MULTI-LEVEL FAST MULTIPOLE METHOD FOR THIN PLATE SPLINE EVALUATION

Ali Zandifar, Ser-nam Lim, Ramani Duraiswami, Nail A. Gumerov and Larry S. Davis

Perceptual Interfaces and Reality Lab
University of Maryland, College Park MD 20742

ABSTRACT

Image registration is an important problem in image processing and computer vision. Much recent work in image registration is on matching non-rigid deformations. Thin Plate Splines are an effective image registration method when the deformation between two images can be modeled as the bending of a thin metal plate on point constraints such that the topology is preserved (non-rigid deformation). However, because evaluating the computed TPS model at all the image pixels is computationally expensive, we need to speed it up. We introduce the use of Multi-Level Fast Multipole Method (MLFMM) for this purpose. Our contribution lies in the presentation of a clear and concise MLFMM framework for TPS, which will be useful for future application developments. The achieved speedup using MLFMM is an improvement from $O(N^2)$ to $O(N \log N)$. We show that the fast evaluation outperforms the brute force method while maintaining acceptable error bound.

1. INTRODUCTION

The image registration problem is to establish correspondences given a pair of images. In general, we have two types of image registration based on the mapping model: rigid and non-rigid. In a rigid model, relative distances of points in 3D are conserved e.g. rotation/translation. In a non-rigid model, points are displaced while maintaining the topology. Examples include the deformations of a talking lip, fingerprint matching etc. In this paper, we consider Thin Plate Spline (TPS) which have been shown as an effective method for non-rigid image registration ([1]). Previous work such as [2, 3, 4, 5] have successfully applied TPS to different applications.

TPS is however computationally expensive. The high cost of using TPS comes from an inverse operation that is $O(N^3)$ and the evaluation process that is $O(N^2)$ using brute force approaches (Sec 2 for details). The first problem has been nicely dealt with in [6] using an approximation approach. For the latter problem, while earlier work such as [7] has proposed the use of Multi-Level Fast Multipole Method (MLFMM), the need remains for a clear, concise and general mathematical framework for using MLFMM in TPS. While [8] has provided an MLFMM framework for evaluation of polyharmonic splines, our work extended upon it specifically for TPS. In addition, we added the formulations for local expansions and local-to-local translation (Sec 3.1 for details) that was missing in [8].

The Fast Multipole Method (FMM) was first introduced by *Greengard* and *Rokhlin* in [9] and was developed for the fast summation of potential fields generated by a large number of sources in gravitational and electro-statistical problems. Lately, this method is extended to other potential problems in computer vision [10], computer graphics, acoustics and so on. MLFMM is in turn a multi-level approach to FMM. In MLFMM, a hierarchy of grids for subdivision of space, nested at multiple scales, and a corresponding hierarchical organization of charge groups and multipole expansions makes a computational complexity of $O(N \log N)$ possible.

The rest of this paper is organized as follows. Sec 2 briefly introduces TPS and the cost of using a brute force approach. Sec 3 describes MLFMM as an effective method for speeding up the evaluation of TPS. The mathematical formulation for the TPS problem is provided in Sec 3.1. Finally, we will present the results and conclude in Sec 4.

2. THIN PLATE SPLINE WARPING

Consider two images $I_1$and I_2. A pixel in I_1 at (x, y) is mapped to a pixel in I_2 at (x', y') by a thin plate spline function as:

$$(x', y') = (f_1(x, y), f_2(x, y)), \tag{1}$$

if functions $f_1$and f_2 are TPS mappings. The TPS model for one of the transformed coordinates is given by parameter vectors $\mathbf{a}$ and $\mathbf{w}$ as

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{p} w_i U(r) \tag{2}$$

where $U(r) = r^2 \log(r)$, r is $||(x, y) - (x_i, y_i)||$, initial landmarks (or control points) are represented as (x_i, y_i), vector a defines the affine part of the mapping and vector w denotes the nonlinear part of the deformation. As with a typical spline function, TPS can interpolate in-between control points while maintaining smoothness criteria. This smoothness is defined as the minimum bending energy ([6]):

$$E_f = \int \int_{R^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dxdy, \tag{3}$$

where E_f is a total bending energy expressed as a function of the second derivatives of a mapping f (geometrically the second derivatives are related to a curvature of f in space). The physical interpretation is that the surface deformed elastically in order to have minimum bending energy (smooth stretching). To minimize the total bending energy (maximum smoothness), it can be shown that:

$$\sum_{i=1}^{p} w_i = \sum_{i=1}^{p} w_i x_i = \sum_{i=1}^{p} w_i y_i = 0. \quad (4)$$

Therefore, we can rewrite Eqn.(2) in a linear matrix equation as:

$$\begin{bmatrix} K & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix}, \quad (5)$$

where O is a 3×3 zero matrix, 0 is 3×1 zero matrix, v are target values, entries of K are $K_{ij} = U(||(x,y) - (x_i, y_i)||$ and $P_{ij} = (1, x_i, y_i)$.

The use of TPS involves two steps: *parameter estimation* and *evaluation*. In the parameter estimation step, the vector parameter w and a^T can be naively computed using direct inverse from an initial set of point correspondences. This is however computationally expensive giving an asymptotic bound of $O(N^3)$. Since [6] provides a nice approximation method to speedup this step, we will not dwell on this problem and refer readers to [6] for details. Another bottleneck comes from the evaluation step in the computation of $\sum_{i=1}^{p} w_i U(||(x,y) - (x_i, y_i)||)$ in Eqn. (2). Using a brute force approach in the evaluation step incurs a cost of $O(N^2)$. By approximating with MLFMM, the complexity can be improved to $O(N \log N)$. A concise MLFMM framework for this evaluation is thus necessary.

3. FAST MULTIPOLE METHOD

The FMM is suitable for evaluating a potential function ϕ which analytically has multipole and local expansions, factorization and grouping properties (such as TPS) as described in [9]. The FMM represents a fundamental change in the way of designing numerical algorithms, such that it solves the problem approximately, and trades exactness for better complexity. However, in practice, the errors resulting from FMM is usually too small to cause any concern. In this section, we will briefly state the FMM method. Consider the sum

$$v(y_j) = \sum_{i=1}^{N} u_i \phi(u_i, y_j), \quad j = 1, ..., N \quad (6)$$

where u_i's are source points and y_j's are evaluation points. Direct evaluation of Eqn. 6 has a complexity of $O(N^2)$(Fig. 1-a). In the FMM, we approximate the function ϕ in multipole series: singular series denoted by M and regular series denoted by L. If the function has grouping and factorization property which can be expanded to singular and regular series, FMM can be used for the evaluation step and reduces the asymptotic bound to $O(N \log N)$. In the FMM literature, singular and regular series expansion are often called multipole and local expansions.

If the sum in Eqn. 6 is evaluated by subdividing the data points into boxes at different levels, it is often referred to as the MLFMM. Each box is evaluated using a multipole expansion of ϕ. The potentials of the boxes are then translated and summed upwards ($M2M$ and $M2L$ translation) before going downwards ($L2L$ translation) again to the lowest levels [11]. We show in Fig. 1, the schematic of the standard method versus single level FMM (SLFMM) and MLFMM.

In the next section, we will provide the analytical representation of near and far field expansions and translations for the MLFMM. We use the notations M and L expansions to represent multipole and local expansions respectively, $M2M$ to represent multipole to multipole translation, $M2L$ for multipole to local translation and $L2L$ for local to local field translation. a_k and b_k are coefficient of M expansion, g_k and h_k are coefficient of L expansion. In 3.1 the function ϕ is $r^2 \log(r)$ of the TPS model where r is $||(x,y) - (x_i, y_i)||$. We will introduce function approximation for $r^2 \log(r)$in terms of multipole and local expansions and translations basis.

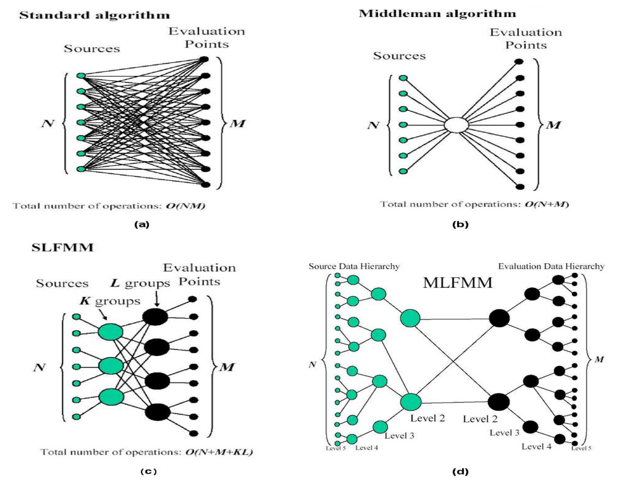

Fig. 1. Schematic representation of matrix-vector product for different methods of a) Standard method b) Middleman method c) FMM d) MLFMM. In standard method, the complexity is $O(NM)$ and we can speed up the algorithm by SLFMM and MLFMM to $O(M \log N)$. For some special cases of the function ϕ (Fig. 1-b), the complexity is $O(M + N)$. For more general functions, we can speed up the function evaluation of Eq. 6 using factorization methods like SLFMM and MLFMM.

3.1. Expansions and Translations

Beatson and *Light* gave the M expansion, $M2M$ translation and $M2L$ translation in [12], which will be restated here without proofs and error bounds. We will add the L-expansion and $L2L$ translation with proof needed for MLFMM. 2-D points are represented as complex numbers. We will also use $\Re\{z\}$ and $\Im\{z\}$ to represent the real and imaginary parts. We will use z to refer to $y - x_{*c}$ where y is the evaluation point and x_{*c} is the center of the current box. We will also use x_i's to refer to the source points and x_{*n} to refer to the center of the box we want to translate to. The multipole expansion of a function $\phi_t(z) = d|z - t| \log(|z - t|)$ for for $|t| < |z|$ is given as

$$s(z) = (\alpha|z|^2 - 2\Re\{\overline{\beta} z\} + \gamma) \log |z| + \Re\{\sum_{k=0}^{\infty} (a_k \overline{z} + b_k) z^{-k}\}, \quad (7)$$

where

$$t = x_i - x_{*c},\ \alpha = d,\ \beta = dt,\ \gamma = d|t|^2,$$
$$a_k = \begin{cases} -\beta, & k = 0 \\ \frac{dt^{k+1}}{k(k+1)}, & k > 0 \end{cases},\ b_k = -\bar{t} a_k,\ \ k \geq 0. \quad (8)$$

In this paper, we derived the local expansion by approximating $\phi_t(z) = d|z-t|\log(|z-t|)$ for $|t| > |z|$ as follows

$$s_1(z) = \Re\{\sum_{l=0}^{\infty}(g_l\overline{z} + h_l)z^l\}, \tag{9}$$

where t in Eqn. 8, g_l and h_l are

$$g_l = \begin{cases} -2dtlog|t|, & l=0 \\ d(1+log|t|), & l=1 \\ \frac{-dt^{-(l-1)}}{l(l-1)}, & l \geq 2 \end{cases},$$

$$h_l = \begin{cases} d|t|^2 log|t|, & l=0 \\ -d\bar{t}, & l=1 \\ \frac{d\bar{t}t^{-(l-1)}}{l(l-1)}, & l \geq 2 \end{cases}. \tag{10}$$

We can write down the local expansion of function ϕ from Eqn.(7) by changing t to z. So, with the $|t| > |z|$, we have

$$s_z(t) = (\alpha|t|^2 - 2\Re\{\overline{\beta}t\} + \gamma)\log|t| + \Re\{\sum_{k=0}^{\infty}(a_k\overline{t} + b_k)t^{-k}\}, \tag{11}$$

where

$$\alpha = d,\ \beta = dz,\ \gamma = d|z|^2$$
$$a_k = \begin{cases} -\beta, & k=0 \\ \frac{dz^{k+1}}{k(k+1)}, & k>0 \end{cases},$$
$$b_k = -\bar{z}a_k, \qquad k \geq 0. \tag{12}$$

Then

$$\begin{aligned} s_z(t) &= (d|t|^2 - 2\Re\{\overline{d}\bar{z}t\} + d|z|^2)\log|t| + \\ &\quad R\{a_0\bar{t} - za_0\} + \Re\{\sum_{k=1}^{\infty}(a_k\overline{t} + b_k)t^{-k}\} \\ &= (d|t|^2 - 2\Re\{\overline{d}\bar{z}t\} + d|z|^2)\log|t| - \\ &\quad R\{dz\bar{t} + zd\bar{z}\} + \Re\{\sum_{k=1}^{\infty}(\frac{dz^{k+1}}{k(k+1)}\bar{t} - \bar{z}\frac{dz^{k+1}}{k(k+1)})t^{-k}\} \\ &= \Re\{d|t|^2\log|t| - \overline{d}t\log|t|\bar{z} + d(1+\log|t|)|z|^2 - d\bar{t}z\} \\ &\quad +\Re\{\sum_{l=2}^{\infty}(\frac{dt^{-(l-1)}}{l(k-1)}\bar{t} - \bar{z}\frac{dt^{-(l-1)}}{l(l-1)})z^l\}. \end{aligned} \tag{13}$$

Therefore g_l and h_l are derived as Eqns.(9) and (10). $M2M$ translation is given by

$$s(z) = (\alpha|z|^2 - 2\Re\{\overline{\tilde{\beta}}z\} + \tilde{\gamma})\log|z| + \Re\{\sum_{k=0}^{\infty}(\tilde{a}_k\overline{z} + \tilde{b}_k)z^{-k}\}, \tag{14}$$

where

$$t = x_{*c} - x_{*n}\ ,\ \tilde{\beta} = \alpha t + \beta, \tilde{\gamma} = \gamma + 2\Re\{\overline{\beta}t\} + \alpha|t|^2,$$
$$\tilde{a}_k = \begin{cases} -\tilde{\beta}, & k=0 \\ \frac{\tilde{\beta}t^k}{k} - \frac{\alpha t^{k+1}}{k+1} + \sum_{j=1}^{k} a_j \begin{pmatrix} k-1 \\ j-1 \end{pmatrix} t^{k-j} & ,k \geq 1 \end{cases},$$
$$\tilde{b}_k = \begin{cases} \tilde{\gamma}, & k=0 \\ \frac{\overline{\tilde{\beta}}t^{k+1}}{k+1} - \frac{\tilde{\gamma}t^k}{k} + \sum_{j=1}^{k}(b_j - a_j\bar{t}) \begin{pmatrix} k-1 \\ j-1 \end{pmatrix} t^{k-j} & ,k \geq 1 \end{cases}. \tag{15}$$

$M2L$ translation is then given as

$$s_1(z) = \Re\{\sum_{l=0}^{\infty}(g_l\overline{z} + h_l)z^l\} \tag{16}$$

where using t in Eqn.(16), $\tilde{\beta}$ and $\tilde{\gamma}$ as given before

$$g_l = \begin{cases} -\tilde{\beta}\log|t| + \sum_{k=0}^{p} a_k(-1)^k\frac{1}{t^k}, & l=0 \\ \alpha\log|t| + \frac{\tilde{\beta}}{t} + \sum_{k=1}^{p} a_k(-1)^k\frac{k}{t^{l+k}}, & l=1 \\ \frac{-\alpha}{(l-1)t^{l-1}} + \frac{\tilde{\beta}}{lt^l} + \sum_{k=1}^{p} a_k(-1)^k \\ \times \begin{pmatrix} l+k-1 \\ k-1 \end{pmatrix} \frac{1}{t^{l+k}}, & l \geq 2 \end{cases},$$

$$h_l = \begin{cases} \tilde{\gamma}\log|t| + \sum_{k=0}^{p}(b_k - a_k\overline{t})(-1)^k\frac{1}{t^k}, & l=0 \\ -\overline{\tilde{\beta}}\log|t| - \frac{\tilde{\gamma}}{t} + \sum_{k=1}^{p}(b_k - a_k\overline{t})(-1)^k\frac{k}{t^{l+k}}, & l=1 \\ \frac{\overline{\tilde{\beta}}}{(l-1)t^{l-1}} - \frac{\tilde{\gamma}}{lt^l} + \sum_{k=1}^{p}(b_k - a_k\overline{t})(-1)^k \\ \times \begin{pmatrix} l+k-1 \\ k-1 \end{pmatrix} \frac{1}{t^{l+k}}, & l \geq 2 \end{cases}. \tag{17}$$

Additionally, our second contribution is to derive $L2L$ translation as follows:

$$s_1(z) = \Re\{\sum_{l=0}^{\infty}(g_l\overline{z} + h_l)z^l\}. \tag{18}$$

We expand $s_1(z-t)$, t in Eqn.(16)

$$s_1(z-t) = \Re\{\sum_{l=0}^{\infty}(g_l(\overline{z} - \overline{t}) + h_l)(z-t)^l\}. \tag{19}$$

and finally, if we can relate Taylor approximation of $s_1(z-t)$ with respect to $s_1(z)$, the coefficients $\tilde{g}_l, \tilde{h}_l$ are the new translated coefficients. Hence:

$$s_1(z-t) = \Re\{\sum_{l=0}^{\infty}(\tilde{g}_l\overline{z} + \tilde{h}_l)z^l\}. \tag{20}$$

We can show that

$$s_1(z-t) = \Re\{\sum_{l=0}^{\infty}(g_l\overline{z} + \hat{h}_l)(z-t)^l\} \tag{21}$$

where $\hat{h}_l = h_l - g_l\bar{t}$.Then, we can conclude after binomial expansion $(z-t)^l$

$$s_1(z-t) = \Re\{\sum_{l=0}^{\infty}(g_l\overline{z} + \hat{h}_l)\sum_{m=0}^{l}\begin{pmatrix} l \\ m \end{pmatrix}(-t)^{l-m}z^m\}. \tag{22}$$

Therefore, we have

$$s_1(z-t) = \Re\{\sum_{m=0}^{\infty}(\tilde{g}_m\overline{z} + \tilde{h}_m)z^m\},$$
$$\tilde{g}_m = \begin{cases} \sum_{l=0}^{p-1} g_l \begin{pmatrix} l \\ m \end{pmatrix}(-t)^{l-m}, & l \geq m \\ 0, & l < m \end{cases},$$
$$\tilde{h}_m = \begin{cases} \sum_{l=0}^{p-1} \hat{h}_l \begin{pmatrix} l \\ m \end{pmatrix}(-t)^{l-m}, & l \geq m \\ 0, & l < m \end{cases}. \tag{23}$$

4. RESULTS AND CONCLUSION

We have provided a concise MLFMM framework that is useful for future development of applications that use TPS. Our main contribution lies in the formulation of a complete MLFMM framework specifically for TPS, including the L-expansion and $L2L$-translation necessary for using MLFMM with TPS. The performance of TPS evaluation MLFMM approximation with truncated series are given in Fig. 2. Clearly, MLFMM outperforms the original evaluation while giving only a small error caused by series truncation. We also show the results of using TPS for image reconstruction Fig. 3 and morphing as shown in Fig. 4.

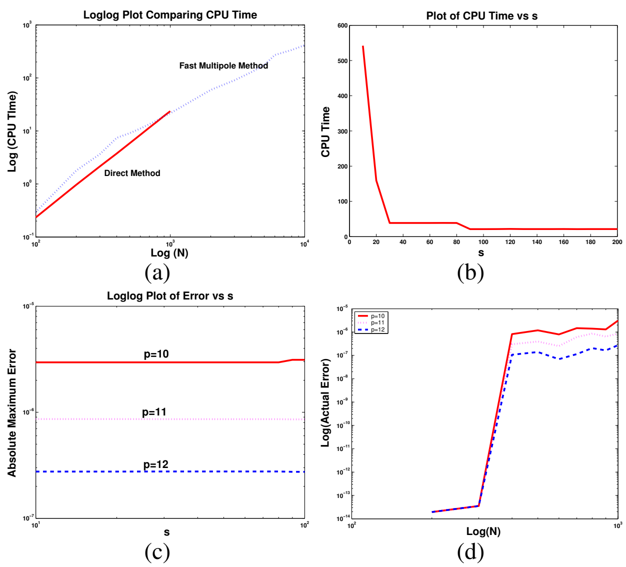

Fig. 2. Implementations are in MATLAB and p, s and N is the term at which series are truncated, the maximum number of data points per box at the finest level and the total number of data points respectively (a) Comparing speed of MLFMM and original evaluation (b) Variation of speed with the maximum number of data points per box at lowest level (c) Variation of error with the maximum number of data points per box at lowest level (d) Actual error with the respect to the number of input sources. Here, we can claim that with acceptable amount of error on the function evaluation (caused by truncation) e.g. $1e-10$ MLFMM outperforms the standard method.

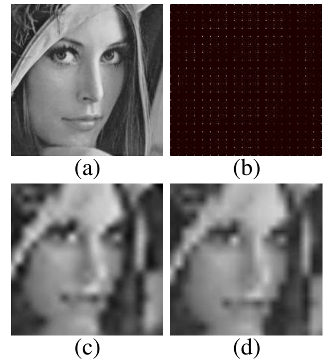

Fig. 3. (a) Original image (b) Subsampled image (c) Reconstructed image from subsampled image using TPS with MLFMM (PSNR=$14.8224db$) (d) Reconstructed image from subsampled image using bilinear interpolation (PSNR=$11.9131db$).

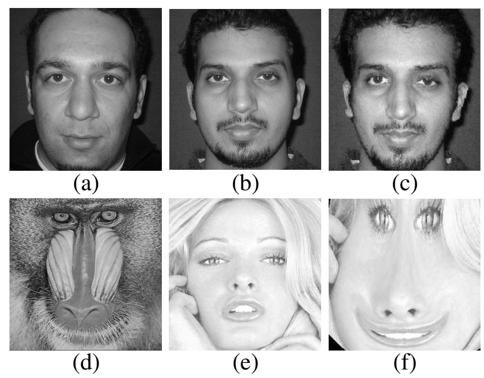

Fig. 4. (a)(b)(d)(e) Original images (c)(f) Morphed image using TPS with MLFMM and 10 landmarks.

5. REFERENCES

[1] F.L. Bookstein, "Principal warps: Thin plate splines and the decomposition of deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, 1989.

[2] N. Arad and D. Reisfeld, "Image warping using few anchor points and radial functions," *Computer Graphics forum, pp. 35–46*, vol. 14, pp. 35–46, 1995.

[3] A.M. Bazen and S.H. Gerez, "Elastic minutiae matching by means of thin-plate spline models," *International Conference on Pattern Recognition*, Aug 2002.

[4] S. Belongie, J. Malik, and J. Puzicha, "Matching shapes," *8th IEEE International Conference on Computer Vision*, Jul 2001.

[5] J.S.M. Vergeest and I. Horvath, "Matching 3d freeform shapes to digitized objects and its application in shape modeling," *7th International Conference on Production Engineering, Design and Control*, Feb 2001.

[6] G. Donato and S. Belongie, "Approximate thin plate spline mappings," in *European Conference on Computer Vision*, 2002.

[7] F. Chen and D. Suter, "Fast multipole method for accelerating the evaluation of splines," *IEEE Computer Science and Engineering*, Sep 1998.

[8] R.K. Beatson and W.A. Light, "Fast evaluation of polyharmonic splines," *IMA Journal on Numerical Analysis*, pp. 343–372, 1997.

[9] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *Journal Computational Physics*, vol. 73, pp. 325–348, 1987.

[10] C. Yang, R. Duraiswami, N. A. Gumerov, and L. S. Davis, "Improved fast gauss transform and efficient kernel density estimation," *Int. Conf. on Computer Vision*, 2001.

[11] N.A. Gumerov, R. Duraiswami, and Y.A. Borovikov, "Data structures, optimal choice of parameters, and complexity results for generalized fast multipole methods in d dimensions," *UMIACS-TR-2003-28, CS-TR-4458, University of Maryland, submitted to J. Comput. Phys.*, 2003.

[12] R.K. Beatson and G.N. Newsam, "Fast evaluation of radial basis function: I," *Computer Mathematics Applications*, vol. 24, no. 12, pp. 7–19, 1992.