Constant Flow Sampling: A Method to Automatically Select the Regularization Parameter in Image Registration

Benoît Compte[1], Adrien Bartoli[1], and Daniel Pizarro[2]

[1] Université d'Auvergne, ISIT, BP10448, F-63000 Clermont-Ferrand.
benoit.compte@u-clermont1.fr, adrien.bartoli@gmail.com
[2] University of Alcala, Alcala de Henares, Spain.
pizarro@depeca.uah.es

Abstract. We present a method to automatically select the regularization parameter in the two-term compound cost function used in image registration. Our method is called CFS (Constant Flow Sampling). It samples the regularization parameter using the constraint that the warp-induced image flow be of constant magnitude on average. Compared to other methods, CFS provably provides a global solution at a specified precision and within a finite number of steps. CFS can be embedded within any algorithm minimizing a two-term compound cost function depending on a regularization parameter. We report experimental results on the registration of several datasets of laparoscopic images.

1 Introduction

A warp $\mathcal{W}$ is a parametric function that allows one to register a source to a target image. We here write $\mathbf{q}' = \mathcal{W}(\mathbf{q};\mathbf{x}) \in \mathbb{R}^2$ the image of a point $\mathbf{q} \in \Omega$ by the warp $\mathcal{W}$ with $\mathbf{x} \in \mathbb{R}^p$ the warp's parameter vector and $\Omega \subset \mathbb{R}^2$ the warp's domain. The optimal warp parameters $\mathbf{x}^* \in \mathbb{R}^p$ are computed by minimizing a cost function containing a data term $\mathcal{E}_d$ and a regularization term $\mathcal{E}_r$ as:

$$\mathbf{x}^*(\lambda) = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \mathcal{E}_d(\mathbf{x}) + \lambda\mathcal{E}_r(\mathbf{x}), \tag{1}$$

where $\lambda \in \mathbb{R}^+$ is the *regularization parameter*, specifying the amount of regularization. Automatically choosing an optimal value for λ is a difficult problem which has not yet received a commonly agreed solution in the scientific community. On the one hand, if the chosen λ is 'too low' the data term will prevail and the warp will overfit the data, including the noise. Consequently, portions of the warp with fewer data will not capture the true deformation. On the other hand, if the chosen λ is 'too large' the warp will be too smooth and will underfit the data.

The general approach to automatically select λ is to construct some test cost function $\mathcal{E}_m : \mathbb{R}^+ \to \mathbb{R}$, $\lambda \mapsto \mathcal{E}_m(\lambda)$ whose value approximates the difference between the warp estimate at λ and the true image deformation, and minimize:

$$\lambda^* = \arg\min_{\lambda\in\mathbb{R}^+} \mathcal{E}_m(\lambda). \tag{2}$$

This raises two difficult problems: *(i)* constructing the test cost function from a limited set of data and *(ii)* finding the global minimimum of the test cost function. While problem *(i)* has been well-studied in the literature, problem *(ii)* still lags behind. For instance, the test cost function can be constructed from the paradigm of CV (Cross-Validation) [11] or by combining landmarks and dense intensity-based error measurements [5]. In any case, *the test cost function is always nonlinear and nonconvex*, making problem *(ii)* extremely difficult to solve efficiently and with guarantees of optimality on the estimated solution. Current approaches use general purpose nonlinear optimization methods such as golden-section search and gradient descent, which cannot cope with the extremely nonlinear behaviour of test cost functions such as Ordinary-CV.

We propose CFS (Constant Flow Sampling), a novel approach to the problem of finding λ by optimizing $\mathcal{E}_m$. The key idea is to sample values of λ over the range of admissible values. The difficulties are obviously to find an upper bound λ_{init} and to sample in such a way that the test cost function's global minimum is not overlooked. Defining an a priori sampling scheme, with regular spacing within the space of λ is not relevant, since $\mathcal{E}_m$ is typically almost constant for 'large' values of λ, and may oscillate for 'small' values of λ. CFS proceeds as follows. We first compute an initial value λ_{init} of λ large enough so that the corresponding warp be the $\lambda \to \infty$ asymptotically regularized warp that minimizes the regularization term. We then sample λ between this initial value λ_{init} and 0. Our key contribution is to sample λ regularly with respect to the magnitude of the flow induced by the warp. More specifically, we select the decrease δ such that the average magnitude of the flow between the warp at λ and at $\lambda - \delta$ be some fixed constant $\tau \in \mathbb{R}^+$. The value of τ is expressed in number of pixels and is thus easily fixed. We typically choose $\tau = 1$ pixel. With CFS, λ undergoes large decreases at the early steps since $\mathcal{E}_m$'s graph is typically flat, and smaller decreases around the global minimum of $\mathcal{E}_m$. Our algorithm is thus guaranteed to sample the range of admissible values of λ evenly and in a finite number of steps. The global minimum is found, assuming that the test cost function is convex within a small region, the size of which being related to the chosen flow magnitude constant τ.

Paper organization. We review the state of the art in §2. We present our CFS method and algorithm in §3. We give experimental results in §4. We finally conclude in §5.

2 State of the Art

The hyperparameter λ is often manually selected by trial and error [3,6]. Here we will describe some methods used to select it automatically.

2.1 Defining the Test Cost Function

The problem of constructing the test cost function $\mathcal{E}_m$ from a limited set of data has been well studied and several criteria have been proposed. The input is a set of n point matches $\{\mathbf{q}_k \leftrightarrow \mathbf{q}'_k\}_{k=1,\dots,n}$.

The first three criteria are feature-based; they are applicable only when 'enough' point matches are available. The fourth criterion is pixel-based; it uses all the raw information available from the images.

Training/Test Splitting (TTS). TTS is the simplest criterion. It consists in splitting the dataset into a training set used for the optimization of the warp parameters given λ and a test set used for the optimization of λ. It is a classical approach in statistical learning [8]. Let $\{\mathbf{r}_k \leftrightarrow \mathbf{r}'_k\}_{k=1,\dots,n_{test}}$ be points matches forming the test set (a subset of the input point matches) and $\mathbf{x}^*_{train}(\lambda)$ the warp parameters obtained using the training set. The TTS score $\mathcal{E}_m^{TTS}$ is defined by:

$$\mathcal{E}_m^{TTS}(\lambda) = \frac{1}{n_{test}} \sum_{k=1}^{n_{test}} \left\| \mathbf{r}'_k - \mathcal{W}\left(\mathbf{r}_k; \mathbf{x}^*_{train}(\lambda)\right) \right\|^2. \tag{3}$$

Ordinary-CV (OCV). OCV is also based on a partition of the dataset. Each point is used in turn as a test set while the others form the training set. For a given regularization parameter λ, let $\mathbf{x}^*_{(k)}(\lambda)$ be the warp parameters estimated from the data with the k-th point left out. The OCV score $\mathcal{E}_m^{OCV}$ is defined by:

$$\mathcal{E}_m^{OCV}(\lambda) = \frac{1}{n} \sum_{k=1}^{n} \left\| \mathbf{q}'_k - \mathcal{W}\left(\mathbf{q}_k; \mathbf{x}^*_{(k)}(\lambda)\right) \right\|^2. \tag{4}$$

This score has been used in [1,7]. Its computation time is low thanks to a closed-form solution [11].

V-fold CV (VCV). An alternative to the OCV score is the VCV score. It consists in splitting the dataset into V subsets of nearly equal size, each of them being used alternatively as a test set while the others form the training set. Let $\mathbf{x}^*_{[v]}(\lambda)$ be the warp parameters obtained from the data with the v-th group left out, m_v the number of point correspondences in the v-th group and $\mathbf{q}_{v,k} \leftrightarrow \mathbf{q}'_{v,k}$ the k-th correspondance of the v-th group. The VCV score $\mathcal{E}_m^{VCV}$ is defined by:

$$\mathcal{E}_m^{VCV}(\lambda) = \sum_{v=1}^{V} \frac{m_v}{n} \sum_{k=1}^{m_v} \frac{1}{m_v} \left\| \mathbf{q}'_{v,k} - \mathcal{W}\left(\mathbf{q}_{v,k}; \mathbf{x}^*_{[v]}(\lambda)\right) \right\|^2. \tag{5}$$

This score has been used in [2].

Photometric Error Criterion (PEC). In this criterion, the point correspondences are used as the training set and the photometric information as the test set. Given a regularization parameter λ and the corresponding warp parameters

$\mathbf{x}^*(\lambda)$ estimated from the point correspondences, the PEC score $\mathcal{E}_m^{PEC}$ is defined by:

$$\mathcal{E}_m^{PEC}(\lambda) = \frac{1}{\mathcal{B}} \sum_{\mathbf{q} \in \mathcal{B}} \|\mathcal{S}(\mathbf{q}) - \mathcal{T}(\mathcal{W}(\mathbf{q}; \mathbf{x}^*(\lambda)))\|^2, \tag{6}$$

where $\mathcal{B}$ is the set of pixels in the region of interest, $\mathcal{S}$ is the source image and $\mathcal{T}$ is the target image.

2.2 Minimizing the Test Cost Function

The algorithm used to minimize the test cost function is often neglected in the literature. Only a few articles mention the minimization algorithm they use, which can be golden-section search, exhaustive search or downhill simplex [1,5]. We assume that other nonlinear optimization methods, like gradient descent, may have been used. Each of these methods have one or both of the following limitations: the region where to search the minimum is user-defined and the local minimization does not guarantee that the global minimum is found.

3 CFS – Constant Flow Sampling

This section introduces our CFS method and algorithm. We first give general points, then study how to find a constant average flow magnitude decrease δ of λ and how to find an upper bound λ_{init} on λ. We finally discuss some characteristics of CFS.

3.1 General Points and Algorithm

Our CFS is meant to be used with any test cost function $\mathcal{E}_m$ and method to train the warp (or more generally the model) parameters. We thus assume that, given some value of the regularization parameter λ, the corresponding warp parameters $\mathbf{x}^*(\lambda)$ can be found by solving problem (1), and that the test cost function is given. Our goal is here to solve problem (2) with a sampling strategy over λ. The CFS algorithm is as follows:

Inputs: test cost function $\mathcal{E}_m : \mathbb{R}^+ \to \mathbb{R}$, tolerance on the flow $\tau \in \mathbb{R}^+$

- Choose an upper bound $\lambda_{\text{init}} \in \mathbb{R}^+$ (see §3.3)
- Set $\lambda \leftarrow \lambda_{\text{init}}$ and $\lambda^* \leftarrow \lambda_{\text{init}}$
- While $\lambda > 0$ do
  - Choose the decrease $\delta \in \mathbb{R}^+$ such that the average flow magnitude between the warp with parameters $\mathbf{x}^*(\lambda)$ and $\mathbf{x}^*(\lambda - \delta)$ equals τ (see §3.2)
  - If $\mathcal{E}_m(\lambda - \delta) < \mathcal{E}_m(\lambda^*)$, Set $\lambda^* \leftarrow \lambda - \delta$
  - Set $\lambda \leftarrow \lambda - \delta$

Outputs: regularization parameter λ^*, warp parameters $\mathbf{x}^*(\lambda^*)$

3.2 Sampling at a Constant Flow Magnitude

Our algorithm samples λ from its upper bound λ_{init} to 0. For each sample value λ we thus have to compute the next value $\lambda - \delta$ such that the displacement of the warp is constant. We now describe how to compute the flow magnitude at a single point $\mathbf{q} \in \Omega$ between the warps with parameters $\mathbf{x}^*(\lambda)$ and $\mathbf{x}^*(\lambda - \delta_{\mathbf{q}})$. For a decrease $\delta_{\mathbf{q}}$ of the regularization parameter λ, the flow difference constraint between the two parameter vectors is:

$$\|\mathcal{W}(\mathbf{q}, \mathbf{x}^*(\lambda)) - \mathcal{W}(\mathbf{q}, \mathbf{x}^*(\lambda - \delta_{\mathbf{q}}))\| = \tau,$$

where we recall that $\tau \in \mathbb{R}^+$ is the specified tolerance on the flow difference magnitude. We here make the assumption that the warp model being used is linear in its parameter vector (but not necessarily in the point coordinates). This is a common requirement, satisfied by most classical warps such as the Thin-Plate Spline [11], the Free-Form Deformation [10] and others such as Moving Least Squares [9]. The training cost function $\mathcal{E}_d(\mathbf{x}) + \lambda \mathcal{E}_r(\mathbf{x})$ of problem (1) can thus be assumed to be of the form $\mathcal{E}_d(\mathbf{x}) \stackrel{\text{def}}{=} \|\mathsf{A}\mathbf{x} - \mathbf{b}\|^2$ and $\mathcal{E}_r(\mathbf{x}) \stackrel{\text{def}}{=} \|\mathsf{K}\mathbf{x}\|^2$. Consequently, we obtain:

$$\mathbf{x}^*(\lambda) = \left(\mathsf{A}^\top \mathsf{A} + \lambda \mathsf{K}^\top \mathsf{K}\right)^{-1} \mathsf{A}^\top \mathbf{b}.$$

We define $\mathbf{a}_{\mathbf{q}} \in \mathbb{R}^p$ to be the lifted coordinates of point $\mathbf{q} \in \Omega$, such that $\mathcal{W}(\mathbf{q}, \mathbf{x}) = \mathbf{a}_{\mathbf{q}}^\top \mathbf{x}$. The flow difference can thus be rewritten as:

$$\left\|\mathbf{a}_{\mathbf{q}}^\top \mathbf{x}^*(\lambda) - \mathbf{a}_{\mathbf{q}}^\top \mathbf{x}^*(\lambda - \delta_{\mathbf{q}})\right\| = \tau.$$

This is a high order polynomial in $\delta_{\mathbf{q}}$. We use Taylor expansion of $\mathbf{x}^*$ around λ to get:

$$\mathbf{x}^*(\lambda - \delta_{\mathbf{q}}) = \mathbf{x}^*(\lambda) + \sum_{n=1}^{n=\infty} \frac{1}{n!} \frac{\partial^n \mathbf{x}^*(\lambda)}{\partial \lambda^n} \delta_{\mathbf{q}}^n$$

$$= \mathbf{x}^*(\lambda) + \sum_{n=1}^{n=\infty} \left(\mathsf{A}^\top \mathsf{A} + \lambda \mathsf{K}^\top \mathsf{K}\right)^{-1} \left(\left(\mathsf{K}^\top \mathsf{K}\right) \left(\mathsf{A}^\top \mathsf{A} + \lambda \mathsf{K}^\top \mathsf{K}\right)^{-1}\right)^n \mathsf{A}^\top \mathbf{b}\, \delta_{\mathbf{q}}^n.$$

An approximate solution is obtained by truncating the above expansion to first order, leading to the following constraint on the flow magnitude difference:

$$\left\|\mathbf{a}_{\mathbf{q}}^\top \left(\mathsf{A}^\top \mathsf{A} + \lambda \mathsf{K}^\top \mathsf{K}\right)^{-1} \left(\mathsf{K}^\top \mathsf{K}\right) \left(\mathsf{A}^\top \mathsf{A} + \lambda \mathsf{K}^\top \mathsf{K}\right)^{-1} \mathsf{A}^\top \mathbf{b}\, \delta_{\mathbf{q}}\right\| = \tau.$$

This allows us to obtain the following expression for $\delta_{\mathbf{q}}$ as a function of the current λ and flow magnitude tolerance τ:

$$\tilde{\delta}_{\mathbf{q}} = \frac{\tau}{\left\|\mathbf{a}_{\mathbf{q}}^\top \left(\mathsf{A}^\top \mathsf{A} + \lambda \mathsf{K}^\top \mathsf{K}\right)^{-1} \left(\mathsf{K}^\top \mathsf{K}\right) \left(\mathsf{A}^\top \mathsf{A} + \lambda \mathsf{K}^\top \mathsf{K}\right)^{-1} \mathsf{A}^\top \mathbf{b}\right\|}.$$

Note that the denominator represents the flow rate with respect to λ. It it of course possible to truncate the Taylor expansion to a higher order. This would

lead to a polynomial root-finding problem in a single variable, $\delta_{\mathbf{q}}$, which can be very easily solved numerically.

In practice, we evaluate the flow for a dense set of points $\mathcal{B} \subset \Omega$ (we use every pixels). Different strategies can be used to select δ. First, the minimum value over all points can be used. This option is the safest by producing a large amount of samples, but still guaranteeing convergence in a finite number of steps. Second, the maximum value over all points can be used. This option will produce fewer samples, and will trade accuracy of the solution for runtime. Third, the average value over all points can be used: this option is a reasonable compromise between accuracy and runtime. Using this third strategy the decrease δ will be:

$$\delta = \frac{1}{\text{size}(\mathcal{B})} \sum_{\mathbf{q} \in \mathcal{B}} \tilde{\delta}_{\mathbf{q}}.$$

3.3 Finding an Upper Bound

When the algorithm begins, λ generally has a very large value, corresponding to an asymptotic regularization. The corresponding rate of displacement will thus be approximately zero, and will lead to $\delta \gg \lambda$, causing the algorithm to immediately terminate. We thus have to compute an upper bound λ_{init} on λ such that the rate of displacement is large enough to yield a suitable decrease δ in λ. To do this we choose λ_{init} such that $\delta(\lambda_{\text{init}})$ lies between two bounds: $\delta_{\min}$ and $\delta_{\max}$. This will ensure that both the rate of displacement of the warp and δ are large enough. We proceed in two steps. First, we start from a high value $\lambda_{\max}$ for λ that we know to be in the asymptotic case (*e.g.* $\lambda_{\max} = 10^{10}$). We then iteratively decrease λ by dividing it by 10 and compute δ at each step. We stop when δ is lower than $\delta_{\max}$ (*e.g.* $\delta_{\max} = \frac{\lambda}{2}$). This gives a lower bound λ_{low} for λ_{init}. Second, we check if δ_{low} is greater than $\delta_{\min}$ (*e.g.* $\delta_{\min} = \frac{\lambda}{3}$). If this holds we stop and set $\lambda_{\text{init}} \leftarrow \lambda_{\text{low}}$. If not, we take $\lambda_{\text{high}} = 10\lambda_{\text{low}}$ and run a simple bisection search to find λ_{init} such that $\delta_{\min} < \delta(\lambda_{\text{init}}) < \delta_{\max}$, as we can see in figure 1.

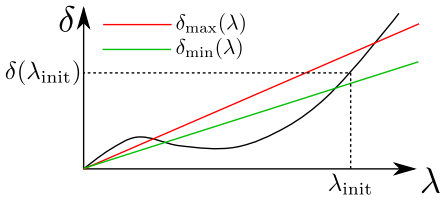

Fig. 1. λ_{init} is chosen so that $\delta_{\min} < \delta(\lambda_{\text{init}}) < \delta_{\max}$.

3.4 Discussion

CFS has several advantages. First, it guarantees that the estimated λ^* matches the global minimum of $\mathcal{E}_m$ provided that the global minimum is not too sharp for the user defined tolerance τ on the warp-induced flow (*e.g.* $\tau = 1$ pixel).

Second, it guarantees that the precision of λ^* with respect to the true global minimum corresponds to the tolerance τ. The trade-off between runtime (less samples) and accuracy (more samples) can be easily specified by changing the value of τ.

4 Experimental Results

4.1 Implementation

In our implementation we use a B-spline warp, also known as the FFD warp [10]. The domain $\Omega \subset \mathbb{R}^2$ of this warp is a rectangle and the warp's shape is controlled by a set of control points which form the warp's parameters.

We present experimental results on two datasets extracted from laparoscopic sequences, with manually-selected point correspondences. For both datasets we tested the TTS and PEC cost functions using the same training set.

4.2 Human Uterus

The first dataset shows a human uterus and has 35 point correspondences, as can be seen in figure 2.

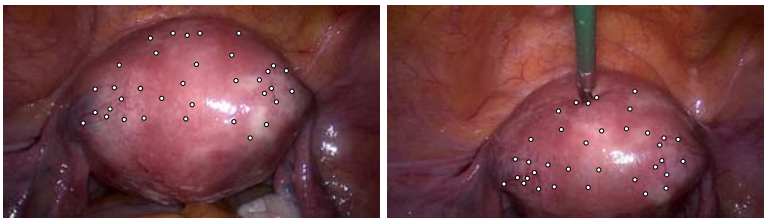

Fig. 2. The uterus image pair with 35 point correspondences.

Figure 3 shows the photometric error obtained using PEC. We can see that the photometric error function has several local minima, and that the part corresponding to 'small' values of λ (which is not visible on the linear scale) has many sharp variations that cannot be handled by traditionnal nonlinear optimization methods. Figure 3 also shows the test error obtained using TTS. We found $\lambda^*_{PEC} = 1.256 \times 10^3$ and $\lambda^*_{TTS} = 0.968 \times 10^3$ and the corresponding training errors: $\mathcal{E}^{PEC}_{\text{train}} = 3.787$px and $\mathcal{E}^{TTS}_{\text{train}} = 3.657$px. The average flow difference between the corresponding warps is 1.54 pixels.

We can see in figure 4 the target points and the warped source points. The difference we can observe is mainly due to the fact that the B-spline warp has difficulties to deal with strong perspective effects like with this couple of images. We could probably obtain better results by using a NURBS warp which has

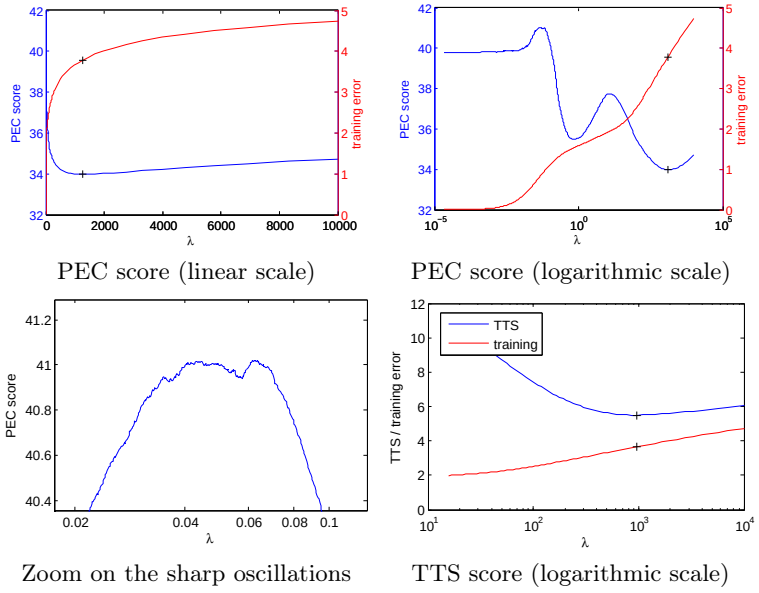

Fig. 3. PEC and TTS scores for the uterus dataset.

been proved to model perspective better than the B-spline warp [4]. However, the NURBS warp is not linear and would need CFS to be extended to handle that case.

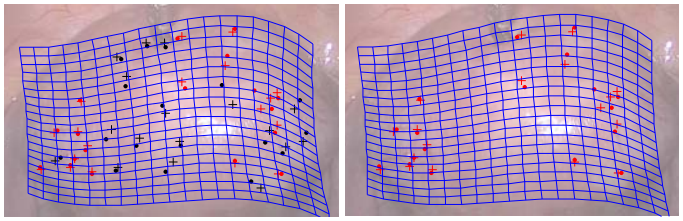

Fig. 4. The deformation grid of the warp, left: TTS criterion, right: PEC criterion, dots: warped source points, crosses: target points, red: training set, black: test set.

4.3 Pig Intestines

The second dataset shows pig intestines and has 54 correspondences, as we can see in figure 5. We can see the test error and the photometric error obtained

with both tested methods in figure 6. In this particular case, the photometric error seems to be smoother than with the uterus dataset.

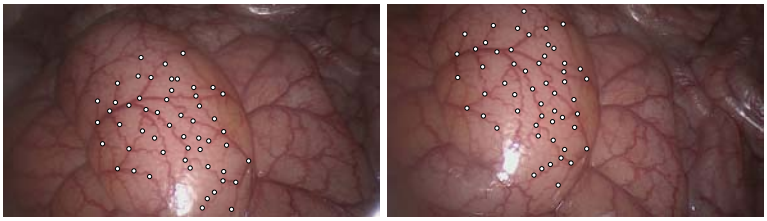

Fig. 5. The intestines image pair with 54 point correspondences.

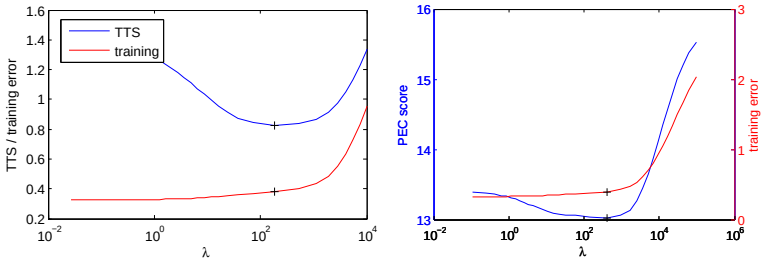

Fig. 6. TTS score (left) and PEC score (right) for the pig intestines dataset.

We can see in figure 7 the target points and the warped source points which are really close to each other. On this example we have $\lambda^*_{PEC} = 0.186 \times 10^3$ and $\lambda^*_{TTS} = 0.413 \times 10^3$, with the corresponding training errors: $\mathcal{E}^{PEC}_{\text{train}} = 0.3972$px and $\mathcal{E}^{TTS}_{\text{train}} = 0.379$px. The difference between these values can be explained by the fact that the photometric error does not vary much near the optimum (optimum: $\lambda^*_{PEC} = 0.413 \times 10^3$, $\mathcal{E}^{PEC}_m = 13.02$; next sample evaluated: $\lambda_{PEC} = 0.141 \times 10^3$, $\mathcal{E}^{PEC}_m = 13.03$).

5 Conclusion

We have presented the CFS (Constant Flow Sampling) method that allows one to find the optimal regularization parameter λ^* of a warp with respect to a given test cost function. It proceeds by sampling the values of λ such that the flow of the warp between two consecutive values is kept approximatively constant. CFS guarantees that the global minimum is found within a user-defined tolerance,

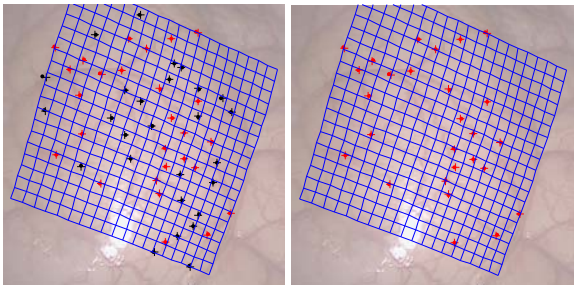

Fig. 7. The deformation grid of the warp, left: TTS criterion, right: PEC criterion, dots: warped source points, crosses: target points, red: training set, black: test set.

under mild constraints on the test cost function. We have successfully tested this method with the photometric error criterion and with the training/test splitting, but it can also be used with any criterion such Ordinary-CV or V-fold CV. A further step will be to implement our method for nonlinear warps, by finding a way to approximate the flow of the warp with respect to λ.

References

1. Bartoli, A.: Maximizing the predictivity of smooth deformable image warps through cross-validation. Journal of Mathematical Imaging and Vision 31(2-3), 135–145 (2008)
2. Brabanter, J.D., Pelckmans, K., Suykens, J., Vandewalle, J., Moor, B.D.: Robust cross-validation score functions with application to weighted least squares support vector machine function estimation. Tech. rep., KU Leuven (2003)
3. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: ECCV (2004)
4. Brunet, F., Bartoli, A., Navab, N., Malgouyres, R.: NURBS warps. In: BMVC (2009)
5. Brunet, F., Bartoli, A., Navab, N., Malgouyres, R.: Pixel-based hyperparameter selection for feature-based image registration. In: VMV (2010)
6. Chambolle, A., Darbon, J.: On total variation minimization and surface evolution using parametric maximal flows. International Journal of Computer Vision 84(3) (2009)
7. Farenzena, M., Bartoli, A., Mezouar, Y.: Efficient camera smoothing in sequential structure-from-motion using approximate cross-validation. In: ECCV (2010)
8. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning. Springer (2003)
9. Lancaster, P., Salkauskas, K.: Surfaces generated by moving least squares methods. Mathematics of Computation 37(155), 141–158 (1981)
10. 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 Transactions on Medical Imaging 18(8), 712–721 (1999)
11. Wahba, G.: Spline Models for Observational Data. Society for Industrial and Applied Mathematics (1990)