Multiple Object Tracking via Prediction and Filtering with a Sobolev-type Metric on Curves

Eleonora Bardelli[1], Maria Colombo[1], Andrea Mennucci[1], and Anthony Yezzi[2]

[1] Scuola Normale Superiore, Pisa, Italy
[2] School of Electrical Engineering, Georgia Institute of Technology, Atlanta, USA

Abstract. The problem of multi-target tracking of deforming objects in video sequences arises in many situations in image processing and computer vision. Many algorithms based on finite dimensional particle filters have been proposed. Recently, particle filters for infinite dimensional Shape Spaces have been proposed although predictions are restricted to a low dimensional subspace. We try to extend this approach using predictions in the whole shape space based on a Sobolev-type metric for curves which allows unrestricted infinite dimensional deformations. For the measurement model, we utilize contours which locally minimize a segmentation energy function and focus on the multiple contour tracking framework when there are many local minima of the segmentation energy to be detected. The method detects figures moving without the need of initialization and without the need for prior shape knowledge of the objects tracked.

1 Introduction

We consider the problem of tracking multiple moving shapes in a video sequence, which has been addressed many times in the past and in many different ways.

1.1 Shape Space

Many *Shape Spaces* have been considered in the past. One common approach is to model Shape Space as a finite dimensional space; as in the case of the *B-splines* approach used for the original *snakes* model in [1–3]. Another common approach is the *level set method* [4], where the shape is represented implicitly by the zero level set of a function. Some authors represent *shapes* explicitly as parametric curves, and then decompose the motion of a shape in a finite dimensional *affine* part, and an infinite dimensional *deformation* part [5, 6]. Others model the shape space as an infinite dimensional Riemannian manifold [7, 8]. Some authors do not model the shapes, but rather a (parametric) estimation of their posterior probability distribution (conditional on the images) [9].

Some of the above choices present problems when tracking. The approach with level sets is not well-apt to fast moving shapes: to predict their motion, we must be able to move the *"shape"* on long range. At the same time, if we model curves as splines, then we must specify the dynamic of the control points, and take care to factor those out of the shape dynamics.

We represent *shapes* as parametric curves. The model that we employ is an Infinite Dimensional Riemannian Manifold, with a Sobolev-type metric $\mathbb{H}$; it has been proposed in [8] and is briefly described in Section 2. The metric can be explained as giving (orthogonal) cost to translation, scaling and deformations of curves. This is a novel approach, in that we will not need, in the tracking model, to address separately the affine and deformation parts: this is implicitly done by the metric $\mathbb{H}$. This also implies that the prediction phase of the tracking algorithm predicts the translation, scaling and deformation parts all together: from the theoretical point of view this improves on previous approaches [6].

1.2 Tracking

We want to track shapes in a series of images I_t, where $t \in \mathbb{N}$, and $I_t \colon \Omega \to \mathbb{R}$ (usually $\Omega = [0,1]^2$). The tracking problem can benefit from prior assumptions: one such prior is the shapes' motion. We model the shapes' dynamics using a simple constant-velocity model. No *a priori* assumption is made on the probability distribution of shapes and of shape velocities.

Tracking is addressed, usually, as a *hidden variable estimation problem.* The tracker has an internal state U_t, usually the *a posteriori* estimate of the position of shape(s) at time t conditional on $I_1 \dots I_t$. To reduce the complexity, a new estimate U_{t+1} is derived from I_{t+1} and U_t. If the tracker includes a dynamical model of the shape motion, then U_t may estimate the velocity of shapes as well.

To compute the estimate U_t, some authors have employed *(extended) Kalman Filtering* [9]; when the Shape Space is not flat, this has known limitations since the predictor/corrector updates are computed only within a linearized vector space (i.e. the tangent space is used to approximate the relevant neighborhood of the underlying shape space). Moreover this cannot be readily adapted to unsupervised tracking of a large, possibly unknown, number of shapes, since a scene including multiple shapes has inherently a multi-modal posterior.

In [6] the authors propose an approach based on *Particle Filtering*; since this requires sampling and predicting in a (theoretically) infinite dimensional space, they split the motion of shapes in a finite dimensional *affine part* and an infinite dimensional *deformation part*; then they predict the affine part alone. We seek instead to carry out prediction within the entire shape space.

1.3 The Proposed Approach

We try to incorporate a simple simple *particle filtering with importance sampling* scheme in the framework provided by the shape space proposed in [8].

For every frame I_t we consider a set of n curves $\gamma_{t,1} \dots \gamma_{t,n}$, that represent the objects in image I_t. The update process for the curves on the frame I_{t+1} consists of three steps. In the first step we generate new curves using a prediction-correction scheme, as typical for a particle filtering approach. We first predict the position and shape of objects in I_{t+1} by shooting multiple geodesic trajectories from the curves $\gamma_{t,i}$. Interpolation between all curves at time t and $t-1$ generates approximately n^2 curves and may be interpreted as a *boosting step.*

We then perform a correction step by evolving each predicted curve via the gradient descent flow of an energy $E = F + E_{reg}$ which is made up of a segmentation energy term F (for example the Chan-Vese energy defined in Section 3), plus a regularizing scale and translation invariant term E_{reg}.

In the second step we generate new curves in a random way. The same gradient flow technique as above is applied to evolve m circles of random centers and radii. These random curves are supposed to find new objects entering the frame and could also individuate fast moving objects, on which it is otherwise difficult to initialize the prediction mechanism.

In the final step we select a subset of curves from those generated during the two previous steps by ranking them according to the segmentation energy F. The selection mechanism guarantees also that the selected curves do not cluster around the same local minimum of the segmentation energy but rather track multiple objects in the frame.

The method has been tested on fixed–camera scenes where multiple objects were moving. It was able to track multiple objects, both in translation and deformation, without the need for prior knowledge of the object shapes nor any special initialization. Results are presented in Section 4 and comparisons/relations with previous literature are discussed in Section 5.

An open-source library has been implemented to test the proposed method. It is available at http://mennucci.sns.it/StiefelCurve/. The source code is well commented and documented and fully clarifies all implementation details.

2 The Curve Model

A planar curve γ is a smooth function from $\mathbb{S}^1$ to $\mathbb{R}^2$ (where $\mathbb{S}^1$ is the unit circle); a curve is immersed when $|\gamma'(\theta)| \neq 0 \, \forall \theta \in \mathbb{S}^1$. We define M to be the space of all *smooth planar immersed curves.*

We define $\mathrm{len}(\gamma)$ to be the length of γ. Given a function $g \colon \mathbb{S}^1 \to \mathbb{R}^2$, we let $D_s g := g'/|\gamma'|$ be the derivative with respect to arc length along γ. We define the integral of g along γ and the average of g as

$$\int_\gamma g(s)\,\mathrm{d}s := \int_{\mathbb{S}^1} g(\theta)|\gamma'(\theta)|\,\mathrm{d}\theta \ , \qquad \fint_\gamma g(s)\,\mathrm{d}s := \frac{1}{\mathrm{len}(\gamma)} \int_\gamma g(s)\,\mathrm{d}s.$$

We also define the *centroid* $\overline{\gamma}$ of γ as $\fint_\gamma \gamma(s)\,\mathrm{d}s$.

We endow the space M with a Riemannian metric $\mathbb{H}$ developed in [8]. Suppose that h is a vector field along γ and decompose it as

$$h = h^t + h^l(\gamma - \overline{\gamma}) + \mathrm{len}(\gamma) h^d \ .$$

Setting

$$p(h) := h - (h \cdot D_s\gamma) D_s\gamma - (h \cdot D_s^2\gamma)(\gamma - \overline{\gamma}) \ ,$$

the components h^t and h^l of h are defined as

$$h^t := \fint_\gamma p(h)\,\mathrm{d}s \in \mathbb{R}^2 \ , \qquad h^l := -\fint_\gamma h \cdot D_s^2\gamma\,\mathrm{d}s \in \mathbb{R} \ .$$

The first component h^t changes the centroid of γ, whereas $h^l(\gamma - \overline{\gamma})$ changes the scale of γ, see [8]. The remaining component is intended to deform γ

$$h^d := \frac{1}{\mathrm{len}(\gamma)}[h - h^t - h^l(\gamma - \overline{\gamma})] \ .$$

Given $h, k \in T_c M$, decomposed as above, the metric is

$$\langle h, k\rangle_{\mathbb{H}} := h^t \cdot k^t + h^l k^l + \mathrm{len}(\gamma)^2 \fint_\gamma D_s h^d \cdot D_s k^d\,\mathrm{d}s \ .$$

The metric $\mathbb{H}$ enjoys the following properties.

- Centroid translations, scale changes and deformations of the curve are orthogonal. Moreover, the space of curves can be decomposed into a product of three spaces representing position, scale, and shape (see Thm 3.4 in [8]).
- Sobolev-type metrics favor *smooth* but otherwise unrestricted infinite–dimensional deformations [10] and they have a coarse-to-fine evolution behavior [11]. They are then quite useful for shape optimization and tracking tasks.
- There is a fast and easy way to compute gradients of commonly used energies with respect to the metric $\mathbb{H}$.
- Geodesics between immersed curves can be numerically computed efficiently. Geodesics connecting immersed curves up to rotation can be computed using simple closed form formulas.

3 The Tracking Algorithm

Given a curve γ, we define its exterior region as the unbounded connected component of $\mathbb{R}^2 \setminus \gamma$ and its interior (denoted by $\mathring{\gamma}$) as the complement in $\mathbb{R}^2$ of the exterior region. We denote by $F(\gamma, I)$ the standard Chan-Vese energy [12]:

$$F(\gamma, I) = \int_{\mathring{\gamma}} (I(x) - \mathrm{avg_{in}} I)^2\ \mathrm{d}x + \int_{\Omega\setminus\mathring{\gamma}} (I(x) - \mathrm{avg_{out}} I)^2\ \mathrm{d}x \qquad (1)$$

where $\mathrm{avg_{in}} I = \fint_{\mathring{\gamma}} I(x)\,\mathrm{d}x$ and $\mathrm{avg_{out}} I = \fint_{\Omega\setminus\mathring{\gamma}} I(x)\,\mathrm{d}x$.

Let $\{I_t\}_{t=0,\dots,N}$ be the frames of the video to be analyzed and $n \in \mathbb{N}$ a fixed parameter. For every t we define curves $\gamma_{t,1}, \dots, \gamma_{t,n} \in M$, which should outline different objects in the video. We expect more than one curve to estimate each moving object in the video in accordance with the *particle filtering* paradigm.

We also use some auxiliary curves $\delta_{t,1}, \dots, \delta_{t,n} \in M$, which will be defined in the following. The curve $\delta_{t,i}$ represents the state of the curve $\gamma_{t,i}$ in the previous frame. The algorithm also depends on some real parameters $\tau_0, \tau_1, d_0 \geq 0$, and a count parameter $m \in \mathbb{N}$.

We define also a *closeness* function f, which will be used in the third step. Given two curves γ and σ, it is the fraction of the area of $\mathring{\gamma}$ covered by $\mathring{\sigma}$,

$$f(\gamma,\sigma) := \frac{\operatorname{Area}(\mathring{\gamma} \cap \mathring{\sigma})}{\operatorname{Area}(\mathring{\gamma})} \ .$$

Each iteration of the algorithm computes $\gamma_{t+1,i}$ and $\delta_{t+1,i}$ for $i = 1, ..., n$ at time $t+1$ starting from the previous two sets of curves at time t. To start, we randomly choose curves $\gamma_{0,1}, ..., \gamma_{0,n}$ and define $\delta_{0,i} = \gamma_{0,i}$ for every $i = 1, \dots, n$. Each full iteration of the algorithm is broken down into three different steps.

Step 1: Generation of new curves via prediction and correction. For every pair $i, j \in \{1, ..., n\}$ let $\varGamma_{i,j} \colon [-1,1] \to M$ be a constant speed geodesic such that $\varGamma_{i,j}(-1) = \delta_{t,i}$, $\varGamma_{i,j}(0) = \gamma_{t,j}$ and $\varGamma_{i,j}$ restricted to $[-1,0]$ is a minimal geodesic between $\delta_{t,i}$ and $\gamma_{t,j}$. An iterative algorithm to compute $\varGamma_{i,j}$ is given in [8]. To shoot a geodesic with a given velocity there is a closed formula, as shown in [13] and [8].

We define the prediction $p_{i,j}$ as the geodesic calculated at time 1, namely

$$p_{i,j} := \varGamma_{i,j}(1) \qquad \forall (i,j) \in \{1, ..., n\}^2 \ .$$

The prediction is made according to a *constant velocity* dynamic of the objects in the video, which is always reasonable on a short time scale. Since geodesics are calculated with respect to the $\mathbb{H}$-metric in M, we do not predict only the position and scale of the new curve, but also its overall shape.

Note that we consider more than n predictions. Since more than one curve is usually tracking any given object, this causes small perturbations in the prediction that give stability to the algorithm. On the other hand we expect predictions made between curves following different objects to be meaningless and to be discarded in the upcoming selection step of the algorithm. Instead of shooting n^2 geodesics, random perturbations may be used but they are difficult to implement in a infinite dimensional shape space.

Then, for every $i, j \in \{1, ..., n\}$, we correct the prediction through a gradient descent flow. We use an energy E_{t+1}, defined as the sum of a Chan-Vese segmentation term F introduced in (1) and a regularizing elastic term with coefficient $k_e > 0$ (which is usually 0.02 in our experiments),

$$E_t(\gamma) := F(\gamma, I_t) + k_e \operatorname{len}(\gamma) \int_\gamma |D_s^2 \gamma|^2 \, \mathrm{d}s \ .$$

Let $\operatorname{GF}(\tau,\gamma) \colon [0,+\infty) \times M \to M$ be the gradient flow of E_{t+1} starting from γ, namely for every $\gamma \in M$ we solve the P.D.E.

$$\begin{cases} \frac{d}{d\tau}\operatorname{GF}(\tau,\gamma) = -\nabla E_{t+1}(\operatorname{GF}(\tau,\gamma)) & \text{for a. e. } \tau \in [0,+\infty) \\ \operatorname{GF}_0(\gamma) = \gamma \end{cases}$$

where ∇E_{t+1} is the gradient of E_{t+1} w.r.t. the metric $\mathbb{H}$. We define the correction as the gradient flow after a fixed flow-time $\tau = \tau_0$,

$$c_{i,j} := \operatorname{GF}(\tau_0, p_{i,j}) \qquad \forall (i,j) \in \{1, ..., n\}^2.$$

Step 2: Generation of random new curves. In order to detect new figures which appear in the frames, we consider m random curves $r_1, ..., r_m \in M_i$. Each r_i is a circle of random center and random radius on the image I_{t+1}.

We then correct the random circles with a gradient flow. Taking E_{t+1} and GF as in the previous paragraph, we define c_i as the gradient flow starting from r_i after a fixed flow-time τ_1

$$c_i := \mathrm{GF}(\tau_1, r_i) \qquad \forall i \in \{1, ..., m\}.$$

Step 3: Selection. In this step we select n curves from the large family of new curves generated in the previous steps

$$C_0 := \left\{ c_{i,j} \mid (i,j) \in \{1, .., n\}^2 \right\} \cup \{ c_i \mid i \in \{1, .., m\}\} \ .$$

We want to select the curves that best fit the image I_{t+1} according to the segmentation energy F, defined in (1). At the same time we prevent the selected curves from clustering around a single mode of the posterior and ignoring all other modes (as noted in [9]). To avoid this form of "collapsing", we employ a *closeness* function $f \colon M \times M \to [0,1]$, which will be described in the following, and a cut-off value $d_0 \in [0,1]$.

We denote by $F_{t+1}(\gamma) = F(\gamma, I_{t+1})$ the segmentation energy on frame I_{t+1} and by $\gamma_{t+1,1}$ the curve that minimizes F_{t+1} within the set C_0. Then, we consider the set of all curves which have *closeness* to $\gamma_{t+1,1}$ smaller than d_0

$$C_1 := \{ c \in C_0 \mid f(\gamma_{t+1,1}, c) < d_0 \} ,$$

and we let $\gamma_{t+1,2}$ be the curve of minimal energy within this set C_1. We repeat the procedure, defining C_2 as the set of curves which have *closeness* to $\gamma_{t+1,1}$ and $\gamma_{t+1,2}$ smaller than d_0; the curve $\gamma_{t+1,3}$ is the one of minimal energy with the set C_2. We repeat this procedure until we have selected n curves $\gamma_{t+1,1}, \ldots, \gamma_{t+1,n}$ or there are no curves left.

Since the sets C_i are decreasing, curves in C_i have F_{t+1} energies greater than $\gamma_{t+1,1} \ldots \gamma_{t+1,i}$. Moreover, a curve $\sigma \in C_i$ is discarded if contains in its interior a curve $\gamma_{t+1,j}$ for some $j \leq i$ because of the definition of f. Indeed, σ is probably a worse segmentation of the same object segmented by $\gamma_{t+1,j}$.

We point out that the energy F_{t+1} used here is different from the energy E_{t+1} used for the gradient flow since we neglect the elastic term in order to select the curves which best segment our moving figures, regardless of their regularity.

Eventually we define the curves $\delta_{t+1,i}$. If $\gamma_{t+1,i} = c_{\tilde{\imath},\tilde{\jmath}}$ for some $(\tilde{\imath}, \tilde{\jmath})$ (i.e. it was obtained via prediction and correction), we define $\delta_{t+1,i}$ as the curve from which the prediction was generated $\delta_{t+1,i} := \gamma_{t,\tilde{\jmath}}$. Otherwise, $\gamma_{t+1,i} = c_{\tilde{\imath}}$ (the result of a gradient flow on a random circle), and we define $\delta_{t+1,i} = \gamma_{t+1,i}$.

Optional Splitting of Curves. While tracking, it happens that curves develop self intersection, in particular when a figure is the superposition of two objects whose trajectories deviate after some time (see Figure 1). For this reason the algorithm has provision for an optional *splitting step*, before the selection step. We divide each curve in all its non-self-intersecting parts and those parts substitute it in the pool $\gamma_{t+1,1}, \ldots, \gamma_{t+1,n}$.

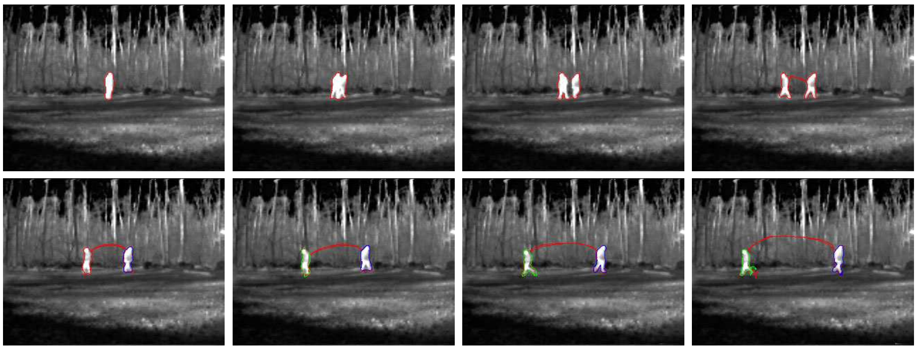

Fig. 1. Evolution with $m = 7$, $n = 3$ without splitting self-intersecting curves

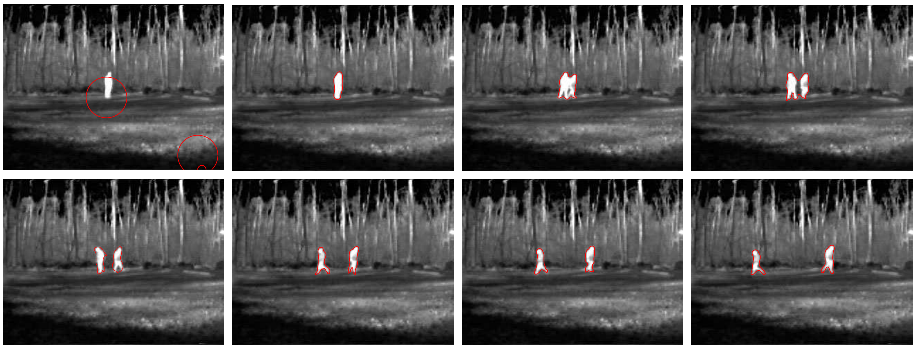

Fig. 2. Evolution with $m = 7$, $n = 3$

4 Experiments

In this section we describe some numerical experiments. Besides testing the algorithm on some simple videos, we run our algorithm disabling some core components, e.g. shape prediction, and present examples of how this affects the quality of the tracking. The variations of the algorithm are the following.

"Gradient flow only". This is the classical Chan-Vese method, implemented on multiple curves. The generation of new curves is made by evolving old curves with a gradient flow on the new frame and selection step is left unchanged.

"Centroid and length prediction". This algorithm differs from the one presented in Section 3 only in the prediction step. In this case the prediction is made about the centroid and length of the curve, leaving the shape unchanged.

Because of the novelty of the shape space and the inherent difficulty of implementing a particle filtering in a infinite dimensional shape space, the main goal of this experimental validation is not to compare the algorithm with the wide literature available nowadays. Instead, we show how the different parts of

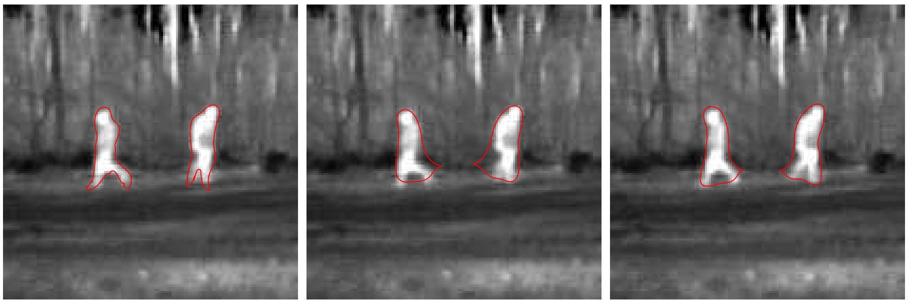

Fig. 3. Enlargement of the sixth image in Figure 2, comparison with the same frame obtained without prediction and with prediction of centroid and length only

the algorithm work and what is the contribution of each, hoping that the result presented here might be the first step towards further studies in this direction.

We consider two different sample videos. In the first one there are two overlapping people in the beginning who then walk in opposite directions. The second video shows a bird's eye view of a plaza with many people walking and a motorcycle which enters the video in the right upper corner.

All our sample videos have been preprocessed in order to eliminate the fixed background. The energies are computed on the preprocessed frames. We show here the curves superimposed on the original frames to provide the scene context.

Figures 1, 2, 4 are examples of the program results. In Figure 1, when the two figures cease to be overlapped two new local minima of the Chan-Vese energy appear. They correspond to the two separate figures and are soon captured by the random curves. Other features can be pointed out in Figures 2 and 4.

Multiple segmentation. The possibility of tracking multiple objects and detecting new objects entering the frame is a key feature of the algorithm. In Figure 2, starting from random circles the central object is detected and then tracked. In Figure 4 there are more objects to follow and because of shadows they have more complex shapes. However, the algorithm works well and the motorcycle which enters the video is quickly detected and followed. Note that only 10 curves are used to follow 5 objects, so the number of needed curves does not grow too much with the number of shapes to track. In both examples once an object has been detected random circles do not influence its tracking any more.

Comparison with "gradient flow only"*.* The prediction produces improvements in the tracking, when compared to simple active-contour based tracking algorithms. For example comparing Figure 4 and 5 we see that figures are detected in both sequences, but in the first they are segmented better than in the second. Indeed, the tracked objects, namely people together with their shadow, have a complicated shape that is quite different from a circle, so it is more difficult for the gradient flow to conform to them after a limited amount of flow time.

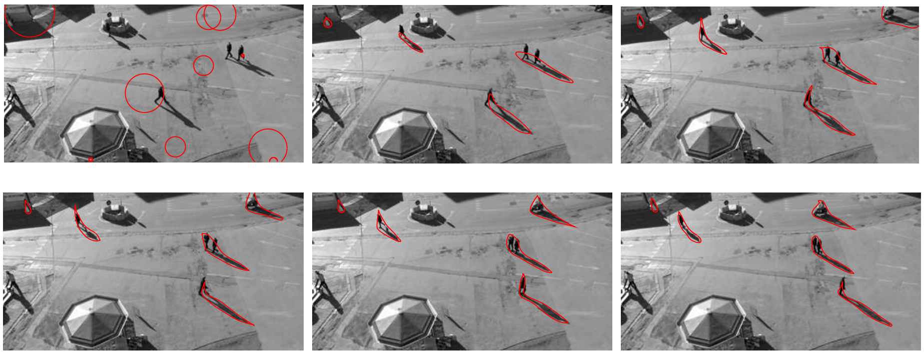

Fig. 4. Evolution with $m = 20$, $n = 10$ (only best 5 curves are drawn)

Shape prediction. The prediction about the shape turns out to be important to delineate small details of the moving shapes. We compared the algorithm with the more limited (and finite dimensional) *"centroid and size prediction"*. Due to space limitation, we omit to include in this paper detailed examples obtained in this way. In Figure 3 we can see a snapshot of the evolution with our algorithm, with *"centroid and size prediction"* and with *"gradient flow only"*. We can observe that the segmentations are much rougher in the last two figures.

5 Conclusions

Our method does not use an a-priori probability model for shapes, as is often done [3, 14], neither *level–set methods*, as is often the case [14] in active contour based trackers. Instead, it uses a metric on curves which allows unrestricted shape deformation and long-range infinite dimensional shape prediction.

The structure of our algorithm overcomes the *motion correspondence problem*. As described in [9], particle filtering is appealing in multiple object tracking because of its ability to carry multiple hypotheses, but establishing the correspondence between objects and observations is not a trivial task.

One current limitation in the proposed algorithm is that it does not enforce temporal coherence in the velocity or the photometry of shapes. This enables the algorithm to easily find and track new objects, but it may be a nuisance in some applications. This is also the reason why the algorithm is applied on pre-filtered, background subtracted frames. However, this limitation is primarily due to our simple choice to use the Chan-Vese model for our segmentation energy and may be significantly improved by using a model that incorporates more photometric details. We are currently testing different choices for the segmentation energy as well as the dynamics so that the algorithm will model and deal with a (possibly non fixed) background, and/or a cluttered scene.

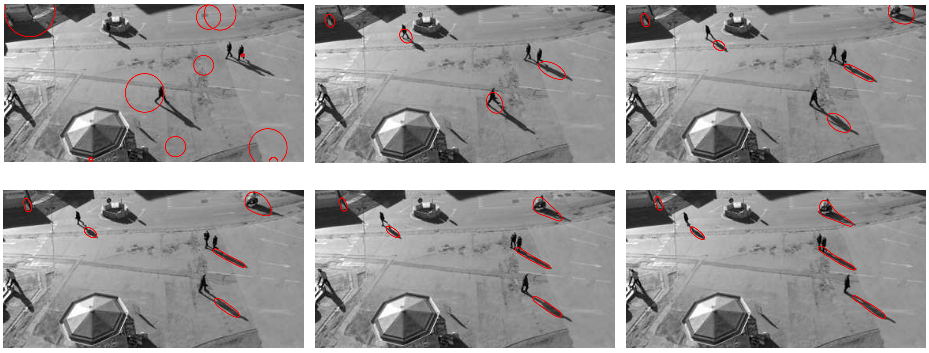

Fig. 5. Evolution with $m = 20$, $n = 10$, without prediction (5 best curves are drawn)

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. Comput. Vis. **1** (1987) 321–331
2. Blake, A., Yuille, A., eds.: Active Vision. MIT Press, Cambridge, Mass. (1992)
3. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. Int. J. Comput. Vis. **1** (1998) 5–28
4. Sethian, J.A.: Level set methods and fast marching methods. Cambridge University Press, Cambridge (1999)
5. Soatto, S., Yezzi, A.J.: DEFORMOTION: Deforming motion, shape average and the joint registration and segmentation of images. In: ECCV (3). (2002) 32–57
6. Rathi, Y., Vaswani, N., Tannenbaum, A., Yezzi, A.: Tracking deforming objects using particle filtering for geometric active contours. IEEE TPAMI **29** (2007) 1470–1475
7. Klassen, E., Srivastava, A., Mio, W., Joshi, S.H.: Analysis of planar shapes using geodesic paths on shape spaces. IEEE TPAMI **26** (2004) 372–383
8. Sundaramoorthi, G., Mennucci, A., Soatto, S., Yezzi, A.: A new geometric metric in the space of curves, and applications to tracking deforming objects by prediction and filtering. SIAM J. Imaging Sci. **4** (2011) 109–145
9. Chang, C., Ansari, R., Khokhar, A.: Multiple object tracking with kernel particle filter. In: CVPR. (2005)
10. Sundaramoorthi, G., Yezzi, A., Mennucci, A.: Sobolev active contours. Int. J. Comput. Vis. **73** (2007) 413–417
11. Sundaramoorthi, G., Yezzi, A., Mennucci, A.: Coarse-to-fine segmentation and tracking using Sobolev Active Contours. IEEE TPAMI **30** (2008) 851–864
12. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. Image Process. **10** (2001) 266–277
13. Edelman, A., Arias, T., Smith, S.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20** (1999) 303–353
14. Zhang, T., Freedman, D.: Tracking objects using density matching and shape priors. In: ICCV, IEEE Computer Society (2003) 1056–1062