

Implicit free-form-deformations for multi-frame segmentation and tracking

Konstantinos Karantzas^{1,2} and Nikos Paragios¹

¹ ATLANTIS, CERTIS, Ecole Nationale des Ponts et Chaussees,
6-8 avenue Blaise Pascal, Cite Descartes, 77455 - Marne-La-Vallee, France
{karank, nikos.paragios}@cermics.enpc.fr
<http://atlantis.enpc.fr/>

² School of Rural and Survey Engineering, National Technical University of Athens
Heron Polytechniou 9, Zographou, 15780, Greece
{karank}@central.ntua.gr

Abstract. In this paper, we propose a novel technique to address motion estimation and tracking. Such technique represents the motion field using a regular grid of thin-plate splines, and the moving objects using an implicit function on the image plane that is a cubic interpolation of a "level set function" defined on this grid. Optical flow is determined through the deformation of the grid and consequently of the underlying image structures towards satisfying the constant brightness constraint. Tracking is performed in similar fashion through the consistent recovery in the temporal domain of the zero iso-surfaces of a level set that is the projection of the Free Form Deformation (FFD) implicit function according to the cubic spline formulation. Such an approach is a compromise between dense motion estimation and parametric motion models, introduces smoothness in an implicit fashion, is intrinsic, and can cope with important object deformations. Promising results demonstrate the potentials of our approach.

1 Introduction

Motion perception is a fundamental task of biological vision with motion estimation and tracking being the most popular and well-addressed applications. To this end, given a sequence of images, one would like to recover the 2D temporal displacement (optical flow) and the position of objects of particular interest. These applications often serve as input to high-level vision tasks, like 3D reconstruction, etc.

Dense optical flow estimation is an ill-posed problem. The problem itself is rather ill-posed since [1] the number of unknowns to be recovered is greater to the number of constraints. Such constraints are determined through the linearization of the visual or intensity preservation constraint [2]. Smoothness constraints [3] are often considered to overcome the ill-posedness of the estimation process and often lead to satisfactory results. A step further refers to the use of parametric motion estimation [4] where the motion in the entire image plane or some

portions of it is represented with a linear function of the pixel coordinates. To this end, robust statistical methods [5] were considered to account for outliers in the estimation process leading to promising results [6, 7] when the assumption on the motion form induced by the model is respected from the data. One can claim that parametric motion models are efficient representations of optical flow, a good compromise between low complexity and reasonable flow estimates that suffer at the object boundaries. Moreover neither the case of non-planar or objects undergoing non-rigid deformations can be addressed through such a formulation.

Tracking non-rigid objects is a task that has gained particular attention in computational vision. Starting from the pioneering formulation of the snake model [8] several attempts to address tracking through the deformation of contours can be found in the literature either model-free [9] or model-based [10]. Level set methods [11] is an established technique [12] to track moving interfaces through model-free [13] or model-based [14] methods with the advantage of being implicit, intrinsic and parameter-free. However they suffer from computational expensive processing [15] while one should preserve the form of the implicit functions through frequent re-initialization steps. Such a limitation was addressed in [16] where a finite element approach was considered to implement a level set flow.

In this paper, we introduce a higher-order polynomial approach to address dense optical flow estimation and tracking within the level set approach. To this end, we represent motion using a free form deformation of a super-imposed regular connected grid, an excellent alternative to dense motion estimation as well as to parametric motion models. Tracking is addressed through the modification of a "level set" function on the FFD space such that its projection on the image space captures the object boundaries. Visual preservation, consistence in the object appearance and smoothness constraints are used to determine the deformation of the implicit grid towards simultaneous motion estimation and tracking of objects in successive frames.

Prior art in joint optical flow estimation and tracking has mostly addressed the case of parametric (mostly affine) motion within the standard level set formulation [17, 18, 14, 19, 20]. The remainder of this paper is organized according to the following fashion; In the next section, we briefly introduce the level set method and the free form deformation model. Our variational model to recover optical flow estimations and perform tracking is described in section 3. The optimization process is presented in section 4, while experimental results and discussion are part of section 5.

2 Free Form Deformations & Implicit Level Sets

Let us consider an image:

$$\mathcal{I}(x, y) = \{(x, y) | 1 \leq x \leq X, 1 \leq y \leq Y\}$$

and a regular lattice of control points superimposed to this image:

$$\mathbf{P}_{m,n} = (\mathbf{P}_{m,n}^x, \mathbf{P}_{m,n}^y); \quad m = 1, \dots, M, \quad n = 1, \dots, N$$

One can introduce a third dimension on this grid, a discrete function $\Phi(\cdot)$, such that input image is approximated through a tensor product of Cubic B-spline:

$$\mathcal{I}(x, y) \approx \sum_{k=0}^3 \sum_{l=0}^3 B_k(u) B_l(v) \Phi_{i+k, j+l}$$

with $i = \lfloor \frac{x}{X} \cdot M \rfloor - 1$, $j = \lfloor \frac{y}{Y} \cdot N \rfloor - 1$ and $B_k(u)$ is the k^{th} basis function of a Cubic B-spline:

$$\begin{aligned} B_0(u) &= (1-u)^3/6, & B_1(u) &= (3u^3 - 6u^2 + 4)/6 \\ B_2(u) &= (-3u^3 + 3u^2 + 3u + 1)/6, & B_3(u) &= u^3/6 \end{aligned}$$

with $u = \frac{x}{X} \cdot M - \lfloor \frac{x}{X} \cdot M \rfloor$ ($B_l(v)$ is defined in a similar fashion with $v = \frac{y}{Y} \cdot N - \lfloor \frac{y}{Y} \cdot N \rfloor$). We assume that (sixteen) adjacent control points are needed to produce the observed value at any given pixel of the image. The parameters of this new representation consist of the position of the grid points and the value embedded function at these points $\Theta = (\mathbf{P}_{m,n}^x, \mathbf{P}_{m,n}^y, \Phi_{m,n})$.

Furthermore one can consider a deformation of this grid (deformation of the image) starting from an initial configuration \mathbf{P} , and the deforming control lattice as

$$\mathbf{P}' = \mathbf{P} + \Delta\mathbf{P}$$

that can be considered as an incremental free form deformation with the deformations of the control points in both directions according to:

$$\Delta\mathbf{P} = \{(\delta\mathbf{P}_{m,n}^x, \delta\mathbf{P}_{m,n}^y)\}; (m, n) \in [1, M] \times [1, N]$$

The essence of FFD is to deform an object by manipulating a regular control lattice \mathbf{P} overlaid on its volumetric embedding space. Once a deformation has been applied, the displacement of a pixel (x, y) given the deformation of the control lattice from \mathbf{P} according to $\Delta\mathbf{P}$, is defined in terms of a tensor product of Cubic B-spline:

$$\begin{aligned} \mathcal{T}(\Delta\mathbf{P}; (x, y)) &= ((x, y)) + \delta T(\Delta\mathbf{P}; (x, y)) \\ &= \sum_{k=0}^3 \sum_{l=0}^3 B_k(u) B_l(v) (\mathbf{P}_{i+k, j+l} + \delta\mathbf{P}_{i+k, j+l}) \end{aligned}$$

Such deformation field $\mathcal{T}(\Delta\mathbf{P}; x, y)$ [21] is a popular approach in graphics, animation and rendering [22]. Opposite to optical flow techniques, FFD techniques support smoothness constraints, exhibit robustness to noise and are suitable for modelling large and small non-rigid deformations. Furthermore, under certain conditions, it can support a dense registration paradigm that is continuous and guarantees a one-to-one mapping.

The level set method [11] consists of representing and evolving an evolving interface $\partial\mathcal{R}(p)$ with the zero-level set of an embedding surface Φ . Such representation can lead to a natural handling of changing the topology of $\partial\mathcal{R}(p)$. Numerical simulations on Φ may be developed trivially and intrinsic geometric

properties of the evolving interface can be estimated directly from the level set function.

Let $\phi : \Omega \rightarrow \mathcal{R}^+$ be a Lipschitz function that refers to a level set representation:

$$\phi(p; t) = \begin{cases} 0 & , p \in \partial\mathcal{R}(t) \\ +\mathcal{D}((p), \partial\mathcal{R}(t)) > 0 & , p \in \mathcal{R}(t) \\ -\mathcal{D}((p), \partial\mathcal{R}(t)) < 0 & , p \in [\Omega - \mathcal{R}(t)] \end{cases} \quad (1)$$

where Ω is the image domain (bounded) and $\mathcal{D}(p, \partial\mathcal{R}(t))$ is the minimum Euclidean distance between the pixel p and the interface $\partial\mathcal{R}(t)$. Then, the level set formulation can be considered as an optimization framework. To this end, one can define the approximations of DIRAC and HEAVISIDE distributions [23, 24]:

$$\delta_a(\phi) = \begin{cases} 0, & |\phi| > \alpha \\ \frac{1}{2\alpha} (1 + \cos(\frac{\pi\phi}{\alpha})), & |\phi| < \alpha \end{cases} \quad (2)$$

$$H_\alpha(\phi) = \begin{cases} 1, & \phi > \alpha \\ 0, & \phi < -\alpha \\ \frac{1}{2} \left(1 + \frac{\phi}{\alpha} + \frac{1}{\pi} \sin\left(\frac{\pi\phi}{\alpha}\right)\right), & |\phi| < \alpha \end{cases}$$

These functions can be used to define contour-based as well as region-based energetic modules for the evolving interface in the level set space [23]:

$$(i) \underbrace{\iint_{\Omega} H_\alpha(\phi(p)) r_1(I(p)) dx dy}_{\text{regional module}}, \quad (ii) \underbrace{\iint_{\Omega} \delta_\alpha(\phi(p)) b(I(p)) |\nabla\phi(p)| dx dy}_{\text{boundary module}}$$

where r and b are *region* and *boundary* positive monotonically decreasing data-driven functions. The first term [i] is a grouping component that accounts for some regional properties (modulo the definition of r) of the area defined by the evolving interface. The second term [ii] is a combination of a boundary attraction term (modulo the definition of b) and a smoothness component [25, 26].

Within the selected representation, one can consider a function Φ defined at the lattice \mathbf{P} to be a level set function, if

$$\phi(x, y) = \sum_{k=0}^3 \sum_{l=0}^3 B_k(u) B_l(v) \Phi_{i+k, j+l}$$

and

$$\phi(x, y) = \begin{cases} 0 & , p \in \partial\mathcal{R}(t) \\ +\mathcal{D}((p), \partial\mathcal{R}(t)) > 0 & , p \in \mathcal{R}(t) \\ -\mathcal{D}((p), \partial\mathcal{R}(t)) < 0 & , p \in [\Omega - \mathcal{R}(t)] \end{cases}$$

One now can use such a formulation to encode motion estimation and tracking. Motion is represented with the deformation of the original lattice \mathbf{P} while tracking will be addressed through the evolution of a "level set function" Φ defined on the same lattice.

3 Optical Flow Estimation

Optical flow estimation is equivalent with recovering a pixel-wise deformation field $\mathcal{T}(\Delta\mathbf{P}; x, y)$ that creates visual correspondences between two consecutive images f and g . Optical flow estimation within FFD is now equivalent with finding the best lattice \mathbf{P} configuration such that the overlaid structures (images) coincide. One can consider the Sum of Squared Differences (SSD) as the data-driven term to recover the deformation field $\mathcal{T}(\Theta; \mathbf{x})$;

$$E_{data}(\Theta) = \iint_{\Omega} (f(\mathbf{x}) - g(\mathcal{T}(\Delta\mathbf{P}; x, y)))^2 dx dy$$

Such an error norm is very sensitive to occlusions as well as to outliers and therefore it can be replaced with a robust estimator, or like an M-estimator. Such a method assigns weights to the constraints at the pixel level that are disproportional to their residual error therefore rejecting the motion outliers. to this end, one should define the influence function, $\psi(x)$ like for example the Tukey's estimator:

$$\rho(x) = \begin{cases} x(K_{\sigma} - x) & \text{if } |x| < K_{\sigma} \\ 0 & \text{otherwise} \end{cases}$$

where K_{σ} characterizes the shape of the robust function and is updated at each iteration leading to the following cost function:

$$E_{data}(\Delta\mathbf{P}) = \iint_{\Omega} \rho(r) dx dy = \iint_{\Omega} \rho(f(\mathbf{x}) - g(\mathcal{T}(\Delta\mathbf{P}; x, y))) dx dy$$

While such a model can be quite efficient it still suffers from the aperture problem. One can consider additional constraints to the constant brightness assumption like the gradient preservation assumption, recently introduced in [27] leading to the following cost function;

$$E_{data}(\Delta\mathbf{P}) = \alpha \iint_{\Omega} \rho(f(\mathbf{x}) - g(\mathcal{T}(\Delta\mathbf{P}; x, y))) dx dy \\ + \beta \iint_{\Omega} \rho(|\nabla f - \nabla g(\mathcal{T}(\Delta\mathbf{P}; x, y))|) dx dy$$

a constraint that improves the estimation of the optical flow on the object boundaries where the visual constancy assumption is often violated.

The use of thin plate splines to represent motion introduces in an implicit form some smoothness constraint that can deal with a limited level of deformation. In order to account for outliers and noise, one can replace the error-two norm with more appropriate robust metrics [5]. In order to further preserve the regularity of the recovered motion flow, one can consider an additional smoothness term on the deformation field $\delta\mathbf{P}$. We consider a computationally efficient smoothness term:

$$E_{smooth}(\Delta\mathbf{P}) = \iint_{\Omega} (|T_x(\Delta\mathbf{P}; x, y)|^2 + |T_y(\Delta\mathbf{P}; x, y)|^2) dx dy$$

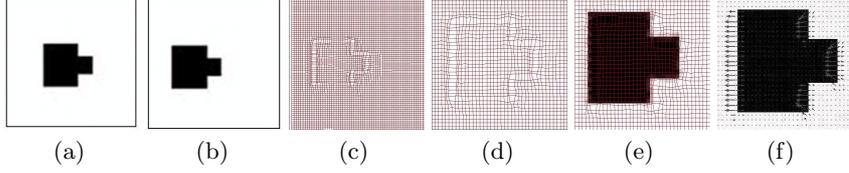


Fig. 1. Binary case with a global one pixel movement to the left : (a) first frame f , (b) second frame g , (c) deformed FFD grid, (d) Zoom on deformed grid, (e) Zoom on deformed grid overlaid to first image (h)Zoom on grid's flow overlaid to second image.

Such smoothness term is based on a classic error norm that has certain known limitations. Within the proposed framework, an implicit smoothness constraint is also imposed by the Spline FFD. Therefore there is not need for introducing complex and computationally expensive regularization components.

Then the global deviations from the data-driven term and the smoothness constraints term can now be integrated to define an objective function that upon optimization will provide a smooth motion field that establishes correspondences between the two images:

$$\begin{aligned}
 E_{flow}(\Delta\mathbf{P}) &= \alpha \iint_{\Omega} \rho(f(\mathbf{x}) - g(\mathcal{T}(\Delta\mathbf{P}; x, y))) dx dy \\
 &\quad + \beta \iint_{\Omega} \rho(\|\nabla f - \nabla g(\mathcal{T}(\Delta\mathbf{P}; x, y))\|) dx dy \\
 &\quad + \gamma \iint_{\Omega} (|T_x(\Delta\mathbf{P}; x, y)|^2 + |T_y(\Delta\mathbf{P}; x, y)|^2) dx dy
 \end{aligned}$$

Multilevel Incremental Free-Form Deformation (MIFFD): A straightforward application of the FFD manipulation cannot always guarantee the successful motion estimation between the two images. One reason for this is that we limit the maximum displacement of a control point to approximately a half of the spacing between control points in order to make the deformation function one-to-one. The correspondences that each time can be caught are according to what level (how coarse or fine) of the FFD's grid has been chosen. Here, we present the MIFFD technique that overcomes the drawbacks of the straightforward method, since it can handle both large and small non-rigid deformations. Multiresolution control lattices are used according to a coarse-to-fine strategy. From a coarser level of the control lattice that can deal better with large displacements we proceed continuously to a finer level. At each level, we can solve for the incremental deformation of the control lattice using the scheme presented in the previous section. In the end, the overall dense deformation field for motion estimation is defined by these incremental deformations from all levels.

Let $\mathbf{P}^1, \dots, \mathbf{P}^K$ denote a hierarchy of control point meshes at different resolutions. Each control mesh \mathbf{P}^k and the associated spline-based FFD defines a transformation $\mathcal{T}^k(\Delta\mathbf{P}; x, y)$ at each level of resolution and the total deformation $\delta\mathcal{T}(x, y)$ for a pixel (x, y) in a hierarchy of K levels is:

$$\delta\mathcal{T}(x, y) = \sum_{k=0}^K \delta\mathcal{T}^k(\Delta\mathbf{P}^k; x, y)$$

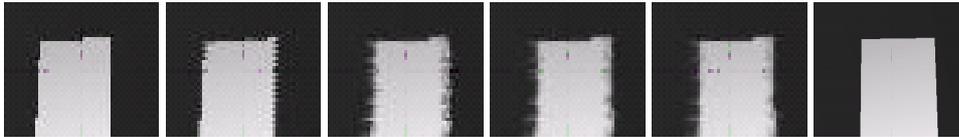


Fig. 2. Multilevel Optical Flow estimation MIFFD (4 levels). From the second image g (top - left), the reconstructed images from the estimated flow are shown, until the first image f (last one) is approximated. Final Energy 7% of initial Energy.

The hierarchy of control lattices can have arbitrary number of levels, but typically 3-4 levels are sufficient to handle both large and small deformations. Such an optimization will lead to successful estimation of the motion field but does not address tracking.

Let us consider without loss of generality that an object is present in the scene. The task of tracking consists of recovering the successive positions of a planar curve $\gamma(\cdot)$ such that the object is properly delineated in time. In order to address this demand we consider a level set curve to represent objects.

4 Object Tracking

Tracking is performed through the consistent recovery in the temporal domain of the zero iso-surfaces of a level set $\gamma(\Delta\mathbf{P})$ that is the projection of the FFD implicit function according to the cubic spline formulation.

Based on region-driven model free image segmentation techniques, objects boundaries are approached through a curve propagation technique (Figure 4). The essence of this approach is to optimize the position and the geometric form of the curve by measuring information along that curve, and within the regions that compose the image partition.

To this end, one can assume without loss of generality that objects are uniform that is also the case for the background. In that case, given an initial position of the curve, one can determine global region-driven $r_{obj}(f)$ and $r_{bg}(f)$ functions provide a statistical description of the inside and outside object area:

$$r_{obj}(f(x, y)) = \frac{(\mu_{obj} - f(x, y))^2}{\sigma_{obj}^2}, \quad r_{bg}(f(x, y)) = \frac{(\mu_{bg} - f(x, y))^2}{\sigma_{bg}^2}$$

where μ_{obj} is the mean and σ_{obj} the covariance matrix of the object appearance (similar definition for the background). In cases where the assumption of Gaussian densities seems unrealistic one can consider a more flexible parametric density function - gaussian mixture - to describe the visual properties of the object and the background.

In the case of static images, one can perform object extraction through the separation of image pixels according to their match with the expected appearance properties of the object and the background. Such an optimization can be

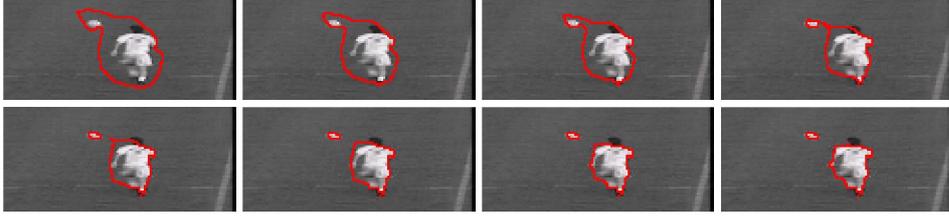


Fig. 3. Curve Propagation on 006 frame of player sequence.

considered on the lattice space, that is

$$\begin{aligned}
 E_{object}(\Phi) = & \alpha \iint \delta \left(\Phi \left(\sum_{k=0}^3 \sum_{l=0}^3 B_k(u) B_l(v) \Phi_{i+k,j+l} \right) \right) \\
 & \left| \nabla \sum_{k=0}^3 \sum_{l=0}^3 B_k(u) B_l(v) \Phi_{i+k,j+l} \right| d\Omega \\
 & + \beta \iint_{\Omega} H \left(\sum_{k=0}^3 \sum_{l=0}^3 B_k(u) B_l(v) \Phi_{i+k,j+l} \right) r_{obj}(f(x, y)) d\Omega \\
 & + \beta \iint_{\Omega} \left[1 - H \left(\sum_{k=0}^3 \sum_{l=0}^3 B_k(u) B_l(v) \Phi_{i+k,j+l} \right) \right] r_{bg}(f(x, y)) d\Omega
 \end{aligned}$$

where the first term imposes smoothness constraints while the second address a background/object separation according to the expected visual properties of the two class. One now can consider the separation of the object/background in both frames f and g given the deformation of the grid through the FFD one can address tracking through the minimization of

$$\begin{aligned}
 E_{tracking}(\Delta \mathbf{P}, \Phi) = & \alpha \iint \delta \left(\Phi \left(\sum_{k=0}^3 \sum_{l=0}^3 B_k B_l \Phi_{i+k,j+l} \right) \right) \\
 & \left| \nabla \sum_{k=0}^3 \sum_{l=0}^3 B_k B_l \Phi_{i+k,j+l} \right| d\Omega \\
 & + \beta \iint_{\Omega} H \left(\sum_{k=0}^3 \sum_{l=0}^3 B_k B_l \Phi_{i+k,j+l} \right) r_{obj}(f(x, y)) d\Omega \\
 & + \beta \iint_{\Omega} \left[1 - H \left(\sum_{k=0}^3 \sum_{l=0}^3 B_k B_l \Phi_{i+k,j+l} \right) \right] r_{bg}(f(x, y)) d\Omega \\
 & + \beta \iint_{\Omega} H \left(\sum_{k=0}^3 \sum_{l=0}^3 B_k B_l \Phi_{i+k,j+l} \right) r_{obj}(g(T(\Delta \mathbf{P}; x, y))) d\Omega \\
 & + \beta \iint_{\Omega} \left[1 - H \left(\sum_{k=0}^3 \sum_{l=0}^3 B_k B_l \Phi_{i+k,j+l} \right) \right] r_{bg}(g(T(\Delta \mathbf{P}; x, y))) d\Omega
 \end{aligned}$$

where α, β are constant coefficients and the assumption that the object/background properties do not change from one frame to the next. One can relax this constraint through the estimation of visual descriptors in both frames.

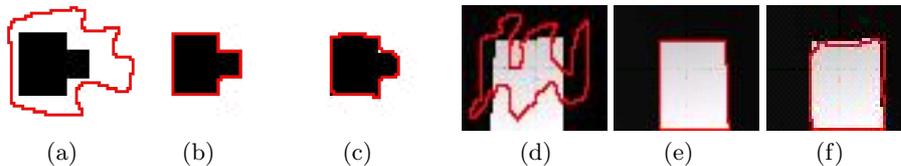


Fig. 4. Tracking result on Test Sequences: (a) Initial contour on first image f of Figure 1 sequence, (b) Curve propagation result on same image, (c) Object boundaries after applying the transformation of MIFFD flow, (d) Initial contour on first image f of Figure 2 sequence, (e) Curve propagation result on same image, (h) Object boundaries after applying the transformation of MIFFD flow.

Such a tracking term can be integrated with the optical flow estimation term to simultaneously address dense optical flow estimation and object tracking.

$$E(\Delta\mathbf{P}, \Phi) = E_{flow}(\Delta\mathbf{P}) + E_{tracking}(\Delta\mathbf{P}, \Phi)$$

The lowest potential of this cost function will provide visual correspondences between the two images, and recover optimal successive positions of objects in time [Figure 5 and 6].

5 Implementation

The calculus of variations and a gradient descent method can be used to optimize such an objective function. A minimizer must fulfill the Euler-Lagrange equation both in the deformation space $[\Delta\mathbf{P}]$ as well as in the implicit space $[\Phi]$;

$$\frac{\partial}{\partial\Delta\mathbf{P}}E(\Delta\mathbf{P}, \Phi) = 0, \quad \frac{\partial}{\partial\Phi}E(\Delta\mathbf{P}, \Phi) = 0$$

One can further develop these conditions using the chain rule;

$$\frac{\partial}{\partial\Delta\mathbf{P}}E(\Delta\mathbf{P}, \Phi) = \frac{\partial E_{flow}(\Delta\mathbf{P})}{\partial\Delta\mathbf{P}} + \frac{\partial E_{tracking}(\Delta\mathbf{P}, \Phi)}{\partial\Delta\mathbf{P}}$$

while in the case of the implicit FFD level set the flow consists only one term;

$$\frac{\partial}{\partial\Phi}E(\Delta\mathbf{P}, \Phi) = \frac{\partial E_{tracking}(\Delta\mathbf{P}, \Phi)}{\partial\Phi}$$

In practice, the proposed framework works in the following fashion. Given an initial contour, the implicit level function is estimated in the lattice space. Then, in parallel one updates the motion parameters of the process as well as deforming the contour. To this end, an adaptive estimation of the regional descriptors is considered as well as frequent re-initializations of the lattice implicit function. Upon a steady state solution, the lattice deformations as well as the object positions are recovered in successive frames. Such positions are used to initialize the process in the next couple of frames and the process is repeated until convergence.

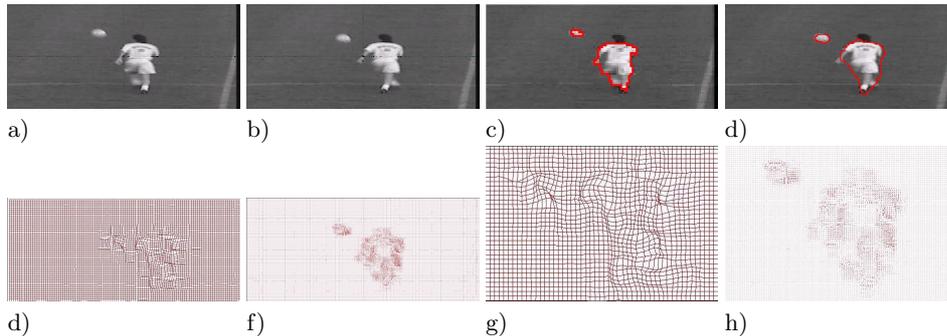


Fig. 5. Player Sequence Recovery (frames R006 -R007): (a) first image f (R006), (b) second image g (R007), (c) Object boundaries on f image from Level Set propagation, (d) Object boundaries on g image after applying the transformation of MIFFD flow, (e) Deformed Grid, (f) Deformed Grid's Flow, (g) Zoom on deformed grid, (h) Zoom on deformed grid's flow.

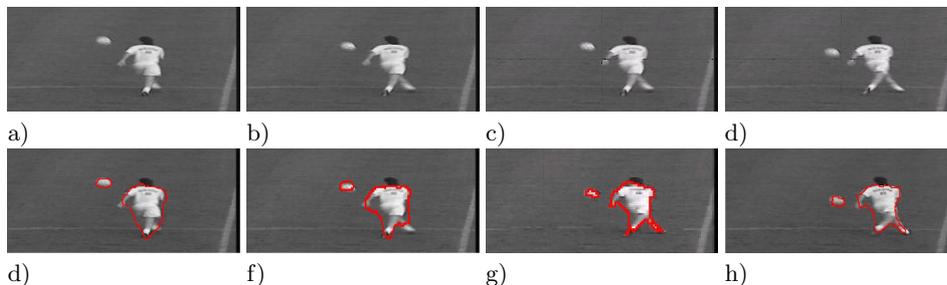


Fig. 6. Player Sequence: Recovered Frames R007-R008 and R009-R010: (a) frame R007, (b) frame R008, (c) Object boundaries from previous recovered frames (Figure 5), (d) Object boundaries after applying the transformation of MIFFD flow, (e) frame R009, (f) frame R010, (g) Object boundaries from Level Set propagation, (h) Object boundaries after applying the transformation of MIFFD flow.

6 Discussion

In this paper we have presented a novel algorithm to optical flow estimation and tracking. Our approach introduces the concept of joint motion estimation and tracking in superimposed spaces of higher order polynomials like thin plane spline level set. The selected representation of motion guarantees one-to-one correspondences, smoothness on the deformation field and is of low complexity. Parallel to that we address tracking through the recovery of explicit correspondences between the object temporal positions in the level set space that is implicit, intrinsic and parameter free. Promising results, as shown in Figure 5 and 6, demonstrate the potentials of the proposed formulation that address in a simultaneous fashion dense optical flow estimation and non-rigid tracking. Classical optical flow test sequences like the Yosemite sequence can not be used for validation because there exist not an apparent object for tracking.

One can consider numerous extensions of the method. The use of FFD that also encode the structure of the image is a prominent one. The grid that was considered to represent motion has a fixed topology and the motion of each image pixel is reproduced using the same number of neighboring elements that are distributed according to the same topology. One can consider modifying the grid dependencies and connections according to the image structure. In terms of tracking, the case of multiple objects is to be addressed. Within the proposed framework one can consider the one-to-one constraint on the correspondences and preserve topology or relax such a constraint to address topological changes from one image to the next. Such a perspective is to be investigated. Last, but not least the use of a 3D deformation grid can be considered to account for motion decomposition in layers.

References

1. J. Barron, D. Fleet, and S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12:43–77, 1994.
2. B. Horn and B. Schunck. Determinating Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.
3. J. Weickert and C. Schnorr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14:245–255, 2001.
4. B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
5. P. Huber. *Robust Statistics*. John Wiley & Sons, 1981.
6. M. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:973–986, 1996.
7. J-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6:348–365, 1995.
8. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. In *IEEE International Conference in Computer Vision*, pages 261–268, 1987.
9. M. Isard and A. Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In *European Conference on Computer Vision*, volume I, pages 343–356, 1996.
10. T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61:38–59, 1995.
11. S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed : Algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
12. S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer Verlag, 2003.
13. N. Paragios and R. Deriche. Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:266–280, 2000.

14. D. Cremers. A Variational Framework for Image Segmentation Combining Motion Estimation and Shape Regularization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 53–58, 2003.
15. J. Sethian. *Level Set Methods*. Cambridge University Press, 1996.
16. M. Weber, A. Blake, and R. Cipolla. Sparse Finite Elements for Geodesic Contours with Level-Sets. In *European Conference on Computer Vision*, pages 391–404, 2004.
17. N. Paragios and R. Deriche. Unifying Boundary and Region-based Information for Geodesic Active Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages II:300–305, 1999.
18. A. Yezzi, L. Zollei, and T. Kapur. A Variational Framework for Joint Segmentation and Registration. In *IEEE Mathematical Methods in Biomedical Image Analysis*, pages 44–51, 2001.
19. I. Dydenko, D. Friboulet, and I. Magnin. A Variational Framework for Affine Registration and Segmentation with Shape Prior: Application in Echocardiographic Imaging. In *IEEE Workshop in Variational and Level Set Methods*, pages 209–217, 2003. Faugeras, O. and Paragios, N. (eds).
20. A. Yilmaz, X. Li, and B. Shah. Contour based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, November 2004.
21. T. Sederberg and S. Parry. Free-Form Deformation of Solid Geometric Models. In *ACM SIGGRAPH*, volume 4, pages 151–160, 1986.
22. P. Faloutsos, M. van de Panne, and D. Terzopoulos. Dynamic Free-Form Deformations for Animation Synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 3:201–214, 1997.
23. H-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational Level Set Approach to Multiphase Motion. *Journal of Computational Physics*, 127:179–195, 1996.
24. T. Chan and L. Vese. Active Contours without Edges. *IEEE Transactions on Image Processing*, 10:266–277, 2001.
25. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. In *IEEE International Conference in Computer Vision*, pages 694–699, 1995.
26. S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *IEEE International Conference in Computer Vision*, pages 810–815, 1995.
27. T. Brox, A. Bruhn, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 158–163, 2004.