

Feature-Driven Direct Non-Rigid Image Registration

V. Gay-Bellile^{1,2}

A. Bartoli¹

P. Sayd²

¹ LASMEA, CNRS / UBP, Clermont-Ferrand, France

² LCEI, CEA LIST, Saclay, France

Vincent.Gay-Bellile@univ-bpclermont.fr

Abstract

The registration problem for images of a deforming surface has been well studied. Parametric warps, for instance Free-Form Deformations and Radial Basis Functions such as Thin-Plate Spline, are often estimated using additive Gauss-Newton-like algorithms. The recently proposed compositional framework has been shown to be more efficient, but can not be directly applied to such non-groupwise warps.

We bring two contributions. First, we propose a Feature-Driven approach making possible the use of compositional algorithms for most parametric warps such as those mentioned above. This is demonstrated by an Inverse Compositional algorithm for Thin-Plate Spline warps. Second, we propose a piecewise linear learning approach to the local registration problem. Experimental results show that the basin of convergence is enlarged, computational cost reduced and alignment accuracy improved compared to previous methods.

1 Introduction

Registering images of a deforming surface is important for tasks such as video augmentation by texture editing, non-rigid Structure-from-Motion and deformation capture. This is a difficult problem since the appearance of imaged surfaces varies due to several phenomena such as camera pose, surface deformation, lighting and motion blur. Recovering a 3D surface, its deformations and the camera pose from a monocular video sequence is intrinsically ill-posed. While prior information can be used to disambiguate the problem, see *e.g.* [8, 13], it is common to avoid a full 3D model by using image-based deformation models, *e.g.* [2, 4, 6, 10]. TPS (Thin-Plate Splines) warps are one possible deformation model proposed in a landmark paper by Bookstein [4], that has been shown to effectively model a wide variety of image deformations in different contexts, including medical images. Recent work shows that TPS warps can be estimated with direct methods, *i.e.* by minimizing the intensity discrepancy between registered images [2, 10]. The Gauss-Newton algorithm with additive update of the parameters is usually used for conducting the minimization. Its main drawback is that the Hessian matrix must be recomputed and inverted at each iteration. More efficient solutions have been proposed by Baker *et al.* [1] based on a compositional update of the parameters, and lead to a constant Hessian matrix.

Most non-rigid warps do not form groups, preventing the use of compositional algorithms since they require to compose and possibly invert the warps. Despite several attempts to relax the groupwise assumption by various approximations [8, 12, 14], there is no simple solution in the literature.

This paper brings two main contributions. The first one is the *Feature-Driven* registration concept, allowing to devise compositional algorithms by relaxing the groupwise requirement for most non-rigid warps such as Radial Basis Functions (with *e.g.* the Thin-Plate Spline [4], a multiquadric [11] or the Wendland kernel function [7]). The main idea is to control the warp by a set of *driving features* (*e.g.* for Radial Basis Function warps, the target centers are used as driving features), and to act on these features directly for operations such as warp reversion and threading, approximating inversion and composition which are not guaranteed to exist or can not be easily computed. For instance, we extend the Inverse Compositional algorithm to TPS warps. The second contribution is an improvement of Cootes’ linear learning approach [5] to local registration. Previous work assumes a linear relationship between the intensity discrepancy and the local parameter update [5, 9]. This assumption induces several practical problems, such as low alignment accuracy. We use a piecewise linear relationship for which a statistical map selection criterion is proposed. We experimentally show that it performs much better than most previous methods in terms of alignment accuracy, computational cost and enlarges the convergence basin. The combination of the Feature-Driven framework with learning-based local registration outperforms other algorithms for most experimental setups.

Notation. The images to be registered are written \mathcal{I}_i with $i = 1, \dots, n$. The template, *e.g.* the region of interest in the first image, is denoted \mathcal{I}_0 . The parameterized warp is written \mathcal{W} . It depends on a parameter vector \mathbf{u}_i for image \mathcal{I}_i and maps a point \mathbf{q} in the template to the corresponding point \mathbf{q}_i in the i -th image: $\mathbf{q}_i = \mathcal{W}(\mathbf{q}; \mathbf{u}_i)$. We write \mathcal{R} the set of pixels of interest and $\text{vect}_{\mathcal{R}}(\mathcal{M})$ the operator that vectorizes the elements of \mathcal{M} indicated in \mathcal{R} .

2 Previous Work

The registration of images of deformable surfaces has received a growing attention over the past decade. Direct registration consists in minimizing the pixel value discrepancy. Registration of an image sequence is posed as a set of nonlinear optimization problems, each of which estimating \mathbf{u}_{i+1} using the registration \mathbf{u}_i of the previous frame as an initial solution. The discrepancy function \mathcal{E} is usually chosen as the two-norm of the difference \mathcal{D} between the template and the current one, warped towards the template, *i.e.* $\mathcal{D}(\mathbf{q}) = \mathcal{I}_0(\mathbf{q}) - \mathcal{I}_{i+1}(\mathcal{W}(\mathbf{q}; \mathbf{u}_{i+1}))$, giving: $\mathcal{E}(\mathbf{u}_{i+1}) = \sum_{\mathbf{q} \in \mathcal{R}} \|\mathcal{D}(\mathbf{q})\|^2$. Other choices are possible, such as Mutual Information.

Using an additive update of the parameter vector, *i.e.* $\mathbf{u}_{i+1} \leftarrow \mathbf{u}_{i+1} + \delta$, Gauss-Newton can be used in a straightforward manner for minimizing \mathcal{E} or in conjunction with complexity tuning schemes as in [2, 10] for TPS warps. A second order approximation of \mathcal{E} , theoretically better than the Gauss-Newton one, is proposed in [3]. The major drawback of these methods is that the image gradient vector for each pixel in \mathcal{R} must be recomputed at each iteration. This is the most expensive step of the process.

A major improvement was proposed by Baker and Matthews [1] through the Inverse Compositional algorithm. The key idea is to replace the additive update by the composition of the current warp \mathcal{W}_{i+1} with the inverse $\tilde{\mathcal{W}}$ of the incremental warp: $\mathcal{W}_{i+1} \leftarrow \mathcal{W}_{i+1} \circ \tilde{\mathcal{W}}$. This leads to a constant Jacobian matrix and a constant Hessian matrix whose

inverse is thus pre-computed. This requires that the warp forms a group. In order to extend the approach to more involved models, several attempts have been made to relax the groupwise requirement. They are reviewed in §3.3. Previous work on learning-based registration are reviewed in §4.3.

3 Feature-Driven Registration

3.1 Feature-Driven Parametrization

The backbone of our approach is to represent the warp by a set of features in the current image, that we call *driving features*. These features have a fixed position in the template, depending on the type of warp that is being used. For RBF warps such as TPS warps, they can be placed anywhere, while for Free-Form Deformations, they must lie on a grid.

Henceforth, we assume that \mathbf{u}_i contains the coordinates of the driving features in \mathcal{S}_i . In our implementation, we use TPS warps, whose Feature-Driven parameterization based on points is described in §A. The target centers of the TPS are used as driving features. The Feature-Driven concept and the registration algorithm we propose are generic in the sense that they are independent of the type of warp that is being used.

In this context, the warp is essentially seen as an interpolant between the driving features. There is obviously an infinite number of such warps. The best one depends on the nature of the observed surface. Loosely speaking, matching the driving features between two images is equivalent to defining a warp since the warp can be used to transfer the driving features from one image to the other, while conversely, the warp can be computed from the driving features.

The Feature-Driven framework has two main advantages. First, it often is better balanced to tune feature positions, expressed in pixels, than coefficient vectors that may be difficult to interpret, as for TPS warps. Second, it allows one to use the efficient compositional framework in a straightforward manner. Indeed, warp composition and inversion can not be directly done for non-groupwise warps. Representing image deformations by TPS warps or Free-Form Deformations is empirical. We propose empirical means for abstracting warp composition and inversion through their driving features, called threading and reversion respectively.

3.1.1 Threading Warps

Given two sets of driving features, \mathbf{v} and \mathbf{v}' , we are looking for a third set \mathbf{v}'' defined such that threading the warps induced by \mathbf{v} and \mathbf{v}' results in the warp induced by \mathbf{v}'' , as shown on figure 1(a). We propose a simple and computationally cheap way to do it, as opposed to previous work. This is possible thanks to the Feature-Driven parametrization. The idea is to apply the \mathbf{v}' induced warp to the features in \mathbf{v} : the resulting set of features is \mathbf{v}'' . This is written: $\mathbf{v}'' = \mathcal{W}(\mathbf{v}; \mathbf{v}')$, where \mathcal{W} is meant to be applied to each feature in \mathbf{v} . In the case of TPS warps, for which we use points as driving features, threading two warps is straightforward. It is also straightforward for all other kinds of RBF warps and for FFD warps.

3.1.2 Reverting Warps

Given a set \mathbf{v} of driving features, we are looking for a set \mathbf{v}' , defined such that the warp induced by \mathbf{v}' is the reversion of the one induced by \mathbf{v} , as illustrated on figure 1(b). As for

the threading, the Feature-Driven framework makes a very simple solution possible. The idea is that applying the \mathbf{v}' induced warp to \mathbf{v} gives \mathbf{u}_0 , *i.e.* the fixed driving features in the template. This is written: $\mathcal{W}(\mathbf{v}; \mathbf{v}') = \mathbf{u}_0$. This is straightforward for TPS warps. This amounts to solving an exactly determined linear system, the size of which is the number of driving features. For some classes of warps, $\mathcal{W}(\mathbf{v}'; \mathbf{v}) = \mathbf{u}_0$ may be more practical to solve for \mathbf{v}' . Note that for all other kinds of RBF warps and FFD warps $\mathcal{W}(\mathbf{v}; \mathbf{v}') = \mathbf{u}_0$ is to be preferred as well.

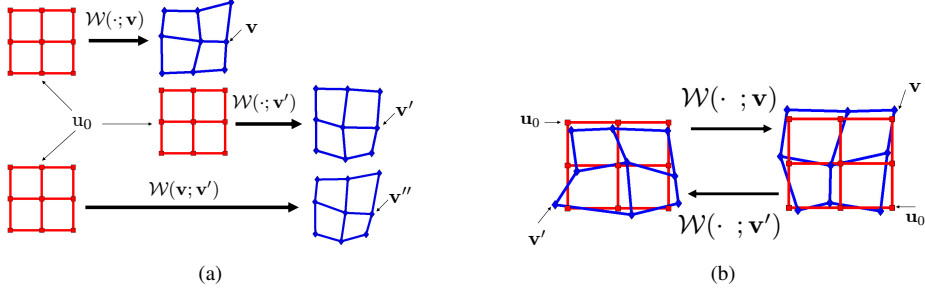


Figure 1: (a) The Feature-Driven warp threading process: \mathbf{v}'' is defined by $\mathbf{v}'' = \mathcal{W}(\mathbf{v}; \mathbf{v}')$. (b) The Feature-Driven warp reversion process: \mathbf{v}' is defined such that $\mathcal{W}(\mathbf{v}; \mathbf{v}') = \mathbf{u}_0$.

3.2 Compositional Feature-Driven Alignment

Benefiting from the Feature-Driven parameterization properties, we extend the compositional algorithms to non-groupwise warps. The following three steps, illustrated on figure 2 are repeated until convergence:

- **Step 1: Warping.** The driving features \mathbf{u}_{i+1} are used to warp the input image \mathcal{I}_{i+1} , thereby globally aligning it to the template: $\mathcal{I}_W(\mathbf{q}) = \mathcal{I}_{i+1}(\mathcal{W}(\mathbf{q}; \mathbf{u}_{i+1}))$.
- **Step 2: Local alignment.** The driving features \mathbf{u} are estimated in the warped image. This is described in §4. Note that for the Inverse Compositional algorithm, warp reversion is done at this step.
- **Step 3: Updating.** The current driving features \mathbf{u}_{i+1} and those in the warped image \mathbf{u} are combined by threading the warps to update the driving features \mathbf{u}_{i+1} in the current image: $\mathbf{u}_{i+1} \leftarrow \mathcal{W}(\mathbf{u}; \mathbf{u}_{i+1})$.

Note that previous work [8, 12, 14] requires a preliminary step before applying the update rule, as reviewed in the next section. In comparison, the Feature-Driven framework makes it naturally included into the third step.

Illumination changes are handled by normalizing the pixel value of the template and those of the warped image at each iteration.

3.3 Relation to Previous Work

Alternative approaches for non-groupwise warp composition as proposed in [8, 12, 14] consist in finding the best approximating warp for the pixels of interest in \mathcal{R} : $\mathbf{u}_{i+1} = \arg \min_{\mathbf{u}_{i+1}} \sum_{\mathbf{q} \in \mathcal{R}} \|\mathcal{W}_i(\tilde{\mathcal{W}}(\mathbf{q})) - \mathcal{W}_{i+1}(\mathbf{q})\|^2$. In [8, 12, 14] the warp is induced by a triangular mesh whose deformations are guided by a parameter vector. This minimization

problem is usually solved in two steps. First the vertices in the current image are computed using the assumption of local rigidity. They usually are not in accordance with a model instance in *e.g.* the case of 3D Morphable Models [8, 14]. Second, the parameter update is recovered by minimizing a prediction error, *i.e.* the distance between the updated vertices and those induced by the parameters. This last step may be time consuming since nonlinear optimization is required. Warp inversion is approximated with first order Taylor expansion in [12], while [14] draws on triangular meshes to avoid linearization. By comparison, our methods revert and thread warps in closed-form: they do not require optimization.

4 Local Registration

The efficiency of compositional algorithms depends on the local alignment step. For instance, the Inverse Compositional algorithm is efficient with the Gauss-Newton approximation for local alignment since this combination makes invariant the Hessian matrix. We show that learning-based local alignment fits in the Forward Compositional framework in a similar manner. Below, we propose an efficient learning-based alignment procedure. This is inspired by previous work modeling the relationship between the local increment δ and the intensity discrepancy \mathbf{D} with an interaction matrix. Using a single interaction matrix has several drawbacks, as reviewed in §4.3. We propose to learn a series $\mathcal{F}_1 \dots \mathcal{F}_\kappa$ of interaction matrices, each of them covering a different range of displacement magnitudes. This forms a piecewise linear approximation to the true relationship. Each matrix thus defines a map. A statistical map selection procedure is learned in order to select the most appropriate matrix \mathcal{F}_s given \mathbf{D} . The update vector is then given by: $\delta = \mathcal{F}_s \mathbf{D}$. Details are given below.

4.1 Learning an Interaction Matrix

An interaction matrix \mathcal{F} is learned by generating artificially perturbed versions of the template \mathcal{A}^j .

Generating training data. The driving features in the template are disturbed from their rest position with randomly chosen direction and magnitude: $\mathbf{u}^j \leftarrow \mathbf{u}_0 + \delta \mathbf{u}^j$. The latter is clamped between a lower and an upper bound, determining the area of validity of the interaction matrix. Our Feature-Driven warp reversion process is used to warp the template.

Learning. The residual vector is computed from the pixels of interest in \mathcal{R} : $\mathbf{D}^j = \text{vect}_{\mathcal{R}}(\mathcal{I}_0 - \mathcal{A}^j)$. The training data are gathered in matrices $\mathcal{U} = (\delta \mathbf{u}^1 | \dots | \delta \mathbf{u}^m)$ and $\mathcal{L} = (\mathbf{D}^1 | \dots | \mathbf{D}^m)$. The interaction matrix \mathcal{F} is computed by minimizing a Least Squares error in the image space, expressed in pixel value unit: $\mathcal{F} = \left(\mathcal{L} \mathcal{U}^\top (\mathcal{U} \mathcal{U}^\top)^{-1} \right)^\dagger$. This is one of the two possibilities for learning the interaction matrix. The other possibility is dual. It minimizes an error in the parameter space, *i.e.* expressed in pixels. Experimental results show that the former gives better results, being in particular much more robust to noise.

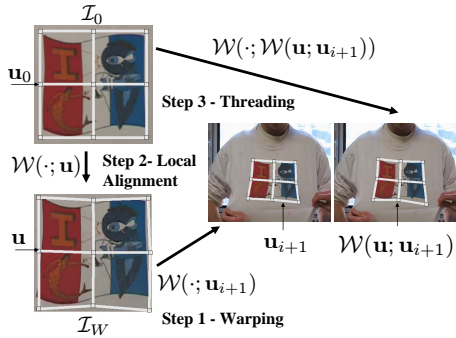


Figure 2: Compositional Feature-Driven Alignment .

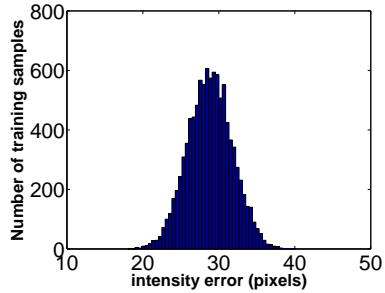


Figure 3: Distribution example of the intensity error magnitude for a perturbation interval: $[7 \dots 13]$ pixels.

4.2 Statistical Map Selection

One issue with the piecewise linear model is to select the best interaction matrix at each iteration. Each of those indeed has a specific domain of validity in the displacement magnitude which unfortunately can not be determined prior to image alignment. We propose to learn a relationship between the intensity error magnitude and the best matrix to use. We express this relationship in terms of probabilities. The intensity error magnitude for a residual error vector \mathbf{D} is defined as its RMS (Root Mean of Squares) $e(\mathbf{D})$. Note that $e^2(\mathbf{D}) \propto \mathcal{L}(\mathbf{u}_{i+1})$. Figure 3 shows that $P(\mathcal{F}_k | e(\mathbf{D}))$ closely follows a Gaussian distribution. The mean and variance of the learned intensity error magnitude are computed for each interaction matrix. Finding the most appropriate interaction matrix given the current intensity error is simply achieved by solving: $s = \arg \max_t P(\mathcal{F}_t | e(\mathbf{D}))$.

4.3 Relation to Previous Work

Learning approaches in the literature often assume that the relationship between the error image and the update parameters is linear [5, 9]. The drawback of those methods is that if the interaction matrix covers a large domain of deformation magnitudes, alignment accuracy is spoiled. On the other hand if the matrix is learned for small deformations only, the converge basin is dramatically reduced. Our piecewise linear relationship solves these issues.

Interaction matrices are valid only locally around the template parameters. Compositional algorithms are thus required, as in [9] for homographic warps. However in [5] the assumption is made that the domain where the linear relationship is valid covers the whole set of registrations. They thus apply the single interaction matrix around the current parameters, avoiding the warping and the composition steps. This does not appear to be a valid choice in practice. Our Feature-Driven framework naturally extends this approach to non-groupwise warps.

5 Experimental Results

We compare four algorithms in terms of convergence frequency, accuracy and convergence rate. Two classical algorithms:

- FA-GN: the Forward Additive Gauss-Newton approach used by [2, 10].
 - FA-ESM: the Efficient Second Order registration algorithm [3], adapted to TPS warps.
- Two algorithms we propose:
- IC-GN: the Feature-Driven Inverse Compositional registration of §3 with Gauss-Newton as local registration engine.
 - FC-Le: the Feature-Driven Forward Compositional registration of §3, with local registration achieved through learning as we propose in §4.

5.1 Simulated Data

In order to assess the algorithms in different controlled conditions, we synthesize images from a template. The driving features are placed on a 3×3 grid, randomly perturbed with magnitude r . We add Gaussian noise, with variance $\sigma\%$ of the maximum greylevel value, to the warped image. We vary each of these parameters independently, using the following default values: $r = 2$ pixels and $\sigma = 1\%$. The noise variance upper bound is 10%, which corresponds to very noisy images. Estimated warps are scored by the mean Euclidean distance between the driving features which generated the warped image, and the estimated ones. Convergence to the right solution is declared if this score is lower than one pixel. The results are means over 500 trials.

Convergence frequency. This is the percentage of convergence to the right solution. Results are shown on figures 4(a) and 4(b). FC-Le has the largest convergence basin closely followed by FA-ESM. On the other hand, FC-Le has the worst performance against noise. However, it always converges for noise amplitude below 8% and converges at 95% for 10% noise which is beyond typical practical values. IC-GN has the smallest convergence basin.

Accuracy. This is measured as the mean residual error over the trials for which an algorithm converged. Results are shown on figures 4(c) and 4(d). The four algorithms are equivalent against displacement magnitude. Concerning noise amplitude, IC-GN and FC-Le are equivalent while FA-ESM is slightly worse and FA-GN clearly worse. For example, at 6% noise, the alignment errors of IC-GN and FC-Le are around 0.2 pixels, FA-ESM at about 0.25 pixels and FA-GN alignment error at 0.35 pixels.

Convergence rate. This is defined by the number of iterations required to converge. Results are shown on figures 4(e) and 4(f). The convergence rate of FC-LE and FA-ESM are almost constant against both displacement and noise amplitudes. However FC-Le does better, with a convergence rate kept below 10. FA-GN and IC-GN are efficient for small displacements, *i.e.* below 5 pixels. The convergence rate increases dramatically beyond this value for both of them. FA-GN is also inefficient for noise amplitude over 4%. This is explained by the fact that the FA-GN Jacobian matrix depends mainly on the gradient of the current image, onto which the noise is added.

5.2 Real Data

The four above described algorithms have been compared on several videos. For two of them (*paper*, *tshirt*), we show results on table 1 and registration visualization samples on figure 5. We measure the average and maximum intensity RMS along the video, computed on the pixels of interest and expressed in pixel value unit, the total number of iterations and the computational time, expressed in seconds. All algorithms have been implemented in

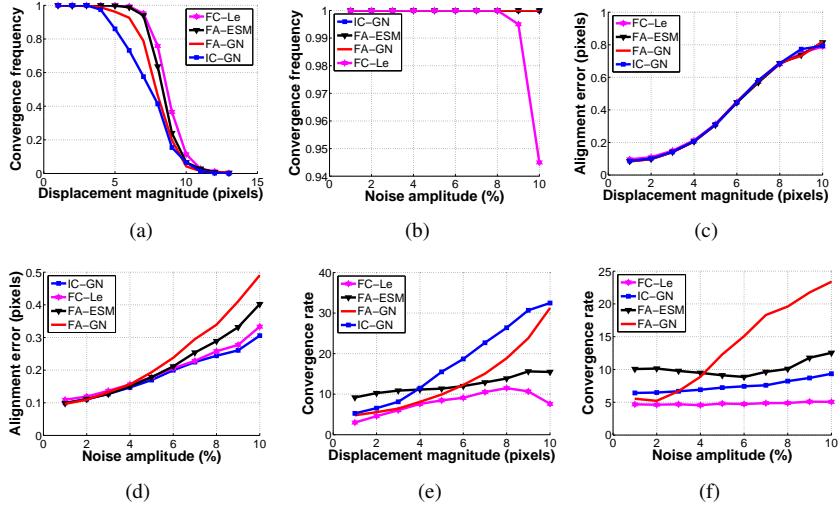


Figure 4: Comparison of the four algorithms in terms of convergence frequency (a) against displacement magnitude and (b) noise magnitude. Comparison of the four algorithms in terms of accuracy (c) against displacement magnitude and (d) noise magnitude. Comparison of the four algorithms in terms of convergence rate (e) against displacement magnitude and (f) noise magnitude.

Matlab. In practice, a great number of driving features with some amount of regularization are used, making the TPS both flexible and well-constrained. In order to illustrate the registration, we define a mesh on the template and transfer it to all other frames. Note that these meshes are different from the estimated driving features. The alignment differences between the four algorithms are visually undistinguishable, when they converge.

Summary. FA-GN is an accurate algorithm. It is however inefficient, especially for important displacements. FA-ESM has almost similar performances compared to FA-GN while being slightly more efficient. The Feature-Driven parametrization yields, via the proposed IC-GN and FC-Le algorithms, fast non rigid registration with TPS warps. While IC-GN loses effectiveness for high displacements, FC-Le has the best behavior. In fact, it is similar to FA-GN for accuracy while being 5 times faster on average and is equivalent or better than IC-GN and FA-ESM in terms of alignment accuracy, computational cost and has a larger convergence basin.

	Mean/max RMS		Iteration #		Total/mean time	
	<i>tshirt</i>	<i>paper</i>	<i>tshirt</i>	<i>paper</i>	<i>tshirt</i>	<i>paper</i>
FA-GN	8.7/13.6	8.98/17.57	9057	2422	2083/5.2	702/2.0
FA-ESM	9.2/14.7	10.22/20.49	3658	2473	877/2.2	708/2.0
IC-GN	9.7/15.8	Diverges	6231	Diverges	436/1.1	Diverges
FC-Le	6.66/12.87	9.44/19.4	3309	1330	380/0.95	176/0.5

Table 1: Results for the *tshirt* and *paper* videos. Bold indicates best performances.

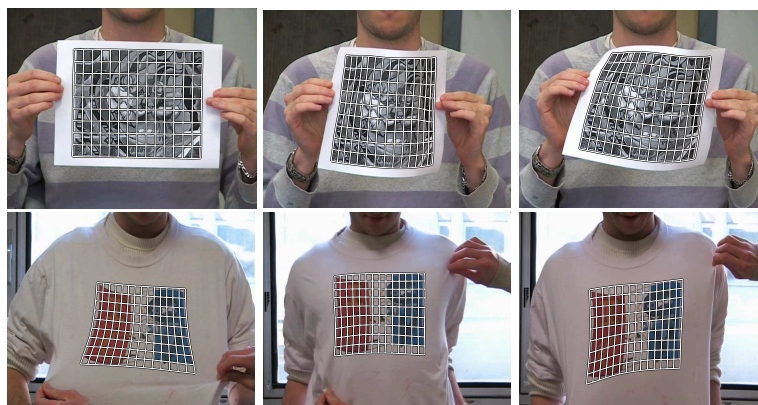


Figure 5: Registration results for FC-Le algorithm. Top: the *paper* video. Bottom: the *tshirt* video.

6 Conclusions

We addressed two important issues for the problem of non-rigid registration. First, we proposed the Feature-Driven framework, relaxing the groupwise requirement for using efficient compositional algorithms. Second, we proposed a statistically motivated piecewise linear local registration engine. Combining these two techniques results in an algorithm outperforming the other ones in terms of alignment accuracy, computational cost and having a larger convergence basin. Real-time surface registration is foreseen with this algorithm. We intend to extend the Feature-Driven approach to more complicated models, *e.g.* 3D Morphable Models [14]. This implies occlusion reasoning, that we intend to do through multiple overlapping patch registration and combination.

References

- [1] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *IJCV*, 2004.
- [2] A. Bartoli and A. Zisserman. Direct estimation of non-rigid registrations. In *BMVC*, 2004.
- [3] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IROS*, 2004.
- [4] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE PAMI*, 1989.
- [5] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *ECCV*, 1998.
- [6] T. F. Cootes, S. Marsland, C. J. Twining, K. Smith, and C. J. Taylor. Groupwise diffeomorphic non-rigid registration for automatic model building. In *ECCV*, 2004.
- [7] Mike Fornefett, Karl Rohr, and H. Siegfried Stiehl. Elastic registration of medical images using radial basis functions with compact support. In *CVPR*, 1999.
- [8] V. Gay-Bellile, M. Perriollat, A. Bartoli, and P. Sayd. Image registration by combining thin-plate splines with a 3D morphable model. In *ICIP*, 2006.
- [9] F. Jurie and M. Dhome. Hyperplane approximation for template matching. *IEEE PAMI*, 2002.

- [10] J. Lim and M.-H. Yang. A direct method for non-rigid motion with thin-plate spline. In *CVPR*, 2005.
- [11] J. A. Little, D. L. G. Hill, and D. J. Hawkes. Deformations incorporating rigid structures. *CVIU*, 1997.
- [12] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 2004.
- [13] J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *CVPR*, 2005.
- [14] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *ICCV*, 2003.

A Feature-Driven Thin-Plate Spline Warps

This parameterization is convenient for centers that remain still in the template. The TPS is an $\mathbb{R}^2 \rightarrow \mathbb{R}$ function driven by assigning target values α_k to l 2D centres \mathbf{c}_k and enforcing several conditions: the TPS is the Radial Basis Function (RBF) that minimizes the integral bending energy. It is usually parameterized by an $l + 3$ coefficient vector $\mathbf{h}_{\alpha,\lambda}^T = (\mathbf{w}^T \ \mathbf{a}^T)$ computed from the target vector α and a regularization parameter $\lambda \in \mathbb{R}^+$. The coefficients in \mathbf{w} must satisfy $\mathbf{P}^T \mathbf{w} = \mathbf{0}$, where the k -th row of \mathbf{P} is $(\mathbf{c}_k^T \ 1)$. These three ‘side-conditions’ ensure that the TPS has square integrable second derivatives. The TPS is defined by:

$$\omega(\mathbf{q}, \mathbf{h}_{\alpha,\lambda}) = \ell_{\mathbf{q}}^T \mathbf{h}_{\alpha,\lambda}, \quad (1)$$

with $\ell_{\mathbf{q}}^T = (\rho(d^2(\mathbf{q}, \mathbf{c}_1)) \cdots \rho(d^2(\mathbf{q}, \mathbf{c}_l)) \ \mathbf{q}^T \ 1)$. Combining the equations obtained for all the l centres \mathbf{c}_r with target values α_r in a single matrix equation gives:

$$\mathbf{K}_\lambda \mathbf{w} + \mathbf{P} \mathbf{a} = \alpha, \quad K_{r,k} = \begin{cases} \lambda & r = k \\ \rho(d^2(\mathbf{c}_r, \mathbf{c}_k)) & r \neq k. \end{cases} \quad (2)$$

Adding $\lambda \mathbf{I}$ acts as a regularizer. Solving for $\mathbf{h}_{\alpha,\lambda}$ using the above equation and the side-conditions is the classical linear method for estimating the TPS coefficients due to Bookstein [4]. The coefficient vector $\mathbf{h}_{\alpha,\lambda}$ is a nonlinear function of the regularization parameter λ and a linear function of the target vector α .

We write $\mathbf{h}_{\alpha,\lambda} = \mathcal{E}_\lambda \alpha$, *i.e.* as a linear ‘back-projection’ of the target vector α . Matrix \mathcal{E}_λ nonlinearly depends on λ . It is given from (2) as a function of \mathbf{K}_λ and \mathbf{P} by:

$$\mathcal{E}_\lambda = \begin{pmatrix} \mathbf{K}_\lambda^{-1} \left(\mathbf{I} - \mathbf{P} (\mathbf{P}^T \mathbf{K}_\lambda^{-1} \mathbf{P})^{-1} \mathbf{P}^T \mathbf{K}_\lambda^{-1} \right) \\ (\mathbf{P}^T \mathbf{K}_\lambda^{-1} \mathbf{P})^{-1} \mathbf{P}^T \mathbf{K}_\lambda^{-1} \end{pmatrix}.$$

This parameterization has the advantages to separate λ and α and introduces units¹. The side-conditions are naturally enforced by this parameterization.

Incorporating this parameterization into the TPS (1) we obtain what we call the *feature-driven* parameterization for the TPS: $\tau(\mathbf{q}; \alpha, \lambda) = \ell_{\mathbf{q}}^T \mathcal{E}_\lambda \alpha$. Standard $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ TPS-Warps are obtained by stacking two $\mathbb{R}^2 \rightarrow \mathbb{R}$ TPS sharing their centres and regularization parameter:

$$\mathcal{W}(\mathbf{q}; \mathbf{u}, \lambda) = (\tau(\mathbf{q}; \alpha_x, \lambda) \ \tau(\mathbf{q}; \alpha_y, \lambda))^T = (\alpha_x \ \alpha_y)^T \mathcal{E}_\lambda^T \ell_{\mathbf{q}}, \quad (3)$$

with $\mathbf{u}^T = (\alpha_x^T \ \alpha_y^T)$. Notation $\mathcal{W}(\mathbf{q}; \mathbf{u})$ is used for $\lambda = 0$.

¹While $\mathbf{h}_{\alpha,\lambda}$ has no obvious unit, α in general has (*e.g.* pixels, meters).