corresponds to the use of the following similarity metric $\rho(p, q)$

$$\begin{aligned} &\rho(p,q) = 1-(p,q)/\sqrt{(p,p)\cdot(q,q)}, \qquad (4)\\ &p = (p(1),..,p(m)), \quad q = (q(1),..,q(m)), \end{aligned}$$

and such a metric will be able to preserve the linear transform of the color brightness, e.g. $I(y) = \alpha I^*(y) + \beta$.

A brute force search for c^t via (3) could be quite inefficient. We hence make use the 2-D-log search method [18] developed under a different setting. The search in the current frame starts from the object centre position of the previous frame, or from the projected centre position estimated from the previous centre moving speed. Each step tests five points in a diamond arrangement and repeats the diamond search in the next step with the centre moved to the best matched point. Nine search points are examined at the last step when the step size is reduced to 1 pixel.

To further reduce the computational load, several strategies can be further considered: 1) Adopt less number of annuli for the object model without decreasing the accuracy significantly; 2) The search for the object centre can be conducted in two, coarse to fine, steps. The first step is to find a suitable candidate location using a smaller m, such as m=3, in (3). The second step is to further check the total difference of two densities between the candidate region and the object template; 3) The object pixels for inspection can be sub-sampled so that only one pixel is selected from every pair of consecutive pixels in space, resulting in a half number of pixels for the model.

4. Object shape verification

Once the centre location is determined, we can proceed to refine the object in the current frame. Most of the work would be to determine if the pixels near the object boundary remain in or outside the object due to the shape deformation during the object motion. Some pixels belonging to the object may have been excluded in the extracted region, and some pixels in the extracted region may no longer be part of the object. The object shape verification here is to examine the pixels near the predicted object boundary, i.e. the previous object boundary projected into the current frame. The area for the shape verification is the area within the distance ρ^v to the candidate object boundary in the current frame, as shown in figure 2, where the area is between Γ^+ and Γ^-, Γ^t is the predicted candidate object border, and Γ^{t-1} denotes the corresponding object border in the previous frame. The search area for an examined pixel is the candidate object region within the distance ρ^s to the examined point, the area marked with the dash lines in figure 2, and the pixel color should be the object template color $I^*(x+d)$ rather than the candidate object color $I(x)$. In other words, if we want to determine whether a pixel $I(y)$ near the candidate boundary actually belongs to the object or not, we compare it with those object pixels $I^*(x+d)$ for $d=c^t- c^{t-1}$ and $||x-y||<\rho^s$. To take into account the neighboring pixels for comparison, we design a neighborhood mask M for selecting the neighboring pixels, see figure 3 for an example. Then the object pixel that has the minimum difference with the examined pixel in term of their respective neighbors can be found within the local search area, and the subsequent difference Δ is obtained for the examined pixel to be considered if it belongs to the object. This difference Δ is expressed as

$$\Delta = \| I(y) - I^*(x_{\min}) \|, \qquad \textbf{(5)}$$
$$x_{\min} = \text{minarg}(\sum_{z \in M} (I(y+z) - I^*(x+z+d)),$$

where the spatial variable z goes over all the selected neighboring locations.

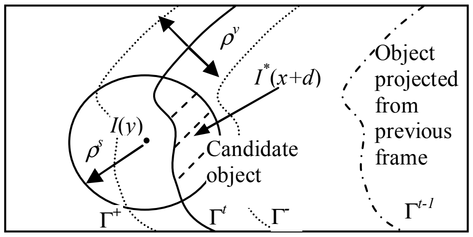


Figure 2. Pixel-wise verification

We note that figure 2 illustrates the verification of pixels on whether they remain on the object. The area on the right of the solid curve is the candidate object region obtained via (3), the area between the round-dotted lines needs to be examined, the slash dash line is the search region for maximum similarity of object pixels, and the dash-dot contour demarcates the object's original location but in the current frame. Finally the verified object areas are extracted via a suitable threshold on the neighborhood similarity along the object boundary.

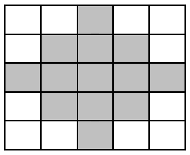


Figure 3. Neighborhood mask.

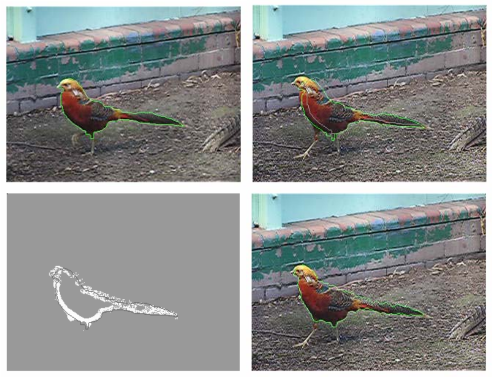
The accuracy of the above verification is affected by the selected parameters ρ^v and ρ^s, as well as the neighborhood mask M. If the object and its surrounding background are well distinguished, then a larger ρ^v is better for the larger object deformation. Otherwise a smaller ρ^v would lead to less interference from the non-object background.

5. Implementation and experiments

The very first thing one needs to decide for implementation, due to (1) and (3), is the choice of the kernel function ψ(·) and the weight function φ(·). For these experiments, we can choose for the kernel function and the weight function the following N-dimensional Gaussian

$$\psi(z) = \frac{1}{(\sqrt{2\pi}\sigma)^N} e^{-\frac{z \cdot z'}{2\sigma^2}}, \qquad \textbf{(6)}$$

or the triangular function $\psi(z)=1-\|z\|$ for $\|z\|\le 1$ and zero otherwise, where z is an N-dimensional row vector and z' is its transpose. The Gaussian is traditionally a popular kernel function in the literature. We note that the function for object template and weight can be quite similar or even the same, and the standard deviation σ should remain the same for both the object template and the candidate region. The object location is thus determined by minimizing the error between the template $p^{t-1}(k)$ and the candidate region $p^t(k)$ via (3).



a	b
c	d

Figure 4. a. Original object. b. Locating object. c. Verifying object. d. Full object extracted.

We now illustrate our experiments on two video sequences. For the first sequence, we take m=8 for the object model, and set the radius r_k =5k+1 pixels for the concentric circles. The density function for the object is obtained via (1). Then we apply the fast search algorithm [18] to locate the new position of the object which has the most similar density as the object model via (3). The result is shown in figure 4 (b), where the brighter boundary indicates the new location while another boundary depicts the original object shape in the previous frame as in figure 4 (a). Next we verify or refine the object shape. The local area to be considered for shape deformation is chosen to be within the distance of ρ^v=8 pixels along the projected boundary for the candidate object. The mask for selected neighbors is a diamond shape of 13 pixels in total as shown in figure 3.

The resulting object contour and the final fully extracted object are shown in figure 4 (c) and (d). The brighter contour in (d) can be regarded as refined or adjusted from the original one in (a). In this shape refinement step, our experiments show that the determination of the pixels on the upper sides of the object is not so effective due to the background being somewhat similar to the upper part of that object. If the object shape deformation is not expected to be large, then the refinement operation can be restricted to a narrower area corresponding to a smaller ρ^v in figure 2. Typical frames with extracted object contours are also depicted in figure 5 below, where the contours are again shown in a bright color.

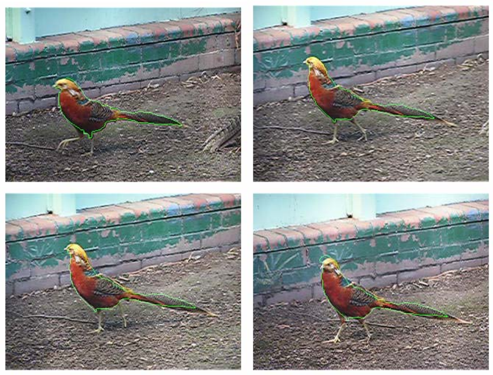
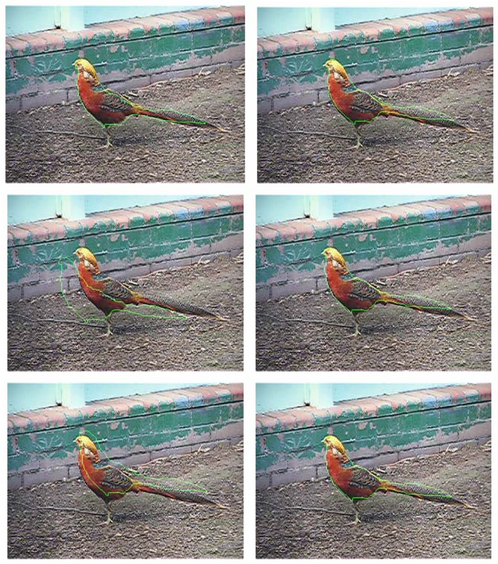


Figure 5. Some tracked movement

There can be various different similarity or distance metrics, like that in (4), which can potentially affect the tracking performance. We thus conduct in figure 6 a comparison by different similarity measures. Figure 6 (a), (c), (e) are based on the standard Euclidean distance and figure 6 (b), (d), (f) are based on our proposed measure in (4) or (3). In figure 6 (a) and (b), both distance measures lead to similarly correct object location in the subsequent frame. However, when the frames are preprocessed to reflect the potential illumination changes by undergoing a certain degree of linear transformation, the results due to the different measures are hugely different as in figure 6 (c) and (d), where latter corresponds to the metric in (4). The experiments showed that the similarity measure (4) can still locate the object properly in (d) whereas the Euclidean distance has led to a significant deviation from its correct position as shown in (c). Further experiments are even conducted on some simple nonlinear transformations. In fact figure 6 (e) and (f), also for the Euclidean distance and our proposed similarity metric (4) respectively, illustrate the tracking results when the color has undergone a quadratic transformation. The latter again holds out well as seen in figure 6 (f) while the former falters as expected. This empirical result for the nonlinear transformation is somehow unexpected, and may be somewhat accidental for this case. In any case, the similarity metric in (4) is more robust at handling the tracking, especially when there are lighting changes or the like in the scene.



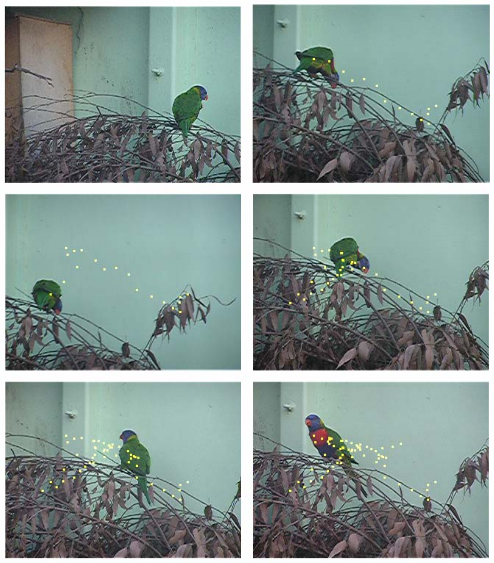
a	b
c	d
e	f

Figure 6. Comparison on locating the object with two different similarity measures.

For a different setting, we conducted another experiment on a parrot video sequence, and the results are summarized in figure 7. To highlight the object traces and avoid the cluttering of the contours, we only plot with a bright dot the centre of the object parrot. The video sequence starts from the frame in figure 7 (a), the bright heavy dots will each denote the object centre of a particular frame. Figure 7 (b), (c), (d) and (e) thus depict the traces of the parrot across a consecutive sequence of frames. In creating the last picture (f) of figure 7, the parrot template has actually been adjusted during the frame by frame object tracking since the appearance of the parrot has undergone relatively larger changes.

6. Conclusion

We proposed an object representation by a kernel-based color probability density function over a collection of concentric annuli at the object centre. Each value of the density function corresponds to the pixels in a specific annulus as a single entity. Although the object representation through features on annuli can be extended to more than one patches of stable color appearance, the rotational invariance of the annuli in the current model simplifies both the representation and the computation. A more noise-tolerant similarity metric has been shown to perform more robustly compared with the traditional Euclidean distance, when determining the similarity between two object representations. The proposed tracking scheme has been shown to be applicable to video sequences containing the motion of a non-rigid object within a non-stationary background. The method is both effective and efficient, due to the associated succinct modeling, in dealing with objects of deforming shapes, and the last verification step further refines the shape of the tracked object in the newer frames.



a	b
c	d
e	f

Figure 7. Parrot movement and the traces of its centre.

References

[1]. Jurie, F. and Dhome, M.: Hyperplane Approximation for Template Matching, IEEE Trans. Patten Analysis and Machine Intelligence, Vol. 24, No. 7, (2002) 996-1000.

[2]. Jain, A.K., Zhong, Y. and Lakshmanan, L.: Object Matching Using Deformable Templates, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 18, No. 3, (1996) 267-278.

[3]. Shen, D. and Davatzikos, C.: An Adaptive-Focus Deformable Model Using Statistical and Geometric Information, IEEE Trans. Pattern Analysis and Machine Intelligent, Vol.22(8), (2000) 906-913.

[4]. Frey, B.: Filling in Scenes by Propagating Probabilities through Layers into Appearance Models, Pro. IEEE Conf. Computer Vision and Pattern Recognition, Vol.1, (2000)185-192.

[5]. Jojie, N. and Frey, B.J.: Learning Flexible Sprites in Video Layers, Proc. IEEE Conf. Computer Vision and Pattern Recognition, Vol. 1, (2001) 199-206.

[6]. Black M.J and Jepson, A.D.,: Eigen Tracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation, Int'l J. Computer Vison, Vol. 26, No. 1, (1998) 63-84.

[7]. Birchfield, S.: Elliptical Hlliptical Head Tracking Using Intensity Gradients and Color Histograms, Proc. IEEE Conf. Computer Vision and Pattern Recognition, (1998) 232-237.

[8]. Hager, G.D. and Belhumeur, P.N.: Efficient Region Tracking with Parametric Models of Geometry and Illumination, IEEE Trans. Pattern Analysis and Machine Intelligent, Vol. 20, No. 10, (1998) 1025-1039.

[9]. Sclaroff, S. and Isidoro, J.: Actice Blobs, Proc. Sixth Int'l Conf. Computer Vision, (1998) 1146-1153.

[10]. Jepson, A.D, Fleet, D.J. and EI-Maraghi, T.F.: Robust Online Appearance Models for Visual Tracking, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 25, No. 10, (2003) 1296-1311.

[11]. Huang, Z. Q. and Jiang, Z,: Target Positioning with Dominant Feature Elements, The 12th International Conf. on Computer Analysis of Image and Patterns, CAIP2007, LNCS 4673, (2007) 69-76.

[12]. Sheikh , Y. and Shah, M.: Bayesian Modeling of Dynamic Scenes for Object Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.27, No. 11, (2005) 1778-1792.

[13]. Elgammal, A. Duraiswami, R. Davis, L.S.: Efficient Kernel Density Estimation Using the Fast Gauss Transform with Applications to Color Modeling and Tracking, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 25, No. 11, (2003) 1499-1504.

[14]. Comaniciu, D. Ramesh, V. and Meer, P.: Kernel-Based Object Tracking, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 25, No. 5, (2003) 564-577.

[15]. Comaniciu, D. and Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 24, No. 5, (2002) 603-619.

[16]. Avidan, S: Support Vector Tracking, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 26, No. 8, (2004) 1064-1072.

[17]. Williams, O., Blake, A. and Cipolla, R.: Sparse Bayesian Learning for Efficient Visual Tracking, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.27, No. 8, (2005) 1292-1304.

[18]. Jain, J. R., and Jain. A. K.: Displacement measurement and its application in interframe image coding, IEEE Trans. Commun. (1981), COM-29: 1799-1808.