Contour-Based Registration and Retexturing of Cartoon-Like Videos

N. P. Tiilikainen[1,2] **A. Bartoli**[2,1] **S. I. Olsen**[1]
[1]DIKU, Copenhagen, Denmark
[2]LASMEA, Clermont-Ferrand, France

Abstract

Retexturing videos of deformable surfaces is an important problem in computer vision as it has a wide variety of applications. A key step in producing visually pleasing retexturing results is registration. Traditional registration methods require a certain amount of texture on the surface in order to capture all the deformation details. However, in cases such as cartoon videos, there is a high number of smooth contours and only little or spurious texture.

We propose a novel method for registering and retexturing cartoon-like videos by means of joint contour detection and point to point curve matching. The main idea is to fit a parametric 3D active surface model in the spatiotemporal space, utilizing a regularization term which limits the change in curvature over time. We show that with cross-validation it is possible to automatically estimate a suitable value for the regularization parameter, controlling the tradeoff between the regularization and the data term.

We report convincing registration and retexturing results on cartoon videos.

1 Introduction

Realistic retexturing of non-rigidly deforming surfaces is an important problem in computer vision as it has a wide variety of real world applications, especially in the movie and entertainment industry.

The main goal of our work is to perform registration and retexturing of challenging videos of non-rigid surfaces where traditional methods would fail. Such challenging cases are often found in cartoons. An example is given in figure 1 where the nose region is to be registered and retextured. This is a particularly difficult task, as there is only little or spurious texture present and the contour of the region is deforming.

For non-rigid surfaces in general the registration problem is difficult, since the shape of the scene changes between the frames. To deal with non-rigid motion Brand [5] represents the non-rigidity as a linear combination of basis shapes. Olsen *et al.* [12] suggests the use of temporal smoothness priors and surface shape priors. These methods require that the amount of texture in the scene is sufficient for feature point detection and unambiguous matching. If the observed surface is predominantly void of texture, or if the texture is poorly distributed, it is not possible to capture sufficient feature point correspondences for correctly modeling the deformations.

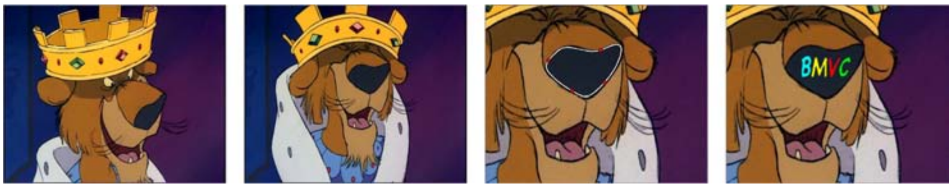

Figure 1: **Image 1 and 2 from left:** Two frames from a cartoon video showing deforming nose region and lack of texture. **Image 3:** One of the closed curves from the 3D active surface model seen encapsulating the deforming region. **Image 4:** The final retexturing.

Modeling the deformations of a sparsely textured surface has recently been investigated by Salzmann *et al.* [15]. However their approach requires that local deformation models are learned in advance.

The possibility of combining point trackers with contour trackers, to handle cases with sparse texture, has also been examined. In [1] Agarwala *et al.* uses a point tracker to track points lying on extracted contours, hereby aiding the creation of cartoon animations from real video footage. A drawback of the method by Agarwala *et al.* is that it requires hand editing of the extracted contour tracks. Bartoli *et al.* [4] takes both point and curve correspondences into account when capturing deformations. This approach partially redeems the requirement for a large number of point correspondences.

In some cases however it is not possible to capture enough point correspondences for the above mentioned method to work reliably. Such cases often arise in cartoons, where the amount of texture is limited and the number of corner points scarce. The two main contributions of this work are:

- First, §3, we present a method for registering videos relying solely on the presence of strong contours. The contour model is based on a 3D parametric active surface model. The standard active contour model was introduced by Kass *et al.* [9] and later extended to 3D [7, 11]. In our case the 3D active surface is a collection of closed curves, each lying in a separate image in the video, creating a tubular shaped surface spanning the spatiotemporal space. We introduce a novel regularization term to the energy functional of the 3D active surface that enables joint contour detection and point to point matching of the curves in the 3D active surface. We match the curves based on their spatial curvature as this is the only visually meaningful cue that is available. We assume that the region of interest is not subject to occlusions and deforms smoothly.

- Second, §4, we address one of the inherent problems with the active contour model: automatically choosing the value of the regularization parameter. The regularization parameter controls the tradeoff between the data and the regularization term. Often the value of this parameter is chosen based on empirical observations [2, 6, 13]. Instead we propose to automatically compute a suitable value for the regularization parameter by maximizing the predictivity of the 3D active surface using *k*-fold cross-validation.

The general idea of fitting active surfaces to videos for retexturing has been tried before by Collomosse *et al.* [8]. Though their use of homographies for mapping between regions limits, their method to the retexturing rigid objects.

In order to perform the final retexturing we retrieve a set of $\mathbb{R}^2 \mapsto \mathbb{R}^2$ Thin-Plate Spline (TPS) warps between a selected reference frame and the remaining frames. The data points for computing the TPS warps are extracted from the curves in our 3D active surface by uniformly sampling each curve at the same interval.

Finally experimental results demonstrating our approach on several challenging datasets are reported in §6.

2 Preliminaries and Background

Notation. Matrices are written in sans-serif, *e.g.* A and vectors in bold fonts *e.g.* $\mathbf{v}$. We denote matrix and vector transpose as A^T and $\mathbf{v}^\mathsf{T}$ and matrix inverse as A^{-1}. The operator $*$ represents convolution, I is the identity matrix and $\|\mathbf{v}\|$ represents vector two-norm. Full and partial derivatives are written using Leibniz's notation.

2.1 Active Contours

The traditional active contour is a parametric curve $\mathbf{u}(s) = [x(s), y(s)]^\mathsf{T}$, defining an $\Omega = [0,1] \mapsto \mathbb{R}^2$ mapping. The curve seeks a position where the energy functional

$$\mathscr{E}_{ac} = \int_\Omega \alpha(s) \left\| \frac{d\mathbf{u}(s)}{ds} \right\|^2 + \beta(s) \left\| \frac{d^2\mathbf{u}(s)}{ds^2} \right\|^2 + \mathscr{P}(\mathbf{u}(s)) ds \tag{1}$$

is at a minimum. The curve is regularized by penalizing its first and second derivatives, which measure the tension and bending respectively. The influence of the regularization terms are controlled by the two regularization parameters $\alpha(s)$ and $\beta(s)$. The data term is given by the potential function $\mathscr{P}$ that is derived from the image, such that its minimum values correspond to desired image features. A potential that encourages the curve to be attracted to regions with high gradient values *i.e.* edges, is $\mathscr{P} = -\|\nabla(G_\sigma * \mathscr{I})\|^2$ where $\mathscr{I}$ is a grey scale input image and G_σ a gaussian with standard deviation σ.

The active contour model is also valid in 3D [7, 11]. In 3D the standard parametric active surface is given by $\mathbf{w}(s,r) = [x(s,r), y(s,r), z(s,r)]^\mathsf{T}$ and defines a $T \times \Omega = [0,1] \times [0,1] \mapsto \mathbb{R}^3$ mapping. The potential function can either be based on true 3D image data such as medical image data [7], or composed from a series of 2D images stacked together to create a volume of 3D data. The standard 3D active surface is regularized through its first and second partial derivatives.

In both the 2D and 3D case the energy functional can be minimized in the variational framework by deriving the associated Euler-Lagrange equation, and applying an iterative gradient descent scheme. The energy functional is however not convex, and the initial solution should therefore lie close to the desired solution.

3 Joint Contour Detection and Point to Point Curve Matching

Active contours in 3D have previously been applied to segment moving objects in image sequences [8]. This can be seen as a batch processing approach to tracking, since a curve is fitted to all the frames simultaneously. One of the benefits of this approach, as opposed to the usual frame by frame tracking approach is that a higher degree of temporal consistency is possible. Our method is based on the same batch processing approach. However, instead of employing the usual regularization terms that penalize the first and second derivatives, we wish to penalize changes in curvature between frames. This is motivated by the fact that the curvature is a visually important cue for distinguishing different parts of a contour, and in our case it is the only cue we have.

3.1 The Proposed 3D Active Surface Model

The discrete time parametric surface $\mathbf{v}(s,t) = [x(s,t), y(s,t), t]^\mathsf{T}$ is represented by a sequence of closed planar curves, one for each value of the discrete temporal variable t *i.e.*, one curve in each frame. The third component of the parametric surface has been replaced by the discrete temporal variable t. This limits the deformation of the surface to two dimensions, fixing the curves of the surface to their respective frames. The surface topology is therefore that of an open ended cylinder spanning the spatiotemporal space.

The energy functional we propose is written as

$$\mathscr{E}_{as} = \int_T \int_\Omega 3\lambda \left\| \frac{\partial}{\partial t} \left(\frac{\partial^2 \mathbf{v}(s,t)}{\partial s^2} \right) \right\|^2 + (1-\lambda)\mathscr{P}(\mathbf{v}(s,t)) ds dt \qquad (2)$$

where $\lambda \in [0,1]$ is the regularization parameter controlling the tradeoff between the regularization term and the potential function, and $\mathscr{P}(\mathbf{v}(s,t))$ is the value of the potential function at position $\mathbf{v}(s,t)$. The potential function is a 3D volume composed by stacking a series of 2D images.

The regularization term, that is the first term in equation (2), is a third order mixed derivative. The inner part of the mixed derivative is a second order derivative measuring curvature, as in the classical active contour formulation of (1). The outer part is a first order derivative over t, enabling us to quantify the difference in spatial curvature along the temporal dimension. When the energy functional is minimized the regularization term penalizes pointwise curvature changes. Therefore, if one of the curves in the surface has high curvature at a specific point, then the curves in the previous and subsequent frames will try to position themselves such that the difference in curvature for this point is minimized. This of course holds for every point along the individual curves of the surface. As a consequence the parametrization of the curves is altered into being perceptually consistent.

By only penalizing the curvature difference over the temporal domain we also decouple the overall motion of the object from its local deformations.

To regulate the influence of each term in the energy functional we have the regularization parameter λ. Large values of λ causes the model to be dominated by the regularization term, forcing the curves in the surface to all have the same curvature in each point. Low values of λ however leads to the potential function being dominant. If the potential function is largely dominant, then we loose the desired property that the regularization term enforces. This also results in the surface not moving at all in parts where the potential function is non-existent.

3.2 Computing the Potential Function

The data term of the energy functional is given by the potential function $\mathscr{P}$. As the input images we are dealing with originate from cartoon clips, it is clear that a good potential can be achieved from exploiting the color information. Color segmentation is done in RGB vector space by first letting the user select a sample region in the first image and then applying the Mahalanobis distance as a similarity measure for all the pixels in all images. Often however, the segmented images contain noise *i.e.* segments that do not belong to the sample region. We remove the noise by requiring that there is an overlap between the segmented region in adjacent frames. With the noise removed we are left with a single segment in each image. We set $\mathscr{P}_t = -\|\nabla(G_\sigma * \mathscr{I}_t)\|^2$ for $t = \{1,...,T\}$ where T is the number of frames and $\mathscr{I}_t$ is the segmented binary image for frame t. The resulting 3D potential function $\mathscr{P}$ is normalized to lie in the range $[-1,0]$.

The placement of the curves making up the initial 3D active surface is computed from the segmented binary images $\mathscr{I}_t$. A circular curve is placed around the segmented region in each binary image. These curves are then connected temporally to form an initial tubular shaped 3D active surface.

3.3 Minimizing the Energy Functional

To find a local minimum of the energy functional in (2) the Euler-Lagrange equation is first derived by means of variational calculus [14]. The derived Euler-Lagrange equation has the form

$$-6\lambda \frac{\partial^6 \mathbf{v}(s,t)}{\partial t^2 \partial s^4} + (1-\lambda)\nabla\mathscr{P}(\mathbf{v}(s,t)) = \mathbf{0}. \tag{3}$$

If the derivatives are approximated by finite differences, the Euler-Lagrange equation can be solved in a manner similar to that of the traditional active contour [9]. Let the vector $\mathbf{v}$ contain all the values of $[x(s,t), y(s,t)]$ at the discrete nodal points of the surface, and let $\mathbf{f}(\mathbf{v})$ be the vector containing the corresponding values of $\nabla\mathscr{P}(\mathbf{v}(s,t))$. This enables us to write equation (3) in the more convenient matrix form

$$\mathsf{A}\mathbf{v} + \mathbf{f}(\mathbf{v}) = \mathbf{0} \tag{4}$$

where A is a block diagonal symmetric positive definite matrix. This matrix contains all the coefficients given by λ. The matrix system is solved iteratively with a gradient descent scheme, taking explicit Euler steps for the potential function and implicit Euler steps for the coefficient matrix. Let i be the evolution index and δ the gradient descent step size. The gradient descent scheme is then expressed as

$$\mathbf{v}^{i+1} = \mathbf{v}^i - \delta(\mathsf{A}\mathbf{v}^{i+1} + \mathbf{f}(\mathbf{v}^i)) \tag{5}$$

isolating the unknown $\mathbf{v}^{i+1}$ terms on the left leads to the iterative evolution equation

$$\mathbf{v}^{i+1} = (\mathsf{A} + (1/\delta)\mathrm{I})^{-1}((1/\delta)\mathbf{v}^i - \mathbf{f}(\mathbf{v}^i)). \tag{6}$$

The solution is assumed to converge when $\|\mathbf{v}^{k+1} - \mathbf{v}^k\|/n$ is below a given threshold, where n is the total number of nodal points.

4 Estimating λ With k-fold Cross-Validation

In [10] Lai and Chin suggest a local minimax approach to automatically estimating the regularization parameter. Their method is however not directly applicable to our case since they use a discrete energy algorithm as in [2].

Instead of manually tuning the regularization weight λ we propose to find a sensible value by employing k-fold cross-validation. Cross-validation has shown promising results in the estimation of regularization weights for TPS warps [4, 3]. We point out that trying to minimize the energy functional (2) over the regularization parameter would not make sense as the trivial solution $\lambda = 0$ would always be selected. Instead we maximize the predictivity of the model over λ. We choose k-fold cross-validation over leave one out cross-validation to reduce the computational expense.

To compute the k-fold cross-validation error we first partition the data *i.e.*, the potential function images, into k groups of equal size. We set $k = 10$, so for instance if we have 50 frames then each group contains 5 frames. Next the model is trained on $k-1$ groups and then evaluated on the remaining group. When we remove a group we do not simply remove the potential function frames. Instead the removed potential function frames are replaced with empty frames. This means that the evolution of the surface is governed only by the regularization term in the held out frames. Once the 3D active surface converges we measure how well the surface predicts the potential function data in the missing frames. This process is repeated for all k possible choices for the held out group and the error from the k runs is averaged. We choose the parameter λ for which the error is at a minimum. In practice we apply a heuristic combining parabolic interpolation with golden section search to find the minimum of the cross-validation error function within the bounds $[0, 1]$.

The experimental results in section 6 show that a suitable value for λ is found.

5 Implementation Details

To obtain a set of TPS warps that describe the deformations, the curve in the reference frame is uniformly sampled at some of the discrete nodal points. The corresponding nodal points in the remaining frames are then simply extracted from the remaining curves in the 3D active surface. This is possible since the parametrization is perceptually consistent

In this manner we obtain a set of J corresponding points $\mathbf{p}_j \leftrightarrow \mathbf{p}'_j$ for each frame. The TPS warps are computed on the basis of these points using the method described in [3]. This method is chosen as it also automatically computes a suitable value for the external smoothing parameter of the TPS compound cost function.

In order to do the initial retexturing of the reference frame, we let the user click 4 points and then compute the homography between these 4 points and the tattoo image that is to be pasted into the video. The tattoo is pasted onto the reference frame by means of the computed homography. Since the homography represents a projective mapping, the reference frame should be selected such that the region being retextured is deformed as little as possible. Based on these criteria the reference frame is chosen using a semi automatic heuristic that presents a selection of candidate frames and lets the user make the final decision. The candidate frames are extracted by examining the roundness and the area of the contour in each frame. We present the frames that have a high degree of roundness or a large area, as we found these to often fulfill the above criteria.

When performing the retexturing we first paste the tattoo onto the reference frame and then transfer the tattoo from the reference frame onto the remaining target frames. When transferring the tattoo from the reference frame to the target frame we cannot simply use the forward warp W as this would lead to missing pixels in the transferred tattoo. Instead the inverse warp W^{-1} is first approximated by a homography. This approximation is subsequently refined by solving a non-linear root finding problem.

To approximate the inverse warp we compute the homography H that gives the projective mapping $\mathsf{H}\mathbf{q} = \mathbf{q}'$, where $\mathbf{q}$ is a point in the target frame and $\mathbf{q}'$ is the transferred point in the reference frame. The source point $\mathbf{q}'$ that exactly map onto the target point $\mathbf{q}$ is then computed by solving the non-linear system; find $\mathbf{q}'$ such that $W(\mathbf{q}') - \mathbf{q} = \mathbf{0}$.

6 Experimental Results

We review the performance of the proposed method by applying it to real cartoon video sequences. The method has successfully been applied to several video sequences. To illustrate this we have picked two sequences, *Lilo and Stitch* and *Robin Hood*, that best as possible demonstrate the capabilities of our method. Furthermore results are shown that verify the use of k-fold cross-validation for estimating the regularization parameter λ.

6.1 Retexturing Results

Figure 2 illustrates our results at some of the different stages in the algorithm. The top row contains 4 unaltered images sampled from the *Robin Hood* sequence. In the remaining rows we have zoomed slightly in on the deforming region of interest, which in this case is the nose. The nose region exhibits non-rigid deformations while being translated, scaled and rotated. In the second row the curve correspondences have been plotted with 4 discrete nodal points marked in red. It should be noted that the red points stay in the same location along the contour of the nose throughout the sequence. The third row displays the deformed warp visualization grid. The deformation grid in the last column is only slightly deformed since this frame is close to the reference frame. In the final row the retextured frames are shown.

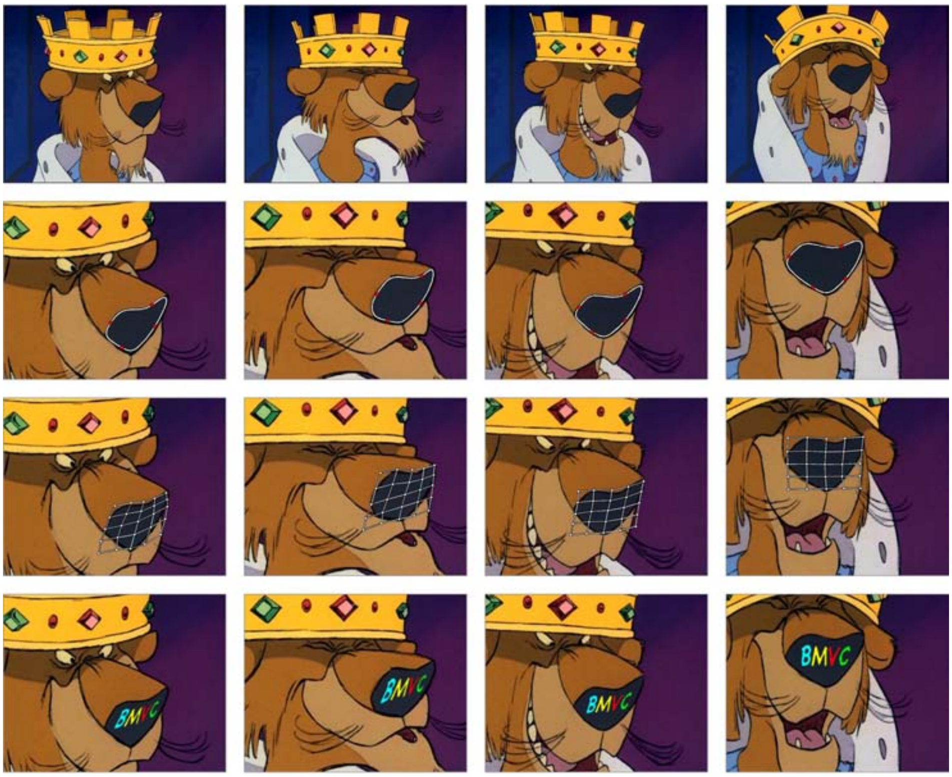

Figure 2: **First row:** 4 sample frames out of a sequence of 100. **Second row:** Corresponding curves with 4 points marked. **Third row:** Grid showing the deformations modeled by the TPS warp. **Fourth row:** The final retexturing.

Figure 3 shows 4 frames from the *Lilo and Stitch* sequence. The top row displays the unaltered frames, while the bottom row shows the final retexturing. The retexturing pattern is that of a lightbulb.

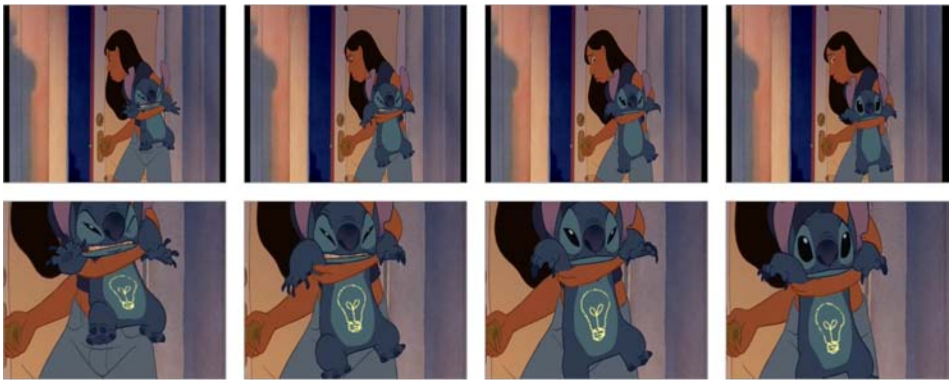

Figure 3: **Top row:** 4 sample frames out of a sequence of 40. **Bottom row:** The same 4 frames as above showing the retexturing.

6.2 *k*-fold Cross-Validation Results

In figure 4 it is confirmed that selecting the λ value which maximizes the predictivity of the model produces visually pleasing results. Setting λ too high or too low causes the curves in the 3D active surface to converge at undesired local minima, while the solution computed by cross-validation is close to what a human user would chose as being optimal. Figure 4 right side, shows the cross-validation error as a function of λ. The red dashed graph is computed on the basis of the *Robin Hood* sequence while the solid blue graph is for the *Lilo and Stitch* sequence. It is clear that the two graphs have well defined global minima, with $\lambda = 0.36$ for the *Robin Hood* sequence and $\lambda = 0.76$ for the *Lilo and Stitch* sequence. Choosing λ too high leads to the potential function energy being mostly ignored giving a solution where each curve in the 3D active surface tends to a circle. On the other hand setting λ too low stops the curves from moving if they are too far from the potential function energy. In both cases the predictivity of the model is poor. It should be noted that the results shown in figure 2 and figure 3 were obtained with the optimal cross-validation values of λ.

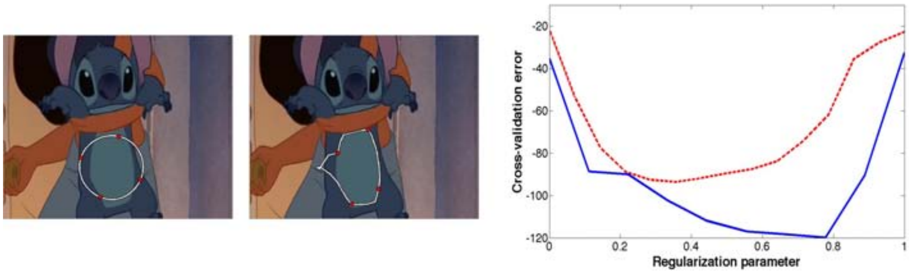

Figure 4: **Left side:** Left, an image from the *Lilo and Stitch* sequence shown with an over regularized solution. Right, same image as on the left but with an under regularized solution. **Right side:** The cross-validation error as a function of the regularization parameter. Red dashed curve; *Robin Hood* sequence, solid blue curve; *Lilo and Stitch* sequence

7 Conclusion

We developed a method for registration and retexturing of cartoon videos that only relies on the presence of strong contours. The image contours were extracted on the basis of color information and a 3D active surface was fitted. The proposed 3D active surface utilized a regularization term which made possible joint contour detection and point to point curve matching. The point to point curve matching was based on minimizing the pointwise difference in curvature between frames. A set of TPS warps were computed, between the selected reference frame and the remaining frames, from points that were uniformly sampled along the curves in the 3D active surface.

Furthermore it was shown that the regularization parameter governing the tradeoff between the regularization term and the potential function could be estimated by *k*-fold cross-validation.

In the experiments we reported visually pleasing results for the retexturing of both the *Lilo and Stitch* and *Robin Hood* sequence.

Acknowledgments The authors wish to thank Pierluigi Taddei for helpful discussions and comments.

References

[1] A. Agarwala. Snaketoonz: a semi-automatic approach to creating cel animation from video. In *NPAR*, pages 139–ff, New York, NY, USA, 2002.

[2] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.

[3] A. Bartoli. Maximizing the predictivity of smooth deformable image warps through cross-validation. *Journal of Mathematical Imaging and Vision*, 31:133–145, July 2008.

[4] A. Bartoli, E. von Tunzelmann, and A. Zisserman. Augmenting images of non-rigid scenes using point and curve correspondences. In *CVPR*, volume 1, pages 699–706, Washington, DC, USA, June 2004.

[5] M. Brand. Morphable 3D models from video. In *CVPR*, volume 2, pages 456–463, 2001.

[6] L. D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218, 1991.

[7] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, November 1993.

[8] J. P. Collomosse, D. Rowntree, and P. M. Hall. Stroke surfaces: Temporally coherent artistic animations from video. *IEEE Transactions on Visualization and Computer Graphics*, 11(5):540–549, 2005.

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[10] K. F. Lai and R. T. Chin. Regularization, formulation and initialization of the active contour models. In *ACCV*, pages 542–545, Osalsa, 1993.

[11] T. McInerney and D. Terzopoulos. A finite element model for 3D shape reconstruction and nonrigid motion tracking. In *ICCV*, Berlin, Germany, 1993.

[12] S. Olsen and A. Bartoli. Implicit non-rigid structure-from-motion with priors. *Journal of Mathematical Imaging and Vision*, 31:233–244, July 2008.

[13] N. Ray, S. T. Acton, and K. Ley. Tracking leukocytes in vivo with shape and size constrained active contours. *IEEE Transactions on Medical Imaging*, 21(10):1222–1235, October 2002.

[14] H. Sagan. *Introduction to the Calculus of Variations*. McGraw-Hill, first edition, 1969.

[15] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3D shape recovery. In *CVPR*, Anchorage, Alaska, June 2008.