Approximate Thin Plate Spline Mappings

Gianluca Donato[1] and Serge Belongie[2]

[1] Digital Persona, Inc., Redwood City, CA 94063
gianlucad@digitalpersona.com
[2] U.C. San Diego, La Jolla, CA 92093-0114
sjb@cs.ucsd.edu

Abstract. The thin plate spline (TPS) is an effective tool for modeling coordinate transformations that has been applied successfully in several computer vision applications. Unfortunately the solution requires the inversion of a $p \times p$ matrix, where p is the number of points in the data set, thus making it impractical for large scale applications. As it turns out, a surprisingly good approximate solution is often possible using only a small subset of corresponding points. We begin by discussing the obvious approach of using the subsampled set to estimate a transformation that is then applied to all the points, and we show the drawbacks of this method. We then proceed to borrow a technique from the machine learning community for function approximation using radial basis functions (RBFs) and adapt it to the task at hand. Using this method, we demonstrate a significant improvement over the naive method. One drawback of this method, however, is that is does not allow for *principal warp* analysis, a technique for studying shape deformations introduced by Bookstein based on the eigenvectors of the $p \times p$ *bending energy matrix*. To address this, we describe a third approximation method based on a classic matrix completion technique that allows for principal warp analysis as a by-product. By means of experiments on real and synthetic data, we demonstrate the pros and cons of these different approximations so as to allow the reader to make an informed decision suited to his or her application.

1 Introduction

The thin plate spline (TPS) is a commonly used basis function for representing coordinate mappings from $\mathbb{R}^2$ to $\mathbb{R}^2$. Bookstein [3] and Davis et al. [5], for example, have studied its application to the problem of modeling changes in biological forms. The thin plate spline is the 2D generalization of the cubic spline. In its regularized form the TPS model includes the affine model as a special case.

One drawback of the TPS model is that its solution requires the inversion of a large, dense matrix of size $p \times p$, where p is the number of points in the data set. Our goal in this paper is to present and compare three approximation methods that address this computational problem through the use of a subset of corresponding points. In doing so, we highlight connections to related approaches in the area of Gaussian RBF networks that are relevant to the TPS mapping

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2352, pp. 21–31, 2002.

problem. Finally, we discuss a novel application of the *Nystrd̃m approximation* [1] to the TPS mapping problem.

Our experimental results suggest that the present work should be particularly useful in applications such as shape matching and correspondence recovery (e.g. [2,7,4]) as well as in graphics applications such as morphing.

2 Review of Thin Plate Splines

Let v_i denote the target function values at locations (x_i, y_i) in the plane, with $i = 1, 2, \ldots, p$. In particular, we will set v_i equal to the target coordinates (x_i', y_i') in turn to obtain one continuous transformation for each coordinate. We assume that the locations (x_i, y_i) are all different and are not collinear. The TPS interpolant $f(x, y)$ minimizes the bending energy

$$I_f = \iint_{\mathbb{R}^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)dxdy$$

and has the form

$$\left[f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{p} w_i U\left(\|(x_i, y_i) - (x, y)\|\right) \right] = w^T x$$

where $U(r) = r^2 \log r$. In order for $f(x, y)$ to have square integrable second derivatives, we require that

$$\sum_{i=1}^{p} w_i = 0 \quad \text{and}$$

$$\sum_{i=1}^{p} w_i x_i = \sum_{i=1}^{p} w_i y_i = 0 \ .$$

Together with the interpolation conditions, $f(x_i, y_i) = v_i$, this yields a linear system for the TPS coefficients:

$$\begin{bmatrix} K & P \\ P^T & O \end{bmatrix} \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} v \\ o \end{bmatrix} \tag{1}$$

where $K_{ij} = U(\|(x_i, y_i) - (x_j, y_j)\|)$, the ith row of P is $(1, x_i, y_i)$, O is a 3×3 matrix of zeros, o is a 3×1 column vector of zeros, w and v are column vectors formed from w_i and v_i, respectively, and a is the column vector with elements a_1, a_x, a_y. We will denote the $(p + 3) \times (p + 3)$ matrix of this system by L; as discussed e.g. in [7], L is nonsingular. If we denote the upper left $p \times p$ block of L^{-1} by L_p^{-1}, then it can be shown that

$$I_f \propto v^T L_p^{-1} v = \boxed{w^T K w} \ .$$

When there is noise in the specified values v_i, one may wish to relax the exact interpolation requirement by means of regularization. This is accomplished by minimizing

$$H[f] = \sum_{i=1}^{n} (v_i - f(x_i, y_i))^2 + \lambda I_f \ .$$

The *regularization parameter* λ, a positive scalar, controls the amount of smoothing; the limiting case of $\lambda = 0$ reduces to exact interpolation. As demonstrated in [9,6], we can solve for the TPS coefficients in the regularized case by replacing the matrix K by $K + \lambda I$, where I is the $p \times p$ identity matrix.

3 Approximation Techniques

Since inverting L is an $O(p^3)$ operation, solving for the TPS coefficients can be very expensive when p is large. We will now discuss three different approximation methods that reduce this computational burden to $O(m^3)$, where m can be as small as $0.1p$. The corresponding savings factors in memory (5x) and processing time (1000x) thus make TPS methods tractable when p is very large.

In the discussion below we use the following partition of the K matrix:

$$K = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \tag{2}$$

with $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{m \times n}$, and $C \in \mathbb{R}^{n \times n}$. Without loss of generality, we will assume the p points are labeled in random order, so that the first m points represent a randomly selected subset.

3.1 Method 1: Simple Subsampling

The simplest approximation technique is to solve for the TPS mapping between a randomly selected subset of the correspondences. This amounts to using A in place of K in Equation (1). We can then use the recovered coefficients to extrapolate the TPS mapping to the remaining points. The result of applying this approximation to some sample shapes is shown in Figure 1. In this case, certain parts were not sampled at all, and as a result the mapping in those areas is poor.

3.2 Method 2: Basis Function Subset

An improved approximation can be obtained by using a subset of the basis functions with all of the target values. Such an approach appears in [10,6] and Section 3.1 of [8] for the case of Gaussian RBFs. In the TPS case, we need to account for the affine terms, which leads to a modified set of linear equations. Starting from the cost function

$$R[\tilde{w}, a] = \frac{1}{2}\|v - \tilde{K}\tilde{w} - Pa\|^2 + \frac{\lambda}{2}\tilde{w}^T A \tilde{w} \ ,$$

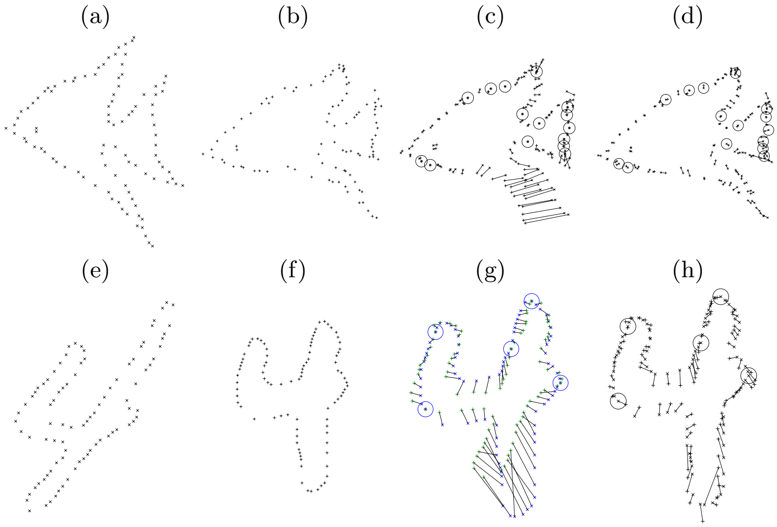

Fig. 1. Thin plate spline (TPS) mapping example. (a,b) Template and target synthetic fish shapes, each consisting of 98 points. (Correspondences between the two shapes are known.) (c) TPS mapping of (a) onto (b) using the subset of points indicated by circles (Method 1). Corresponding points are indicated by connecting line segments. Notice the quality of the mapping is poor where the samples are sparse. An improved approximation can be obtained by making use of the full set of target values; this is illustrated in (d), where we have used Method 2 (discussed in Section 3.2). A similar mapping is found for the same set of samples using Method 3 (see Section 3.3). In (e-h) we observe the same behavior for a pair of handwritten digits, where the correspondences (89 in all) have been found using the method of [2].

we minimize it by setting $\partial R/\partial\tilde{w}$ and $\partial R/\partial a$ to zero, which leads to the following $(m+3)\times(m+3)$ linear system,

$$\begin{bmatrix} \tilde{K}^T\tilde{K}+\lambda A & \tilde{K}^T P \\ P^T\tilde{K} & P^T P \end{bmatrix} \begin{bmatrix} \tilde{w} \\ a \end{bmatrix} = \begin{bmatrix} \tilde{K}^T v \\ P^T v \end{bmatrix} \tag{3}$$

where $\tilde{K}^T = [A \quad B^T]$, $\tilde{w}$ is an $m\times 1$ vector of TPS coefficients, and the rest of the entries are as before. Thus we seek weights for the reduced set of basis functions that take into account the full set of p target values contained in v. If we call $\tilde{P}$ the first m rows of P and $\tilde{I}$ the first m columns of the $p\times p$ identity matrix, then under the assumption $\tilde{P}^T\tilde{w}=0$, Equation (3) is equivalent to

$$\begin{bmatrix} \tilde{K}+\lambda\tilde{I} & P \\ \tilde{P}^T & O \end{bmatrix} \begin{bmatrix} \tilde{w} \\ a \end{bmatrix} = \begin{bmatrix} v \\ o \end{bmatrix}$$

which corresponds to the regularized version of Equation (1) when using the subsampled $\tilde{K}$ and $\tilde{P}^T$ in place of K and P^T.

The application of this technique to the fish and digit shapes is shown in Figure 1(d,h).

3.3 Method 3: Matrix Approximation

The essence of Method 2 was to use a subset of exact basis functions to approximate a full set of target values. We now consider an approach that uses a full set of approximate basis functions to approximate the full set of target values. The approach is based on a technique known as the Nyström method.

The Nyström method provides a means of approximating the eigenvectors of K without using C. It was originally developed in the late 1920s for the numerical solution of eigenfunction problems [1] and was recently used in [11] for fast approximate Gaussian process regression and in [8] (implicitly) to speed up several machine learning techniques using Gaussian kernels. Implicit to the Nyström method is the assumption that C can be approximated by $B^T A^{-1} B$, i.e.

$$\hat{K} = \begin{bmatrix} A & B \\ B^T & B^T A^{-1} B \end{bmatrix} \tag{4}$$

If $\mathrm{rank}(K) = m$ and the m rows of the submatrix $[A \quad B]$ are linearly independent, then $\hat{K} = K$. In general, the quality of the approximation can be expressed as the norm of the difference $C - B^T A^{-1} B$, the Schur complement of K.

Given the $m \times m$ diagonalization $A = U \Lambda U^T$, we can proceed to find the approximate eigenvectors of K:

$$\hat{K} = \tilde{U} \Lambda \tilde{U}^T, \quad \text{with} \quad \tilde{U} = \begin{bmatrix} U \\ B^T U \Lambda^{-1} \end{bmatrix} \tag{5}$$

Note that in general the columns of $\tilde{U}$ are not orthogonal. To address this, first define $Z = \tilde{U} \Lambda^{1/2}$ so that $\hat{K} = Z Z^T$. Let $Q \Sigma Q^T$ denote the diagonalization of $Z^T Z$. Then the matrix $V = Z Q \Sigma^{-1/2}$ contains the leading orthonormalized eigenvectors of $\hat{K}$, i.e. $\hat{K} = V \Sigma V^T$, with $V^T V = I$.

From the standard formula for the partitioned inverse of L, we have

$$L^{-1} = \begin{bmatrix} K^{-1} + K^{-1} P S^{-1} P^T K^{-1} & -K^{-1} P S^{-1} \\ -S^{-1} P^T K^{-1} & S^{-1} \end{bmatrix}, \quad S = -P^T K^{-1} P$$

and thus

$$\begin{bmatrix} w \\ a \end{bmatrix} = L^{-1} \begin{bmatrix} v \\ o \end{bmatrix} = \begin{bmatrix} (I + K^{-1} P S^{-1} P^T) K^{-1} v \\ -S^{-1} P^T K^{-1} v \end{bmatrix}$$

Using the Nyström approximation to K, we have $\hat{K}^{-1} = V \Sigma^{-1} V^T$ and

$$\hat{w} = (I + V \Sigma^{-1} V^T P \hat{S}^{-1} P^T) V \Sigma^{-1} V^T v \ ,$$
$$\hat{a} = -\hat{S}^{-1} P^T V \Sigma^{-1} V^T v$$

with $\hat{S} = -P^T V \Sigma^{-1} V^T P$, which is 3 × 3. Therefore, by computing matrix-vector products in the appropriate order, we can obtain estimates to the TPS

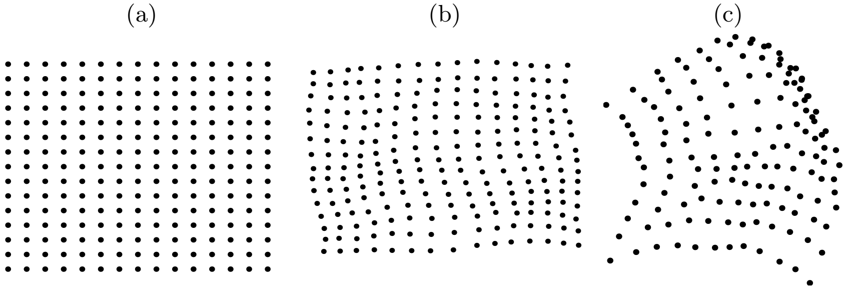

Fig. 2. Grids used for experimental testing. (a) Reference point set $\mathcal{S}_1$: 12 × 12 points on the interval $[0, 128] \times [0, 128]$. (b,c) Warped point sets $\mathcal{S}_2$ and $\mathcal{S}_3$ with bending energy 0.3 and 0.8, respectively. To test the quality of the different approximation methods, we used varying percentages of points to estimate the TPS mapping from $\mathcal{S}_1$ to $\mathcal{S}_2$ and from $\mathcal{S}_1$ to $\mathcal{S}_3$.

coefficients without ever having to invert or store a large $p \times p$ matrix. For the regularized case, one can proceed in the same manner, using

$$(V \Sigma V^T + \lambda I)^{-1} = V(\Sigma + \lambda I)^{-1} V^T \ .$$

Finally, the approximate bending energy is given by

$$w^T \hat{K} w = (V^T w)^T \Sigma (V^T w)$$

Note that this bending energy is the average of the energies associated to the x and y components as in [3].

Let us briefly consider what $\hat{w}$ represents. The first m components roughly correspond to the entries in $\tilde{w}$ for Method 2; these in turn correspond to the columns of $\hat{K}$ (i.e. $\tilde{K}$) for which exact information is available. The remaining entries weight columns of $\hat{K}$ with (implicitly) filled-in values for all but the first m entries. In our experiments, we have observed that the latter values of $\hat{w}$ are nonzero, which indicates that these approximate basis functions are not being disregarded. Qualitatively, the approximation quality of methods 2 and 3 are very similar, which is not surprising since they make use of the same basic information. The pros and cons of these two methods are investigated in the following section.

4 Experiments

4.1 Synthetic Grid Test

In order to compare the above three approximation methods, we ran a set of experiments based on warped versions of the cartesian grid shown in Figure 2(a). The grid consists of 12 × 12 points in a square of dimensions 128 × 128. Call this set of points $\mathcal{S}_1$. Using the technique described in Appendix A, we generated

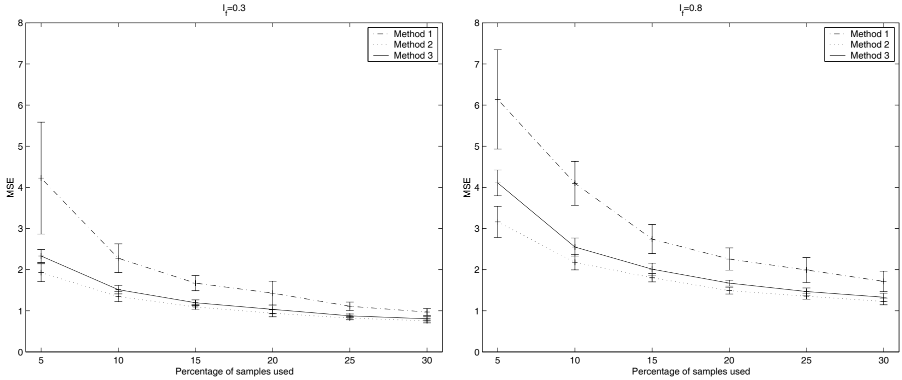

Fig. 3. Comparison of approximation error. Mean squared error in position between points in the target grid and corresponding points in the approximately warped reference grid is plotted vs. percentage of randomly selected samples used. Performance curves for each of the three methods are shown in (a) for $I_f = 0.3$ and (b) for $I_f = 0.8$.

point sets $\mathcal{S}_2$ and $\mathcal{S}_3$ by applying random TPS warps with bending energy 0.3 and 0.8, respectively; see Figure 2(b,c). We then studied the quality of each approximation method by varying the percentage of random samples used to estimate the (unregularized) mapping of $\mathcal{S}_1$ onto $\mathcal{S}_2$ and $\mathcal{S}_3$, and measuring the mean squared error (MSE) in the estimated coordinates. The results are plotted in Figure 3. The error bars indicate one standard deviation over 20 repeated trials.

4.2 Approximate Principal Warps

In [3] Bookstein develops a multivariate shape analysis framework based on eigenvectors of the *bending energy matrix* $L_p^{-1} K L_p^{-1} = L_p^{-1}$, which he refers to as *principal warps*. Interestingly, the first 3 principal warps always have eigenvalue zero, since any warping of three points in general position (a triangle) can be represented by an affine transform, for which the bending energy is zero. The shape and associated eigenvalue of the remaining principal warps lend insight into the bending energy "cost" of a given mapping in terms of that mapping's projection onto the principal warps. Through the Nyström approximation in Method 3, one can produce approximate principal warps using $\hat{L}_p^{-1}$ as follows:

$$\begin{aligned}
\hat{L}_p^{-1} &= \hat{K}^{-1} + \hat{K}^{-1} P S^{-1} P^T \hat{K}^{-1} \\
&= V \Sigma^{-1} V^T + V \Sigma^{-1} V^T P S^{-1} P^T V \Sigma^{-1} V^T \\
&= V(\Sigma^{-1} + \Sigma^{-1} V^T P S^{-1} P^T V \Sigma^{-1}) V^T \\
&\triangleq V \hat{\Lambda} V^T
\end{aligned}$$

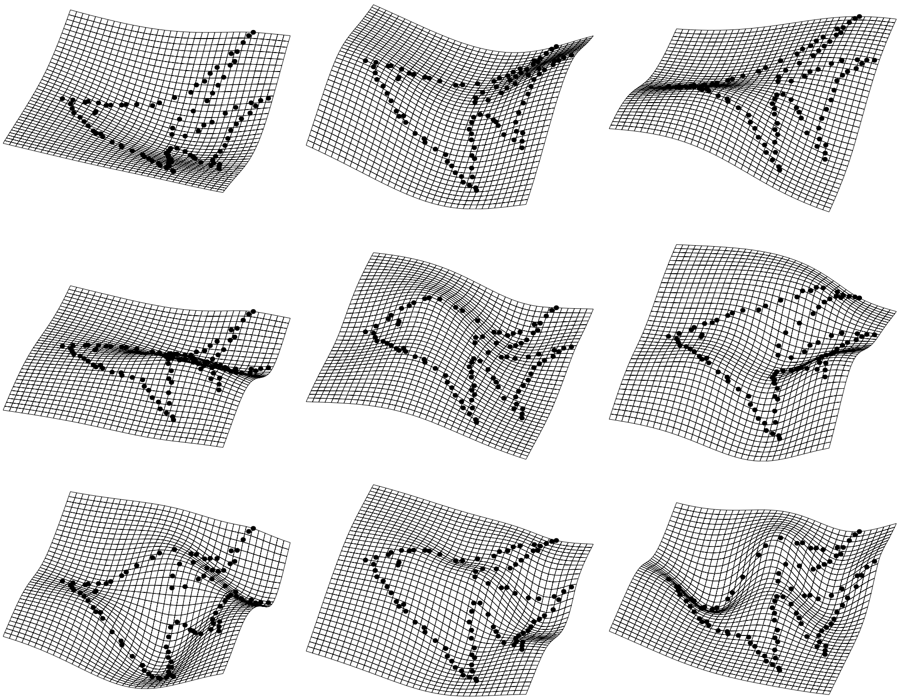

Fig. 4. Approximate principal warps for the fish shape. From left to right and top to bottom, the surfaces are ordered by eigenvalue in increasing order. The first three principal warps, which represent the affine component of the transformation and have eigenvalue zero, are not shown.

where

$$\hat{\Lambda} \stackrel{\Delta}{=} \Sigma^{-1} + \Sigma^{-1} V^T P S^{-1} P^T V \Sigma^{-1} = W D W^T$$

to obtain orthogonal eigenvectors we proceed as in section 3.3 to get

$$\hat{\Lambda} = \hat{W} \hat{\Sigma} \hat{W}^T$$

where $\hat{W} \stackrel{\Delta}{=} W D^{1/2} Q \hat{\Sigma}^{1/2}$ and $Q \hat{\Sigma} Q^T$ is the diagonalization of $D^{1/2} W^T W D^{1/2}$. Thus we can write

$$\hat{L}_p^{-1} = V \hat{W} \hat{\Sigma} \hat{W}^T V^T$$

An illustration of approximate principal warps for the fish shape is shown in Figure 4, wherein we have used $m = 15$ samples. As in [3], the principal warps are visualized as continuous surfaces, where the surface is obtained by applying a warp to the coordinates in the plane using a given eigenvector of $\hat{L}_p^{-1}$ as the nonlinear spline coefficients; the affine coordinates are set to zero. The

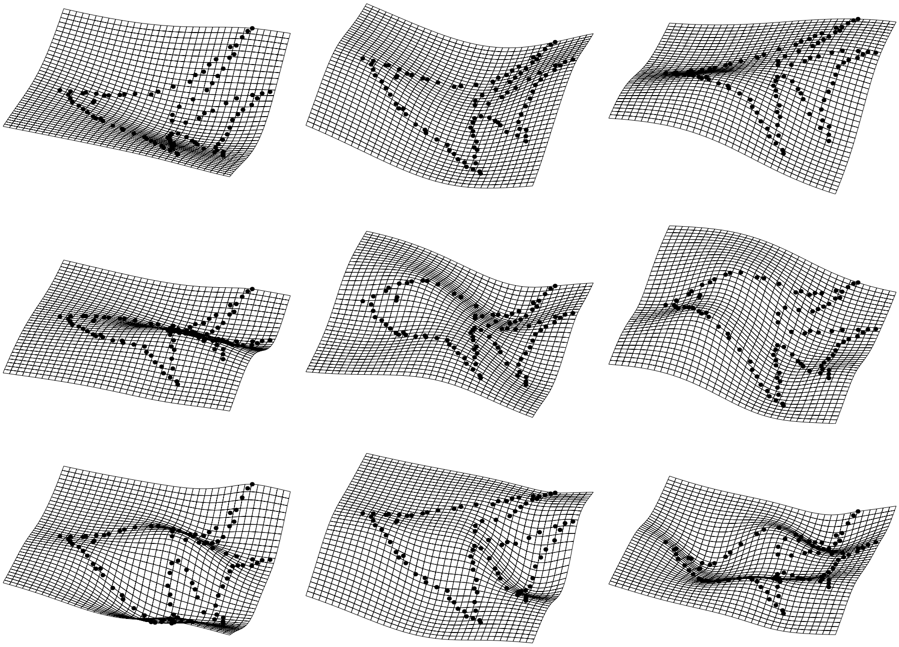

Fig. 5. Exact principal warps for the fish shape.

corresponding exact principal warps are shown in Figure 5. In both cases, warps 4 through 12 are shown, sorted in ascending order by eigenvalue.

Given a rank m Nyström approximation, at most $m-3$ principal warps with nonzero eigenvalue are available. These correspond to the principal warps at the "low frequency" end, meaning that very localized warps, e.g. pronounced stretching between adjacent points in the target shape, will not be captured by the approximation.

4.3 Discussion

We now discuss the relative merits of the above three methods. From the synthetic grid tests we see that Method 1, as expected, has the highest MSE. Considering that the spacing between neighboring points in the grid is about 10, it is noteworthy, however, that all three methods achieve an MSE of less than 2 at 30% subsampling. Thus while Method 1 is not optimal in the sense of MSE, its performance is likely to be reasonable for some applications, and it has the advantage of being the least expensive of the three methods.

In terms of MSE, Methods 2 and 3 perform roughly the same, with Method 2 holding a slight edge, more so at 5% for the second warped grid. Method 3 has a disadvantage built in relative to Method 2, due to the orthogonalization

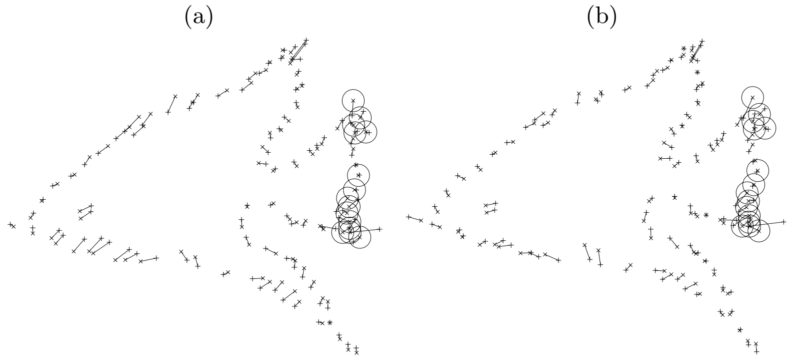

Fig. 6. Comparison of Method 2 (a) and 3 (b) for poorly chosen sample locations. (The performance of Method 1 was terrible and is not shown.) Both methods perform well considering the location of the samples. Note that the error is slightly lower for Method 3, particularly at points far away from the samples.

step; this leads to an additional loss in significant figures and a slight increase in MSE. In this regard Method 2 is the preferred choice.

While Method 3 is comparatively expensive and has slightly higher MSE than Method 2, it has the benefit of providing approximate eigenvectors of the bending energy matrix. Thus with Method 3 one has the option of studying shape transformations using principal warp analysis.

As a final note, we have observed that when the samples are chosen badly, e.g. crowded into a small area, Method 3 performs better than Method 2. This is illustrated in Figure 6, where all of the samples have been chosen at the back of the tail fin. Larger displacements between corresponding points are evident near the front of the fish for Method 2. We have also observed that the bending energy estimate of Method 2 ($\tilde{w}^T A \tilde{w}$) exhibits higher variance than that of Method 3; e.g. at a 20% sampling rate on the fish shapes warped using $I_f = 0.3$ over 100 trials, Method 2 estimates I_f to be 0.29 with $\sigma = 0.13$ whereas Method 3 gives 0.25 and $\sigma = 0.06$. We conjecture that this advantage arises from the presence of the approximate basis functions in the Nyström approximation, though a rigorous explanation is not known to us.

5 Conclusion

We have discussed three approximate methods for recovering TPS mappings between 2D pointsets that greatly reduce the computational burden. An experimental comparison of the approximation error suggests that the two methods that use only a subset of the available correspondences but take into account the full set of target values perform very well. Finally, we observed that the method based on the Nyström approximation allows for principal warp analysis and performs better than the basis-subset method when the subset of correspondences is chosen poorly.

Acknowledgments. The authors wish to thanks Charless Fowlkes, Jitendra Malik, Andrew Ng, Lorenzo Torresani, Yair Weiss, and Alice Zheng for helpful discussions. We would also like to thank Haili Chui and Anand Rangarajan for useful insights and for providing the fish datasets.

Appendix: Generating Random TPS Transformations

To produce a random TPS transformation with bending energy ν, first choose a set of p reference points (e.g. on a grid) and form L_p^{-1}. Now generate a random vector u, set its last three components to zero, and normalize it. Compute the diagonalization $L_p^{-1} = U \Lambda U^T$, with the eigenvalues sorted in descending order. Finally, compute $w = \sqrt{\nu} U \Lambda^{1/2} u$. Since I_f is unaffected by the affine terms, their values are arbitrary; we set translation to $(0, 0)$ and scaling to $(1, 0)$ and $(0, 1)$.

References

1. C. T. H. Baker. *The numerical treatment of integral equations.* Oxford: Clarendon Press, 1977.
2. S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Proc. 8th Int'l. Conf. Computer Vision*, volume 1, pages 454–461, July 2001.
3. F. L. Bookstein. Principal warps: thin-plate splines and decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.
4. H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 44–51, June 2000.
5. M.H. Davis, A. Khotanzad, D. Flamig, and S. Harms. A physics-based coordinate transformation for 3-d image matching. *IEEE Trans. Medical Imaging*, 16(3):317–328, June 1997.
6. F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.
7. M. J. D. Powell. A thin plate spline method for mapping curves into curves in two dimensions. In *Computational Techniques and Applications (CTAC95)*, Melbourne, Australia, 1995.
8. A.J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *ICML*, 2000.
9. G. Wahba. *Spline Models for Observational Data.* SIAM, 1990.
10. Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 520–526, 1997.
11. C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, pages 682–688, 2001.