



HAL
open science

Augmenting Images of Non-Rigid Scenes Using Point and Curve Correspondences

Adrien Bartoli, Eugénie von Tunzelmann, Andrew Zisserman

► **To cite this version:**

Adrien Bartoli, Eugénie von Tunzelmann, Andrew Zisserman. Augmenting Images of Non-Rigid Scenes Using Point and Curve Correspondences. 2004, pp.699-706. hal-00094765

HAL Id: hal-00094765

<https://hal.science/hal-00094765>

Submitted on 14 Sep 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Augmenting Images of Non-Rigid Scenes Using Point and Curve Correspondences

Adrien Bartoli, Eugénie von Tunzelmann and Andrew Zisserman

Department of Engineering Science, University of Oxford

{Bartoli, az}@robots.ox.ac.uk

Abstract

Our goal is to augment images of non-rigid scenes coming from single-camera footage. We do not assume any a priori information about the scene being viewed, such as for example a parameterized 3D model or the motion of the camera. One possible solution is to use non-rigid factorization of points, from which a dense interpolating function modeled by a thin-plate spline can be computed. However, in many cases, point correspondences fail to capture precisely all the deformations occurring in the scene. Examples include the eyebrows or the lips when augmenting sequences of a face. Such deformations can be captured by tracking curves, but then point correspondences are not obtained directly due to the aperture problem.

We propose an integrated method for non-rigid factorization and thin-plate spline interpolant estimation using point and curve correspondences over multiple views. The main novelties lie in the introduction of curves into the non-rigid factorization framework and in a direct global solution for the registration map, obtained by minimizing the registration error over all points and curves while taking all the images into account. The parameters of the registration map are set using cross-validation. The fidelity of the map is demonstrated by augmenting video footage undergoing various types of deformation.

1. Introduction

Augmenting images coming from pre-shot monocular footage is a major issue in the domain of special effects. One approach is to compute a dense mapping function between a reference frame and all other frames of the sequence. The augmentation is then performed by the user only on the reference image and is transferred automatically to the rest of the sequence.

When the observed scene is rigid, the problem is equiv-

alent to computing a 3D surface. Techniques such as dense matching [10] and multiple-view stereo [7], or interpolation from a set of feature correspondences, can be employed to estimate this surface.

However, the assumption of rigidity is violated in many cases of interest, such as faces changing expression or clothing deforming. The problem is then particularly challenging because a different shape is observed in each image. A major step forwards for such cases was made by Torresani *et al.* [13, 14] and Brand [3]. Building on the work of [1, 6], they developed and demonstrated factorization of non-rigid scenes, where the non-rigidity was represented as a linear combination of basis shapes.

Unfortunately, for real sequences of smooth surfaces there is often insufficient texture to establish point correspondences which capture accurately all the deformations. However, curves encapsulating such deformations *are* often available. In this paper we give a solution to this problem. We describe a method for computing a dense interpolating function between images of a non-rigid motion sequence. The mapping function is computed from both point *and* curve correspondences, using the non-rigid factorization framework. An example of using the computed mapping is shown in figure 1 (the sequence is shown in figure 5). In this case there are only a few areas where points can be reliably tracked, e.g. the corner of the eyes, but there are several curves (the hairline, the eyebrows) which may be used to determine the mapping.

In outline our method proceeds by computing an initial mapping based on point correspondences. From this mapping, we introduce virtual point correspondences, chosen such that the registration error of curves is minimized. Finally, the mapping parameters and the virtual points positions are globally tuned by minimizing a non-linear error function, and the process is iterated until the registration of curves is satisfactory. Thin-plate spline image interpolants are used for the mapping. The non-rigid factorization, based on a low-rank shape assumption, mainly serves to enforce



Figure 1. (left) Close-up of two frames of a 40 frame sequence from the film ‘Run Lola Run’. (middle) the frames overlaid with real point and curve correspondences. (right) the frames augmented with a logo on the forehead. The top row shows the reference frame.

global shape consistency by defining a low-rank subspace for reconstructing the virtual points.

This paper is organized as follows. §2 gives preliminaries and background. §3 describes our approach and §4 reports experimental results. Finally, §5 gives conclusions and extensions.

2. Preliminaries and Background

2.1. Notation

Without loss of generality, we use the first image as the reference image, i.e. the image that we will relate to all other ones to perform augmentation. We do not use homogeneous coordinates, i.e. image point coordinates are 2-vectors. Vectors and matrices are respectively typeset using bold and sans-serif fonts, e.g. \mathbf{x} and \mathbf{X} . We denote *real points* as \mathbf{x}_{ij} , for the j -th ($j = 1, \dots, m$) point in the i -th view ($i = 1, \dots, n$). Index $k = 1, \dots, l$ is used for curves, i.e. C_{ik} is the i -th image of the k -th curve. The *virtual points* are denoted \mathbf{y}_{ikp} where $p = 1, \dots, d_k$. Points on a curve are obtained by $C_{ik}(t)$, where $t \in [0, 1]$ is a parameter.

2.2. Non-Rigid Factorization

We sketch the assumptions and ideas of non-rigid factorization and the way we use it. We denote \mathbf{X} the $(2n \times m)$

matrix defined as:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_{11} & \mathbf{x}_{12} & \cdots & \mathbf{x}_{1m} \\ \mathbf{x}_{21} & \mathbf{x}_{22} & \cdots & \mathbf{x}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{n1} & \mathbf{x}_{n2} & \cdots & \mathbf{x}_{nm} \end{pmatrix}.$$

Non-rigid factorization assumes the affine camera model and that centroids have been subtracted in each image. In the rigid case, matrix \mathbf{X} has rank $r = 3$ if the scene is 3D [12], $r = 2$ if the scene is 2D. In the non-rigid case, it has been shown that \mathbf{X} can still be of low rank, e.g. [14]. This result is obtained by assuming that the observed 3D shapes are linear combinations of basis shapes. For example, if a 3D scene is observed, and is a linear combination over 2 basis shapes, then $r = 6$ since matrix \mathbf{X} factorizes as:

$$\mathbf{X} = \underbrace{\begin{pmatrix} \lambda_{11}\mathbf{P}_1 & \lambda_{12}\mathbf{P}_1 \\ \lambda_{21}\mathbf{P}_2 & \lambda_{22}\mathbf{P}_2 \\ \vdots & \vdots \\ \lambda_{n1}\mathbf{P}_n & \lambda_{n2}\mathbf{P}_n \end{pmatrix}}_{\mathbf{M}} \underbrace{\begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \cdots & \mathbf{B}_{1m} \\ \mathbf{B}_{21} & \mathbf{B}_{22} & \cdots & \mathbf{B}_{2m} \end{pmatrix}}_{\mathbf{S}},$$

where \mathbf{P}_i are the leading (2×3) submatrices of affine projection matrices, the λ_{iu} are the weights of the linear combinations and \mathbf{B}_{uj} is the u -th 3D basis point for the j -th point. We call \mathbf{M} the *motion matrix* and \mathbf{S} the *shape matrix*. This factorization can be achieved by Singular Value Decomposition $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. The motion and shape matrices are given by the r first columns of $\mathbf{U}\sqrt{\mathbf{\Sigma}}$ and the r first rows of $\sqrt{\mathbf{\Sigma}}\mathbf{V}^T$ respectively, which, in the presence of noise, corresponds to nullifying all but the r first singular values of \mathbf{X} . Recomposing the motion and shape matrices gives the reprojected points $\hat{\mathbf{X}} = \mathbf{M}\mathbf{S}$. Each column \mathbf{s} of the shape matrix \mathbf{S} corresponds to a physical point. More precisely, it is a *basis point* in \mathbb{R}^3 : it encapsulates the low-rank information which, coupled to the motion matrix \mathbf{M} , enables all images of the point to be predicted. While the basis shapes are obtained in a straightforward manner from the shape matrix, the weights and the projection matrices are difficult to extract properly from the motion matrix. Moreover, it is sometimes difficult to choose between 2D and 3D deforming scenes. Finally, the non-rigidity induces additional ambiguities in the reconstruction, see [9], and bundle adjustment needs therefore strong regularization terms.

Our algorithm relies on non-rigid factorization. However, it does not need the motion matrix to be explicitly decomposed into projection matrices and weight factors, nor the shape matrix to be cast as sets of 2D or 3D points. This avoids making an explicit choice between a 2D or 3D scene.

In practice, there is the issue of choosing the rank r . Previous solutions to this problem include [9] (using model selection), and [6] (by examining the singular values). In the

experimental section §4, we use cross-validation measures to choose r .

2.3. Thin-Plate Spline Image Interpolants

A $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ radial basis function has the form $f(\mathbf{x}) = (f^x(\mathbf{x}) \ f^y(\mathbf{x}))^\top$, where, if we define $* \in \{x, y\}$:

$$f^*(\mathbf{x}) = \alpha^* + \gamma^* x + \delta^* y + \sum_{j=1}^m w_j^* E(\|\mathbf{x}_j - \mathbf{x}\|). \quad (1)$$

A 2D thin-plate spline is obtained by choosing the basis functions as $E(\rho) = \rho^2 \log \rho$. This choice minimizes the *bending energy* [2]:

$$\Phi(f) = \iint \left(\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right) dx dy.$$

Interpolating or approximating mappings can be constructed from m point correspondences $\mathbf{x}_j \leftrightarrow \mathbf{x}'_j$. The parameters of a mapping are encapsulated into a $(m+3) \times 2$ matrix $\mathbf{h} = (\mathbf{h}^x \ \mathbf{h}^y)$ where the $m+3$ vectors \mathbf{h}^x and \mathbf{h}^y are defined by $\mathbf{h}^{*\top} = (w_1^* \ \dots \ w_m^* \ \alpha^* \ \gamma^* \ \delta^*)$.

In practice, parameters \mathbf{h} are computed by minimizing the following cost function [4]:

$$\psi(f, \mathbf{x}_j, \mathbf{x}'_j, \sigma) = \sigma \Phi(f) + \sum_{j=1}^m d^2(\mathbf{x}'_j, f(\mathbf{x}_j)).$$

The regularization parameter $\sigma \in \mathbb{R}^+$ controls the trade-off between the smoothness of the mapping and the interpolation. In the limiting case $\sigma = 0$, f interpolates the point correspondences. When $\sigma > 0$, they are approximated, and when $\sigma \rightarrow +\infty$, the mapping tends to an affine transformation.

By taking the partial derivatives of ψ with respect to \mathbf{h} , one obtains the following linear system:

$$\mathbf{C}(\mathbf{x}_j) \mathbf{h} = \mathbf{D}(\mathbf{x}'_j). \quad (2)$$

where the $(3+m) \times (3+m)$ matrix $\mathbf{C}(\mathbf{x}_j)$ depends only on the coordinates in the source image, while the $(3+m) \times 2$ matrix $\mathbf{D}(\mathbf{x}'_j)$ contains the coordinates of the points in the target image:

$$\mathbf{C}(\mathbf{x}_j) = \begin{pmatrix} \mathbf{K} + \sigma \mathbf{I} & \mathbf{P} \\ \mathbf{P}^\top & \mathbf{0} \end{pmatrix}, \quad \mathbf{D}(\mathbf{x}'_j)^\top = (\mathbf{x}'_1 \ \dots \ \mathbf{x}'_m \ \mathbf{0} \ \mathbf{0} \ \mathbf{0}),$$

where the (r, c) -th entry of matrix \mathbf{K} is $E(\|\mathbf{x}_r - \mathbf{x}_c\|)$ and the r -th row of matrix \mathbf{P} is $(1 \ \mathbf{x}_r^\top)$. The last three equations are the ‘side conditions’, which ensure that the computed mapping has square integrable second derivatives, i.e. that it behaves smoothly outside the region of interest.

The outcome is a vector valued function $f(\mathbf{x})$, which exactly maps the corresponding points $\mathbf{x}_j \leftrightarrow \mathbf{x}'_j$ onto each other (for $\sigma = 0$) and is a smooth interpolant for other points.

3. Computing the Mappings

We are given an image sequence of a deforming surface, where there may be relative motion between the surface and camera. Our goal is to recover a set of thin-plate spline mappings between the reference frame and each of the other frames of the sequence. We wish to exploit curve, as well as point, correspondences in computing these maps. Whilst mappings can be constructed from point correspondences as reviewed in §2.3, the extension to curve correspondences is not trivial.

1. *Tracking*. Compute point \mathbf{x}_{ij} and curve \mathcal{C}_{ik} tracks. Gather the points in matrix \mathbf{X} .
2. *Rank and regularization parameter selection*, §3.1. Compute the rank r to be used for non-rigid factorization and the regularization parameter σ of the thin-plate spline mappings by cross-validation.
3. *Non-rigid factorization*, §2.2. Factorize \mathbf{X} into motion \mathbf{M} and shape \mathbf{S} . The reprojections are $\hat{\mathbf{X}} = \mathbf{M}\mathbf{S}$.
4. *Thin-plate spline mappings estimation*, §2.3. Use reprojected points in matrix $\hat{\mathbf{X}}$ to estimate the f_i .
5. *Curves registration assessment*, §3.2. For each curve k , if the registration error $\mathcal{D}_k^2 > \mu^2$, introduce a virtual point on this curve, as described in §3.3. If all curves are well-registered, converge.
6. *Basis points reconstruction*, §3.4. For all virtual points, reconstruct a basis point by appending a column \mathbf{s}_{kp} to the shape matrix \mathbf{S} . Append the corresponding column to matrix \mathbf{X} .
7. *Global refinement (optional)*, §3.5. Minimize the global error function \mathcal{E} , equation (6), over the mappings and all virtual basis points \mathbf{s}_{kp} .
8. *Iteration*. Loop on 3.

Table 1. Point- and curve-based multiple-view dense registration algorithm. The curve registration threshold μ is typically chosen as 1/10 pixels.

The idea is to first use measured point correspondences throughout the sequence to compute an initial estimate of the cameras and basis points via non-rigid factorization. A thin-plate spline mapping can then be built from the estimated point correspondences between the reference image and any other frame. Note that the rank r used in the non-rigid factorization and the regularization parameter σ used for the mappings are computed at this stage by minimizing a cross-validation error, as described in §3.1. The computed thin-plate spline provides an initial estimate of the desired

map. However, this initial estimate will induce a registration error on the curves, and the map is then refined by minimizing this registration error. Our method differs from that of [4] principally in this use of points to provide an initial estimate that is consistent over multiple frames.

More precisely, the algorithm proceeds as follows. The initial parameters h_i of mappings f_i are computed using the real point correspondences \mathbf{x}_{ij} , where f_i is the mapping between the reference image and the i -th image. For each curve correspondence \mathcal{C}_{ik} , $i = 1, \dots, n$, a multi-view registration error is computed, as the mean distance between the reference curve \mathcal{C}_{1k} mapped with f_i to images $i = 2, \dots, n$ and the corresponding curves \mathcal{C}_{ik} . The computation of the curve registration error is described in §3.2. If the registration error of a curve is too high, say the distance is greater than $1/10$ pixels, this means that the image area surrounding the curve is badly mapped, and that the curve may provide useful constraints to improve the mapping. To incorporate this information, we introduce a virtual point \mathbf{y}_{ikq} on the curve, as described in §3.3. This process is iterated until the registration of all curves becomes satisfactory. The role of the virtual point correspondences, chosen on corresponding curves, is to make the mappings computed throughout the iterations register the curves better and better. In other words, the virtual point correspondences are used to incorporate into the mapping the information provided by the curve correspondences.

The iterative process is important since it ensures that all curves are well-registered by the mapping, and that the set of virtual points is minimal. The alternative solution of introducing a fixed number of virtual points on each curve is not satisfactory, since this set of points could be redundant for some curves, while other curves could be badly registered by the mapping.

Introducing point correspondences on curves yields one major problem: nothing guarantees that they are actually consistent, in the sense of being the image of a unique 3D point. We deal with this problem using the non-rigid factorization reviewed in §2.2. A similar idea is followed in [13] for the reconstruction of unreliable point correspondences. Using non-rigid factorization, we factorize the real point correspondences into a motion and a shape matrix. Given the motion matrix, instead of computing directly the positions of all virtual corresponding points \mathbf{y}_{ikq} along the curves in all images, we instead compute the corresponding 3D basis points \mathbf{B} , which makes the virtual points consistent with the low-rank shape constraint. More details are given in §3.4.

Our global refinement step turns into minimizing the distance between the virtual points predicted by the mappings and those predicted by the reconstruction, and the global registration error on curves. The cost function is described in §3.5 and the optimization procedure in appendix A. The

algorithm is summarized in table 1.

3.1. Computing the Rank and Regularization

We propose to choose the rank r used in the non-rigid factorization and the regularization parameter σ of the thin-plate splines by minimizing a cross-validation error $\tau(r, \sigma)$. Cross-validation reflects how well the model can interpolate the data. It consists of applying the algorithm to the data but leaving out one of the correspondences. The error in the predicted positions of this correspondence is then measured using the mappings computed from all the remaining correspondences. We perform non-rigid factorization, and use the reprojected points to compute thin-plate splines. The point left out is then transferred from the first view to all other views using the thin-plate splines, and the difference between its actual and predicted positions is computed. The cross-validation error $\tau(r, \sigma)$ is obtained by averaging these differences over all views and all points. By varying the rank r used in non-rigid factorization and the regularization parameter σ of the thin-plate splines, different cross-validation errors are obtained. We choose r and σ such that $\tau(r, \sigma)$ is minimized.

3.2. Assessing Curve Registration

This section deals with computing the multi-view registration error \mathcal{D}_k of a curve correspondence \mathcal{C}_{ik} , $i = 1, \dots, n$, given a multi-view mapping f_i , $i = 2, \dots, n$. As explained previously, we use a transfer error, given by the mean of the distances between $f_i(\mathcal{C}_{1k})$, the reference curve \mathcal{C}_{1k} mapped to image $i > 1$, and the corresponding curve \mathcal{C}_{ik} :

$$\mathcal{D}_k^2 = \frac{1}{n-1} \sum_{i=2}^n \varepsilon^2(f_i(\mathcal{C}_{1k}), \mathcal{C}_{ik}), \quad (3)$$

where ε is a distance measure between two curves. A natural distance measure is the average of the distances between e regularly sampled points on \mathcal{C} with parameters t_q , $q = 1, \dots, e$, and \mathcal{C}' :

$$\varepsilon^2(\mathcal{C}, \mathcal{C}') = \frac{1}{e} \sum_{q=1}^e \left(\min_{t' \in [0,1]} d^2(\mathcal{C}(t_q), \mathcal{C}'(t')) \right). \quad (4)$$

By substituting equation (4) into equation (3), we obtain:

$$\mathcal{D}_k^2 = \nu \sum_{i=2}^n \sum_{q=1}^e \left(\min_{t' \in [0,1]} d^2(f_i(\mathcal{C}_{1k}(t_{kq})), \mathcal{C}_{ik}(t')) \right), \quad (5)$$

where $\nu = \frac{1}{e(n-1)}$ is a normalizing factor. This expression can be evaluated by computing $e(n-1)$ independent one-dimensional minimisations over t' . We use the Newton algorithm with analytic differentiation. Reliable initial solutions are provided by $t' = t_{kq}$, and the constraint $t' \in [0, 1]$ is enforced by a simple clamping. Note that for certain

curve parameterizations, direct specific solutions may exist to compute \mathcal{D}_k .

3.3. Choosing the Virtual Points

At each iteration of the algorithm, the current estimated mapping is tested against each curve. Assume that the registration error \mathcal{D}_k of the k -th curve is not satisfactory. We drop the curve index k for clarity throughout this section.

A virtual point $\mathbf{y}_{1q} = \mathcal{C}_1(s_{1q})$ is introduced on the curve in the reference image (see below). The corresponding virtual points in the other views are chosen on the corresponding curves, such the mapping computed at the next iteration will tend to better match the curves. A trivial solution to chose the corresponding points \mathbf{y}_{iq} is to evaluate all images of the curve with the same parameters, i.e. $\mathbf{y}_{iq} = \mathcal{C}_i(s_{1q})$. However, this solution highly depends on the parameterisation of the curves. We prefer to choose for the \mathbf{y}_{iq} the closest points to the points predicted by the current mapping:

$$\mathbf{y}_{iq} = \mathcal{C}(s_{iq}) \text{ with } s_{iq} = \arg \min_s d^2(f_i(y_{1q}), \mathcal{C}_i(s)).$$

The virtual point introduced in the reference image is chosen such that it has a high probability to reduce the curve registration error, i.e. such the total error $\sum_{i=2}^n d^2(f_i(y_{1q}), \mathcal{C}_i(s_{iq}))$ induced by choosing the corresponding points is maximized.

3.4. Reconstructing the Basis Points

We drop the curve index k for clarity throughout this section. We tackle the problem of finding the virtual corresponding points of \mathbf{y}_{ip} , and reconstructing the underlying basis point \mathbf{s}_p . Point \mathbf{y}_{1p} lies on \mathcal{C}_1 in the reference image and has been introduced to incorporate information from this curve correspondence in the mapping. Computing the corresponding points $\mathbf{y}_{2p}, \dots, \mathbf{y}_{np}$ is difficult due to the aperture problem: they may be located anywhere along the curves $\mathcal{C}_2, \dots, \mathcal{C}_n$ in frames $2, \dots, n$ respectively. This problem is $(n-1)$ -dimensional, where n is the number of views. To reduce this high-dimensional problem, we consider the previously computed non-rigid factorization $\hat{\mathbf{X}} = \mathbf{M}\mathbf{S}$. Each column \mathbf{s} of the shape matrix \mathbf{S} corresponds to a particular point. For any virtual point \mathbf{y}_{1p} , we append a column \mathbf{s}_p to matrix \mathbf{S} . The corresponding image points are given by $(\hat{\mathbf{y}}_{1p}^\top \dots \hat{\mathbf{y}}_{np}^\top)^\top = \mathbf{M}\mathbf{s}_p$. We have reduced our $(n-1)$ -dimensional problem to a r -dimensional one, where r is the rank used for non-rigid factorization, e.g. from 39 to 3 on the 40 frame Lola sequence. The accurate estimation of \mathbf{s}_p is part of the global non-linear refinement described in the next section. We compute \mathbf{s}_p such that the distances between the predicted points $\hat{\mathbf{y}}_{ip}$ and the points \mathbf{y}_{ip} is minimized for $i = 2, \dots, n$, and such that $\hat{\mathbf{y}}_{1p} = \mathbf{y}_{1p}$:

$$\min_{\mathbf{s}_p \mid \hat{\mathbf{y}}_{1p} = \mathbf{y}_{1p}} \sum_{i=2}^n d^2(\hat{\mathbf{y}}_{ip}, \mathbf{y}_{ip}).$$

This is a linear least squares problem under linear constraints, that we solve using the framework described in e.g. [5, A3.4.4].

3.5. Non-Linear Global Refinement

We deal with the last step of our iterative algorithm, the global refinement of the mappings parameters and virtual points positions. We minimize an error function consisting of two terms:

$$\min_{\mathbf{h}_i, \mathbf{s}_{kp}} \mathcal{E} \text{ with } \mathcal{E}^2 = \mathcal{E}_{rec}^2 + \zeta^2 \mathcal{E}_{reg}^2, \quad (6)$$

where \mathbf{h}_i are the parameters of the i -th mapping f_i and ζ^2 is a weight that we currently choose as 1. The reconstruction term \mathcal{E}_{rec} and the registration term \mathcal{E}_{reg} are described below. The appendix shows how the optimization can be carried out using sparse matrix inversion.

The reconstruction term concerns the reconstruction of the basis points for the virtual points. It measures the difference between the virtual image points predicted by the reconstruction and those predicted by the mapping: $\mathcal{E}_{rec}^2 = \sum_{k=1}^l \sum_{i=2}^n \sum_{p=1}^{d_k} d^2(\hat{\mathbf{y}}_{ikp}, f_i(\mathbf{y}_{1kp}))$, where d_k is the number of virtual points on curve k .

The registration term accounts for the registration of the curves across the images. It is based on the multi-view curve registration error \mathcal{D}_k of §3.2: $\mathcal{E}_{reg}^2 \propto \sum_{k=1}^l \mathcal{D}_k^2$.

4. Experimental Results

4.1. Implementation Details

We mark real points and curves in the first image, and use the Shi-Tomasi point tracker [11] to get the real point correspondences. Curves are modeled by natural splines. We track each curve by computing a local non-rigid transformation g by minimizing an intensity-based registration error on the image patch \mathcal{X} surrounding the curve:

$$\min_{g_i} \sum_{\mathbf{x} \in \mathcal{X}} (\mathcal{I}_1(\mathbf{x}) - \mathcal{I}_i(g_i(\mathbf{x})))^2,$$

where $\mathcal{I}_i(\mathbf{x})$ is the intensity or colour of the pixel with coordinates \mathbf{x} in image \mathcal{I}_i , and g_i is a thin-plate spline with a fixed number of centres, chosen as 4 in our implementation. Transformations g_i are used to transfer the control points of the curve, which are finely tuned by maximizing the normal image gradient along the curve.

More sophisticated curve trackers, see e.g. [8], based on geodesic snakes, could be used as well.

4.2. Computing the Rank and Regularization

We use the procedure of §3.1 based on cross-validation to determine the rank r to be used in non-rigid factorization and the regularization parameter of the thin-plate splines.



Figure 2. (top row) Sample frames (5 out of 105) of the Bears sequence. (second row) the point and curve correspondences. (third row) the computed flow field. (last row) the augmented sequence.

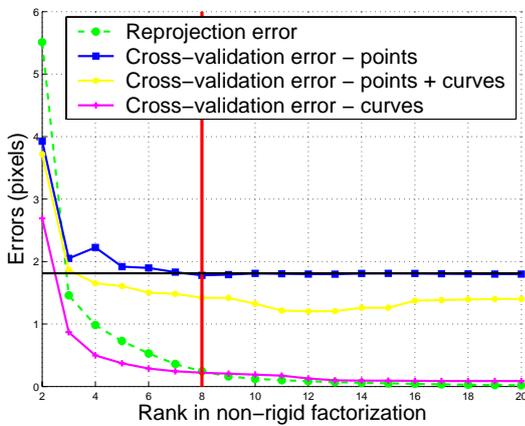


Figure 3. Cross-validation on the Bears sequence, measured against the regularization parameter σ of the thin-plate spline mappings and the rank r used in non-rigid factorization. The vertical line indicates the selected rank $r = 8$. The horizontal line indicates the cross-validation computed on raw point tracks. The cross-validation for the curves clearly shows that they improve the computed mappings.

Cross-validation also provides a quantitative analysis. We use the 105 frames of the Bears sequence shown in figure 2. We chose this sequence since many stable point and curve correspondences can be obtained. More precisely, we tracked 94 points and 6 curves.

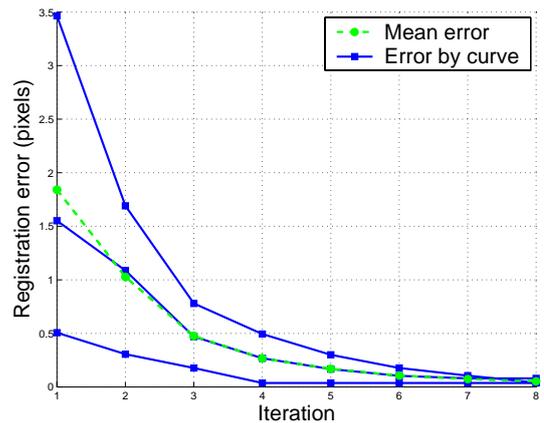


Figure 4. Multi-view curve registration error through the iterations of the Lola sequence, from top to bottom: hairline, left and right eyebrows.

Figure 3 shows the cross-validation plotted against different values of r . For each value of r , the σ minimizing the cross-validation is employed.

The same experiment run on the Lola sequence, not shown here, gave $r = 3$.

4.3. Augmenting Images

We give a qualitative evaluation of the registration algorithm by augmenting sequences. We first consider the Lola sequence, consisting of 40 frames, from which samples are given in figure 5. The 33 real points and 3 curves used are shown on figure 1. The algorithm requires a total of 7 iterations to converge, to a 0.052 pixels average registration error on the curves, and 16 virtual points to be inserted – 7 on the hairline, 6 on the left eyebrow and 3 on the right eyebrow. Figure 4 shows the evolution of the registration error for each curve and the mean over all curves, as the iterations proceed. Figure 6 shows the curves mapped from the reference image to another image of the sequence through the iterations. We observe that the quality of the mapping in the vicinity of the curves clearly improves through the iterations, until finally the mapped curve becomes indistinguishable from the tracked curve. Figure 7 shows images of the Lola sequence augmented with a logo on the forehead. There is a significant gain of quality between the result obtained using only the real points and the result obtained by applying our algorithm to real points and curves.



Figure 7. (left) The augmented reference image. (middle) The augmentation transferred based on real points. (right) Using real points and curves. In the former case, the logo is deformed and shifted towards the hairline, while in the latter case, it is centred on the forehead, as in the reference image.

5. Conclusions and Extensions

One drawback of using the reference image registration approach is that occlusions can not be handled easily. This limits the number of sequences that can be dealt with. A more explicit 3D surface representation would address this problem.

We plan to introduce a final step designed to tune the mapping's parameters based on a direct method, i.e. intensity-based, that should capture more completely all the fine deformations of the surface.

A. Sparse Optimization Algorithm

We show how to carry out the optimization of criterion \mathcal{E} , equation (6), using a sparse Levenberg-Marquart algorithm. By merging the two sums over k and on i , and incorporating the inner minimization over t' in the main outer minimization by introducing the parameters t'_{ikq} , the problem becomes:

$$\min_{\mathbf{h}_i, \mathbf{s}_{kp}, t'_{ikq} \in [0,1]} \sum_{k=1}^l \sum_{i=2}^n \left(\sum_{q=1}^e d^2(f_i(C_{1k}(t_{kq})), C_{ik}(t')) \right) + \left(\sum_{p=1}^{d_k} d^2(\hat{y}_{ikp}, f_i(y_{1kp})) \right).$$

Let J denotes the Jacobian matrix of \mathcal{E} . The Levenberg-Marquart algorithm consists in iteratively solving normal equations $(H + \theta I)\delta = -\mathbf{g}$, where $\mathbf{g} = J^T \mathbf{r}$ is the gradient and $H = J^T J$ the Gauss-Newton approximation of the Hessian matrix. Parameter $\theta \in \mathbb{R}$ is tuned heuristically. We

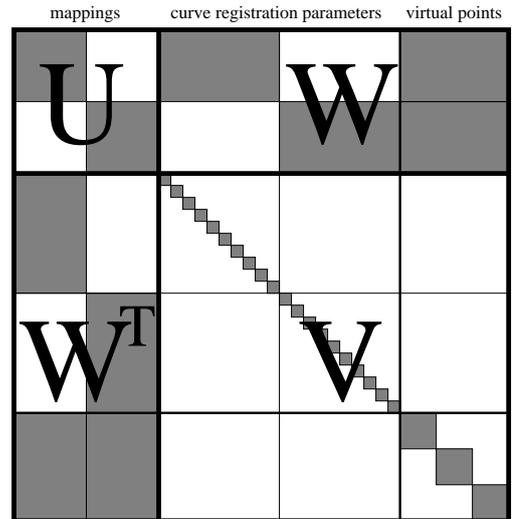


Figure 8. Shape of the Hessian matrix H (Gauss-Newton approximation) for a toy example with $n = 3$ images, $m = 3$ real points, $l = 2$ curves, $e = 5$ sampled points for curve registration error estimation, $d_1 = 2$ virtual points on curve 1, $d_2 = 1$ virtual point on curve 2 and rank $r = 3$ for non-rigid factorization.

refer to e.g. [5, A4.2] for more details. In order to solve the normal equations efficiently, we investigate the shape of matrix H , shown in figure 8. As can be seen, this matrix has



Figure 5. Sample frames (5 out of 40) from the film ‘Run Lola Run’.

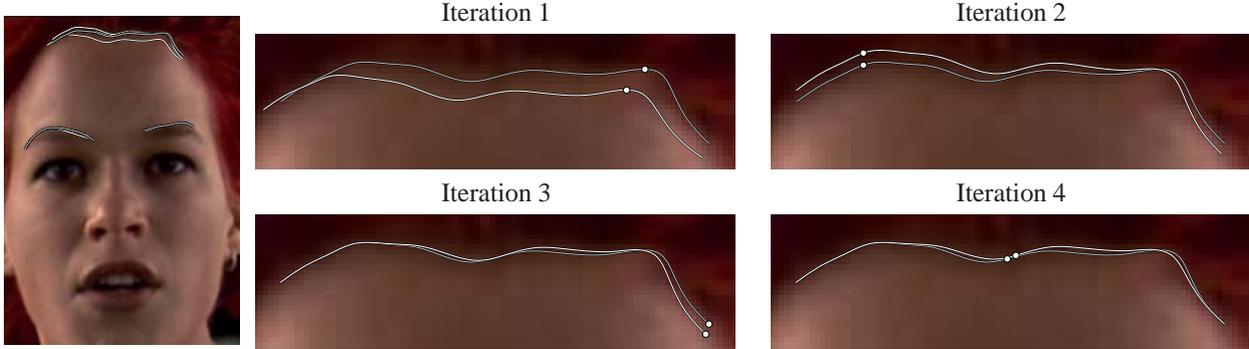


Figure 6. Evolution of the mapping of the curves through the iterations. The grey curve is the tracked curve, while the white curves are predicted by the mappings. (left) All curves. (right) close-up on the hairline shows that the mapping is improved through the iterations in the vicinity of the curve. On each image, we show the virtual point correspondence that is incorporated at the next iteration.

a strong sparse block structure that we exploit to solve the normal equations using partitioning techniques from bundle adjustment, as described in e.g. [5, A4.3], based on the U , V and W blocks shown on figure 8. Assuming that the initial solution is in the best region of convergence, we enforce the constraints $t'_{ikq} \in [0, 1]$ using a simple clamping. The derivatives are computed in a very simple manner since $C(\mathbf{x}_j)$, equation (2) is a constant matrix and hence the thin-plate spline mapping, equation (1), is linear in the unknowns.

References

- [1] B. Bascle and A. Blake. Separability of pose and expression in facial tracing and animation. *International Conference on Computer Vision, Bombay, India*, pages 323–328, 1998.
- [2] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.
- [3] M. Brand. Morphable 3D models from video. *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, pages II: 456–463, 2001.
- [4] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2):114–141, February 2003.
- [5] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, June 2000.
- [6] M. Irani. Multi-frame optical flow estimation using subspace constraints. *International Conference on Computer Vision*, volume I, pages 626–633, September 1999.
- [7] R. Koch, M. Pollefeys, and L. J. Van Gool. Realistic surface reconstruction of 3D scenes from uncalibrated image sequences. *Journal of Visualization and Computer Animation*, 11(3):115–127, 2000.
- [8] N. Paragios and R. Deriche. Geodesic active regions for tracking. *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 688–694, 1999.
- [9] H. Aanæs and F. Kahl. Estimation of deformable structure and motion. *Proceedings of the Vision and Modelling of Dynamic Scenes Workshop, Copenhagen, Denmark*, June 2002.
- [10] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, May 2002.
- [11] J. Shi and C. Tomasi. Good features to track. *Proceedings of the Conference on Computer Vision and Pattern Recognition, Seattle, Washington, USA*, pages 593–600, 1994.
- [12] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.
- [13] L. Torresani and C. Bregler. Space-time tracking. *Proceedings of the European Conference on Computer Vision, Copenhagen, Denmark*, pages 801–812, June 2002.
- [14] L. Torresani, D. Yang, G. Alexander, and C. Bregler. Tracking and modelling non-rigid objects with rank constraints. *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, 2001.