Chapter 18
Incremental Level Set Tracking

Shay Dekel, Nir Sochen, and Shai Avidan

Abstract We consider the problem of contour tracking in the level set framework. Level set methods rely on low level image features, and very mild assumptions on the shape of the object to be tracked. To improve their robustness to noise and occlusion, one might consider adding shape priors that give additional weight to contours that are more likely than others. This works well in practice, but assumes that the class of object to be tracked is known in advance so that the proper shape prior is learned. In this work we propose to learn the shape priors on the fly. That is, during tracking we learn an eigenspace of the shape contour and use it to detect and handle occlusions and noise. Experiments on a number of sequences reveal the advantages of our method.

18.1 Introduction

Contour tracking is a task in which the contour of the object(s) of interest has to be extracted for each frame in a video in a way which is robust to noise and clutter. This task is different from object tracing in which a bounding box that contains the object is sought or segmentation in which the contour in a given image is extracted. The main issues that we address in this paper is the robustness to noise, clutter and occlusions on one hand and the ability to deal with shape/behavior change. Accordingly, this paper introduces a way to achieve such a robustness as a matter of principle without paying too much attention to the quality of segmentation. The way it is done is via variational and level-set methods.

S. Dekel (✉) · S. Avidan
Department of Electrical Engineering, Tel Aviv University, Tel Aviv, Israel

N. Sochen
Department of Applied Mathematics, Tel Aviv University, Tel Aviv, Israel
e-mail: sochen@math.tau.ac.il

M. Breuß et al. (eds.), *Innovations for Shape Analysis*, Mathematics and Visualization,
DOI 10.1007/978-3-642-34141-0_18, © Springer-Verlag Berlin Heidelberg 2013

Level set methods are a convenient way to parameterize and track object contours. They work by evolving a contour which tightly encloses the deformable object to be tracked. However this method cannot handle missing or misleading information due to noise, clutter or occlusions in the input images. In order to overcome these problems one can derive a parametric model for implicit representation of the segmentation curve by applying low dimensional subspace representation, such as Principle Component Analysis (PCA) to a specific collection of training set images before the tracking begins. In this case the evolving curve has limited degrees of freedom since the curve lies on the PCA subspace. This property enables the segmentation to be robust to noise and partial occlusions. However this model relies on a fixed training set and assumes that the object class to be tracked is known in advance.

We present an extension that deals with these shortcomings. Our approach learns, on line, a shape prior that is then used during tracking. This enables our tracker to overcome occlusions and, unlike previous algorithms, it does not demand a specific predefined training set.

We formulate the tracking process as a mixture of two major models: In the **On-line Learning Model**, we perform region-based segmentation on each frame using the Chan-Vese approach [3] together with the edge-based approach which is based on the Geodesic Active Contour [2]. The segmentation results are then used to incrementally learn an on-line low dimensional subspace representation of the objects' contour, efficiently adapting to changes in the appearance of the target.

In case of occlusion, the eigencoordinates of the segmented shape will differ considerably from those obtained so far and, in that case, we switch to the **PCA Representation Model** that tracks the object using the currently available PCA space. This PCA eigenbase representation together with the temporal prior representation allows us to limit the degree of freedom of the evolving contour which enables it to cope with missing or misleading information due to occlusions, partial occlusions and noise. Once the object reappears we switch back to the online learning model and keep updating our representation model. Hence, we can properly track deformable objects through occlusion and noise.

We provide experimental results that present several properties of our method: We show that our method can cope with partial or total occlusions, as well as examples in which the images are severely contaminated with strong Gaussian noise. In addition, we show that our algorithm can adapt to considerable deformations in shape.

18.2 Background

Contour tracking via variational methods and level-sets is based on the seminal works [1, 3, 18] and many more; see [6] for a very nice overview and for further references on level-set based tracking.

Several authors combined prior shape information into level-set-based segmentation. Leventon et al. [11] incorporated a training set information as a prior model to restrict the flow of the geodesic active contour using Principle Component Analysis (PCA). Tsai et al. [25] used only the first few eigenmodes by performing optimization. Rousson et al. [20, 21] introduced shape information on the variational level. Chen et al. [4] imposed shape constraints directly on the contour. However these authors ignored the temporal coherence of the shapes which leads to degredation in performance when dealing with occlusions.

Cremers [5] proposes to model the embedding functions by a Principle Component Analysis (PCA) and to use dynamical shape prior. He learns a specific set of training shapes before the tracking begins and also exploits the temporal correlations between consecutive shapes. This enables him to handle occlusions and large amounts of noise. His method is well-suited for specific tracking missions where a pre-defined training set can be performed off-line.

Another approach is presented in the work of Fussenegger et al. [7]. In that work a level-set method is combined with PCA decomposition of shape space. This is very similar and relevant to this paper. The difference is in the aim and type of video treated. In Fussengger et al. there are many, mainly rigid, objects to segment such that each individual shape doesn't change too much from frame to frame. Our paper deals with mainly one object with **changing shape** and the main focus is on the way to deal with occlusions and changes of shape along the video.

Our work is motivated in part by the power of subspace representation and exploits the temporal correlations between consecutive shapes following the work of Cremers [5]. But in contrast to eigentracking algorithms, our algorithm does not require a specific training set before tracking begins. It learns the eigenbase on-line during the object tracking process, thus eliminating the need to collect the training images prior to tracking.

18.2.1 Integrated Active Contours

We start with a generic algorithm for data-based segmentation. The model is formulated in variational way and we use the Integrated Active Contour model [19, 22] that combines region-based and edge-based segmentation via the level-set formulation. In order to perform region-based segmentation in each frame we use the Chan-Vese algorithm which attempts to partition the image into two regions according to common image properties. Then we add to the functional an edge-based term which is based on the Geodesic Active Contour (GAC).

Let $I_t : \Omega \to \mathbb{R}$ be the image at time t that assigns for each pixel $x \in \Omega \subset \mathbb{R}^2$ a real value grey level. A contour that separates the object (or objects) from the background is encoded as a zero level-set of a function $\phi_t : \Omega \to \mathbb{R}$. The contour at frame t is $C_t = \{(x, y)|\phi_t(x, y) = 0\}$. We denote the region inside the zero-level set by $\Omega_+ = \{(x, y)|\phi_t(x, y) > 0\}$ and similarly the region outside the zero level-set $\Omega_- = \{(x, y)|\phi_t(x, y) < 0\}$. The probability of the contour ϕ_t, given the previous contours and all the measurements $[I_0(\mathbf{x}) \dots I_t(\mathbf{x})]$ using the Bayes rule is:

$$\mathscr{P}(\phi_t|I_{0:t}, \phi_{0:t-1}) \propto \underbrace{\mathscr{P}_+(I_t|\phi_t)}_{Target} \cdot \underbrace{\mathscr{P}_-(I_t|\phi_t)}_{Background} \cdot \underbrace{\mathscr{P}(\phi_{0:t-1}|\phi_t)}_{Shape} \cdot \underbrace{\mathscr{P}(\phi_t)}_{Smoothness} \tag{18.1}$$

Here $P_\pm$ are the probability distributions of the grey value intensities inside and outside of the zero level-set of ϕ_t.

While $P_\pm$ can be quite involved in real-life applications we choose here to stick to the simple Gaussian model in order to concentrate on the tracking part. This choice leads to the Chan-Vese model. In this approach we find a contour, represented by $\phi(x)$, that partitions the image into two regions Ω_+ and Ω_-, that describe an optimal piecewise constant approximation of the image. We also assume that the intensities of the shape and the background are independent samples from two Gaussian probabilities, therefore:

$$\mathscr{P}_\pm(I_t|\phi_t) = \prod_{x \in \Omega_\pm} \frac{1}{\sqrt{2\Pi\sigma_\pm^2}} \exp^{-\frac{(I_t - C_\pm)^2}{2\sigma_\pm^2}} \tag{18.2}$$

Thereby $C_\pm$ and $\sigma_\pm$ are the mean and standard deviation of the intensities inside and outside of the zero level-set of ϕ_t.

The region-based energy is defined as

$$E_{RB}(\phi_t, C_+, C_-) = -\log\{\mathscr{P}_+(I_t|\phi_t) \cdot \mathscr{P}_-(I_t|\phi_t)\} \tag{18.3}$$

The contour with the highest probability is the one that minimizes the following region-based energy functional:

$$\begin{aligned} E_{RB}(\phi_t, C_+, C_-) &= \int_\Omega \{\frac{1}{2\sigma_+^2}(I_t(x) - C_+)^2 + \frac{1}{2}\log(2\pi\sigma_+^2)\}H(\phi_t(x))dx \\ &+ \int_\Omega \{\frac{1}{2\sigma_-^2}(I_t(x) - C_-)^2 + \frac{1}{2}\log(2\pi\sigma_-^2)\}\{1 - H(\phi_t(x))\}dx \end{aligned} \tag{18.4}$$

where $H(\phi_t(x))$ is the Heaviside step function:

$$H(\phi_t(x)) = \begin{cases} 1 \text{ if } \phi_t(x) > 0 \\ 0 \text{ if } \phi_t(x) \leqslant 0 \end{cases} \tag{18.5}$$

The smoothness prior is taken to be the Geodesic Active Contour (GAC) term [2]. This term defines the object boundaries as a (locally) minimal length weighted by the local gradients. In other words it is a geodesic over a Riemannian manifold whose metric is defined via the gradients of the image. It leads to the following functional:

$$E_{GAC}(\phi_t) = \int_{\Omega} g_{GAC}(|\nabla I|)|\nabla H(\phi_t(x))|dx \tag{18.6}$$

where $g_{GAC} = 1/(1 + |\nabla I|^2)$.

Finally, the Integrated Active Contour functional E_{IAC} is obtained by the summation of the region-based energy E_{RB} (18.4) and the edge-based geodesic active contour energy E_{GAC} (18.6) as:

$$E_{IAC}(\phi_t, C_+, C_-) = E_{RB}(\phi_t, C_+, C_-) + \mu E_{GAC}(\phi_t) \tag{18.7}$$

with the Euler-Lagrange equation:

$$\frac{\delta E_{IAC}}{\delta \phi_t} = \Big[\log(\frac{\sigma_+}{\sigma_-}) + \frac{1}{2\sigma_+^2}(I_t(x) - C_+)^2 \\ -\frac{1}{2\sigma_-^2}(I_t(x) - C_-)^2 - \mu \mathrm{div}(g_{GAC}(|\nabla I|)\frac{\nabla \phi_t(x)}{|\nabla \phi_t(x)|})\Big]\delta(\phi_t(x)) \tag{18.8}$$

18.2.2 Building the PCA Eigenbase

The shape term, as explained earlier, is an on-line learning model that produces and updates an eigenbase representation during tracking. We work here with a PCA decomposition, where we first build a PCA eigenbase from the first n frames of the sequence and then incrementally update it as new m observations arrive. For efficiency we use an incremental PCA algorithm and only keep the top k eigenvalues. The corresponding eigenvectors are denoted by ψ_i. This PCA eigenbase, ψ_i, will help us cope with occlusions in the PCA representation model. Each shape is represented as:

$$\phi_i(x) = \bar{\phi}_0(x) + \sum_{j=1}^{k} \alpha_{ij} \psi_j \tag{18.9}$$

where $\phi_i(x)$ represents the i-th shape from the PCA subspace model, $\bar{\phi}_0(x)$ is the mean shape and α_{ij} is the PCA coefficient of the i-th shape.

18.2.2.1 First PCA Eigenbase

We produce the first PCA eigenbase from the previous n segmentation results of the On-Line Learning model. Let $A = \{\phi_1(x), \phi_2(x) \dots \phi_n(x)\}$ be the previous n level set function segmentations. Each data element, $\phi_i(x)$, is a $d \times 1$ vector that contains the level set function of the i-th shape. We calculate the mean shape $\phi_A(x)$ as:

$$\bar{\phi}_A(x) = \frac{1}{n}\sum_{i=1}^{n}\phi_i(x) \tag{18.10}$$

Then we apply singular value decomposition (SVD) on the centered n previous level set functions

$$U_A \Sigma_A V_A^T = SVD\{\phi_1 - \bar{\phi}_A, \phi_2 - \bar{\phi}_A, \ldots, \phi_n - \bar{\phi}_A\} \tag{18.11}$$

Here U_A and Σ_A contain the eigenvectors and the eigenvalues, respectively. Then the first PCA eigenbase denoted as $\psi_A(x) = U_A^{(1:k)}$ contains the eigenvectors corresponding to the k largest eigenvalues, i.e. vectors of U_A. These terms $\psi_A(x)$ and Σ_A will serve as initialization to the incremental PCA algorithm.

18.2.2.2 Updating the Eigenspace

Incremental PCA combines the current eigenbase with new observations without recalculating the entire SVD. Numerous algorithms have been developed to efficiently update an eigenbase as more data arrive [8, 9]. However, most methods assume a fixed mean when updating the eigenbase. We use the Sequential Karhunen-Loeve (SKL) algorithm of Levy and Lindenbaum [12]. They present an efficient method that incrementally updates the eigenbase as well as the mean when new observations arrive. They also add a forgetting factor $f \in [0, 1]$ that down-weights the contribution of the earlier observations. This property plays an important role in the on-line learning. As time progresses the observation history can become very large and the object may change its appearance, and the forgetting factor allows us to strengthen the contribution of the current data such that the updated PCA eigenbase will be able to cope with that change. This algorithm allows us to update the PCA eigenbase online while tracking using the segmentation results during the on-line learning model.

18.2.2.3 Detection of Occlusion

In the on-line learning model we performed region-based segmentation in each frame and incrementally updated the PCA eigenbase. We want to know when the current contour encounters an occlusion in order to switch to the PCA representation model. Then, after the occlusion ends we switch back to the on-line learning model to keep updating our representation model.

For this purpose, we rely on the PCA coefficients that represent the current shape and observe that under occlusions these coefficients are farther away from the mean PCA coefficients.

During the on-line learning model we project each contour segmentation ϕ_t on the current PCA subspace $\Psi(x)$ to obtain its current PCA coefficient α_t. Then we measure the Mahalanobis distance between the current PCA coefficient α_t and the mean PCA coefficient $\bar{\alpha}$:

$$D_t(\alpha_t) = \sqrt{(\alpha_t - \bar{\alpha})^T S^{-1}(\alpha_t - \bar{\alpha})} \tag{18.12}$$

Here $\bar{\alpha}$ is the mean PCA coefficient and S is the covariance matrix. These two terms were obtained by collecting only the good PCA coefficients every frame during the on-line learning model. From (18.12), if $D_t(\alpha_t) > Th$ our method switches to the PCA representation model and when $D_t(\alpha_t) < Th$ we return back to the on-line learning model.

Experimental results show that if the scene is free of occlusions, the Mahalanobis distance is usually $D_t(\alpha_t) < Th$. Figure 18.1 shows the Mahalanobis distance in each frame during the two models. We can see that during the on-line learning model the contour encounters an occlusion and its appropriate Mahalanobis distance is above Th (the peaks in the blue bars in the middle), in that moment we switch to the PCA representation model and see the improvement in the Mahalanobis distances during the occlusion (the red bars in the middle). Then we switch back to the on-line learning model when the coefficients are below Th.

18.2.3 Dynamical Statistical Shape Model

Once the algorithm detects an occlusion, it switches to the PCA representation model with the updated PCA eigenbase. But before we switch to the PCA representation model, we want to exploit the temporal correlations between the shapes. As explained in Eq. (18.9) we can represent each shape using the PCA eigenbase and the mean shape. Therefore the segmentation in the current frame $\phi_t(x)$ can be represented by the PCA coefficient vector α_t. This will lead us to write the probability of the shape prior from (18.1) as: $\mathscr{P}(\alpha_t|\alpha_{0:t-1})$ instead of $\mathscr{P}(\phi_t|\phi_{0:t-1})$. In addition during the on-line learning model, we ignored the correlation between the frames since we assumed that the object wasn't occluded and therefore no temporal prior information was needed. In the PCA representation model, where the deformable object may be occluded, we have to obtain more powerful prior which relies on the correlation between consecutive frames. For this reason we can represent each shape by a Markov chain of order q in a manner similar to Cremers [5]. More formally, the current shape at time t can be represented by the previous shapes using an Auto Regressive (AR) model as follows:

$$\alpha_t = \vartheta + A_1\alpha_{t-1} + A_2\alpha_{t-2} \ldots + A_q\alpha_{t-q} + \eta \tag{18.13}$$

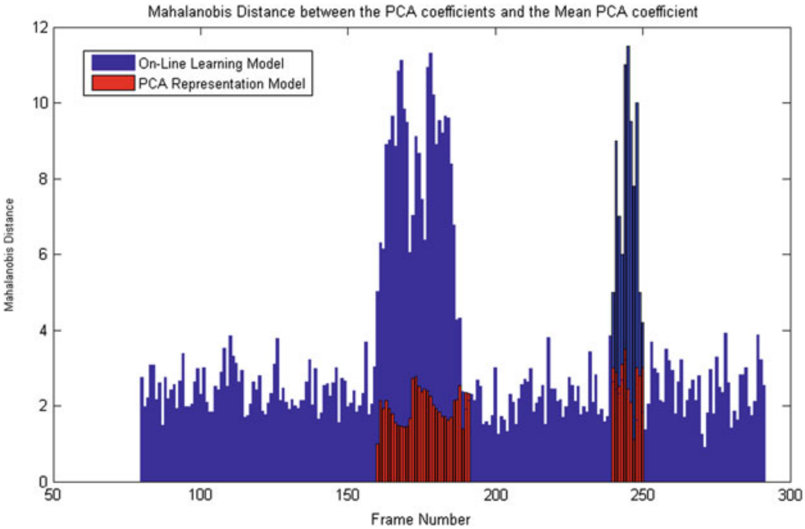

Fig. 18.1 The Mahalanobis distance between the current PCA coefficient α_t and the mean PCA coefficient $\bar{\alpha}$ in each frame during the two models on-line learning model (*blue*) and PCA representation model (*red*)

Here η is Gaussian noise with zero mean and covariance matrix Λ, A_i are the transition matrices of the AR model. and ϑ is the mean of the process. With this AR model we can determine the probability $\mathscr{P}(\alpha_t|\alpha_{0:t-1})$ for observing a particular shape α_t at time t given the shapes estimated on the previous frames as follows:

$$\mathscr{P}(\alpha_t|\alpha_{0:t-1}) \propto \exp(-\frac{1}{2}\omega^T\Lambda^{-1}\omega) \tag{18.14}$$

Where:

$$\omega = \alpha_t - \vartheta - A_1\alpha_{t-1} - A_2\alpha_{t-2}\ldots - A_q\alpha_{t-q} \tag{18.15}$$

Various methods have been proposed in the literature to estimate the model parameters: η, Λ and A_i. We applied a Stepwise Least Squares algorithm as proposed in [16]. The order q determines the accuracy of the AR model approximation and its value depends on the input sequence. In order to estimate its value we use the Schwarz Bayesian Criterion [23].

18.3 PCA Representation Model

The algorithm switches to that model only if it detects an occlusion. Once detecting an occlusion, it continually tracks the same target using the PCA eigenbase and the AR parameters which were obtained in the on-line learning model. As explained in Sect. 18.2.2, according to (18.9) the segmentation in the current frame $\phi_t(x)$ can be represented by the PCA coefficient vector α_t. Therefore we actually exchange in (18.1) each level set function ϕ by the appropriate shape vector representation α:

$$\mathscr{P}(\alpha_t|I_{0:t},\alpha_{0:t-1}) \propto \underbrace{\mathscr{P}_+(I_t|\alpha_t)}_{\textit{Target}} \cdot \underbrace{\mathscr{P}_-(I_t|\alpha_t)}_{\textit{Background}} \cdot \underbrace{\mathscr{P}(\alpha_t|\alpha_{0:t-1})}_{\textit{Shape Prior}} \tag{18.16}$$

This will lead us in this model to focus on estimating the shape vector representation α_t by minimizing the following energy functional:

$$\begin{aligned} E_{PCA}(\alpha_t, C_+, C_-) &= -log(\mathscr{P}(\alpha_t|I_{0:t},\alpha_{0:t-1})) \\ &= \underbrace{-\log \mathscr{P}_+(I_t|\alpha_t) \cdot \mathscr{P}_-(I_t|\alpha_t)}_{\textit{Data}} - \log \underbrace{\mathscr{P}(\alpha_t|\alpha_{0:t-1})}_{\textit{Shape Prior}} \end{aligned} \tag{18.17}$$

Thereby the probabilities $\mathscr{P}_\pm(I_t|\alpha_t)$ are similar to the Chan-Vese probabilities (18.2), except that $\phi_t(x)$ is determined by α_t as (18.9).

Applying these probabilities and (18.14)–(18.17) leads to the following energy functional:

$$\begin{aligned} E_{PCA}(\alpha_t, C_+, C_-) = \frac{1}{2\sigma_+^2}\int_\Omega (I_t(x) - C_+)^2 H(\phi(\alpha_t))dx + \frac{1}{2}\log(2\pi\sigma_+^2)\int_\Omega H(\phi(\alpha_t))dx \\ + \frac{1}{2\sigma_-^2}\int_\Omega (I_t(x) - C_-)^2(1 - H(\phi(\alpha_t)))dx + \frac{1}{2}\log(2\pi\sigma_-^2)\int_\Omega (1 - H(\phi(\alpha_t)))dx + \lambda\frac{1}{2}\omega^T\Lambda^{-1}\omega \end{aligned} \tag{18.18}$$

Here λ is an additional parameter that allows relative weighting between data and prior and $\phi(\alpha_t)$ is the level set estimation which is determined by α_t. The segmentation in each frame requires the estimation of the shape vector α_t, which is done by minimizing (18.18) with respect to α_t using gradient descent strategy.

18.4 Motion Estimation

In each frame we estimate the translation positions $(u, v)_t$ and use this to translate the previous contour ϕ_{t-1} as initialization to estimate the current contour. In the On-Line Learning model the segmentations aren't sensitive to the initial contour since we assume that the object isn't occluded. Therefore we use Lucas-Kanade approach [13] to estimate the translation position $(u, v)_t$ by minimizing

$$E_{LK}(u, v)_t = K_W * (I_x u + I_y v + I_t)^2 \tag{18.19}$$

where $K_W * (\cdot)$ denotes the convolution with an integration window of size W. I_x, I_y are the x, y derivatives of the image in each axis and I_t is the derivative

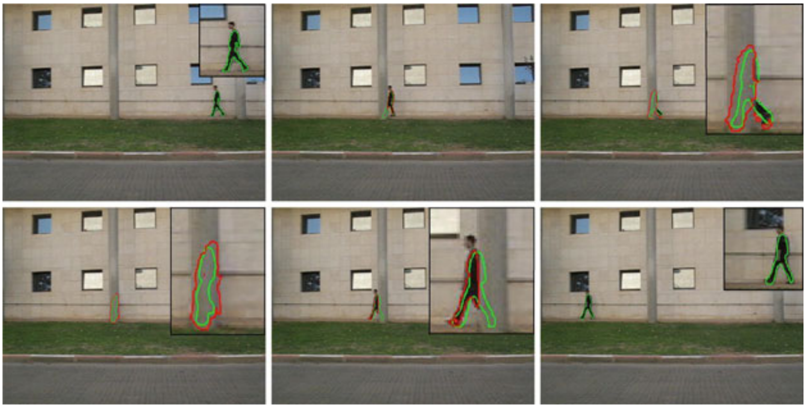

Fig. 18.2 Comparison between our algorithm (*Green*) and Chan-Vese (*Red*) on walking man sequence (319 frames) with full occlusion. In the on-line learning model (frame 22) and in the PCA representation model (frames: 153,161,165,176,266). As can be seen, the Chan-Vese model cannot handle the case of occlusion

between two consecutive frames. In the on-line learning model we also learn the temporal translations between consecutive frames to build a motion prior. This is done by collecting all the translations seen so far and build it into a AR model in the same way as we build the shape prior (18.13)

$$\begin{bmatrix} u \\ v \end{bmatrix}_t = \begin{bmatrix} \bar{u} \\ \bar{v} \end{bmatrix} + B_1 \begin{bmatrix} u \\ v \end{bmatrix}_{t-1} + B_2 \begin{bmatrix} u \\ v \end{bmatrix}_{t-2} + \ldots B_q \begin{bmatrix} u \\ v \end{bmatrix}_{t-q} + \eta_{pos}$$

Here η_{pos} is Gaussian noise with zero mean with covariance matrix Λ_{pos}, B_i are the transitions of the AR model, and $(\bar{u}, \bar{v})$ are the mean values of u and v.

In the PCA representation model, when the object is occluded we use the learned AR motion parameters B_i, Λ_{pos} and $(\bar{u}, \bar{v})$ to estimate u and v in each frame, (u_p, v_p), as a prior and combine this to the LK functional (18.19):

$$E_{LK-PRIOR}(u, v)_t = E_{LK}(u, v) + \xi((u - u_p)^2 + (v - v_p)^2) \tag{18.20}$$

This addition prevents the estimation of (u, v) during occlusion from being too far from their prior estimations.

18.5 Results

We tested our algorithm on different sequences with a deformable shape that are partially or fully occluded.

In each example the on-line learning model provides the contour based segmentations of the deformable shape and incrementally constructs the PCA eigenbase.

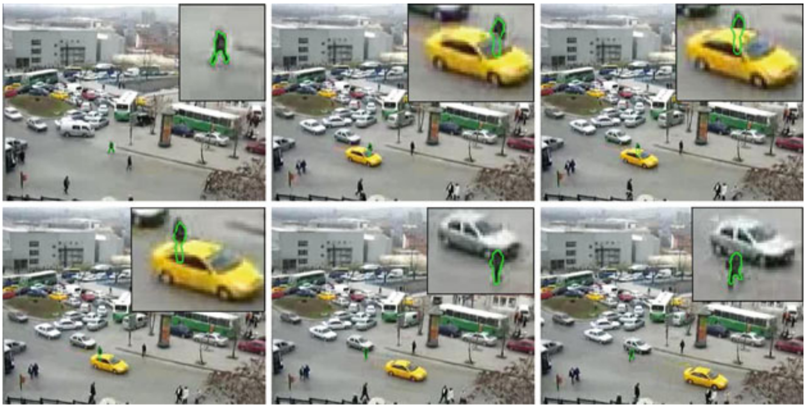

Fig. 18.3 Results of our algorithm on walking man sequence (200 frames) with partial traffic occlusions (*yellow* and *silver car*). The algorithm automatically switches from on-line learning (frame 60) to PCA representation as soon as occlusion is detected (frames: 11,66,68,70,77,84)

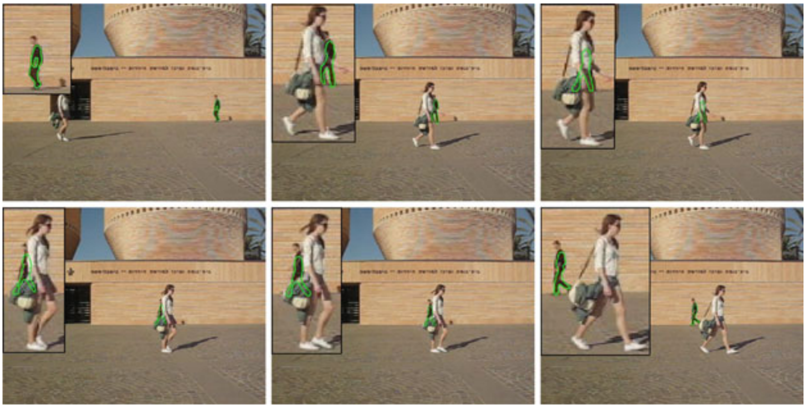

Fig. 18.4 Results of our algorithm on walking man sequence (173 frames) with full occlusion (walking woman hides the target). The algorithm automatically switches from on-line learning (frame 95) to PCA representation as soon as occlusion is detected (frames: 35,100,101, 103,104,109)

When it detects an occlusion, it estimates the AR parameters that capture the temporal dynamics of the shapes evolution seen so far and switches to the PCA representation model. The PCA model uses the current PCA eigenbase and the estimated AR prior parameters to keep segmenting the deformable shape during occlusion. Finally, when the target reappears it switches back to the on-line learning model and keeps tracking the target and updating the PCA eigenbase. We can see that it maintains the appropriate contours when the shape is totally or partially occluded.

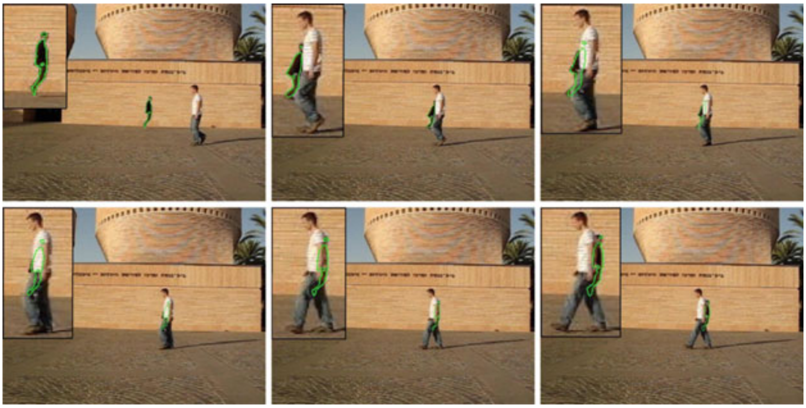

Fig. 18.5 Results of our algorithm on jumping man sequence (225 frames) with full occlusion (walking man hides the target). Frames: 103,120,121,122,124,125

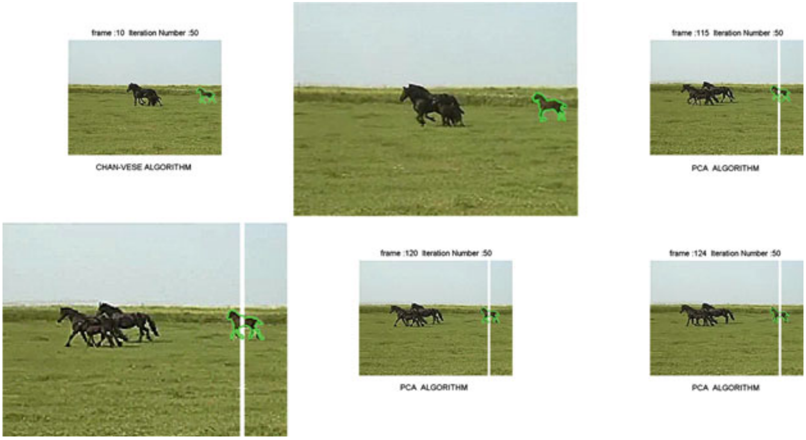
frame :10 Iteration Number :50
CHAN-VESE ALGORITHM
frame :115 Iteration Number :50
PCA ALGORITHM
frame :120 Iteration Number :50
PCA ALGORITHM
frame :124 Iteration Number :50
PCA ALGORITHM

Fig. 18.6 Results of our algorithm on running horse sequence (290 frames) with one partially long period (synthetic) occlusion (20 frames with white label hides partially the target). We showed that our method was able to remain locked onto the target and illustrate the appropriate contours. Frames: 10,89,115,119,120,124

First, we compared our method to a stand-alone Chan-Vese algorithm on a sequence of walking man with one occlusion (left column). As can be seen in Fig. 18.2, the Chan-Vese model could not handle the occlusion properly, while our method kept tracking the person through the entire sequence and was able to illustrate the appropriate shapes when the man was totally occluded by the left column (Figs. 18.3–18.6).

Finally, in Fig. 18.7 we examined our method on a noisy sequence of a jumping man with additive Gaussian noise (SNR =15), and, as can be seen, our algorithm is able to cope with Gaussian noise and occlusion as well.

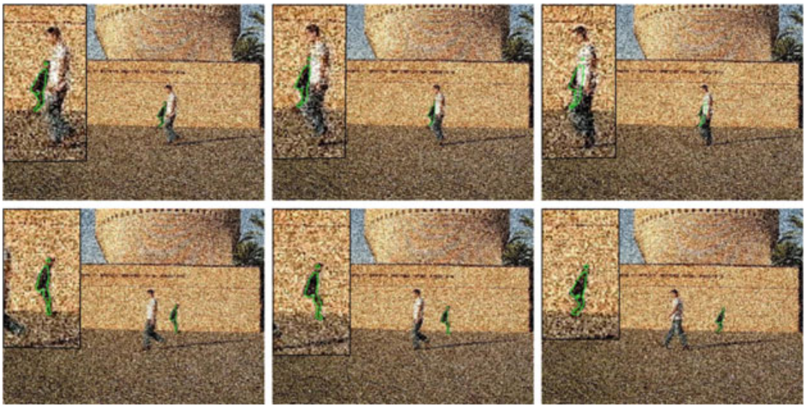

Fig. 18.7 Results of our algorithm on jumping man sequence with additive Gassuan noise (SNR = 15) and full occlusion (walking man hides the target). Frames: 119,120,121,132,134,142

18.6 Conclusions

We have extended level-set tracking to learn an eigenbase on the fly. This was then used to handle occlusions by switching from a Chan-Vese-based algorithm to a PCA-based representation that is more robust to noise and occlusions. In addition, we have shown that the proposed incremental level-set tracking can adjust to changes in the appearance of the object. This results in a robust tracker that can handle never-seen-before objects and deal with partial or full occlusions and noise.

References

1. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proceeding of IEEE International Conference on Computer Vision, Boston, USA, pp. 694–699 (1995)
2. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vis. **22**(1), 61–79 (1997)
3. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Trans. Image Process. **10**(2), 266–277 (2001)
4. Chen, Y., Tagare, H., Thiruvenkadam, S., Huang, F., Wilson, D., Gopinath, K.S., Briggs, R.W., Geiser, E.: Using shape priors in geometric active contours in a variational framework. Int. J. Comput. Vis. **50**(3), 315–328 (2002)
5. Cremers, D.: Dynamical statistical shape priors for level set based tracking. IEEE Trans. Pattern Anal. Mach. Intell. **28**(8), 1262–1273 (2006)
6. Cremers, D., Rousson, M., Deriche, R.: A Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion and Shape, IJCV (2007)
7. Fussenegger, M., Roth, P., Bischof, H., Deriche, R., Pinz, A.: A level set framework using a new incremental, robust active shape model for object segmentation and tracking. Image Vis. Comput. **27**, 1157–1168 (2009)

8. Golub, G.H., Van Loan, C.F.: Matrix Computations. The Johns Hopkins University Press, Baltimore (1996)
9. Hall, P., Marshall, D., Martin, R.: Incremental eigenanalysis for classification. In: Proceedings of British Machine Vision Conference, pp. 286–295 (1998)
10. Kichenassamy, S., Kunar, A., Olver, P.J., Tannenbaum, A., Yezzi, A.J.: Gradient flows and geometric active contour model. In: IEEE International Conference on Computer Vision, pp. 810–815 (1995)
11. Leventon, M., Grimson, W., Faugeras, O.: Statistical shape influence in geodesic active contours. In: CVPR, vol. 1, Hilton Head Island, SC, pp. 316–323 (2000)
12. Levy, A., Lindenbaum, M.: Sequential Karhunen-Loeve basis extraction and its application to images. IEEE Trans. Image Process. **9**(8), 1371–1374 (2000)
13. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
14. Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: a level set approach. IEEE Trans. Pattern Anal. Mach. Intell. **17**(2), 158–175 (1995)
15. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Commun. Pure Appl. Math. **42**, 577–685 (1989)
16. Neumaier, A., Schneider, T.: Estimation of parameters and eigenmodes of multivariate autoregressive models. ACM Trans. Math. Softw. **27**(1), 27–57 (2001)
17. Osher, S.J., Sethian, J.A.: Fronts propagation with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**, 12–49 (1988)
18. Paragios, N., Derichle, R.: Geodestic active regions and level set methods for supervised texture segmentation. Int. J. Comput. Vis. **46**(3), 223–247 (2002)
19. Paragios, N., Deriche, R.: Geodesic active regions and level set methods for motion estimation and tracking. Comput. Vis. Image Underst. **97**(3), 259–282 (2005)
20. Rousson, M., Paragios, N.: Shape priors for level set representations. In: Heyden, A. et al. (eds.) European Conference on Computer Vision. Volume 2351 of Lecture Notes in Computer Science, pp. 78–92. Springer, Berlin (2002)
21. Rousson, M., Paragios, N., Deriche, R.: Implicit active shape models for 3d segmentation in MRI imaging. In: MICCAI, pp. 209–216 (2004)
22. Sagiv, C., Sochen, N., Zeevi, Y.Y.: Integrated active contours for texture segmentation. IEEE Trans. Image Process. **1**(1), 1–19 (2006)
23. Schwarz, G.: Estimating the dimension of a model. Ann. Stat. **6**, 461–464 (1978)
24. Tsai, A., Yezzi, A., Wells, W., Tempany, C., Tucker, D., Fan, A., Grimson, E., Willsky, A.: Model-based curve evolution technique for image segmentation. In: Computer Vision and Pattern Recognition, Kauai, Hawaii, pp. 463–468 (2001)
25. Tsai, A., Yezzi, A.J., Willsky, A.S.: Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. IEEE Trans. Image Process. **10**(8), 1169–1186 (2001)