IPSJ Transactions on Computer Vision and Applications Vol. 1 209–219 (Sep. 2009)

Research Paper

A Fast Registration Method Using IP and Its Application to Ultrasound Image Registration

Bo Zheng,[†1] Ryo Ishikawa,[†2] Takeshi Oishi,[†1]
Jun Takamatsu[†3] and Katsushi Ikeuchi[†1]

This paper presents a fast registration method based on solving an energy minimization problem derived by implicit polynomials (IPs). Once a target object is encoded by an IP, it will be driven fast towards a corresponding source object along the IP's gradient flow without using point-wise correspondences. This registration process is accelerated by a new IP transformation method. Instead of the time-consuming transformation to a large discrete data set, the new method can transform the polynomial coefficients to maintain the same Euclidean transformation. Its computational efficiency enables us to improve a new application for real-time Ultrasound (US) pose estimation. The reported experimental results demonstrate the capabilities of our method in overcoming the limitations of a noisy, unconstrained, and freehand US image, resulting in fast and robust registration.

1. Introduction

The task of registration aims to build a transformation relationship between a given (source) object and a target object. It is often an intermediate but crucial step inside the whole computer vision complexity.

Considerable efforts have been made in this area. One of the most popular approaches is the Iterative Closest Point (ICP) method [2], whose original form minimized the distances between the closest point pairs. Zhang [24] proposed using k-D trees to speed up the closest point search. Several groups [4)–7),12)–14),17)] improved the original version by replacing the Euclidean distance or finding more robust correspondences. However, the ICP-based methods inevitably require extra computation to find the correspondences that are sensitive to local correspondence.

†1 Institute of Industrial Science, The University of Tokyo
†2 Medical Imaging Project, Canon Inc.
†3 Nara Institute of Science and Technology

Avoiding the computation for searching for point-wise correspondences, the coarse (global) registration methods [18),22)] solve the problem by single (non-iterative) computation, using the central and oriented information of two objects being extracted from their polynomial coefficients. These methods are very computationally efficient but difficult to deal with in cases where the two objects are partially overlapped, so they are limited to some specific applications [16)].

Recently, consideration has been given to constructing a distance field for achieving the registration, such as the work in Refs. 9), 11). These methods spend much extra memory to preserve the distance field for the models and then register them by the evolution generated in the distance field. The methods also avoid correspondence searching and thus speed up the convergence, but they are liable to take too much memory space, especially for a dense model or a 3D object, and registration is limited in the regions where distance field has been generated.

In this paper, we first use an implicit polynomial (IP) to model the target object, and then we register this IP model to a discrete source object for quickly finding the transformation relationship. The advantages of our method over the prior methods are that: i) unlike the ICP-based methods, it avoids the extra computation for point-wise correspondences; ii) unlike the coarse registration methods, it totally supports partial-overlapping registration; iii) unlike the registration methods of preserving a discrete distance field, it needs very little memory space for preserving a few IP coefficients, and the algebraic model can generate an infinite distance field to support registration in a wider space.

The remainder of this paper is organized as follows: After introducing IP modeling technique in Section 2, we present our registration method using the IP model in Section 3. Then in Section 4, in order to further accelerate our algorithm to be suitable for large data set cases we propose an IP transformation method. Sections 5 and 6 present reports on experimental results and extend the proposed method to Ultrasound image application. Conclusion and future work follow in Section 7.

2. IP Modeling

First we concentrate on how to model the target object with an implicit poly-

nomial. In this section, we first introduce the mathematical formulation of IP, and then present a modeling technique.

2.1 Formulation of IP

IP is an implicit function defined in a multivariable polynomial form. For example, a 3D IP of degree n can be denoted by:

$$f_n(\mathbf{x}) = \sum_{0\le i,j,k;i+j+k\le n} a_{ijk}x^i y^j z^k = (\underbrace{1 \;\; x \;\; \ldots \;\; z^n}_{\mathbf{m}(\mathbf{x})^T})(\underbrace{a_{000} \;\; a_{100} \;\; \ldots \;\; a_{00n}}_{\mathbf{a}})^T, \quad (1)$$

where variable $\mathbf{x} = (x\ y\ z)$ denotes the Cartesian coordinate of a data point and an IP can be represented as an inner product between the monomial vector and the coefficient vector as $\mathbf{m}(\mathbf{x})^T\mathbf{a}$. We employ the inverse *lexicographical order* [22] for monomial indices $\{i, j, k\}$. The *homogeneous binary polynomial* of degree r in x, y, and z, $\sum_{i+j+k=r} a_{ijk}x^i y^j z^k$, is called the r-th degree form of the IP.

2.2 Distance Field Constraints

Although in the past two decades, IP was attractive to shape representation for object recognition purpose [3),10),19)–21),26)], in this paper we try to take advantage of IP to model the distance field in order to solve the registration problem. To do this, we define optimization constraints that let a polynomial be a given signed distance $\mathcal{D}$ at the data point $\mathbf{x}$ as:

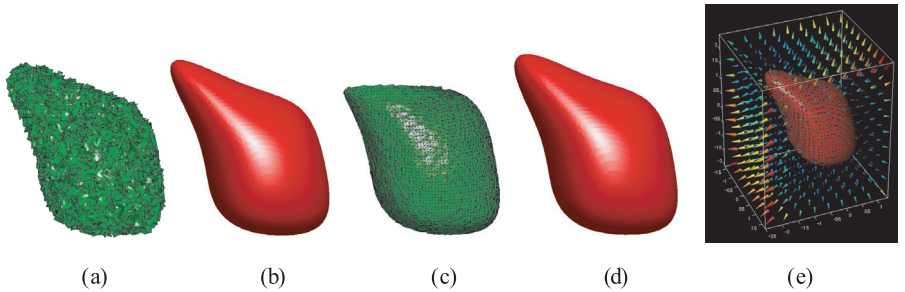
$$f_n(\mathbf{x}) = \mathcal{D}, \quad (2)$$

where $\mathcal{D}$ is defined as:

$$\mathcal{D}(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \mathcal{M} \\ +d(\mathbf{x}, \mathcal{M}), & \mathbf{x} \in \mathcal{M}_{inner} \\ -d(\mathbf{x}, \mathcal{M}), & \mathbf{x} \in \mathcal{M}_{outer} \end{cases}, \quad (3)$$

where we suppose $\mathcal{M}$ represents a closed surface of target object; $\mathcal{M}_{inner}$ and $\mathcal{M}_{outer}$ represent its finite inner and outer regions respectively; and $d(\mathbf{x}, \mathcal{M})$ is a the minimum Euclidean distance: $d(\mathbf{x}, \mathcal{M}) = \min_{\mathbf{x}_i \in \mathcal{M}} ||\mathbf{x} - \mathbf{x}_i||$.

For constructing the signed distance field (SDF), we employ the approximation method proposed in Ref. 15) that generates two level sets of SDF along the surface normals. The difference to their method is just that we generate SDF around the surface more densely and in a wider region.



(a) (b) (c) (d) (e)

Fig. 1 (a) noisy data set; (b) fitting result of 4-degree IP; (c) data set with missing data; (d) fitting result of 4-degree IP; (e) resulting gradient field of IP model of the same object.

2.3 Fitting

Using the optimization constraints (2), the IP coefficients can be solved by the linear least-squares method formulated as:

$$\mathbf{a} = (M^{\mathrm{T}}M)^{-1}M^{\mathrm{T}}\mathbf{d}, \quad (4)$$

where matrix M consists of the monomial vectors, i.e., the i-th row of M is $\mathbf{m}(\mathbf{x}_i)$ (see Eq. (1)), and $\mathbf{d}$ is the right-hand vector whose ith element is $\mathcal{D}(\mathbf{x}_i)$. For solving Eq. (4), we adopt our previous modeling method proposed in Ref. 26), which allows us to adaptively determine the moderate degree for IP different objects according to the shape complexity. **Figure 1** shows the modeling examples: for the data set shown in Figs. 1 (a) and (c), the resulting fits are shown in Figs. 1 (b) and (d); Figure 1 (e) shows the resulting gradient field generated by $f \cdot \frac{\nabla f}{\|\nabla f\|}$. We can see that two important characteristics are: i) IP is robust against noises and occlusions, and ii) IP can provide a stable gradient field around its zero-level set.

3. IP Registration

Having an IP model representing the target object, the next problem is how to register this IP to the source object. To achieve that, we first define the registration problem as an energy minimization problem, and then, by minimizing this energy function, the motion of the source object can be driven to the target model (IP model).

3.1 Energy Functional

To achieve the registration, an energy functional E which will be minimized to

find the proper transformation parameters defined as:

$$\mathbf{p} = \arg\min_{\mathbf{p}} E(\mathbf{p}). \tag{5}$$

where $\mathbf{p}$ is the transformation parameters ($\mathbf{p} \in \mathcal{R}^6$ for rigid transformation). In general, the energy functional E evaluates the registration by minimizing the distance between the data set and IP, defined as:

$$E = \sum_i dist(T(\mathbf{p}, \mathbf{x}_i), f_n), \forall \mathbf{x}_i \in \Omega, \tag{6}$$

where $T(\mathbf{p}, \mathbf{x}_i)$ is the function $\mathcal{R}^3 \to \mathcal{R}^3$, which returns the transformed point of $\mathbf{x}_i$ by the rigid-transform operation respected to parameter $\mathbf{p}$; $dist(\mathbf{x}, f_n)$ means a certain distance from the data point $\mathbf{x}$ to the zero set of f_n; and Ω represents the 3D region of the source model.

There are multiple choices for the distance $dist(\mathbf{x}, f_n)$ in Eq. (6), and the common one is to use $\mathcal{L}_2$ norm. Thus it becomes possible to form least-squares regression with $dist(\mathbf{x}, f_n) = f_n(\mathbf{x})^2$. However, in this work, we choose more meaningful approximation for the distance representation of IP (see Ref. 21)):

$$dist(\mathbf{x}, f_n) = \frac{f_n(\mathbf{x})^2}{\parallel \bigtriangledown f_n(\mathbf{x}) \parallel^2}, \forall \mathbf{x} \in \Omega. \tag{7}$$

3.2 Minimizing Energy Functional

To minimize the energy functional (6), the method employs the following two steps. First, it minimizes the function without any constraint in the transformation. This means all points can move freely towards IP along their gradients during the first minimization. Next, it determines the transformation parameters to maintain the Euclidean transformation. These two steps are repeated alternately until the convergence. The efficiency of this minimization benefits from its success in avoiding time-consuming computation for finding point-wise correspondence.

3.2.1 First Step: Free Deformation

At the first step, by calculus of variations (see Ref. 8)), the Gateaux derivative (first variation) of the functional E in Eq. (6) to point $\mathbf{x}$ can be approximately formulated as:

$$\frac{\partial E}{\partial \mathbf{x}} = \frac{\partial dist(\mathbf{x}, f)}{\partial \mathbf{x}} \approx 2f(\mathbf{x}) \frac{\bigtriangledown f_n(\mathbf{x})}{\parallel \bigtriangledown f_n(\mathbf{x}) \parallel^2} = 2g(\mathbf{x}), \tag{8}$$

if we assume $\parallel \bigtriangledown f_n(\mathbf{x}) \parallel$ is a constant and let the gradient vector $g(\mathbf{x}) = f(\mathbf{x}) \frac{\bigtriangledown f_n(\mathbf{x})}{\parallel \bigtriangledown f_n(\mathbf{x}) \parallel^2}$.

Therefore, we need to minimize this functional to satisfy the Euler-Lagrange equation $\frac{\partial E}{\partial \mathbf{x}} = 0$. Thus the steepest descent process is executed in the following gradient flow for each point $\mathbf{x}$:

$$\frac{\partial \mathbf{x}}{\partial t} \approx -2g(\mathbf{x}), \tag{9}$$

where t denotes the time step. From the view of implementation, each point can be updated as:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - 2g(\mathbf{x}^k), \tag{10}$$

where $\mathbf{x}^k$ denotes the point at k-th time step.

Similarly, if we let $X \in \mathcal{R}^{N \times 3}$ represent the integral data set consisting of the point coordinates as rows, and $G \in \mathcal{R}^{N \times 3}$ be the integral gradients of X, then X^k at the k-th time step can be updated as the following recurrence:

$$X^{k+1} = X^k - 2G(X). \tag{11}$$

This shows the fact that, under this operation, all points in the discrete data set of source object will be moved towards the zero set of polynomials along their gradients.

3.2.2 Second Step: Rigid Deformation

In the first step, although every point in the source model can move close to the IP surface, the transformation dose not maintain the rigid transformation that is necessary for rigid parameter estimation. Therefore, in the second step, we alter the outcome of the first step so that the result is rigid transformation. For achieving this, since from the Eq. (11) we can obtain data sets represented by two matrices X^k and X^{k+1} in two continuous steps, the covariant matrix of these two matrices can be calculated as:

$$A = (X^k - \bar{X}^k)^{\mathrm{T}} (X^{k+1} - \bar{X}^{k+1}), \tag{12}$$

where $\bar{X}$ is a matrix in which each row consists of the mean value (mass center point) of X. If A ($\in \mathcal{R}^{3 \times 3}$) is SVD-decomposed as $A = USV^{\mathrm{T}}$, where S is the diagonal matrix and U and V are the unitary matrices, then rigid transformation

is obtained by:

$$R = UV^{\mathrm{T}},\ \mathbf{t} = \bar{X}^k - \bar{X}^{k+1}R^{\mathrm{T}}, \tag{13}$$

where R and $\mathbf{t}$ are rotation and translation parameters respectively (see Ref. 23)). Thus, with R and $\mathbf{t}$, we can update the transformation in the first step shown in Eq. (11) to be rigid transformation as in:

$$X^{k+1} = X^k R^{\mathrm{T}} + \mathbf{t}. \tag{14}$$

Therefore, the above two steps can be alternately repeated until the discrete data set of the source model is moved near the IP model with desired accuracy.

4. Acceleration Using IP Transformation

In this section, we present an acceleration method that makes registration more efficient when the number of source data points is much larger than the number of IP coefficients of the target model. Since in this case moving the large-scale data sets to the IP model is obviously time-consuming, instead we inversely move the IP model towards the discrete model.

To drive this motion of IP requires a transformation method through which the IP's coefficients can be transformed to maintain an inverse transformation to what was done in Eq. (14). Therefore the acceleration method can be described as a process where, instead of updating the discrete data set in Eq. (14), we update the coefficients of IP at each step:

$$\mathbf{a}^{k+1} = V(R, \mathbf{t})\mathbf{a}^k, \tag{15}$$

where we suppose $\mathbf{a}^{k+1}$ and $\mathbf{a}^k$ are IP coefficients at the $(k+1)$ and k-th time step, and V is a square transformation matrix corresponding to the Euclidean transformation of rotation R and translation $\mathbf{t}$. Thus the remained problem is how to calculate the transformation matrix V in Eq. (15).

4.1 IP Transformation Method

To our knowledge, although prior literatures such as Ref. 22) have introduced the existence of IP transformation matrix V from the aspect for extracting algebraic invariants of IP, unfortunately computational implementation of IP transformation was not explicitly described. Tarel, et al.[18] showed a tensor-based transformation that represents 3D IP in 4D homogeneous form, and then transformed the tensor elements. The method is good for IP invariants extraction but still suffers from heavy computational cost for tensor transformation, since for transforming an r-degree 3D IP, it need to transform 4^d tensor elements.

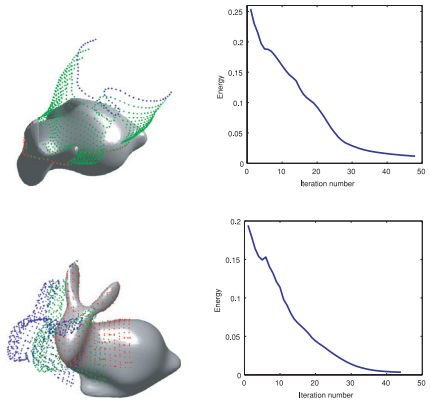
We describe our implementation based on symbolic computation in Appendix A.1. This solution can be seen as a kind of incremental calculation that increases, beginning from the transformation for first-degree terms, until it obtains a transformation for n-th degree terms.

5. Numerical Experiments

In this section, we report results of experiments dealing with some synthetic data sets to evaluate the method on computational performance. All the experiments were implemented in Matlab 7 with a PC having an Intel core 2 CPU, 2.4 GHz, and 2 GB memory.

5.1 Registration for Synthesized Data Set

Figure 2 shows the registrations in a simple case that registers two synthetic data sets to target object "bunny" modeled by an IP of degree 8. Also their



(a) (b)

Fig. 2 IP registration with 2D and 3D feature points: (a) registration process between "bunny" IP model and points (blue points: initial position; green points: selected mid-step positions; red points: final position). (b) the mean absolute distance $\frac{1}{N}\sum|dist(\mathbf{x}, f)|$ versus iteration number.

energy convergence behaviors is shown in Fig. 2 (b) respectively. From this result, we can see that our method can totally support the partially overlapped objects.

5.2 Comparison between ICP Method and Our Method

We solved a registration task shown in which we initially rotated and translated the target model to the position shown as blue points in **Fig. 3**. Then we tested three methods, standard ICP method [2], ICP method with KD tree [24] and our method, to let the process iterate until it satisfed the same accuracy for registration.

Since they are set with the same stopping criterion (same threshold for registration accuracy), we only compared the computational time and memory in **Tables 1**, **2** and **3**, which show the registration tasks using 1k points, 2.5k points and 10k points respectively for both target and source data sets.

It is worth noting that, in the factor of accuracy, IP makes it difficult to achieve better registration accuracy, because IP often models the data set "coarsely" by the global algebraic description. But, on the contrary, the IP model has few coefficients and little robustness against noises, which make it faster and more stable when facing noise-condition applications, such as the ultrasound image application described in the following section.

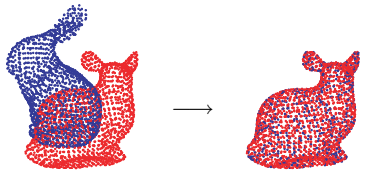


Fig. 3 Illustration for registration: blue points for target object and red points for source object.

Table 1 Registration using 1k points.

	ICP	ICP with KD tree	IP
CPU Time (sec)	11.95	0.91	0.43
Memory	2K points	2K points + KD tree	1K points + 165 coef.

Table 2 Registration using 2.5k points.

	ICP	ICP with KD tree	IP
CPU Time (sec)	30.54	6.56	0.51
Memory	5K points	5K points + KD tree	2.5K points + 165 coef.

Table 3 Registration using 10k points.

	ICP	ICP with KD tree	IP
CPU Time (sec)	123.01	35.12	1.07
Memory	20K points	20K points + KD tree	10K points + 165 coef.

6. Application to US Image Registration

Ultrasound (US) imaging is widely used for assistance with surgical operations and clinical diagnosis. The relative pose estimation for a US image to another modality, such as MRI or CT, is desired and helpful for diagnostic guidance. But such estimation suffers from difficulties due to poor image quality with speckle noises, low signal-to-noise ratio, uniform brightness, and the fact that only cross-section images can be obtained. Fortunately, considering the characteristics of our method as described above, we can confront this registration problem by making use of IP.

6.1 Registration for US Images

Registration for US images can be simply viewed as 2D-3D registration, if the source and target models, 2D feature points, and a 3D organ model represented by an IP are given.

Three-dimensional organ models are often obtained from a medical modality, such as CT, MRI or PET, by 3D image segmentation. For example, segmentation using the Graph Cut or Level set method can extract 3D surface models from volumetric data. Then the 3D surface models can be modeled by IPs with the modeling method described in Section 2, and the modeling process can be achieved offline. On the other hand, 2D feature points extraction must be done online. The 2D segmentation method for a single image, such as the Level set method, is not suitable to a real-time requirement.

In this paper, we employed our previous method [25] to register an IP model to a single US image. The process of this method can be simply viewed as: i) using an edge detector, it extracts two kinds of feature points around the IP model's surface from the US image plane: boundary-like feature points and region-like feature points, and ii) by our registration method described in Sections 3 and 4, it drives the motion of boundary-like feature points to the IP model's surface

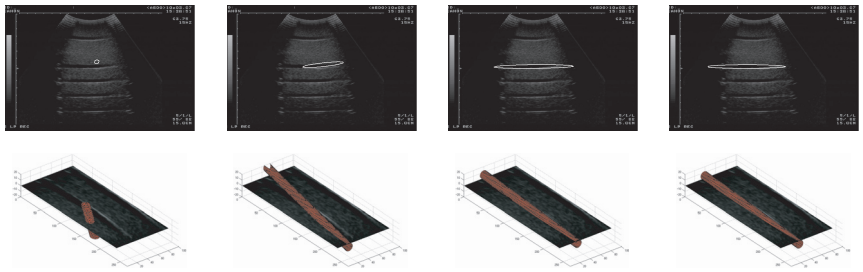


Fig. 4 Pose estimation for a single US image.

(zero level set) and also drives the motion of region-like feature points into the IP model's inner region (negative level set), but the registration process is the same.

6.2 Experimental Results

6.2.1 Pose Estimation for Phantom ATS514

Figure 4 shows pose estimation process for a US image, where the US images are obtained by measuring phantom ATS514 [1)] consisting of the cylinder-like shapes inside in different materials. Then we use a cylinder-like IP model (2-degree IP shown in red in Fig. 4) to register the US images. The initial position is set as: the cylinder-like IP model crosses the US image plane with an angle of 85°. For this image, 8 iterations are required, and in the process the motion of the IP is driven to fit to the hole-like region in the image. We show this process in Fig. 4 with cross-section contour and relative pose between the IP model and the US plane at each iteration.

Figure 5 shows three US image frames with the same initialization and in the second row of Fig. 5 the iteration number required and final relative pose are shown. The third row shows the cross-section contour between the IP and the US plane at the final iteration. For each case, the consumed CPU time is within 30 ms.

6.2.2 Pose Estimation for Duck Toy

Figure 7 shows another result of the US image pose estimation where the images are obtained by scanning a duck toy made of rubber in the cistern shown in

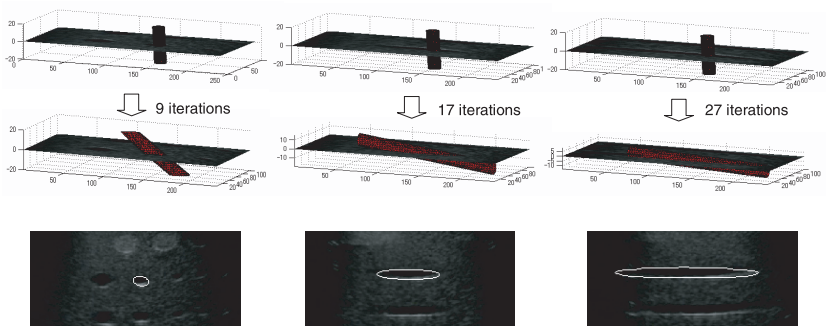


Fig. 5 Pose estimation for phantom ATS514. First row: initial position; Second row: final position; Third row: cross-section contour of IP and US image (white points).

Fig. 6 Left: photograph of measuring a duck toy. Right: IP model of duck object.

Fig. 6 left. We modeled the duck object with an 8-degree IP shown in Fig. 6 right. **Figure 7** shows the relative position and the cross-section contour at each iteration resulting from the registration. In this case the consumed CPU time is about 180 ms.

6.2.3 Pose Estimation and Tracking for CT Data

The third example is to test by solving a registration problem between real CT data and a US image. To do this, we first segmented the CT data to obtain the desired organ object by modern segmentation methods such as Graph Cut (see **Fig. 8**). Then we model the organ object by a 4-degree IP as shown in Fig. 8 right. Finding the relative pose to CT data for a US image now becomes finding the relative pose to the IP model.

The first example is pose estimation for a single US image shown in **Fig. 9** where we show the registration process with relative pose and cross-section contour at

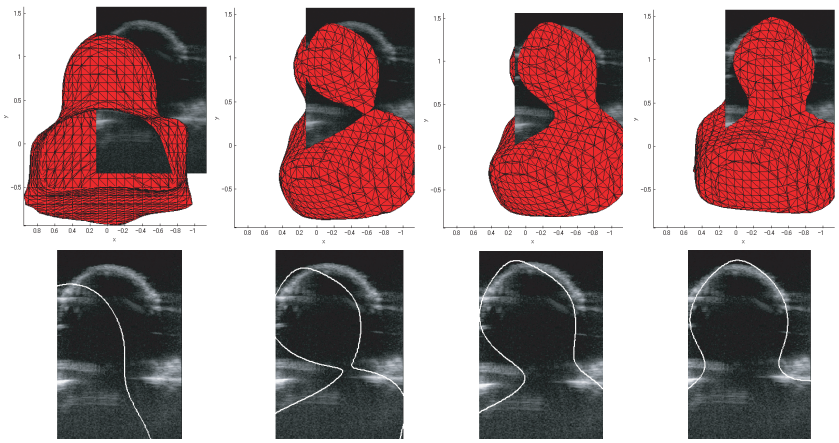


Fig. 7 Pose estimation for duck toy. First row: relative poses of IP and US images at each iteration. Second row: cross-section contour at each iteration.

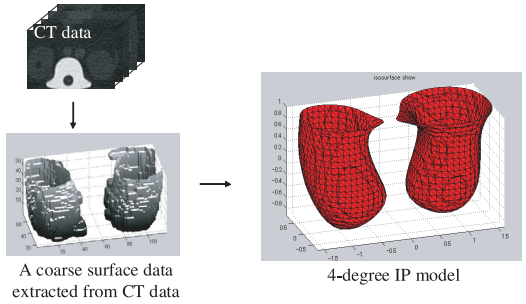


Fig. 8 Segmenting kidney data from the CT scans and modeling with a 4-degree IP.

each selected iteration.

We also tried a tracking problem for a US image sequence by tracking the position of the kidney object with the above IP model. Results are shown in **Fig. 10** where we show the cross-section contours of IP for selected frames.

7. Conclusion and Future Work

A fast registration method using the IP gradient field has been proposed. It is

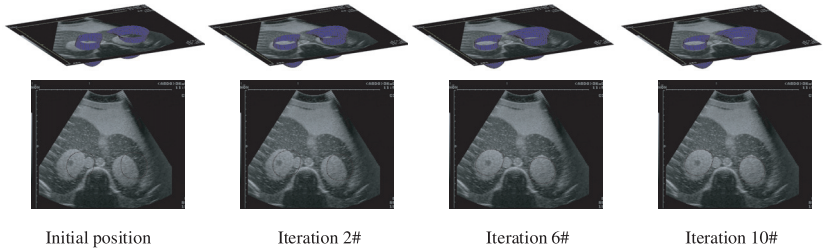


Fig. 9 US pose estimation for CT data with 4-degree IP.

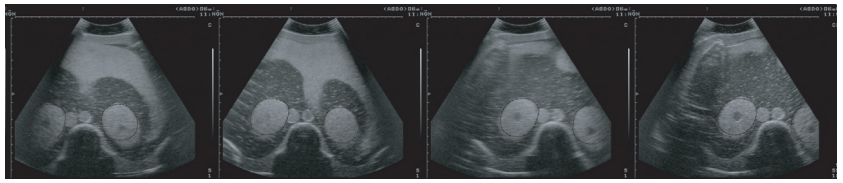


Fig. 10 Tracking for US image sequence. Cross-section contours of frames are shown in black points.

computationally efficient due to the adoption of a two-step minimization process without point matching, and acceleration supported by IP transformation. We demonstrated its high performance by applying it into real-time US image registration, which shows robustness against the presence of weak boundaries and strong noise, and is possibility suitable for real-time implementation. In future work, we are considering modeling multiple organ objects using IPs to enhance the robustness of registration.

Acknowledgments This work was supported by Canon Inc., under the project: Physics-based vision theories for next-generation medical image processing.

References

1) http://www.atslabs.com
2) Besl, P. and McKay, N.: A Method for Registration of 3-D Shapes, *IEEE Trans. Patt. Anal. Mach. Intell.* (*TPAMI*), Vol.14, No.2, pp.239–256 (1992).
3) Blane, M., Lei, Z.B. and Cooper, D.B.: The 3L Algorithm for Fitting Implicit Polynomial Curves and Surfaces to Data, *IEEE Trans. Patt. Anal. Mach. Intell.*

(*TPAMI*), Vol.22, No.3, pp.298–313 (2000).
4) Boughorbel, F., Koschan, A., Abidi, B. and Abidi, M.: Gaussian Energy Functions for Registration without Correspondences, *17th International Conference on Pattern Recognition* (*ICPR*) (2004).
5) Champleboux, G., Lavallee, S., Szeliski, R. and Brunie, L.: From Accurate Range Imaging Sensor Calibration to Accurate Model-Based 3D Object Localization, *Proc. IEEE Conf. Computer Vision and Patt. Rec.* (*CVPR*), pp.83–89 (1992).
6) Chen, Y. and Medioni, G.: Surface Description of Complex Objects from Multiple Range Images, *Proc. IEEE Conf. Computer Vision and Patt. Rec.* (*CVPR*), pp.153–158 (1994).
7) Chetverikov, D., Svirko, D., Stepanov, D. and Krsek, P.: The trimmed iterative closest point algorithm, *International Conference on Pattern Recognition* (*ICPR*) (2002).
8) Evans, L.: *Partial Differential Equations*, Providence: American Mathematical Society (1998).
9) Iwashita, Y., Kurazume, R., Konishi, K., Nakamoto, M., Aburaya, N., Sato, Y., Hashizume, M. and Hasegawa, T.: Fast Model-Image Registration using 2D Distance Map for Surgical Navigation System, *Advanced Robotics*, Vol.21, No.7, pp.751–770 (2007).
10) Keren, D. and Cooper, D.: Describing Complicated Objects by Implicit Polynomials, *IEEE Trans. Patt. Anal. Mach. Intell.* (*TPAMI*), Vol.16, No.1, pp.38–53 (1994).
11) Munim, H.E. and Farag, A.: Shape Representation and Registration using Vector Distance Functions, *Proc. IEEE Conf. Computer Vision and Patt. Rec.* (*CVPR*), pp.Y1–Y8 (2007).
12) Nguyen, V.-D., Nzomigni, V. and Stewart, C.V.: Fast and Robust Registration of 3D Surfaces Using Low Curvature Patches, *2nd International Conference on 3-D Imaging and Modeling* (*3DIM '99*), p.201 (1999).
13) Oishi, T., Nakazawa, A., Kurazume, R. and Ikeuchi, K.: Fast Simultaneous Alignment of Multiple Range Images using Index Images, *Proc. 5th International Conference on 3-D Digital Imaging and Modeling* (*3DIM 2005*), pp.476–483 (2005).
14) Rusinkiewicz, S. and Levoy, M.: Efficient Variants of the ICP Algorithm, *International Conference on 3D Digital Imaging and Modeling* (*3DIM 2001*) (2001).
15) Sahin, T. and Unel, M.: Fitting Globally Stabilized Algebraic Surfaces to Range Data, *Proc. IEEE Conf. Int. Conf. on Computer Vision* (*ICCV*), Vol.2, pp.1083–1088 (2005).
16) Salvi, J., Matabosch, C., Fofi, D. and Forest, J.: A review of recent range image registration methods with accuracy evaluation, *Image and Vision Computing*, Vol.25, No.5, pp.578–596 (2007).
17) Stewart, C.: Covariance-based registration, Technical Report RPI-CS-TR 02-8, Department of Computer Science Rensselaer Polytechnic Institute (2002).
18) Tarel, J.-P., Civi, H. and Cooper, D.B.: Pose Estimation of Free-Form 3D Objects without Point Matching using Algebraic Surface Models, *Proc. IEEE Workshop Model Based 3D Image Analysis*, pp.13–21 (1998).
19) Tarel, J.-P. and Cooper, D.B.: The Complex Representation of Algebraic Curves and Its Simple Exploitation for Pose Estimation and Invariant Recognition, *IEEE Trans. Patt. Anal. Mach. Intell.* (*TPAMI*), Vol.22, No.7, pp.663–674 (2000).
20) Tasdizen, T., Tarel, J.-P. and Cooper, D.B.: Improving the Stability of Algebraic Curves for Applications, *IEEE Trans. Imag. Proc.* (*TIP*), Vol.9, No.3, pp.405–416 (2000).
21) Taubin, G.: Estimation of Planar Curves, Surfaces and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation, *IEEE Trans. Patt. Anal. Mach. Intell.* (*TPAMI*), Vol.13, No.11, pp.1115–1138 (1991).
22) Taubin, G. and Cooper, D.: *Symbolic and Numerical Computation for Artificial Intelligence*, *chapter 6*, Computational Mathematics and Applications. Academic Press (1992).
23) Umeyama, S.: Least-Squares Estimation of Transformation Parameters Between Two Point Patterns, *IEEE Trans. Patt. Anal. Mach. Intell.* (*TPAMI*), Vol.13, No.4, pp.376–380 (1991).
24) Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces, *Int. J. Computer Vision* (*IJCV*), Vol.13, No.2, pp.119–152 (1992).
25) Zheng, B., Ishikawa, R., Oishi, T., Takamatsu, J. and Ikeuchi, K.: 6-DOF Pose Estimation form Single Ultrasound Image using 3D IP Models, *Proc. IEEE Comput. Vision Patt. Recog. Workshop 2008*, pp.1–8 (2008).
26) Zheng, B., Takamatsu, J. and Ikeuchi, K.: Adaptively Determining Degrees of Implicit Polynomial Curves and Surfaces, *Proc. Conf. Asian Conf. on Computer Vision* (*ACCV*), Vol.LNCS 4844, pp.289–300 (2007).

Appendix

A.1 Implementation for IP Transformation

A.1.1 Pure Rotation Transformation

First, let us explain some notations for polynomial operations which were first introduced by Taubin and Cooper (see Ref. 22)). Let a coefficient a_{ijk} of an IP in Eq. (1) be presented as $\frac{\Phi_{ijk}}{i!j!k!}$, and a vector $\Phi_{[l]} = \left(\frac{\Phi_{l00}}{\sqrt{l!0!0!}} \ \frac{\Phi_{l-1,1,0}}{\sqrt{(l-1)!1!0!}} \ \cdots \ \frac{\Phi_{00l}}{\sqrt{0!0!l!}}\right)^{\mathrm{T}}$ be according to the l-th form of IP. An operation on this vector is defined as: $\Phi_{[l,m]} = \Phi_{[l]} \star \Phi_{[m]}^{\mathrm{T}}$, where $\star$ represents the classic matrix multiplication, with the difference that the individual elements

$\frac{\Phi_{ijk}}{\sqrt{i!j!k!}}$ and $\frac{\Phi_{abc}}{\sqrt{a!b!c!}}$ in each vector are multiplied as $\frac{\Phi_{i+a,j+b,k+c}}{\sqrt{i!j!k!}\sqrt{a!b!c!}}$. For example,

$$\Phi_{[1]} = \left(\frac{\Phi_{100}}{\sqrt{1!0!0!}} \ \frac{\Phi_{010}}{\sqrt{0!1!0!}} \ \frac{\Phi_{001}}{\sqrt{0!0!1!}} \right)^{\mathrm{T}}, \tag{16}$$

and

$$\Phi_{[1,1]} = \Phi_{[1]} \star \Phi_{[1]}^{\mathrm{T}} = \begin{pmatrix} \Phi_{200} & \Phi_{110} & \Phi_{101} \\ \Phi_{110} & \Phi_{020} & \Phi_{011} \\ \Phi_{101} & \Phi_{011} & \Phi_{002} \end{pmatrix}. \tag{17}$$

In Ref. 22), it was pointed out that under a non-singular coordinate transformation A, e.g., $\Phi'_{[l]} = A_{[l]}\Phi_{[l]}$, the transformed coefficient matrix is given by

$$\Phi'_{[l,m]} = A_{[l]}^{-\mathrm{T}} \Phi_{[l,m]} A_{[m]}^{-1}, \tag{18}$$

where $A_{[l]}$ is a non-singular $h_l \times h_l$ transformation matrix (assuming h_l be the number of monomial in the l-th form), and $A_{[m]}$ is the same. From this equation, we know that if $A_{[l]}$ and $A_{[m]}$ are given, then a linear relationship between $\Phi'_{[l,m]}$ and $\Phi_{[l,m]}$ can be found; that is, the element-wise correspondence can be linearly expressed as

$$\Phi'_{ijk} = \sum_{\beta;i,j,k\le l+m} a_{\alpha\beta}\Phi_{ijk} = \mathbf{a}_\alpha^{\mathrm{T}} \Phi_{[l+m]}. \tag{19}$$

Then since all of the elements in vector $\Phi_{[l+m]}$ are contained in matrix $\Phi_{[l,m]}$, and similarly all of the elements in $\Phi'_{[l+m]}$ are contained in $\Phi'_{[l,m]}$, a new linear correspondence can be constructed between $\Phi_{[l+m]}$ and $\Phi'_{[l+m]}$ by arranging the necessary elements in the right order into $\Phi'_{[l+m]}$ as

$$\begin{aligned} \Phi'_{[l+m]} &= (\Phi'_{l+m,0,0}, \Phi'_{l+m-1,1,0}, \Phi'_{l+m-2,2,0}, \ldots)^{\mathrm{T}} \\ &= (\mathbf{a}_1^{\mathrm{T}}\Phi_{[l+m]}, \mathbf{a}_2^{\mathrm{T}}\Phi_{[l+m]}, \mathbf{a}_3^{\mathrm{T}}\Phi_{[l+m]}, \ldots)^{\mathrm{T}} \\ &= A_{[l+m]}\Phi_{[l+m]}, \end{aligned} \tag{20}$$

if the $(l+m)$-th transformation matrix is constructed as: $A_{[l+m]} = [\mathbf{a}_1^{\mathrm{T}}, \mathbf{a}_2^{\mathrm{T}}, \mathbf{a}_3^{\mathrm{T}}, \ldots]^{\mathrm{T}}$.

As a simple example, the elements of $\Phi_{[2]}\left(= \left(\frac{\Phi_{200}}{2} \ \Phi_{110} \ \Phi_{101} \ \frac{\Phi_{020}}{2} \ \Phi_{011} \ \frac{\Phi_{002}}{2}\right)^{\mathrm{T}}\right)$ are contained in $\Phi_{[1,1]}$ shown in Eq. (17), and the elements of $\Phi'_{[2]}$ are contained in $\Phi'_{[1,1]}$. From the equation $\Phi'_{[1,1]} = R_{[1]}\Phi_{[1,1]}R_{[1]}^{\mathrm{T}}$ (supposing $R_{[1]}$ is a pure rotation matrix), it is easy to find out the relationship of a linear expansion for the elements, e.g., the first element of $\Phi'_{[1,1]}$ is expanded as: $\Phi'_{200} = \mathbf{r}_1\Phi_{[1,1]}\mathbf{r}_1^{\mathrm{T}} = r_{11}^2\Phi_{200} + 2r_{11}r_{12}\Phi_{110} + 2r_{11}r_{13}\Phi_{101} + r_{12}^2\Phi_{020} + 2r_{12}r_{13}\Phi_{011} + r_{13}^2\Phi_{002}$, where $\mathbf{r}_1$ is the first row of $R_{[1]}$ and r_{ij} is an element of $R_{[1]}$. Then since all the elements in $\Phi_{[1,1]}$ are contained in $\Phi_{[2]}$, we can find a linear relationship $\frac{1}{2}\Phi'_{200} = \mathbf{a}_1^{\mathrm{T}}\Phi_{[2]}$, where $\mathbf{a}_1 = (r_{11}^2 \ r_{11}r_{12} \ r_{11}r_{13} \ r_{12}^2 \ r_{12}r_{13} \ r_{13}^2)^{\mathrm{T}}$ is viewed as the first row of $A_{[2]}$.

Therefore, transformation matrices for different forms can be calculated in an incremental manner as

Algorithm	
1)	Initialization: given $A_{[1]}$;
2)	$\Phi'_{[1,1]} = A_{[1]}^{-\mathrm{T}}\Phi_{[1,1]}A_{[1]}^{-1} \longrightarrow A_{[2]}$;
3)	$\Phi'_{[1,2]} = A_{[1]}^{-\mathrm{T}}\Phi_{[1,2]}A_{[2]}^{-1} \longrightarrow A_{[3]}$;
	$\vdots$
n)	$\Phi'_{[1,n-1]} = A_{[1]}^{-\mathrm{T}}\Phi_{[1,n-1]}A_{[n-1]}^{-1} \longrightarrow A_{[n]}$.

Here, $\longrightarrow$ represents constructing the transformation matrix for the next step, after finding out the linear mapping relationship between $\Phi'_{[1,l]}$ and $\Phi_{[1,l]}$. Note if the initial argument $A_{[1]}$ represents the pure rotation, then $A_{[l]}$ is an orthogonal matrix (see Ref. 22)), and thus Eq. (18) degenerates to

$$\Phi'_{[l,m]} = A_{[l]}\Phi_{[l,m]}A_{[m]}^{\mathrm{T}}.$$

A.1.2 Affine Transformation

By using homogeneous coordinates, an IP of n variables in Euclidean space can be described in projective space by a corresponding homogeneous IP of $n+1$ variables. To convert a ternary (i.e., 3D) IP of degree d

$$f_{3D}^d(x,y,z) = \sum_{0\le i,j,k;i+j+k\le d} a_{ijk}x^i y^j z^k \tag{21}$$

into its homogeneous representation, a new component $t = 1$ is added to the 3D IP as

$$f_{4D}^d(x,y,z,t) = \sum_{0\le i,j,k,l;i+j+k+l=d} a_{ijk}x^i y^j z^k t^l. \tag{22}$$

Therefore a homogeneous polynomial corresponding to a 3D IP of degree d is a form of degree d in a 4D IP. Thus the procedure mentioned in the last section can be used to transform the homogeneous IP (4D IP). The different is that the incremental algorithm starts from a 4×4 affine transformation matrix $A_{[1]}$, e.g., in the Euclidean case

$$A_{[1]} = \begin{pmatrix} R_{3\times3} & \mathbf{t} \\ \mathbf{0}_{1\times3} & 1 \end{pmatrix},$$

where $R_{3\times3}$ and $\mathbf{t}$ are a pure 3D rotation matrix and a translation vector. As a result, a 3D IP of degree d can be affine-transformed only by $A_{[d]}$, once it is worked out. Furthermore, the incremental scheme can be modified for saving the computational cost, e.g., for calculating $A_{[9]}$ we can select the incremental order as:

$A_{[1]} \to A_{[2]} \to A_{[4]} \to A_{[8]} \to A_{[9]}$.

(Received October 5, 2008)
(Accepted February 24, 2009)
(Released September 24, 2009)

(Communicated by *Ryo Kurazume*)

Bo Zheng received his M.E. and Ph.D. in the Applied Physics School of Engineering in 2005 and Information Science and Technology in 2008 respectively, from the University of Tokyo. He worked for Fujitsu Co., Ltd. during the Summer of 2005. He is currently a project research associate in the Institute of Industrial Science, the University of Tokyo. His research interests include algebraic representation, object recognition, and solving large-scale and sparse linear systems.

Ryo Ishikawa received his M.E. and D.E. degrees in system engineering from University of Electro-Communications, Japan, in 2002 and 2005 respectively. He has been working in Canon Inc. since 2005 and now is a researcher of Medical Imaging Project of Canon. He is a member of IEEE, JSST and SOFT.

Takeshi Oishi is a Project Lecturer at the Interfaculty Initiative in Information Studies, the University of Tokyo. He received the B.Eng. degree in Electrical Engineering from Keio University in 1999, and the Ph.D. degree in Interdisciplinary Information Studies from the University of Tokyo in 2005. His research interests are in modeling from reality, digital archive and augmented reality.

Jun Takamatsu received the Ph.D. degree in Computer Science from the University of Tokyo, Japan, in 2003. After working in the Institute of Industrial Science, the University of Tokyo, he joined Nara Institute of Science and Technology (NAIST), Japan in 2008. At present, he is an associate professor in Graduate School of Information Science, NAIST. His research interests are in 3D shape modeling and analysis, physics-based vision, and robotics including learning-from-observation and task planning using feasible motion analysis.

Katsushi Ikeuchi received the Ph.D. degree in Information Engineering from the University of Tokyo in 1978. After working at the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Electrotechnical Laboratory of Japanese Ministry of International Trade and Industry, and the School of Computer Science, Carnegie Mellon University, he joined the university in 1996. At present, he is a Professor at the Interfaculty Initiative in Information Studies, the University of Tokyo. His research interests include modeling-from-reality, object recognition, and learning-from-observation. He has received several awards, including the David Marr Prize in computational vision, and IEEE R&A K-S Fu memorial best transaction paper award. In addition, in 1992, his paper, "Numerical Shape from Shading and Occluding Boundaries" was selected as one of the most influential papers to have appeared in Artificial Intelligence Journal within the past ten years. His IEEE activities include General Chair, IROS95, ITSC00, IV01; Program Chair, CVPR96, ICCV03; Associate Editor, IEEE TRA, IEEE TPAMI; distinguished lecture SPS (2000–2002), RAS (2004–2006). Dr. Ikeuchi was selected to IEEE Fellow in 1998. He is the EIC of International Journal of Computer Vision.