J Math Imaging Vis
DOI 10.1007/s10851-007-0062-1

Maximizing the Predictivity of Smooth Deformable Image Warps through Cross-Validation

Adrien Bartoli

Abstract Estimating smooth image warps from landmarks is an important problem in computer vision and medical image analysis. The standard paradigm is to find the model parameters by minimizing a compound energy including a data term and a smoother, balanced by a 'smoothing parameter' that is usually fixed by trial and error.

We point out that warp estimation is an instance of the general supervised machine learning problem of fitting a flexible model to data, and propose to learn the smoothing parameter while estimating the warp. The leading idea is to depart from the usual paradigm of minimizing the energy to the one of maximizing the predictivity of the warp, *i.e.* its ability to do well on the entire image, rather than only on the given landmarks. We use cross-validation to measure predictivity, and propose a complete framework to solve for the desired warp. We point out that the well-known non-iterative closed-form for the leave-one-out cross-validation score is actually a good approximation to the true score and show that it extends to the warp estimation problem by replacing the usual vector two-norm by the matrix Frobenius norm. Experimental results on real data show that the procedure selects sensible smoothing parameters, very close to user selected ones.

Keywords Image registration · Landmarks · Warps · Cross-validation

A. Bartoli (✉)
LASMEA, CNRS/Université Blaise Pascal, Clermont-Ferrand, France
e-mail: Adrien.Bartoli@gmail.com

A. Bartoli
DIKU, University of Copenhagen, Copenhagen, Denmark

1 Introduction

The image registration problem is important since it directly relates to numerous applications, for instance deformable surface augmentation in computer vision, see *e.g.* [21], or multimodal image fusion in medical imaging, see *e.g.* [15].

The problem has been tackled in several different ways. A commonly agreed paradigm is to minimize some compound energy including a data term and a smoother [18]. The latter is weighted so that the estimated warp is smooth but still close to interpolating the landmarks. Most of the work uses trial and error to manually set an acceptable value for this weight, called the *smoothing parameter*. The energy can obviously not be minimized over the smoothing parameter since the result would always be zero.

The purpose of this paper is to bring a simple method that jointly learns the warp and the smoothing parameter. The key idea is to make the warp as general as possible in the sense of making it able to explain the deformation of the entire image, given a restricted set of landmarks. This is different from the classical approach that makes the warp interpolate the landmarks as best as possible, given some smoothing parameter. This is strongly inspired by the machine learning paradigm of supervised learning from examples: the source image landmarks are the inputs and the target image landmarks are the corresponding outputs. In this setting, the classical approach is an empirical risk minimization algorithm. The smoothing parameter controls the model complexity since increasing smoothness decreases the number of effective model parameters.

Determining smoothing weights and other parameters such as kernel widths is a common machine learning problem. A successful approach is to consider the expected prediction error, also termed test or generalization error, which, as the smoothing weight varies, measures the bias-variance trade-off, see *e.g.* [22]. For the warp estimation problem, the generalization error can not be computed exactly since the number of landmarks is usually low and their distribution is unknown. There are however several ways to approximate the generalization error. The so-called model selection criteria such as BIC, AIC and GRIC have been successfully applied to pick up the best model in a discrete set of possible models. For instance, given two images of a rigid scene, one must choose between, say a homography and the fundamental matrix, see [14, 26]. Determining the smoothing parameter is however not a model selection problem since it does not change the actual warp model, but the estimation method.[1] A related approach is MDL, that has been used in medical image registration to register sets of multiple images, see *e.g.* [16], and for the Structure-from-Motion problem in [17].

The approach we follow is to split the data points in a training and a test set, and select the smoothing parameter for which the trained model minimizes the test error. Since the number of landmarks is usually small, we follow the approach of recycling the test set, in a leave-one-out cross-validation (LOOCV) manner. This technique was introduced in [1, 28]. It is related to the Jackknife and bootstrap techniques of sampling the dataset so that statistics can be drawn from it, and has been widely applied in machine learning, see *e.g.* [2]. For linear least squares (LLS) problems, there exists a non-iterative closed-form giving the LOOCV score. It is very close to the prediction sum of squares (PRESS) statistic and the studentized residuals.[2] We show that, while exact for the PRESS, this closed-form is actually an approximation of the LOOCV score, which turns out to be a very good approximation for typical parameter values. For the warp estimation problem, each landmark brings two equations through its two-dimensional coordinates. These two equations are said to be linked since they must be handled jointly (it would be meaningless to select one coordinate of a landmark for the training set and the other one for the test set). We show that the existing LOOCV closed-form extends to the linked measurement case by replacing the usual vector two-norm by the matrix Frobenius norm, and that this holds true for any dimension of the target space.

We point out that cross-validation is very different from the Random Sample Consensus (RANSAC) paradigm [9]. The latter trains the model using randomly sampled sets of minimal data, test on the rest of the data, and keeps the model with the largest 'consensus set'. It is meant to robustly estimate the model parameters, while cross-validation aims at quantifying the predictivity of the model. It is not obvious how RANSAC could be used to estimate image warps since there is not a clear definition of what a minimal data set is in this case. The proposed method using cross-validation is not robust, in the sense that it does not cope with mismatched landmarks.

We implement the idea of using cross-validation to register images through a parametric registration framework based on landmarks. The warp is assumed to be linear for some nonlinearly lifted source landmark coordinates. This includes warps such as Free-Form Deformations (FFD) based on tensor products [24, 25] and Radial Basis Functions (RBF), see *e.g.* [3, 10]. Experimental results are reported for Thin-Plate Spline (TPS) warps [3] which are the bending energy minimizing RBF.

Paper Organization Section 2 reviews the standard LLS estimation of warps from landmarks. Section 3 derives our approximate non-iterative closed-form to the LOOCV score and shows how it relates to the generalized cross-validation (GCV) score. Section 4 reports experimental results and Sect. 5 concludes. Finally, Appendix 1 reviews the TPS and derives our feature-driven parameterization, Appendix 2 brings a proof of the LOOCV lemma, and Appendix 3 an experimental evaluation of the closed-form LOOCV formula.

Notation Vectors are in bold fonts, *e.g.* $\mathbf{p}$, and matrices in sans-serif, *e.g.* A. Matrix, vector transpose and matrix inverse are written as in $\mathbf{p}^\mathsf{T}$, A^T and A^{-1}. Vector two-norm is denoted as in $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\mathsf{T}\mathbf{x}}$ and matrix Frobenius norm as in $\|\mathsf{A}\|_\mathcal{F} = \sqrt{\mathrm{tr}(\mathsf{A}^\mathsf{T}\mathsf{A})}$, where tr is the matrix trace operator. We stress that $\|\mathsf{A}\|_\mathcal{F}^2 = \|\mathbf{a}_1\|_2^2 + \|\mathbf{a}_2\|_2^2 + \cdots$, where $\mathbf{a}_1, \mathbf{a}_2, \ldots$ are the columns of matrix A. The real and projective spaces of dimension n are respectively written $\mathbb{R}^n$ and $\mathbb{P}^n$.

2 Landmark-Based Warp Estimation

Let $\mathbf{p} \in \mathbb{R}^2$ be a landmark coordinate vector in the source image. The warp $\mathcal{W} : \mathbb{R}^2 \times \mathbb{R}^{l\times 2} \mapsto \mathbb{R}^2$ maps a point from the source to the target image and depends on a set of parameters (often a set of l control points) in matrix $\mathsf{L} \in \mathbb{R}^{l\times 2}$ as:

$$\mathcal{W}(\mathbf{p};\mathsf{L}) \stackrel{\mathrm{def}}{=} \mathsf{L}^\mathsf{T}\nu(\mathbf{p}), \tag{1}$$

with $\nu : \mathbb{R}^2 \mapsto \mathbb{R}^l$ some nonlinear lifting function, which outputs an l-vector representing the lifted coordinates of a landmark. The lifted coordinates are linearly projected to $\mathbb{R}^2$ to give the predicted point in the target image. This general model encompasses FFDs and general RBFs. As an example, the lifting function for TPS warps is derived in Appendix 1.

[1] Another reason is that most of the model selection criteria requires that the distribution of the data point to model residuals has a known parametric form, which is clearly not the case in general for empirical smooth deformable warps.

[2] The PRESS statistic is similar to the LOOCV score but for a cost function with a data term only.

Let $\mathbf{p}_j \leftrightarrow \mathbf{q}_j$, $j = 1, \dots, m$ be m landmark correspondences between the two images. Let ϵ_j be some random variable representing the noise and the deviation between the physics and the warp model, *i.e.* $\mathbf{q}_j = \mathcal{W}(\mathbf{p}_j; \mathsf{L}) + \epsilon_j$, from which the mean sum of squared residuals (MSR) is:

$$\mathcal{E}_d^2(\mathsf{L}) \overset{\text{def}}{=} \frac{1}{m} \sum_{j=1}^{m} \|\mathcal{W}(\mathbf{p}_j; \mathsf{L}) - \mathbf{q}_j\|_2^2.$$

It plays the role of a *data term* as it measures the transfer error, *i.e.* the discrepancy between the predicted and the measured target landmarks. It is used in conjunction with a smoother $\mathcal{E}_s$ in a compound cost function:

$$\mathcal{E}^2(\mathsf{L}; \mu) \overset{\text{def}}{=} \mathcal{E}_d^2(\mathsf{L}) + \mu^2 \mathcal{E}_s^2(\mathsf{L}),$$

with μ the smoothing parameter. The smoothing term is usually based on partial derivatives of the warp, such as the second derivatives:

$$\mathcal{B}^2(\mathsf{L}) \overset{\text{def}}{=} \int_{\mathbb{R}^2} \left\| \frac{\partial^2 \mathcal{W}}{\partial \mathbf{p}^2}(\mathbf{p}; \mathsf{L}) \right\|_{\mathcal{F}}^2 d\mathbf{p}. \tag{2}$$

Other examples are elastic registration which uses spring terms [6] and fluid registration which uses viscosity [4]. A different way of controlling the smoothness is to directly change the number of warp parameters, such as the number of control points in FFD-based registration [24]. Brownian warps are proposed in [20] along with a smoother constraining the estimated warp to be invertible [19]. Depending on the warp being used, the integral in (2) needs to be discretized. Note that using TPS warps allows to solve the integral in closed-form, as is shown in Appendix 1. We assume that it can anyway be fairly approximated by a discrete differential operator or any other matrix operator, and define:

$$\mathcal{E}_s^2(\mathsf{L}) \overset{\text{def}}{=} \|\mathsf{ZL}\|_{\mathcal{F}}^2 \approx \mathcal{B}^2(\mathsf{L}). \tag{3}$$

The compound cost function thus writes as:

$$\mathcal{E}^2(\mathsf{L}; \mu) = \frac{1}{m} \|\mathsf{NL} - \Xi\|_{\mathcal{F}}^2 + \mu^2 \|\mathsf{ZL}\|_{\mathcal{F}}^2$$

with

$$\mathsf{N}^\mathsf{T} \overset{\text{def}}{=} \begin{pmatrix} \nu(\mathbf{p}_1) & \cdots & \nu(\mathbf{p}_m) \end{pmatrix}$$

and

$$\Xi^\mathsf{T} \overset{\text{def}}{=} \begin{pmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_m \end{pmatrix}.$$

Using the matrix Frobenius norm is a natural choice since, as the vector two-norm, it is based on summing squared matrix or vector elements. Given the smoothing parameter μ, the warp parameters $\hat{\mathsf{L}}(\mu)$ are solved for through:

$$\hat{\mathsf{L}}(\mu) = \arg\min_{\mathsf{L}} \mathcal{E}^2(\mathsf{L}; \mu) \tag{4}$$

$$= \arg\min_{\mathsf{L}} \left\| \begin{pmatrix} \mathsf{N} \\ \sqrt{m}\mu\mathsf{Z} \end{pmatrix} \mathsf{L} - \begin{pmatrix} \Xi \\ \mathsf{0} \end{pmatrix} \right\|_{\mathcal{F}}^2$$

$$= \underbrace{(\mathsf{N}^\mathsf{T}\mathsf{N} + m\mu^2\mathsf{Z}^\mathsf{T}\mathsf{Z})^{-1}\mathsf{N}^\mathsf{T}}_{\mathsf{T}(m\mu^2)} \Xi. \tag{5}$$

The *influence matrix* T maps the target landmark coordinates in Ξ to the warp coefficients $\hat{\mathsf{L}}$ and plays an important role in the cross-validation technique given in the next section. We note that the matrix Frobenius norm naturally allows handling the linked equations induced by the two dimensions of landmark coordinates.

3 Maximizing Predictivity by Cross-Validation

The idea of cross-validation is to approximate the generalization error by splitting the data in a training and a test set, and average the test error over several such partitionings. There are different kinds of cross-validations, including leave-one-out (LOOCV), v-fold and generalized cross-validation (GCV). The two latter ones are usually preferred for computation efficiency. We use LOOCV and show that it can be very efficiently approximated in closed-form, for models in the form (1). The formula for LOOCV is the same as for the PRESS [1], except that the hat matrix is replaced by the *influence matrix*, incorporating the smoother, and that an approximation needs to be made in the derivation.

The LOOCV score is defined as a function of the smoothing parameter μ:

$$\mathcal{E}_g^2(\mu) \overset{\text{def}}{=} \frac{1}{m} \sum_{j=1}^{m} \|\mathcal{W}(\mathbf{p}_j, \hat{\mathsf{L}}_{(j)}(\mu)) - \mathbf{q}_j\|_2^2, \tag{6}$$

where $\hat{\mathsf{L}}_{(j)}(\mu)$ are the model parameters estimated with all but the j-th landmark:

$$\hat{\mathsf{L}}_{(j)}(\mu) \overset{\text{def}}{=} \arg\min_{\mathsf{L}} \mathcal{E}_{(j)}^2(\mathsf{L}; \mu). \tag{7}$$

We therefore have to solve the following nested optimization problem to get the most predictive solution $\hat{\mathsf{L}}$, obtained by plugging the optimal $\hat{\mu}$ in (4), giving:

$$\hat{\mathsf{L}} \overset{\text{def}}{=} \arg\min_{\mathsf{L}} \mathcal{E}^2(\mathsf{L}; \arg\min_{\mu} \mathcal{E}_g^2(\mu)).$$

At first glance, the LOOCV score seems computationally expensive, making its minimization over μ extremely costly if not infeasible in a reasonable amount of time on a standard computer. It turns out that there actually is a non-iterative closed-form for the LOOCV score which does not require solving the system m times as a trivial, greedy application of (6) requires.

The closed-form is based on the so-called LOOCV lemma, demonstrated in Appendix 2. Consider an LLS problem, and a reduced problem with only a subset of the measurements. The lemma says that replacing in the full dataset problem the measurements by their prediction with the solution to the reduced problem makes the solution to this modified problem the same as for the reduced one. In other words, define $\tilde{\Xi}^j$ as Ξ except that the j-th row, corresponding to the j-th landmark, is replaced by the prediction $\mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu))^{\mathsf{T}}$, *i.e.*:

$$\tilde{\Xi}^j \overset{\text{def}}{=} \Xi - \mathbf{e}_j(\mathbf{q}_j - \mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu)))^{\mathsf{T}}, \tag{8}$$

with $\mathbf{e}_j$ a zero vector with one at the j-th entry and $\hat{\mathsf{L}}_{(j)}(\mu)$ the solution to the reduced problem. The lemma states:

$$\hat{\mathsf{L}}_{(j)}(\mu) = \mathsf{T}\big((m-1)\mu^2\big)\tilde{\Xi}^j. \tag{9}$$

In other words, the solution is a constant linear function of a slightly modified right-hand side matrix. Although it could seem weird that the unknown estimate $\hat{\mathsf{L}}_{(j)}(\mu)$ is used to make a prediction in order to artificially create a problem to solve for this estimate, it actually is essential for deriving the non-iterative closed-form we are aiming at, as is clearly shown below.

Recall that matrix T maps the target landmarks to the model parameters while matrix N maps the model parameters to the predicted landmarks. We therefore construct the influence matrix H which maps the target landmarks to the predicted ones as:

$$\mathsf{H}(\gamma) \overset{\text{def}}{=} \mathsf{N}\mathsf{T}(\gamma) = \mathsf{N}(\mathsf{N}^{\mathsf{T}}\mathsf{N} + \gamma\mathsf{Z}^{\mathsf{T}}\mathsf{Z})^{-1}\mathsf{N}^{\mathsf{T}}.$$

Matrix H has size $(m \times m)$, *i.e.* it has as many rows and columns as there are landmark correspondences, and is symmetric. We write $\mathbf{h}_j(\gamma)$, $j = 1, \ldots, m$ the columns (or rows) of $\mathsf{H}(\gamma)$. This allows us to write:

$$\mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}(\mu)) = \Xi^{\mathsf{T}}\mathbf{h}_j(m\mu^2)$$

$$\mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu)) = (\tilde{\Xi}^j)^{\mathsf{T}}\mathbf{h}_j((m-1)\mu^2).$$

Taking the difference between the two equations and approximating $\mathbf{h}_j \overset{\text{def}}{=} \mathbf{h}_j(m\mu^2) \approx \mathbf{h}_j((m-1)\mu^2)$ gives:

$$\mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}(\mu)) - \mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu)) \approx (\Xi - \tilde{\Xi}^j)^{\mathsf{T}}\mathbf{h}_j.$$

Substituting the definition (8) of $\tilde{\Xi}^j$ gives:

$$\mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}(\mu)) - \mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu)) \approx \mathbf{h}_j^{\mathsf{T}}\mathbf{e}_j(\mathbf{q}_j - \mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu))).$$

Writing $h_{j,j} \overset{\text{def}}{=} \mathbf{h}_j^{\mathsf{T}}\mathbf{e}_j$ the diagonal elements of $\mathsf{H}(m\mu^2)$, we get:

$$\mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}(\mu)) - \mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu)) \approx h_{j,j}(\mathbf{q}_j - \mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu))),$$

that we rearrange to:

$$h_{j,j}\mathbf{q}_j + (1 - h_{j,j})\mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu)) \approx \mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}(\mu)).$$

Subtracting $\mathbf{q}_j$ on each side gives:

$$h_{j,j}\mathbf{q}_j + (1 - h_{j,j})\mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu)) - \mathbf{q}_j \approx \mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}(\mu)) - \mathbf{q}_j,$$

$$(1 - h_{j,j})(\mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu)) - \mathbf{q}_j) \approx \mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}(\mu)) - \mathbf{q}_j,$$

$$\mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}_{(j)}(\mu)) - \mathbf{q}_j \approx \frac{1}{1 - h_{j,j}}(\mathcal{W}(\mathbf{p}_j; \hat{\mathsf{L}}(\mu)) - \mathbf{q}_j).$$

We thus have obtained an analytical, non-iterative expression giving each term in the sum for the LOOCV score (6), that we can rewrite as:

$$\mathcal{E}_g^2(\mu) \approx \frac{1}{m}\left\| \mathrm{diag}\left(\frac{1}{\mathbf{1} - \mathrm{diag}(\mathsf{H}(m\mu^2))}\right)(\mathsf{N}\hat{\mathsf{L}}(\mu) - \Xi)\right\|_{\mathcal{F}}^2, \tag{10}$$

where $\mathrm{diag}(\mathsf{M})$ is a vector containing the diagonal entries of matrix M and $\mathrm{diag}(\mathbf{v})$ is a diagonal matrix with as diagonal entries the elements of vector $\mathbf{v}$, and $\mathbf{1}$ is a vector of ones.

Minimizing the LOOCV score is done through the closed-form (10). Most of the methods in the literature are specific to the GCV score, which uses the approximation $\mathrm{diag}(\mathsf{H}(m\mu^2)) \approx \mathrm{tr}(\mathsf{H}(m\mu^2))\mathrm{I}$, with I the identity matrix, which allows simplifying the closed-form $\mathcal{E}_g$ further, see [27]. The minimization problem however remains nonlinear, and most of the methods for the GCV score can be applied to the LOOCV score, eventhough it is often neglected in the literature. Possible methods range from golden section search [5] and sampling (with optional local polynomial interpolation), *e.g.* [12, 13]. We tried several different methods. Most of them find the correct minimum in all cases. The fastest one is downhill simplex, which has typical computation times of less than half a second for $m \approx 50$ landmarks and $l \approx 25$ deformation centres on a standard PC running our MATLAB implementation.[3] This computation time, although not prohibitive, is much higher than that of a straightforward fitting of the warp, given the smoothing parameter.

The approximation based on $\mathbf{h}_j = \mathbf{h}_j(m\mu^2)$ in the above derivation allows to derive the closed-form (10). We call it the m-approximation. We compared its value against the direct evaluation of (6), giving the 'true' LOOCV score, on a bunch of typical values, and with another candidate approximation using $\mathbf{h}_j = \mathbf{h}_j((m-1)\mu^2)$, called the

[3] The downhill simplex or Nelder-Mead algorithm is implemented within the `fminsearch` MATLAB function.

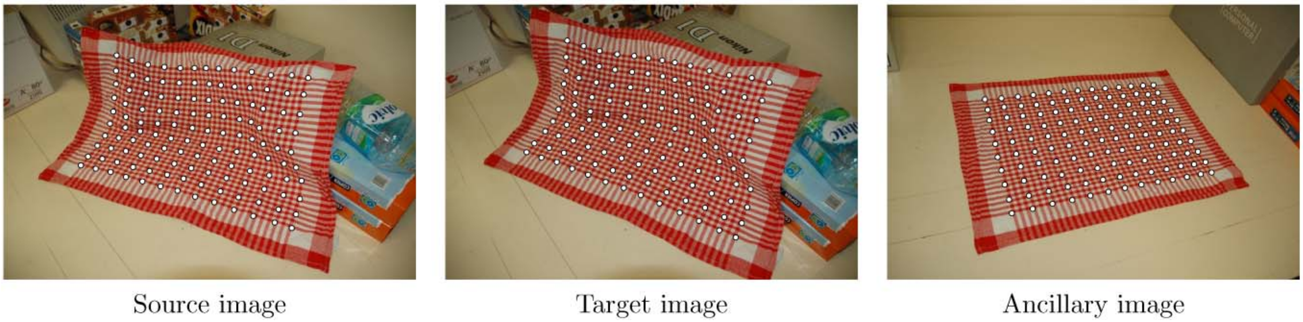

Source image Target image Ancillary image

Fig. 1 (*left*, *middle*) The source and target images in the dishcloth dataset overlaid with the 130 manually clicked point correspondences. (*right*) The ancillary image, showing the dishcloth flat, which is used to create a warp visualization grid, as shown on Fig. 2

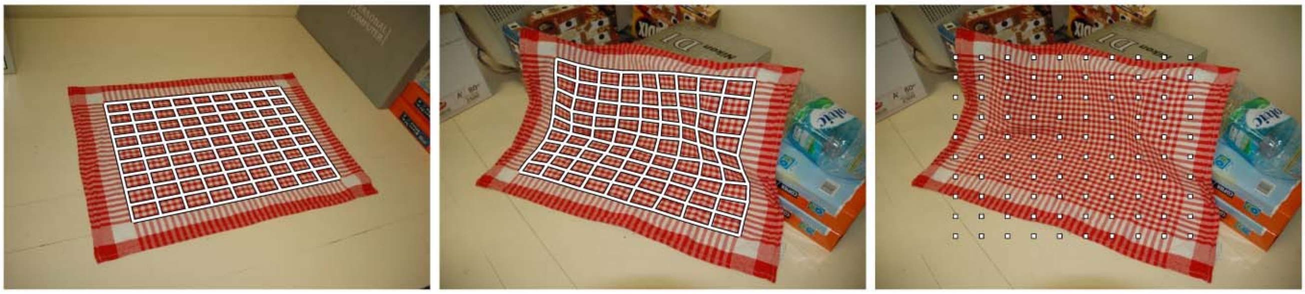


Fig. 2 (*left*) The ancillary image showing the warp visualization grid over the region of interest. (*middle*) The warp visualization grid transferred from the ancillary image to the source image. (*right*) The 10 × 10 deformation centre grid in the source image

$(m-1)$-approximation. The results are reported in Appendix 3. Our conclusions are that there is no significant difference between the two approximations, albeit that the m-approximation has a better behavior than the $(m-1)$-approximation in the sense that its minimum is located closer to the true one, and has the same value as the true minimum LOOCV score. We also tried approximations based on $\mathbf{h}_j = \mathbf{h}_j((\eta m + (1-\eta)(m-1))\mu^2) = \mathbf{h}_j((m-1+\eta)\mu^2)$ with $\eta \in [0,1]$ – none of them did better than the m-approximation.

4 Experimental Results

We evaluated our algorithm on several datasets. For three of them we show results. Most of the other methods in the literature assume that the smoothing parameter is given and estimate the warp parameters, whereas the proposed algorithm estimate both the warp and smoothing parameters.

4.1 The Dishcloth Dataset

This dataset has three images of a dishcloth for which 130 corresponding points were manually marked, see Fig. 1. The two images that we use for testing our algorithm show the dishcloth with the same deformation but from a different viewpoint. The point correspondences cover the entire dishcloth, which remains entirely visible. This dataset is thus easy in the sense that many point correspondences are available and that the two images are quite similar.

The third image in this dataset shows the dishcloth flatten on a table, and is called the ancillary image. It is used to create a warp visualization grid in the source image, as shown on Fig. 2 and explained below. First, we mark the four corners of the region of interest in the ancillary image, and use them to create a homography of $\mathbb{P}^2$ mapping the canonical unit square to these four corners. This is used to transfer a regular grid from the canonical unit square to the ancillary image. Second, we use the point correspondences to compute a deformable warp from the ancillary to the source image, and use it to transfer the visualization grid to the source image. This visualization grid, although similar to the data points, is very useful to visualize the behavior of the warp independently of the actual data points.

We proceed to register the images and use a regular grid of 10 × 10 deformation centres, as shown in Fig. 2. Figure 3 shows the LOOCV score and the RMSR as functions of the smoothing parameter μ. We observe that they both asymptotically tend to respectively the PRESS statistic and RMSR of a two-dimensional affine image transformation, which we measured to be respectively 3.14 pixels and 3.06 pixels. The minimization finds the LOOCV optimal smoothing parameter $\hat{\mu} \approx 0.55$. Zooming onto the graph shows that

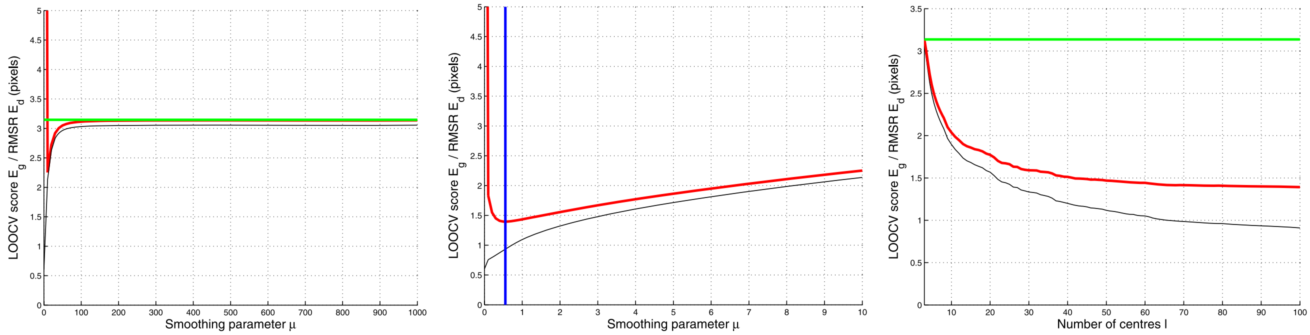
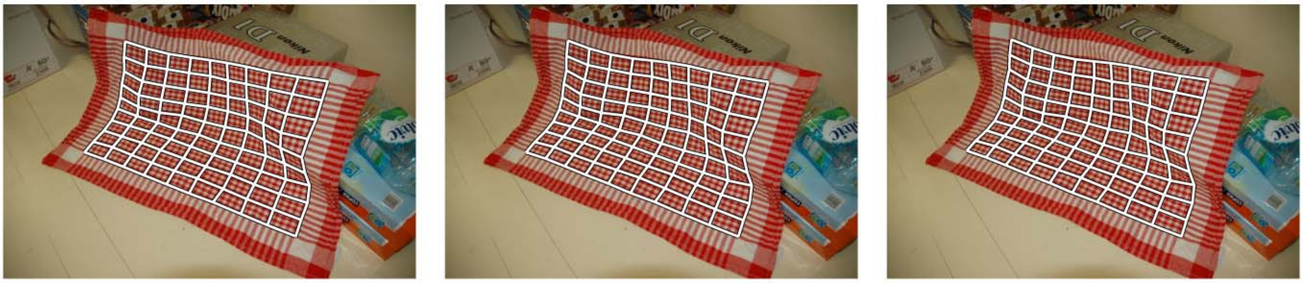


Fig. 3 (*left*) The LOOCV score (*thick*, *red curve*) and RMSR (*thin*, *black curve*) as functions of the smoothing parameter μ—the *green horizontal line* is the PRESS for an affinity. (*middle*) Zoom onto the left graph—the *blue vertical line* shows the selected optimal smoothing parameter $\hat{\mu}$. (*right*) The mean LOOCV score (*thick*, *red curve*) and RMSR (*thin*, *black curve*) as functions of the number of deformation centres l



LOOCV optimal solution
$\mu = \hat{\mu} \approx 0.55$
RMSR ≈ 0.93 pixels
LOOCV ≈ 1.40 pixels

Over-smoothed solution
$\mu = 10\hat{\mu} \approx 5.53$
RMSR ≈ 1.77 pixels
LOOCV ≈ 1.91 pixels

Extremely over-smoothed solution
$\mu = 10,000$
RMSR ≈ 3.06 pixels
LOOCV ≈ 3.14 pixels

Fig. 4 The visualization grid predicted by the warp for (*left*) the LOOCV optimal solution parameter, (*middle*) an exaggerated smoothing parameters and (*right*) an extreme smoothing parameter corresponding to the asymptotically affine behavior of the warp

it actually has a shallow minimum. This is explained by the fact that this dataset is 'simple', in the sense that the image deformation is limited. Once a sufficient amount of smoothness is reached, it is not that critical to oversmooth. As expected, the RMSR is a monotonic function of μ: the smoother the warp, the lower the effective number of parameters and so the higher the training RMSR error. Figure 3 also shows the LOOCV score and the RMSR as functions of the number of centres l. These curves were obtained by randomly sampling centres in the convex hull of the source landmarks, and for each set of centres, finding the smoothing parameter minimizing the LOOCV score. It can be seen that both the LOOCV score and the RMSR are decreasing functions of m. This is explained by the fact that since an adaptive smoothing parameter is used, adding more parameters can not degrade the quality of the warp, since the extra parameters just get smoothed out. This means that without any prior information, the number of deformation centres should be chosen large. On this particular example, it is clear that choosing more than 40 deformation centres, say, does not bring a significant improvement to the quality of the warp.

Finally, Fig. 4 shows the visualization grid transferred to the target image, for different smoothing parameters. As was expected from the shape of the LOOCV score in Fig. 3, oversmoothing has a limited effect on the estimated warp. Note however that the LOOCV score grows by more than a third, from 1.40 pixels to 1.91 pixels, when 10 times the optimal smoothing parameter is used.

4.2 The Paper Sheet Dataset

This dataset has two images of a paper sheet shown in Fig. 5. One of the images shows the paper sheet flat, with substantial radial distortion. The other image shows the paper smoothly bent in such a way that a self-occlusion shows up, *i.e.* part of the surface is being occluded by itself. We manually clicked 53 points on both images as shown in Fig. 5. We clicked the four corners or the paper sheet in the flat paper image, and, as for the dishcloth ancillary image, created a regular visualization grid. It is used to visually assess the quality of an estimated warp.

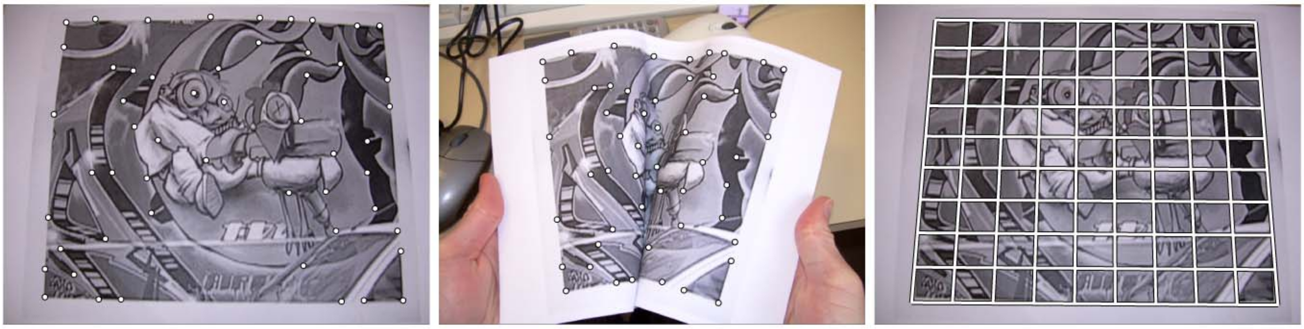


Fig. 5 (*left*, *middle*) The source and target images of the self-occluding paper sheet dataset overlaid with the 53 manually clicked point correspondences. (*right*) The warp visualization grid covering the region of interest

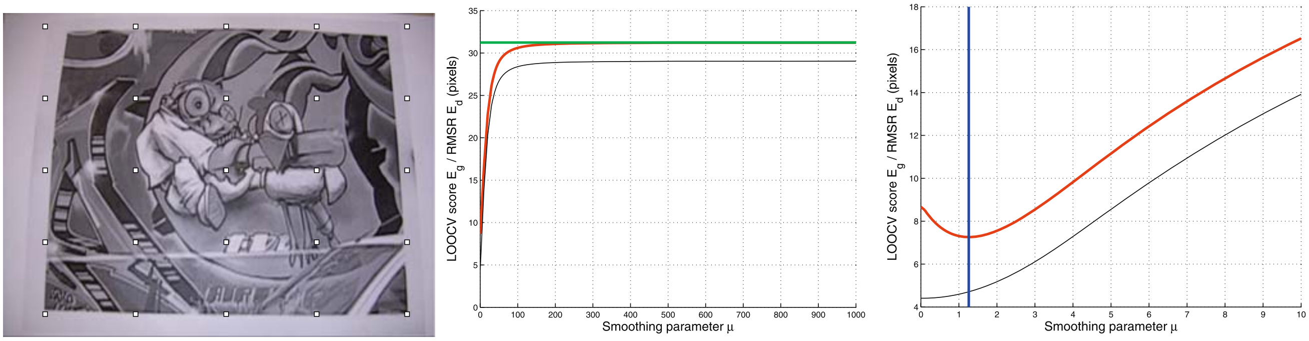


Fig. 6 (*left*) The 5 × 5 deformation centre grid. (*middle*) The LOOCV score (*thick*, *red curve*) and RMSR (*thin*, *black curve*) as functions of the smoothing parameter μ—the *green horizontal line* is the PRESS for an affinity. (*right*) Zoom onto the middle graph—the *blue vertical line* shows the selected optimal smoothing parameter $\hat{\mu}$

This dataset is much more difficult than the dishcloth dataset, in the sense that due to surface self-occlusion in the target image, a large part of the region of interest visible in the source image disappears in the target image.

Figure 6 shows the 5 × 5 grid of deformation centres we selected over the source image. This figure also shows the LOOCV score as a function of the smoothing parameter μ. We observe that it asymptotically converges to the PRESS score for an affine image transformation, which is 31.24 pixels. The RMSR has the same behavior in that it converges to the RMSR for the affine transformation (not shown on the graph), which is 29.05 pixels. Zooming onto the beginning of the LOOCV curve shows that it actually has a well defined minimum, and that this is what our algorithm selects as optimal smoothing parameter $\hat{\mu}$.

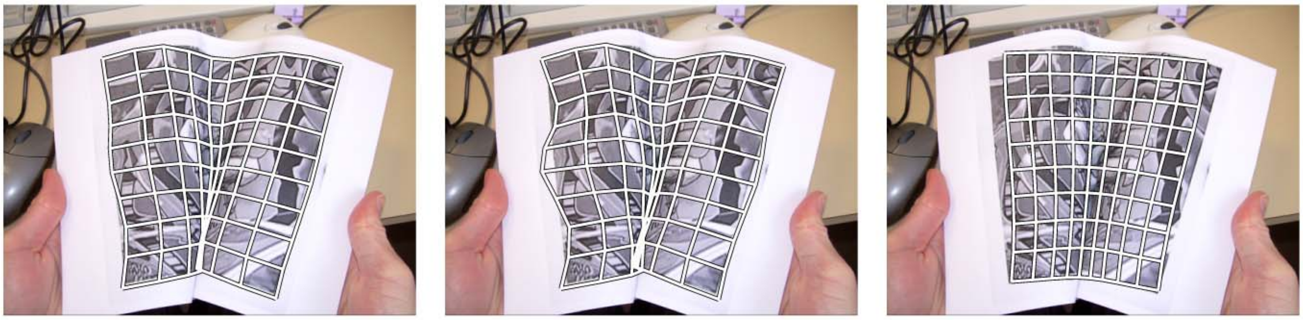
The LOOCV optimal warp we obtain is shown on Fig. 7. An under- and an over-smoothed solutions are also shown for comparison. The selected $\hat{\mu}$ clearly corresponds to what one would have chosen by tweaking, since it is visually very satisfying.

We observed that the LOOCV score and the RMSR are, as for the dishcloth example, monotonic decreasing functions of the number of deformation centres l (this graph is not shown). Choosing $l = 25$ for this particular example is a sensible choice.

4.3 The Spine Dataset

This dataset is extracted from the one used in [7]. It consists of lateral, lumbar spine X-ray images, similar to the pair of example images shown in Fig. 8 for two different patients. Each image has been annotated by experienced radiologists who placed 6 points on the corners and in the middle of the vertebra endplates. This provides a total of 36 landmarks since L1 to L4, and the 2 neighboring vertebrae are used in every image. They also manually marked the outlines of the L1 to L4 vertebrae in each image. As can be seen from the outlines, the vertebrae show different degrees of fracture at follow up. On the source image, L3 and L4 show a moderate biconcave deformity, while on the target image, L1 shows a severe wedge deformity.

We estimated a warp with a grid of 3 × 3 deformation centres, as show on Fig. 8. The LOOCV score we obtained is 12.60 pixels and the RMSR is 9.71 pixels. This is quite high, as the noticeable discrepancy between the target and predicted landmarks shows.



LOOCV optimal solution
$\mu = \hat{\mu} \approx 1.26$
RMSR ≈ 4.72 pixels
LOOCV ≈ 7.27 pixels

Under-smoothed solution
$\mu = 0$
RMSR ≈ 4.41 pixels
LOOCV ≈ 8.67 pixels

Over-smoothed solution
$\mu = 10\hat{\mu} \approx 12.58$
RMSR ≈ 15.97 pixels
LOOCV ≈ 18.53 pixels

Fig. 7 The visualization grid predicted by the warp for (*left*) the LOOCV optimal smoothing parameter, (*middle*) no smoothing at all and (*right*) an exaggerated smoothing parameter

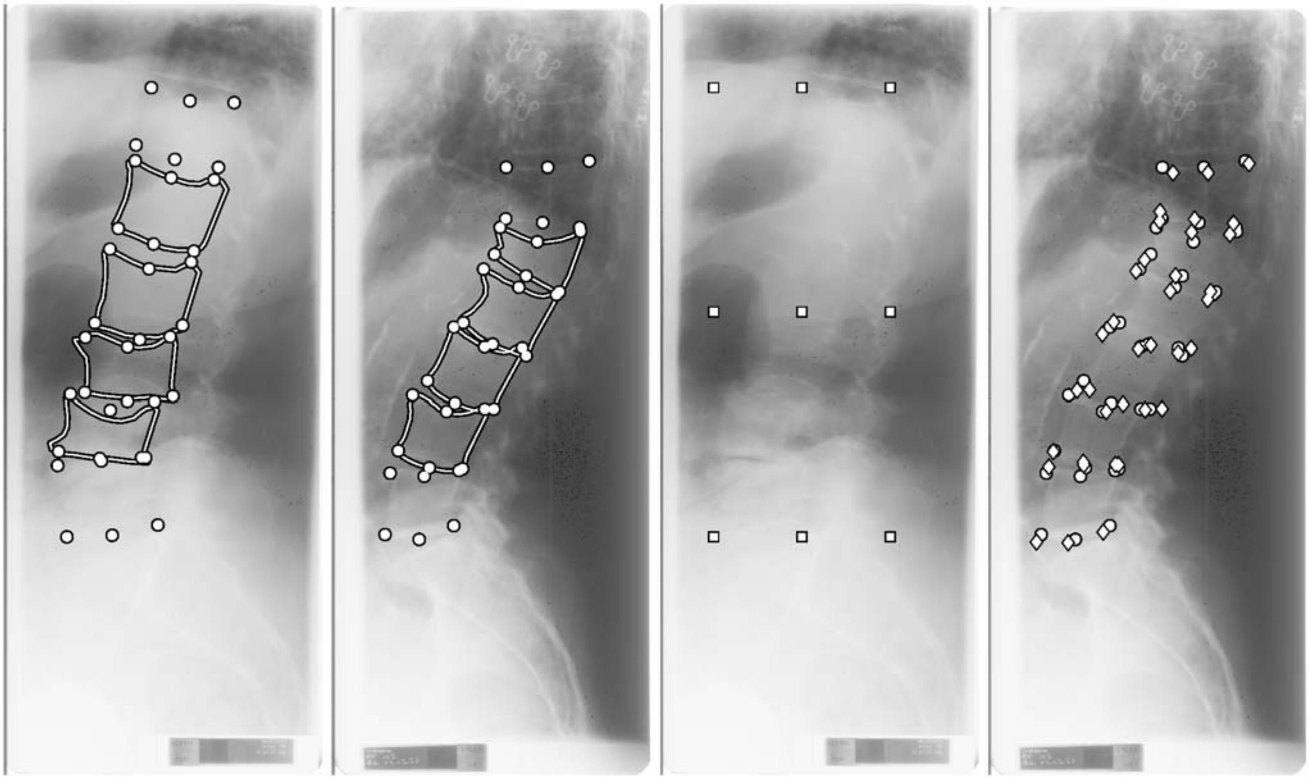
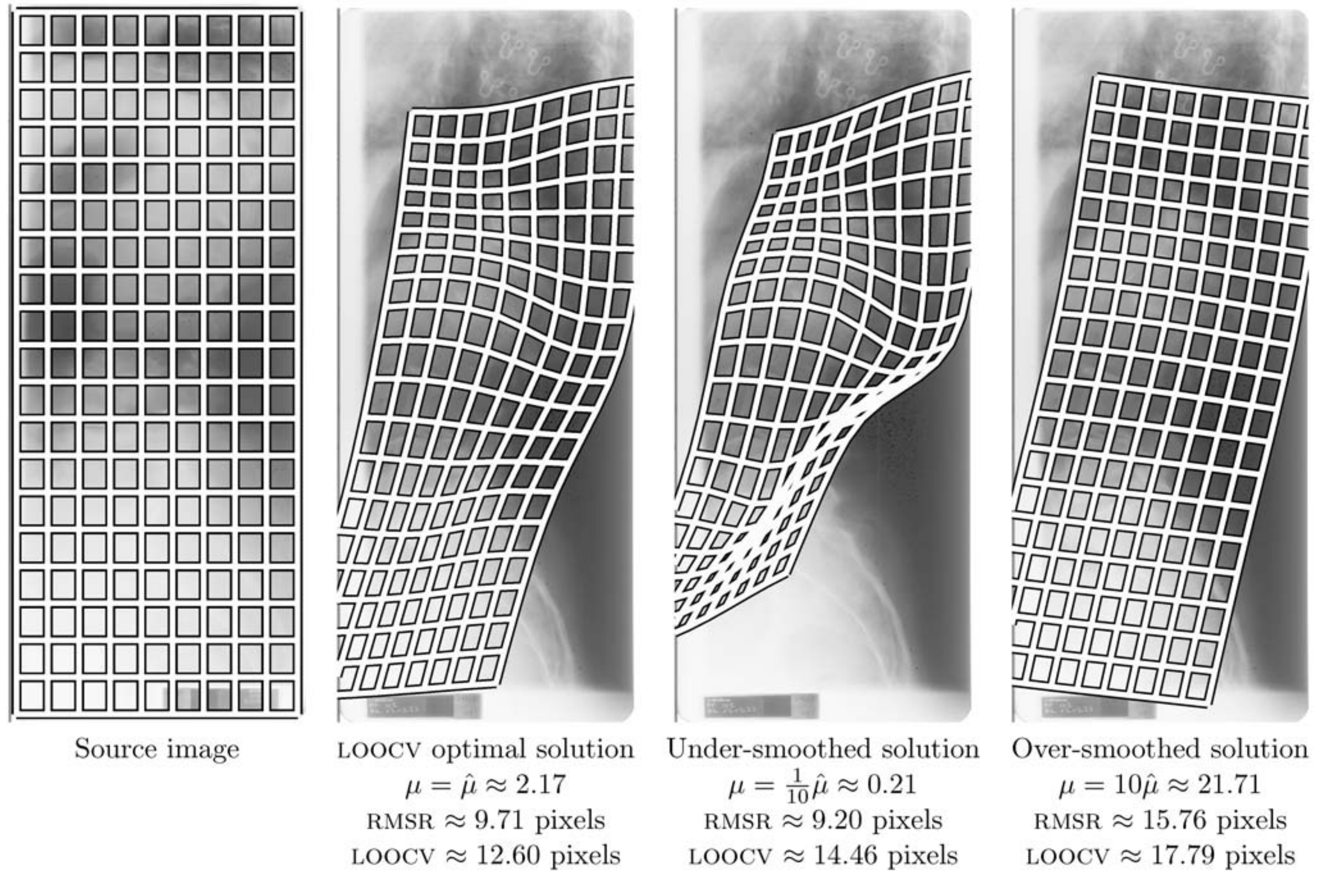


Fig. 8 (from *left to right*) The source and target images overlaid with the 36 landmarks (*circles*) and the vertebrae boundaries (see main text for details), the 3 × 3 grid of deformation centres we use in the source image, and the landmarks predicted by the LOOCV optimal warp (*diamonds*)

Figure 9 shows the visualization grid for different values of the smoothing parameter. We observe that the LOOCV optimal solution has a nice visual behavior. The under-smoothed one almost folds on itself, while the over-smoothed one is very rigid.

Figure 10 shows the LOOCV score and RMSR as functions of the smoothing parameter. The LOOCV has a well-defined minimum $\hat{\mu} \approx 2.17$, and the curves have the same shape as for the two previous datasets. These two graphs however show a novel curve, representing the target to transferred boundary distance. This is computed as follow. Given a warp estimate, we transfer each point on the source boundary to the target image, and measure the distance to the closed point onto the target boundary. Averaging over the source boundary points gives the 'boundary error'. What we observe is that this boundary error has a minimum,



Source image

LOOCV optimal solution
$\mu = \hat{\mu} \approx 2.17$
RMSR ≈ 9.71 pixels
LOOCV ≈ 12.60 pixels

Under-smoothed solution
$\mu = \frac{1}{10}\hat{\mu} \approx 0.21$
RMSR ≈ 9.20 pixels
LOOCV ≈ 14.46 pixels

Over-smoothed solution
$\mu = 10\hat{\mu} \approx 21.71$
RMSR ≈ 15.76 pixels
LOOCV ≈ 17.79 pixels

Fig. 9 (*from left to right*) The source image with the visualization grid, the LOOCV optimal smoothing parameter, an under-smoothed and an over-smoothed solutions

which is located at a slightly lower value than the LOOCV optimal smoothing parameter. We tried different combinations of images: the LOOCV score and the boundary error exhibited the same behavior in all cases, *i.e.* minimizing the LOOCV score slightly overestimates the location of the minimum for the boundary error. Recall that the warp, and thus the LOOCV score, are computed only from the 36 landmarks. This means that these landmarks and the vertebra outlines are strongly correlated, which was expected since those landmarks actually form the basis for classical semi-quantitative vertebra fracture grading strategies, see [11].

Figure 11 shows the vertebra boundaries, and allows one to visually compare the marked and the predicted boundaries in the target image. It is seen that the LOOCV optimal and the under-smoothed solutions are both visually satisfying. They obviously fail to capture all the subtle shape changes, but account for the main deformations. This was expected since the LOOCV minimum over-estimates the boundary error minimum. The over-smoothed solution clearly misses important shape changes.

5 Conclusion

We described a framework for estimating a deformable image warp from landmarks based on a compound cost function including a data term and a smoother. The method, based on leave-one-out cross-validation, automatically determines the smoothing parameter balancing the data term and the smoother. We showed that a simple closed-form solution exists for computing the leave-one-out cross-validation score given the smoothing parameter, and minimize it with a downhill simplex algorithm, yielding reasonable computation time, typically much less than a second. We report convincing experimental results on various datasets.

Generally speaking, one possible issue with leave-one-out cross-validation is the “testing-on-training data” problem. This does not occur with the kind of data we use in this paper since the landmarks are usually sparse, but should be considered if more data are available, *e.g.* a pixel-wise displacement field, by using for instance an exclusion zone around each training point. There also exist pathological cases, for which the leave-one-out cross-validation score has

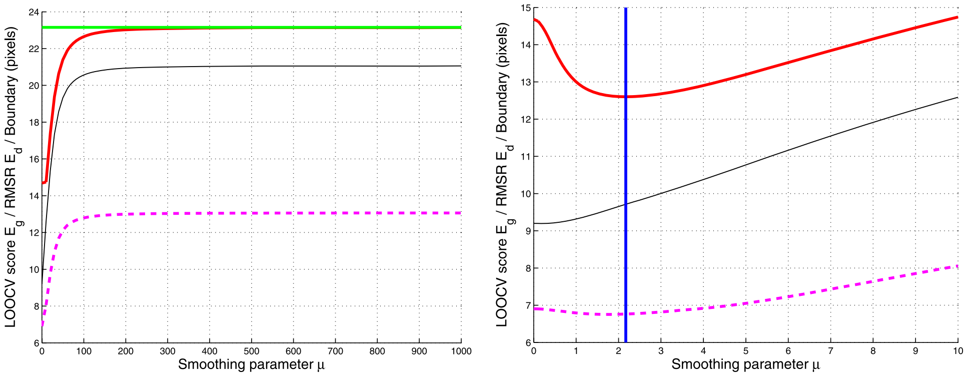
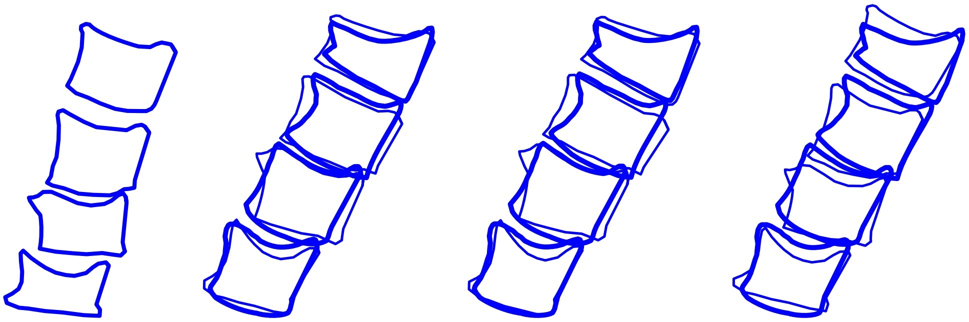


Fig. 10 (*left*) The LOOCV score (*thick*, *red curve*) and RMSR (*thin*, *black curve*) as functions of the smoothing parameter μ, as well as the boundary transfer error (*thick dashed*, *purple curve*)—the *green horizontal line* is the PRESS for an affinity. (*right*) Zoom onto the left graph—the *blue vertical line* shows the selected optimal smoothing parameter $\hat{\mu}$



Source LOOCV optimal solution Under-smoothed solution Over-smoothed solution

Fig. 11 (*from left to right*) The vertebra boundaries in the source image, and in the target image. The manually marked boundary is shown (*thick curve*), as well as the one transferred by the warp from the source image (*thin curve*) for different smoothing parameters

several local minima. How to find the optimal minimum in practice in a guaranteed manner is an open research topic.

Acknowledgements I want to thank Vincent Gay-Bellile for the self-occluding paper dataset, Florent Brunet for pointing me some references, Rabia Granlund and the authors of [7], in particular Mads Nielsen, for the spine dataset, and also Søren Olsen who has made my regular visits to Copenhagen possible.

Appendix 1: The Thin-Plate Spline

The TPS is an $\mathbb{R}^2 \to \mathbb{R}$ function driven by assigning target values α_k to control centres $\mathbf{c}_k$ with $k = 1, \ldots, l$ and enforcing several conditions: the TPS is the Radial Basis Function that minimizes the integral bending energy. The idea of using the thin-plate equation as an interpolation map is due to Duchon [8]. Standard $\mathbb{R}^2 \mapsto \mathbb{R}^2$ TPS-Warps are obtained by stacking two TPSs sharing their centres, as proposed by Bookstein [3]. This is described below, along with our feature-driven parameterization.

6.1 Standard Parameterization

The TPS is usually parameterized by an $l + 3$ coefficient vector $\boldsymbol{\eta}^\mathsf{T} = (\mathbf{w}^\mathsf{T}\ \mathbf{a}^\mathsf{T})$ and an internal smoothing parameter $\lambda \in \mathbb{R}^+$. There are l coefficients in $\mathbf{w}$ and three coefficients in $\mathbf{a}$. These coefficients can be computed from the $(l \times 1)$ target vector $\boldsymbol{\alpha}$. The TPS is given by:

$$\omega(\mathbf{p}, \boldsymbol{\eta}_{\boldsymbol{\alpha},\lambda}) \overset{\text{def}}{=} \left(\sum_{k=1}^{l} w_k\ \rho(d^2(\mathbf{p}, \mathbf{c}_k))\right) + \mathbf{a}^\mathsf{T}\tilde{\mathbf{p}}, \qquad (11)$$

where $\rho(d) = d \log(d)$ is the TPS kernel function for the squared distance and $\tilde{\mathbf{p}}^\mathsf{T} = (\mathbf{p}^\mathsf{T}\ 1)$. The coefficients in $\mathbf{w}$ must satisfy $\tilde{\mathsf{C}}^\mathsf{T}\mathbf{w} = \mathbf{0}$, where the k-th row of $\tilde{\mathsf{C}}$ is $\tilde{\mathbf{c}}_k$. These three 'side-conditions' ensure that the TPS has square integrable second derivatives. It is convenient to define the $(l + 3)$-vector $\boldsymbol{\ell}_\mathbf{p}$ as:

$$\boldsymbol{\ell}_\mathbf{p}^\mathsf{T} \overset{\text{def}}{=} (\rho(d^2(\mathbf{p}, \mathbf{c}_1)) \cdots \rho(d^2(\mathbf{p}, \mathbf{c}_l))\ \tilde{\mathbf{p}}^\mathsf{T}), \qquad (12)$$

allowing the TPS (11) to be rewritten as a dot product:

$$\omega(\mathbf{p}, \boldsymbol{\eta}_{\boldsymbol{\alpha},\lambda}) = \boldsymbol{\ell}_\mathbf{p}^\mathsf{T}\boldsymbol{\eta}_{\boldsymbol{\alpha},\lambda}. \qquad (13)$$

Equation (12) thus represents the first step in the nonlinear lifting function making the TPS-warp fit in the general warp definition (1) used in this paper.

6.2 Standard Estimation

Applying the TPS (11) to the centre $\mathbf{c}_r$ with target value α_r gives:

$$\left(\sum_{k=1}^{l} w_k\ \rho(d^2(\mathbf{c}_r, \mathbf{c}_k))\right) + \mathbf{a}^\mathsf{T}\tilde{\mathbf{c}}_r = \alpha_r.$$

Combining the equations obtained for all the l centres with the side-conditions $\tilde{\mathsf{C}}^\mathsf{T}\mathbf{w} = \mathbf{0}$ in a single matrix equation gives:

$$\underbrace{\begin{pmatrix} \mathsf{K}_\lambda & \tilde{\mathsf{C}} \\ \tilde{\mathsf{C}}^\mathsf{T} & 0 \end{pmatrix}}_{\mathcal{D}} \underbrace{\begin{pmatrix}\mathbf{w}\\ \mathbf{a}\end{pmatrix}}_{\boldsymbol{\eta}_{\boldsymbol{\alpha},\lambda}} = \begin{pmatrix}\boldsymbol{\alpha}\\ \mathbf{0}\end{pmatrix}$$

$$\text{with } K_{r,k} \overset{\text{def}}{=} \begin{cases}\lambda & r=k,\\ \rho(d^2(\mathbf{c}_r,\mathbf{c}_k)) & \text{otherwise.}\end{cases}$$

Adding $\lambda\mathsf{I}$ to the leading block of the design matrix $\mathcal{D}$ to give K_λ acts as an internal smoother. An *ad hoc* method for finding λ is described in [23]. Solving for $\boldsymbol{\eta}_{\boldsymbol{\alpha},\lambda}$ by inverting $\mathcal{D}$ is the classical linear method for estimating the TPS coefficients [3]. The coefficient vector $\boldsymbol{\eta}_{\boldsymbol{\alpha},\lambda}$ is thus a nonlinear function of the internal smoothing parameter λ and a linear function of the target vector $\boldsymbol{\alpha}$.

6.3 A Feature-Driven Parameterization

We express $\boldsymbol{\eta}_{\boldsymbol{\alpha},\lambda}$ as a linear 'back-projection' of the target value vector $\boldsymbol{\alpha}$. This is modeled by the matrix $\mathcal{E}_\lambda$, nonlinearly depending on λ, given by the l leading columns of $\mathcal{D}^{-1}$:

$$\boldsymbol{\eta}_{\boldsymbol{\alpha},\lambda} = \mathcal{E}_\lambda\boldsymbol{\alpha}$$

$$\text{with } \mathcal{E}_\lambda \overset{\text{def}}{=} \begin{pmatrix} \mathsf{K}_\lambda^{-1}(\mathsf{I}-\tilde{\mathsf{C}}(\tilde{\mathsf{C}}^\mathsf{T}\mathsf{K}_\lambda^{-1}\tilde{\mathsf{C}})^{-1}\tilde{\mathsf{C}}^\mathsf{T}\mathsf{K}_\lambda^{-1}) \\ (\tilde{\mathsf{C}}^\mathsf{T}\mathsf{K}_\lambda^{-1}\tilde{\mathsf{C}})^{-1}\tilde{\mathsf{C}}^\mathsf{T}\mathsf{K}_\lambda^{-1}\end{pmatrix}. \tag{14}$$

This parameterization has the advantages to separate λ and $\boldsymbol{\alpha}$ and to introduce units.[4] The side-conditions are naturally enforced by this parameterization.

Incorporating the parameterization (14) into the TPS (13) we obtain what we call the *feature-driven* parameterization $\tau(\mathbf{p};\boldsymbol{\alpha},\lambda) = \omega(\mathbf{p};\boldsymbol{\eta}_{\boldsymbol{\alpha},\lambda})$ for the TPS:

$$\tau(\mathbf{p};\boldsymbol{\alpha},\lambda) \overset{\text{def}}{=} \boldsymbol{\ell}_\mathbf{p}^\mathsf{T}\mathcal{E}_\lambda\boldsymbol{\alpha}. \tag{15}$$

The square integral bending energy $\kappa = \int_{\mathbb{R}^2}\|\frac{\partial^2\tau}{\partial\mathbf{p}^2}(\mathbf{p};\boldsymbol{\alpha},\lambda)\|_\mathcal{F}^2 \, d\mathbf{p} = 8\pi\mathbf{w}^\mathsf{T}\mathsf{K}_\lambda\mathbf{w}$ is given by $\kappa = 8\pi\boldsymbol{\alpha}^\mathsf{T}\bar{\mathcal{E}}_\lambda\boldsymbol{\alpha}$, where $\bar{\mathcal{E}}_\lambda$ is the $(l\times l)$ *bending energy matrix* given by amputating $\mathcal{E}_\lambda$ of its last three rows:

$$\bar{\mathcal{E}}_\lambda \overset{\text{def}}{=} \mathsf{K}_\lambda^{-1}(\mathsf{I}-\tilde{\mathsf{C}}(\tilde{\mathsf{C}}^\mathsf{T}\mathsf{K}_\lambda^{-1}\tilde{\mathsf{C}})^{-1}\tilde{\mathsf{C}}^\mathsf{T}\mathsf{K}_\lambda^{-1}). \tag{16}$$

The bending energy matrix is symmetric and in the absence of internal regularization, *i.e.* for $\lambda = 0$, has rank $l-3$. The eigenvectors corresponding to the $l-3$ nonzero eigenvalues are the *principal warps*, the corresponding eigenvalues indicating their bending energy, as defined by Bookstein [3].

The TPS-warp is obtained by stacking two $\mathbb{R}^2 \mapsto \mathbb{R}$ TPSs. From (11), we get:

$$\begin{pmatrix}\tau(\mathbf{p};\boldsymbol{\alpha}_x,\lambda)\\ \tau(\mathbf{p};\boldsymbol{\alpha}_y,\lambda)\end{pmatrix} = (\boldsymbol{\ell}_\mathbf{p}^\mathsf{T}\mathcal{E}_\lambda\mathsf{L})^\mathsf{T},$$

where $\boldsymbol{\alpha}_x$ and $\boldsymbol{\alpha}_y$ are the first and second columns of L. The TPS warp is thus expressed in the form (1), *i.e.* $\mathcal{W}(\mathbf{p};\mathsf{L}) = \mathsf{L}^\mathsf{T}\nu(\mathbf{p})$, with the following nonlinear lifting function:

$$\nu(\mathbf{p}) = \mathcal{E}_\lambda^\mathsf{T}\boldsymbol{\ell}_\mathbf{p}.$$

The internal smoothing parameter λ is chosen small to ensure that matrix $\mathcal{E}_\lambda$ is well-conditioned.

Finally, the second derivative based smoother in (2) has the form:

$$\mathcal{B}^2(\mathsf{L}) = 8\pi\|\sqrt{\bar{\mathcal{E}}}\mathsf{L}\|_\mathcal{F}^2,$$

and we thus just choose Z such that $\mathsf{Z}^\mathsf{T}\mathsf{Z} = 8\pi\bar{\mathcal{E}}$ in the matrix form (3) to achieve the exact integral. Note that in practice, one does not need to compute Z since only $\mathsf{Z}^\mathsf{T}\mathsf{Z}$ is needed, *e.g.* for building the influence matrix T in (5).

Appendix 2: The LOOCV Lemma

This lemma states that replacing a target value with its prediction by the model estimated with this equation omitted does not change the result. In other words, adding equations to an LLS problem with as right-hand side the prediction by the model solving the initial problem, does not change the result.

Define $\mathsf{D}_j = \mathsf{I} - \operatorname{diag}(\mathbf{e}_j)$. Our goal is to show that (9) gives $\hat{\mathsf{L}}_{(j)}(\mu)$ as from (7). Following (5) we rewrite (7) as:

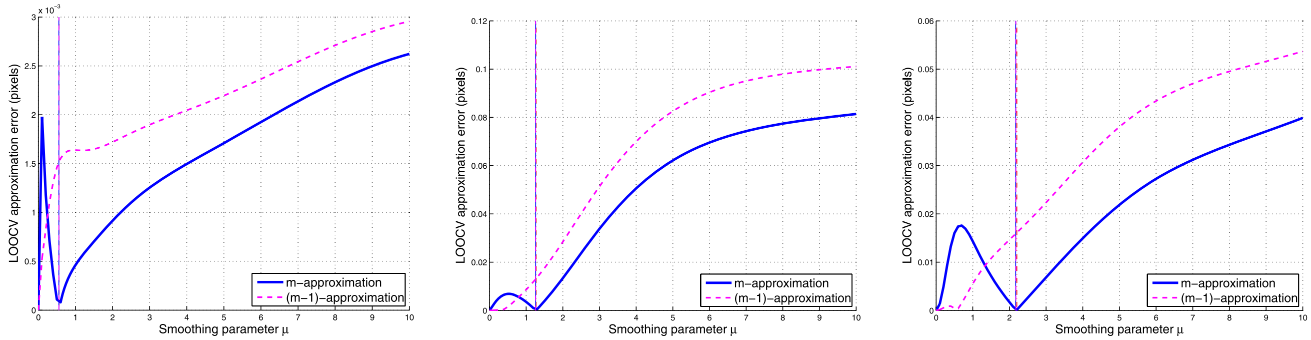
$$\begin{aligned}\hat{\mathsf{L}}_{(j)}(\mu) &= \arg\min_\mathsf{L}\left\|\begin{pmatrix}\mathsf{D}_j\mathsf{N}\\ \sqrt{m-1}\mu\mathsf{Z}\end{pmatrix}\mathsf{L} - \begin{pmatrix}\mathsf{D}_j\Xi\\ 0\end{pmatrix}\right\|_\mathcal{F}^2\\ &= (\mathsf{N}^\mathsf{T}\mathsf{D}_j\mathsf{N} + (m-1)\mu^2\mathsf{Z}^\mathsf{T}\mathsf{Z})^{-1}\mathsf{N}^\mathsf{T}\mathsf{D}_j\Xi,\end{aligned} \tag{17}$$

since $\mathsf{D}_j^\mathsf{T} = \mathsf{D}_j$ and $\mathsf{D}_j\mathsf{D}_j = \mathsf{D}_j$. We rewrite $\tilde{\Xi}^j$ from (8) as $\tilde{\Xi}^j = \mathsf{D}_j\Xi + (\mathsf{I}-\mathsf{D}_j)\mathsf{N}\hat{\mathsf{L}}_{(j)}$. We expand equation (9) by replacing T from (5) and $\tilde{\Xi}^j$ from just above, giving:

$$\begin{aligned}&\mathsf{T}((m-1)\mu^2)\tilde{\Xi}^j\\ &\quad = (\mathsf{N}^\mathsf{T}\mathsf{N} + (m-1)\mu^2\mathsf{Z}^\mathsf{T}\mathsf{Z})^{-1}\mathsf{N}^\mathsf{T}\\ &\qquad \cdot(\mathsf{D}_j\Xi + \mathsf{N}\hat{\mathsf{L}}_{(j)} - \mathsf{D}_j\mathsf{N}\hat{\mathsf{L}}_{(j)}).\end{aligned} \tag{18}$$

The second term rewrites to:

[4] While $\boldsymbol{\eta}_{\boldsymbol{\alpha},\lambda}$ has no obvious unit, $\boldsymbol{\alpha}$ in general has (*e.g.* pixels, meters).



The dishcloth dataset
LOOCV ≈ 1.40 pixels

The paper sheet dataset
LOOCV ≈ 7.27 pixels

The spine dataset
LOOCV ≈ 12.60 pixels

Fig. 12 Zoom around the minimum onto the difference between the true LOOCV score from the greedy formula (6) and the m- and $(m-1)$-approximations. The *vertical lines* show the minima (the *dashed red line* is the true minima)

$$\begin{aligned}&(\mathsf{N}^\mathsf{T}\mathsf{N} + (m-1)\mu^2\mathsf{Z}^\mathsf{T}\mathsf{Z})^{-1}\mathsf{N}^\mathsf{T}\mathsf{N}\hat{\mathsf{L}}_{(j)}\\ &\quad = \hat{\mathsf{L}}_{(j)} - (m-1)\mu^2(\mathsf{N}^\mathsf{T}\mathsf{N} + (m-1)\mu^2\mathsf{Z}^\mathsf{T}\mathsf{Z})^{-1}\mathsf{Z}^\mathsf{T}\mathsf{Z}\hat{\mathsf{L}}_{(j)}.\end{aligned} \tag{19}$$

Substituting in (18) gives:

$$\begin{aligned}&\mathsf{T}((m-1)\mu^2)\tilde{\Xi}^j\\ &\quad = \hat{\mathsf{L}}_{(j)} + (\mathsf{N}^\mathsf{T}\mathsf{N} + (m-1)\mu^2\mathsf{Z}^\mathsf{T}\mathsf{Z})^{-1}\\ &\qquad \cdot (\mathsf{N}^\mathsf{T}\mathsf{D}_j\Xi - (m-1)\mu^2\mathsf{Z}^\mathsf{T}\mathsf{Z}\hat{\mathsf{L}}_{(j)} - \mathsf{N}^\mathsf{T}\mathsf{D}_j\mathsf{N}\hat{\mathsf{L}}_{(j)}).\end{aligned} \tag{20}$$

This concludes the proof since the right-most factor vanishes, as shown below. Substitute $\hat{\mathsf{L}}_{(j)}$ from (17), this gives:

$$\begin{aligned}&\mathsf{N}^\mathsf{T}\mathsf{D}_j\Xi - (m-1)\mu^2\mathsf{Z}^\mathsf{T}\mathsf{Z}\hat{\mathsf{L}}_{(j)} - \mathsf{N}^\mathsf{T}\mathsf{D}_j\mathsf{N}\hat{\mathsf{L}}_{(j)}\\ &\quad = \mathsf{N}^\mathsf{T}\mathsf{D}_j\Xi - ((m-1)\mu^2\mathsf{Z}^\mathsf{T}\mathsf{Z} + \mathsf{N}^\mathsf{T}\mathsf{D}_j\mathsf{N})\\ &\qquad \cdot (\mathsf{N}^\mathsf{T}\mathsf{D}_j\mathsf{N} + (m-1)\mu^2\mathsf{Z}^\mathsf{T}\mathsf{Z})^{-1}\mathsf{N}^\mathsf{T}\mathsf{D}_j\Xi\\ &\quad = \mathsf{N}^\mathsf{T}\mathsf{D}_j\Xi - \mathsf{N}^\mathsf{T}\mathsf{D}_j\Xi\\ &\quad = 0.\end{aligned}$$

Appendix 3: The Non-Iterative Approximation to LOOCV

In order to compare the m-approximation and the $(m-1)$-approximation we plot the difference between the true LOOCV score from (6) and each of the two approximations. This is shown for the three datasets in Fig. 12. As can be seen, both approximations are very close to the true LOOCV score. The m-approximation is in general better than the $(m-1)$-approximation, except at some points for $\mu < \hat{\mu}$. The minimum value of the m-approximation coincides with the true value at the true minimum, albeit that the location of the approximated minimum is slightly shifted from the true location. The $(m-1)$-approximation has a larger shift. Using the m-approximation is thus the best option, although the difference is very small. The order of magnitude on the location of the minimum is between 10^{-2} and 10^{-4}. The error on the minimum LOOCV score for the $(m-1)$-approximation is at 10^{-1} pixels.

References

1. Allen, D.M.: The relationship between variable selection and data augmentation and a method for prediction. Technometrics **16**, 125–127 (1974)
2. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
3. Bookstein, F.L.: Principal warps: thin-plate splines and the decomposition of deformations. IEEE Trans. Pattern Anal. Mach. Intell. **11**(6), 567–585 (1989)
4. Bro-Nielsen, M., Gramkow, C.: Fast fluid registration of medical images. In: Visualization in Biomedical Imaging (1996)
5. Burrage, K., Williams, A., Erhel, J., Pohl, B.: The implementation of a generalized cross validation algorithm using deflation techniques for linear systems. Technical report, Seminar fur Angewandte Mathematik, July 1994
6. Christensen, G.E., He, J.: Consistent nonlinear elastic image registration. In: Workshop on Mathematical Methods in Biomedical Image Analysis (2001)
7. de Bruijne, M., Lund, M.T., Tankó, L.B., Pettersen, P.C., Nielsen, M.: Quantitative vertebral morphometry using neighbor-conditional shape models. Med. Image Anal. **11**(5), 503–512 (2007)
8. Duchon, J.: Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. RAIRO Anal. Numér. **10**, 5–12 (1976)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Comput. Vis. Graph. Image Process. **24**(6), 381–395 (1981)
10. Fornefett, M., Rohr, K., Stiehl, H.S.: Radial basis functions with compact support for elastic registration of medical images. Image Vis. Comput. **19**(1), 87–96 (2001)
11. Genant, H., Wu, C., van Kuijk, C., Nevitt, M.: Vertebral fracture assessment using a semiquantitative technique. J. Bone Miner Res. **8**(9), 1137–1148 (1993)

12. Golub, G.H., von Matt, U.: Generalized cross-validation for large-scale problems. J. Comput. Graph. Stat. **6**(1), 1–34 (1997)
13. Hawkins, D.M., Yin, X.: A faster algorithm for ridge regression of reduced rank data. Comput. Stat. Data Anal. **40**(2), 253–262 (2002)
14. Kanatani, K.: Geometric information criterion for model selection. Int. J. Comput. Vis. **26**(3), 171–189 (1998)
15. Maintz, J.B.A., Viergever, M.A.: A survey of medical image registration. Med. Image Anal. **2**(1), 1–36 (1998)
16. Marsland, S., Twining, C.J., Taylor, C.J.: A minimum description length objective function for groupwise non-rigid image registration. In: Image and Vision Computing (2007)
17. Maybank, S., Sturm, P.: MDL, collineations and the fundamental matrix. In: British Machine Vision Conference (1999)
18. Modersitzki, J.: Numerical Methods for Image Registration. Oxford Science, Oxford (2004)
19. Nielsen, M., Johansen, P.: A PDE solution of Brownian warping. In: European Conference on Computer Vision (2004)
20. Nielsen, M., Johansen, P., Jackson, A.D., Lautrup, B.: Brownian warps: a least committed prior for non-rigid registration. In: Medical Image Computing and Computer-Assisted Intervention (2002)
21. Pilet, J., Lepetit, V., Fua, P.: Fast non-rigid surface detection, registration and realistic augmentation. Int. J. Comput. Vis. **76**(2), 109–122 (2008)
22. Poggio, T., Rifkin, R., Mukherjee, S., Niyogi, P.: General conditions for predictivity in learning theory. Nature **458**, 419–422 (2004)
23. Rifkin, R.M., Lippert, R.A.: Notes on regularized least squares. Technical Report MIT-CSAIL-TR-2007-025, MIT, May 2007
24. Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L.G., Leach, M.O., Hawkes, D.J.: Nonrigid registration using free-form deformations: application to breast MR images. IEEE Trans. Med. Imaging **18**(8), 712–721 (1999)
25. Szeliski, R., Coughlan, J.: Spline-based image registration. Int. J. Comput. Vis. **22**(3), 199–218 (1997)
26. Torr, P.H.S.: Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. Int. J. Comput. Vis. **50**(1), 27–45 (2002)
27. Wahba, G.: Splines Models for Observational Data. SIAM, Philadelphia (1990)
28. Wahba, G., Wold, S.: A completely automatic French curve: fitting spline functions by cross-validation. Commun. Stat. **4**, 1–17 (1975)

Adrien Bartoli is a permanent CNRS research scientist at the LASMEA laboratory in Clermont- Ferrand, France, since October 2004 and a visiting professor at DIKU in Copenhagen, Denmark for 2006-2009. Before that, he was a postdoctoral researcher at the University of Oxford, UK, in the Visual Geometry Group, under the supervision of Prof. Andrew Zisserman. He did his PhD in the Perception group, in Grenoble at INRIA, France, under the supervision of Prof. Peter Sturm and Prof. Radu Horaud. He received the 2004 INPG PhD Thesis prize and the 2007 best paper award at CORESA. Since September 2006, he is co-leading the ComSee research team. His main research interests are in Structure-from-Motion in rigid and non-rigid environments and machine learning within the field of computer vision.