A Unifying Framework for Non-linear Registration of 3D Objects

Zsolt Sánta and Zoltan Kato
Image Processing and Computer Graphics Dept., University of Szeged
H-6701 Szeged, PO. Box 652., Hungary, Fax: +36 62 546-397
Email: {santazs, kato}@inf.u-szeged.hu

***Abstract*—An extension of our earlier work is proposed to find a non-linear aligning transformation between a pair of deformable 3D objects. The basic idea is to set up a system of nonlinear equations whose solution directly provides the parameters of the aligning transformation. Each equation is generated by integrating a nonlinear function over the object's domains. Thus the number of equations is determined by the number of adopted nonlinear functions yielding a flexible mechanism to generate sufficiently many equations. While classical approaches would establish correspondences between the shapes, our method works without landmarks. Experiments with 3D polynomial and thin plate spline deformations confirm the performance of the framework.**

I. Introduction

A wide range of application areas requires the alignment of deformable objects [1], [2]. For example, the registration of 3D laser scans [3] or volumetric medical image alignment [4], but it is a crucial task also in object matching and change detection applications. When registering a pair of objects, first we have to characterize the possible deformations. From this point of view, registration techniques can be classified into two main categories: physical model-based and parametric or functional representation [5]. Herein, we deal with the latter representation, which typically originate from interpolation and approximation theory. Two broadly used classes of such parametric models are polynomials and splines, in particular thin plate splines (TPS) [6], [7].

Polynomial deformations are governed by fewer parameters and are acting *globally* on the shapes, hence regularization is not needed. Moreover, many non-polynomial transformation can be approximated by a polynomial one *e.g.* via a Taylor expansion [8].

Thin plate spline models are quite useful whenever a parametric free-form registration is required but the underlying physical model of the object deformation is unknown or too complex. Furthermore, TPS models can be extended to include various regularizations, such as the bending energy [6].

From a methodological point of view, we can differentiate *landmark-based* and *area-based* (or *feature-less*) approaches. Landmark-based methods are challenged by the correspondence problem, which is particularly difficult to solve in the case of non-linear deformations. Area-based approaches are typically relying on the availability of rich radiometric information which is used to construct a similarity measure based on some kind of intensity correlations. The aligning transformation is then found by maximizing similarity between the objects, which usually yields a complex non-linear optimization procedure. In many cases, reliable radiometric information may not be available (*e.g.* range data or multimodal medical images), therefore purely shape-based methods are particularly interesting [1], [2], [3], [4], [9].

Most of these approaches, however, are focusing on point set registration. In [9], a probabilistic model is proposed where a Gaussian mixture with centroids corresponding to the first set is fit to the second point set by maximizing the likelihood. In [1], both point sets are represented as a Gaussian mixture model and then the L2 distance of the two mixtures is minimized. While point-based registration has the advantage of tolerating rather high occlusions, they are typically inefficient for large point sets. In many cases, however, the goal is the accurate alignment of whole volumetric objects of several megavoxels (*e.g.* medical applications). Therefore object level registration methods are needed.

While elastic registration of planar shapes has been addressed by many researchers [10], [8], the extension of 2D methods for 3D object registration is far from trivial. Herein, we propose the extension of the correspondence-less approach in [8] for aligning 3D deformable objects subject to a diffeomorphic deformation. Following [8], we construct a system of nonlinear equations by integrating a set of nonlinear functions over the object volumes and then solve it by classical *Levenberg-Marquardt* algorithm. We have conducted a series of quantitative tests on a synthetic dataset to demonstrate the performance and robustness of the proposed approach.

II. Registration framework

The task is to determine the parameters of a general $\varphi : \mathbb{R}^3 \to \mathbb{R}^3$ *diffeomorphism*, which transforms the *template* shape to the *observation* shape. Let us denote the coordinates of the *template* and the *observation* by $\mathbf{x} = [x_1, x_2, x_3]^T$ and $\mathbf{y} = [y_1, y_2, y_3]^T$, respectively. Furthermore, let us define the inverse of φ as $\varphi^{-1} : \mathbb{R}^3 \to \mathbb{R}^3$. Then, the following relation holds:

$$\varphi(\mathbf{x}) = \mathbf{y} \quad \Longleftrightarrow \quad \mathbf{x} = \varphi^{-1}(\mathbf{y}). \tag{1}$$

The transformation function can be decomposed as $\varphi = [\varphi_1, \varphi_2, \varphi_3]^T$. In order to avoid the solving of the correspondence problem, we will integrate both sides of (1) over the foreground volumes, $\mathcal{F}_t$ and $\mathcal{F}_o$ of the *template* and *observation*, respectively [8]

$$\int_{\mathcal{F}_o} \mathbf{y} d\mathbf{y} = \int_{\mathcal{F}_t} \varphi(\mathbf{x}) |J_\varphi(\mathbf{x})| d\mathbf{x}, \tag{2}$$

where $|J_\varphi| : \mathbb{R}^3 \to \mathbb{R}$ is the Jacobian determinant of the deformation, defined as

$$|J_\varphi(\mathbf{x})| = \begin{vmatrix} \frac{\partial \varphi_1}{\partial x_1} & \frac{\partial \varphi_1}{\partial x_2} & \frac{\partial \varphi_1}{\partial x_3} \\ \frac{\partial \varphi_2}{\partial x_1} & \frac{\partial \varphi_2}{\partial x_2} & \frac{\partial \varphi_2}{\partial x_3} \\ \frac{\partial \varphi_3}{\partial x_1} & \frac{\partial \varphi_3}{\partial x_2} & \frac{\partial \varphi_3}{\partial x_3} \end{vmatrix} \tag{3}$$

Note that, since φ is a diffeomorphism, the Jacobian determinant will be non-vanishing over $\mathcal{F}_t$.

From (2) we could determine three parameters of φ. Unfortunately, in most cases the number of parameters are much higher than three.

The identity relation of (1) remains valid when a properly chosen function $\omega : \mathbb{R}^3 \to \mathbb{R}$ is acting on both sides of the equations. Using the integral equations from (2), we get the following [8]

$$\int_{\mathcal{F}_o} \omega(\mathbf{y}) d\mathbf{y} = \int_{\mathcal{F}_t} \omega(\varphi(\mathbf{x})) |J_\varphi(\mathbf{x})| d\mathbf{x}. \tag{4}$$

The basic idea is to generate a sufficient number of equations by making use of non-linear ω functions [8]. Let the number of parameters denoted by N and $\{\omega\}_{i=1}^{\ell}$, $\omega_i : \mathbb{R}^3 \to \mathbb{R}$ denote the set of adopted functions. To solve the problem, at least N equations are needed, hence $\ell \geq N$. From (4) these set of functions generates the following equations

$$\int_{\mathcal{F}_o} \omega_i(\mathbf{y}) d\mathbf{y} = \int_{\mathcal{F}_t} \omega_i(\varphi(\mathbf{x})) |J_\varphi(\mathbf{x})| d\mathbf{x}. \tag{5}$$

where $i = 1, \ldots, \ell$.

III. Transformation models

Although, the proposed approach can be used with any differentiable transformation function, herein we will focus on polynomial transformations and thin plate splines.

A. Polynomial transformations

A widely used class of deformations is the polynomial family, where the deformation field $p : \mathbb{R}^3 \to \mathbb{R}^3$, $p(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), p_3(\mathbf{x})]$ is given by three polynomial functions $p_i : \mathbb{R}^3 \to \mathbb{R}$. Without loss of generality, we can assume that $d = deg(p_1) = deg(p_2) = deg(p_3)$ [11]:

$$p_1(\mathbf{x}) = \sum_{i=0}^{d} \sum_{j=0}^{d-i} \sum_{k=0}^{d-i-j} a_{ijk} x_1^i x_2^j x_3^k$$
$$p_2(\mathbf{x}) = \sum_{i=0}^{d} \sum_{j=0}^{d-i} \sum_{k=0}^{d-i-j} b_{ijk} x_1^i x_2^j x_3^k$$
$$p_3(\mathbf{x}) = \sum_{i=0}^{d} \sum_{j=0}^{d-i} \sum_{k=0}^{d-i-j} c_{ijk} x_1^i x_2^j x_3^k.$$

The transformation has a total of $N = (d+3)(d+2)(d+1)/2$ parameters. The Jacobian determinant of p composed of the following partial derivatives [11]:

$$\frac{\partial p_1}{\partial x_1} = \sum_{i=1}^{d} \sum_{j=0}^{d-i} \sum_{k=0}^{d-i-j} i a_{ijk} x_1^{i-1} x_2^j x_3^k$$
$$\frac{\partial p_1}{\partial x_2} = \sum_{j=1}^{d} \sum_{i=0}^{d-j} \sum_{k=0}^{d-i-j} j a_{ijk} x_1^{i} x_2^{j-1} x_3^k$$
$$\frac{\partial p_1}{\partial x_3} = \sum_{k=1}^{d} \sum_{i=0}^{d-k} \sum_{j=0}^{d-i-k} k a_{ijk} x_1^{i} x_2^{j} x_3^{k-1}. \tag{6}$$

For p_2 and p_3, we get a similar formula.

B. Thin plate spline

Thin plate splines (TPS) [6], [7] are commonly used as parametric models for elastic deformations using radial basis functions. A TPS transformation $\varsigma : \mathbb{R}^3 \to \mathbb{R}^3$ can also be decomposed as three coordinate functions $\varsigma(\mathbf{x}) = [\varsigma_1(\mathbf{x}), \varsigma_2(\mathbf{x}), \varsigma_3(\mathbf{x})]^T$. Using a set of control points $c_k \in \mathbb{R}^3$, let us denote the linear parameters by $a_{ij} \in \mathbb{R}$, and the coefficients of the radial basis functions by $w_{ki} \in \mathbb{R}$ with $i = 1, \ldots, 3, j = 1, \ldots, 4$ and $k = 1, \ldots, K$. Then, the TPS functions can be written as

$$\varsigma_i(\mathbf{x}) = a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + a_{i4} + \sum_{k=1}^{K} w_{ki} Q(\|c_k - \mathbf{x}\|) \tag{7}$$

where $Q : \mathbb{R} \to \mathbb{R}$ is the radial basis function, which has the following form in 3D [6]:

$$Q(r) = |r|.$$

The number of the necessary parameters are $N = 3(K + 4)$ consisting of 12 linear parameters a_{ij} and 3 coefficients w_{ki} for each of the K control points c_k. The Jacobian determinant of ς composed of the partial derivatives of the transformation:

$$\frac{\partial \varsigma_i}{\partial x_j} = a_{ij} - \sum_{k=1}^{K} w_{ki} \frac{c_{kj} - x_j}{\|\mathbf{c}_k - \mathbf{x}\|} \qquad i, j = 1, 2, 3$$

The w_{ki} coefficients are also required to satisfy the following additional constraints [7], to ensure that the plate would not move or rotate under the imposition of the loads. The sum of the loads applied to the plate and the moments with respect to all axes should be 0:

$$\sum_{k=1}^{K} w_{ki} = 0 \quad \text{and} \quad \sum_{k=1}^{K} c_{k_j} w_{ki} = 0, \qquad i, j = 1, 2, 3. \tag{8}$$

In classical correspondence based approaches, control points are placed at extracted point matches, and the deformation at other positions is interpolated by the TPS. In this paper, however we are considering TPS as a parametric model to *approximate* the physical deformation. The control points can be placed *e.g.* on a uniform grid, to capture the local deformations everywhere.

IV. Numerical implementation

In practice, we only have a digital representation of the continuous objects $\mathcal{F}_t$ and $\mathcal{F}_o$, hence the equations of (4) are only approximately valid. In addition, objects extracted from images are subject to various segmentation errors, which again introduces errors in (4). Therefore, an overdetermined system is constructed, *i.e.* $\ell > N$ in (4), which is then solved in the least-squares sense via a *Levenberg-Marquardt* algorithm.

To increase numerical stability the *template* and *observation* coordinates are normalized into $[-0.5, 0.5]$ [8], [11], by applying normalizing transformations N_t and N_o, respectively:

$$N_o = \begin{pmatrix} o_1 & 0 & 0 & -o_1O_1 \\ 0 & o_2 & 0 & -o_2O_2 \\ 0 & 0 & o_3 & -o_3O_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{9}$$

$$N_t = \begin{pmatrix} t_1 & 0 & 0 & -t_1T_1 \\ 0 & t_2 & 0 & -t_2T_2 \\ 0 & 0 & t_3 & -t_3T_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{10}$$

where T_i, O_i $(i = 1, \ldots, 3)$ are the centroids of the template and the observation, respectively and t_i, o_i $(i = 1, \ldots, 3)$ are the scale factors. This will basically transform the input objects into a unit sphere centered at the origin.

Moreover, the range of the ω_i functions are also normalized into $[-1, 1]$ to equalize the contributions of each functions to the algebric error. Following [8], this is achieved by dividing the integrals with the maximal magnitude of the integral over a sphere with center in the origin and a radius $\frac{\sqrt{3}}{2}$:

$$N_i = \int_{\|\mathbf{x}\| \le \frac{\sqrt{3}}{2}} |\omega_i(\mathbf{x})| d\mathbf{x} \tag{11}$$

We thus obtain the following system of equations

$$\frac{1}{N_i} \int_{N_o(\mathcal{F}_o)} \omega_i(\mathbf{y}) d\mathbf{y} = \frac{1}{N_i} \int_{N_t(\mathcal{F}_t)} \omega(\varphi(\mathbf{x})) |J_\varphi(\mathbf{x})| d\mathbf{x}, \tag{12}$$

where $i = 1, \ldots, \ell$.

Using discrete voxel images, the integrals can be approximated via sums over the coordinates of the center of the voxels. For the ω_i functions we used simple power functions of the form of $\omega_i(\mathbf{x}) = x_1^{m_i} x_2^{n_i} x_3^{o_i}$. The equations system from (12) can be written as

$$\frac{1}{N_i} \sum_{\mathbf{y} \in F_o} |N_o|\, y_1^{m_i} y_2^{n_i} y_3^{o_i} = \frac{1}{N_i} \sum_{\mathbf{x} \in F_t} |N_t|\, \varphi_1(\mathbf{x})^{m_i} \varphi_2(\mathbf{x})^{n_i} \varphi_3(\mathbf{x})^{o_i} |J_\varphi(\mathbf{x})|, \tag{13}$$

where F_o and F_t are the normalized coordinates of the *observations* and the *template*, respectively and $|N_o|$ and $|N_t|$ are the determinants of the normalizing transformations (aka. the Jacobian determinants of the transformations).

The outline of the algorithm is given in Algorithm 1. The computational complexity of the proposed algorithm is

$$\mathcal{O}(\ell|F_o| + I|F_t|\ell), \tag{14}$$

where ℓ is the number of ω functions and I denotes the number of function calls executed by the *Levenberg-Marquardt* solver.

Algorithm 1 Pseudo code of the proposed algorithm

```
Input: template and observation voxels
Output: The transformation parameters of $\varphi$
1: Choose a set of $\ell > N$ nonlinear functions $\{\omega_i\}_{i=1}^{\ell}$, and
   for each $\omega_i$ compute the normalizing constant $N_i$ using (11).
2: Compute the normalizing transformations $N_t$ and $N_o$
   which maps voxel coordinates into $[-0.5, 0.5]$.
3: Construct the system of equations (12) using the approximations of (13).
4: Find a least-squares solution of the system using the
   Levenberg-Marquardt algorithm. Use the identity transformation for initialization.
5: Unnormalizing the solution gives the parameters of the
   aligning transformation.
```

As mentioned in the previous section, TPS has to satisfy the additional constrains of (8). Since these constraints are linear in the parameters of the transformation, the solution of the above system can be solved as a constrained least-square problem with linear constraints.

Note that, the transformation models can be non-diffeomorphic with the appropriate parameter sets. In our approach we have the assumption that the underlying transformation is diffeomorphic, but there is no guarantee to find a diffeomorphic solution. Although, in our experiments diffeomorphic solutions were found, this can be enforced by adding some further regularizations to the equation system (*e.g.* minimization of the bending energy) or adding penalties on nearly zero or negative Jacobians to the *Levenberg-Marquardt* solver.

Similarly to the two dimensional case, these transformation models can be written as $\varphi(\mathbf{x}) = \sum_{i=1}^{N} a_i \phi_i(\mathbf{x})$, where $a_i \in \mathbb{R}$ (the parameters) and $\phi_i : \mathbb{R}^3 \to \mathbb{R}$ are basis functions. Furthermore, when the ω functions are polynomial, the right-hand side of (13) can be separated with respect to a_i and $\phi_i(\mathbf{x})$, $i = 1, \ldots, N$ [8]. In the polynomial case the resulting equations will be polynomial equations, which gives the opportunity to precalculate the integrals over the images before the iterations [11]. However, the degree of these new polynomial equations can be enormously high (*e.g.* 33, for a third order polynomial transformation). The separation can be computed for the TPS also, but the model could have more than a hundred parameters, which makes the separation impractical.

V. Experimental results

The proposed approach has been tested on synthetic datasets generated by the corresponding transformation model.

The algorithm has been implemented in `C++` using the `levmar` library written by Lourakis [12]. All tests were ran under a Linux system running on a virtualized Core i5 3.1 GHz architecture. The registration error has been quantitatively evaluated based on the dice coefficient of the aligned objects:

$$\delta = \frac{|F_r \bigtriangleup F_o|}{|F_r| + |F_o|} \cdot 100\%, \qquad (15)$$

where F_o and F_r denote the set of foreground voxels of the *observation* and *registered* objects respectively. Some of our results are presented in Fig. 1, where *observation* and *registered* objects were overlayed, overlapping voxels are shown in yellow, while non-overlapping ones in red and green. In practice, a $\delta < 10\%$ corresponds to a visually good alignment.

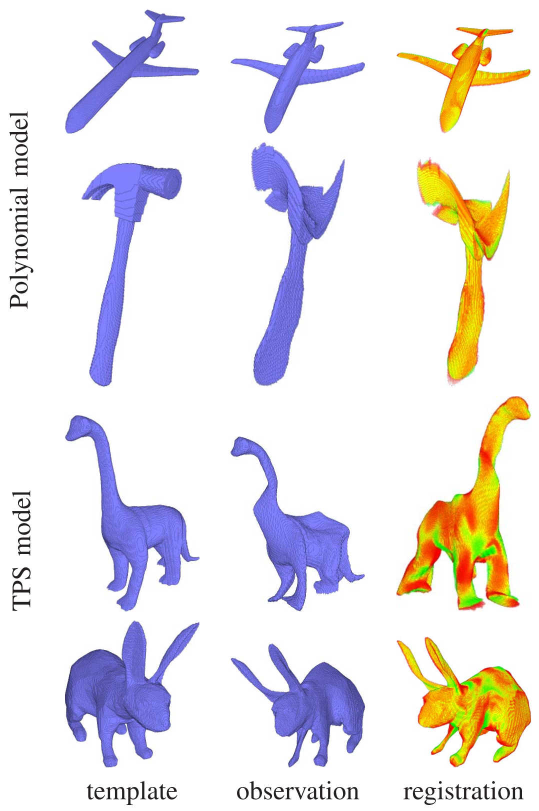

Fig. 1. Some results on the synthetic images. *Observation* and *registered* objects were overlayed, overlapping voxels are shown in yellow, while non-overlapping ones in red and green.

Model	Runtime (min)			$\delta(\%)$		
	m	μ	σ	m	μ	σ
Polynomial [11]	13.96	6.85	17.84	7.02	6.06	4.58
TPS	31.42	59.77	67.74	7.19	7.05	2.3

TABLE I
RESULTS ON THE SYNTHETIC IMAGES (M – MEDIAN, μ – MEAN AND σ – STANDARD DEVIATION).

For both models, random transformations were generated. In order to guarantee that the resulting transformation is diffeomorphic, the Jacobian of the generated transformation has been computed and it has been accepted only if it's Jacobian were positive everywhere over the object [5].

The registered objects, as well as synthetic observations, were generated by the following procedure (in Matlab): First, a smooth triangular mesh has been created from the object's surface using Matlab's internal `isosurface` function. Then the transformation was applied to the vertices yielding the transformed triangular mesh, from which the final voxelized object was obtained by the `binvox` program available from http://www.cs.princeton.edu/~min/binvox/ [13].

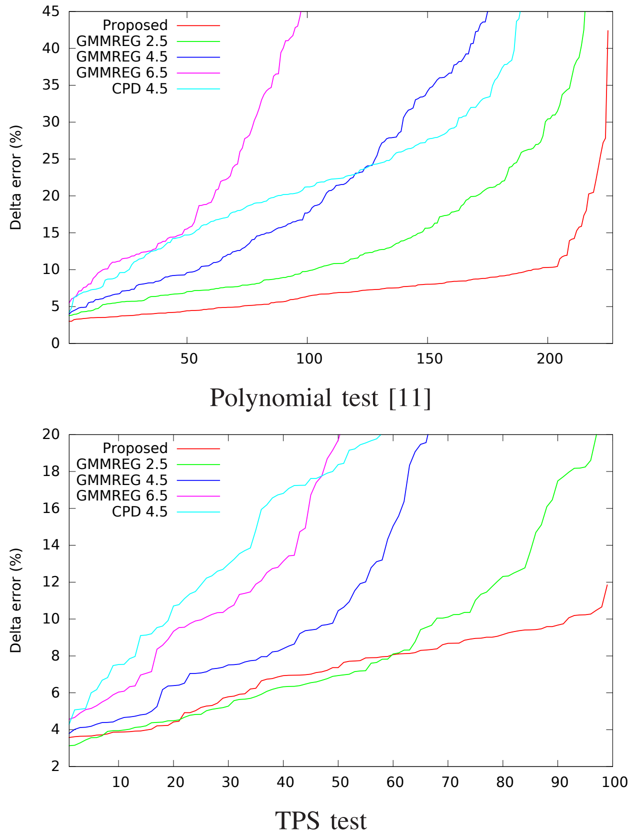

Fig. 2. Comparison with GMMREG [1] and CPD [9].

We have compared our results to the point-based registration frameworks in [1] (GMMREG) and [9] (CPD). We used the `C++` implementation of these methods available from http://code.google.com/p/gmmreg and set the parameters to their default values. At a first stage, we used all of the surface voxels as the input point set to the algorithm. For these sets consisting of about 0.5 megavoxel, the algorithms were running more than 12 hours without finding the transformation. Therefore the size of the point sets was reduced by using the vertices of an approximating triangular surface with various resolutions [14]. The mesh resolution was controlled by the maximal radius r of the corresponding Delaunay sphere. In Fig. 2, quantitative results for $r = \{2.5, 4.5, 6.5\}$ indicate that these methods provide inferior alignments against our approach with both transformation models.

In practice, segmentation never produces perfect shapes, therefore robustness against segmentation errors was also evaluated on simulated data: we randomly added or removed squares uniformly around the boundary of each slice of the observations (see sample slices in Fig. 3). The robustness test in [8] showed that the method does not tolerate occlusions well. The algorithm could not separate the occlusions from the real observation and due to the non-linearity of the transformation model, it will deform the initial shape to the occluded object.

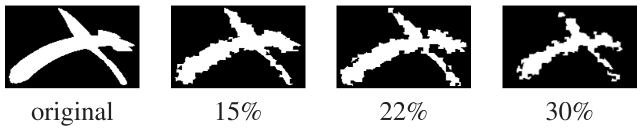

Fig. 3. Sample surface errors on a slice.

A. Polynomial deformations

In the polynomial experiments [11], we used a third order polynomial deformation model (*i.e.* $d = 3$), which has a total of $k = (d+3)(d+2)(d+1)/2 = 60$ parameters. A system of 64 equations were generated by the set $\omega_i(\mathbf{x}) = x_1^{n_i} x_2^{m_i} x_3^{o_i}$, where $\{(n_i, m_i, o_i)\}_{i=1}^{64} = \{(a, b, c) \mid 0 \leq a, b, c \leq 3\}$

The synthetic database for polynomial deformations consisted 500 deformed objects. The observations have been generated by applying second and third order polynomial deformations to different *template* objects. The transformation parameters were randomly picked from the following intervals: $a_{100}, b_{010}, c_{001} \in [0.5; 1.5]$, $a_{010}, a_{001}, b_{100}, b_{001}, c_{100}, c_{010} \in [-0.25; 0.25]$ and all other parameters are from $[-0.5; 0.5]$. Note that $a_{000} = b_{000} = c_{000} = 0$ (*i.e.* no translations), because initial normalization would remove any larger translations. Some of the results from [11] are presented in Fig. 1 and the statistics of our test are described in Table I.

For this dataset, the surface error of 15%, 22% and 30% of the original object volume were tested and Fig. 4 shows the quantitative evaluation of the alignment error δ on more than 115 objects. The proposed approach is quite robust up to as high as 22% surface error [11].

B. Thin plate spline

In the thin plate spline experiments we used a TPS model with 64 control points placed on a uniform grid, yielding a total of 204 parameters and a system of 216 equations were generated by the set $\omega_i(\mathbf{x}) = x_1^{m_i} x_2^{n_i} x_3^{o_i}$, where $(m_i, n_i, o_i)_{i=1}^{216} = \{(a, b, c) \mid 0 \leq a, b, c \leq 6\}$.

This database consisted 750 deformed objects of size $0.1 - 2.5$ megavoxels. Deformations were generated based on thin plate splines with 16, 32, and 64 control points placed on corresponding grids. Random translations with elements from $[-0.2, 0.2]$ were applied to the control points yielding a random free-form deformation of the objects. Some examples of our registration results on this dataset can be seen in Fig. 1 and the statistics are in Table I.

For the TPS experiments, the surface error of 10%, 20% and 30% of the original object volume were generated. The

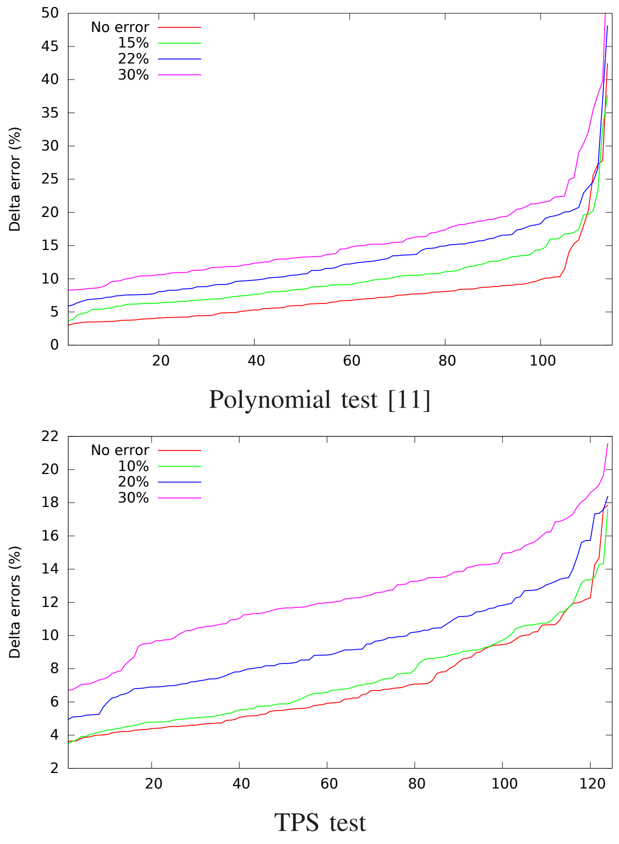

Fig. 4. Robustness test results for various degree of synthetically generated surface errors.

second plot in Fig. 4 shows the quantitative evaluation of the alignment error δ on 125 objects. The TPS model approach is quite robust up to as high as 20% surface error.

VI. Conclusion

We have proposed an extension of the 2D elastic registration method from [8], which works without established correspondences. As in [8], we set-up a system of non-linear equations whose solution directly provides the parameters of the aligning transformation. Herein, we considered a polynomial and a thin plate spline deformation model, but other diffeomorphism can also be used. The efficiency and robustness of the proposed approach have been evaluated on a large synthetic dataset. Our method compares favorably to two recent 3D matching algorithms [1], [9].

Acknowledgment

This research was partially supported by the grant CNK80370 of the National Innovation Office (NIH) & the Hungarian Scientific Research Fund (OTKA); the European Union and co-financed by the European Regional Development Fund within the project TÁMOP-4.2.1/B-09/1/KONV-2010-0005.

References

[1] B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011. [Online]. Available: http://gmmreg.googlecode.com

[2] C. Papazov and D. Burschka, "Deformable 3D shape registration based on local similarity transforms," *Computer Graphics Forum*, vol. 30, no. 5, pp. 1493–1502, Aug. 2011.

[3] R. Sagawa, K. Akasaka, Y. Yagi, H. Hamer, and L. Van Gool, "Elastic convolved ICP for the registration of deformable objects," in *Proceedings of International Conference on Computer Vision*, IEEE. Kyoto, Japan: IEEE, Oct. 2009, pp. 1558–1565.

[4] F. Michel, M. M. Bronstein, A. M. Bronstein, and N. Paragios, "Boosted metric learning for 3D multi-modal deformable registration," in *International Symposium on Biomedical Imaging: From Nano to Macro*, IEEE. Chicago, Illinois, USA: IEEE, Mar. 2011, pp. 1209–1214.

[5] M. Holden, "A review of geometric transformations for nonrigid body registration," *IEEE Trans. Med. Imaging*, vol. 27, no. 1, pp. 111–128, 2008.

[6] F. L. Bookstein, "Principal warps: Thin-Plate Splines and the Decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, 1989.

[7] L. Zagorchev and A. Goshtasby, "A comparative study of transformation functions for nonrigid image registration," *Image Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 529 –538, march 2006.

[8] C. Domokos, J. Nemeth, and Z. Kato, "Nonlinear Shape Registration without Correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 943–958, 2012.

[9] A. Myronenko, X. B. Song, and M. Á. Carreira-Perpiñán, "Non-rigid point set registration: Coherent point drift," in *Proceedings of the Annual Conference on Neural Information Processing Systems*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. Vancouver, British Columbia, Canada: MIT Press, Dec. 2006, pp. 1009–1016.

[10] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002. [Online]. Available: http://dx.doi.org/10.1109/34.993558

[11] Z. Sánta and Z. Kato, "Elastic Registration of 3D Deformable Objects," in *Proceedings of International Conference on Digital Image Computing: Techniques and Applications*. Fremantle, Western Australia: IEEE, dec 2012.

[12] M. Lourakis, "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++," [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, Jul. 2004, [Accessed on 14 Jun. 2012.].

[13] F. S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 2, pp. 191 – 205, Apr. 2003.

[14] Q. Fang and D. Boas, "Tetrahedral mesh generation from volumetric binary and gray-scale images," in *International Symposium on Biomedical Imaging: From Nano to Macro*, IEEE. Boston, Massachusetts, USA: IEEE, Jun. 2009, pp. 1142–1145.