Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition Letters 24 (2003) 2967–2976

Pattern Recognition Letters

www.elsevier.com/locate/patrec

An accurate and fast point-to-plane registration technique

Soon-Yong Park *, Murali Subbarao

Department of Electrical and Computer Engineering, College of Engineering and Applied Science, State University of New York at Stony Brook, Stony Brook, NY 11794-2350, USA

Received 29 January 2003; received in revised form 1 July 2003

Abstract

This paper addresses a registration refinement problem and presents an accurate and fast *point-to-*(*tangent*) *plane* technique. *Point-to-plane* approach is known to be very accurate for registration refinement of partial 3D surfaces. However, the computation complexity for finding the intersection point on a destination surface from a source control point is hindering the algorithm from real-time applications. We introduce a novel *point-to-plane* registration technique by combining the high-speed advantage of *point-to-projection* technique. In order to find the intersection point fast and accurately, we forward-project the source point to the destination surface and reproject the projection point to the normal vector of the source point. We show that iterative projections of the projected destination point to the normal vector converge to the intersection point. By assuming the destination surface to be a monotonic function in a new 2D coordinate system, we show contraction mapping properties of our iterative projection technique. Experimental results for several objects are presented for both pair-wise and multi-view registrations.

Keywords: Range registration; point-to-plane; Contractive projection point

1. Introduction

Registration refinement of multiple range images is an essential step in multi-view 3D modeling. When the range images are coarsely registered by a priori knowledge of registration parameters, registration refines the rigid transformation parameters to minimize alignment error between overlapping range surfaces. There have been many investigations on the refinement problem (Campbell and Flynn, 2001; Rusinkiewicz and Levoy, 2001). For a pair-wise registration, the registration problem can be considered to be an error minimization of the rigid body transformation between control point sets (Arun et al., 1987) on the pair of surfaces. Control point sets are decided by matching either geometric (Besl and McKay, 1992), photometric (Lensch et al., 2000; Weik, 1997), or geometric and photometric (Bernadini et al., 2001; Johnson and Kang, 1997) structures of the range surfaces. In this paper, we address a registration problem in the first category, which considers only geometric structure for control point matching.

* Corresponding author. Tel.: +1-6316329149; fax: +1-6316328494.

E-mail addresses: parksy@ece.sunysb.edu (S.-Y. Park), murali@ece.sunysb.edu (M. Subbarao).

0167-8655/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0167-8655(03)00157-0

Based on the method of control point matching, three approaches can be considered in general. *Point-to-point*, e.g. iterative closest point (ICP) algorithm (Besl and McKay, 1992), *point-to-(tangent) plane* (Chen and Medioni, 1992; Bergevin et al., 1996), and *point-to-projection* (Blais and Levine, 1995) techniques are well-known registration techniques. The ICP algorithm is one of the common techniques for refinement of partial 3D surfaces (or models) and many variant techniques have been investigated. However, searching the closest point in the ICP algorithm is a computationally expensive task. In order to accelerate the speed of closest point searching, some searching techniques are commonly employed, for example a kd-tree searching, z-buffering, or a closest-point caching (Nishino and Ikeuchi, 2002; Benjemaa and Schmitt, 1999).

In contrast, *point-to-projection* approach finds the correspondence of a source control point by projecting the source point onto a destination surface from the point of view of the destination (Blais and Levine, 1995; Neugebauer, 1997). When the source point is backward projected to the image plane of the destination surface, the destination control point is the forward projection of the image point. This approach makes registration very fast, because it does not involve any searching step to find the correspondence. However, one of its disadvantages is that the result of registration is not as accurate as those of the others (Rusinkiewicz and Levoy, 2001).

Among three approaches, *point-to-(tangent)-plane* technique is known to be the most accurate (Pulli, 1999; Rusinkiewicz and Levoy, 2001). From a source control point, the matching control point is the projection of the source point onto the tangent plane at a destination surface point which is the intersection of the normal vector of the source point (Bergevin et al., 1996; Chen and Medioni, 1992; Gagnon et al., 1994). However, finding the intersection on the destination surface is also computationally expensive. One of the acceleration techniques is first searching the closest point, and finding the intersecting surface (or the triangle) from its neighboring triangles (Rusinkiewicz and Levoy, 2001). Gagnon (Gagnon et al., 1994) tries to find the intersection on a 2D grid of range images along the projection of the normal vector of the source point.

In this paper, we address the registration problem to refine a pair of range surfaces as well as multi-view range surfaces. We propose an accurate and fast *point-to-plane* registration technique. We combine advantages of *point-to-plane* and *point-to-projection* techniques for fast control point searching. We employ the accuracy from the *point-to-plane* technique and the speed from the *point-to-projection* technique. In order to find the intersecting control point, we project a source point to the destination surface, re-project the projection point to the normal vector of the source point. We show that iterative normal projections converge to the intersection point. By assuming the destination surface as a monotonic function in a new 2D coordinate system, we show a contraction mapping property of our registration technique (Naylor and Sell, 1982). Therefore, we call our technique as *contractive projection point* (CPP) algorithm. Experimental results for several 3D models are presented for many pair registrations as well as a multi-view registration.

2. Description of registration techniques

In this section, we briefly describe the basic principle of three registration refinement techniques. Suppose there is a control point set $\boldsymbol{P}$ on a source surface $\mathscr{S}_P$, and another control point set $\boldsymbol{Q}$ on a destination surface $\mathscr{S}_Q$. If we have K control points on each surface and $k = 0, \ldots, K-1$, then the registration problem is estimating a rigid body transformation $\boldsymbol{T} = [\boldsymbol{R}|\boldsymbol{t}]$ which minimizes an alignment error measure ϵ such that

$$\epsilon = \sum_{k=1}^{K} \|\boldsymbol{Q}_k - (\boldsymbol{R}\boldsymbol{P}_k + \boldsymbol{t})\|^2. \quad (1)$$

In general, source control point set $\boldsymbol{P}$ is selected by sampling the source surface-randomly or uniformly, and filtered by some constraints to delete unreliable control points. The destination control point set $\boldsymbol{Q}$ is the conjugate of the source point set, which is determined by a matching criterion.

Point-to-point technique (or ICP technique) is the most common technique. From a source control point $\boldsymbol{p}$ on the source surface, the ICP algorithm searches the closest point $\boldsymbol{q}$ on the destination surface. Fig. 1(a) shows a basic diagram of the ICP algorithm. An error metric d_s is the distance between two control points. In order to search the closest point, a searching technique, e.g. kd-tree, is commonly used. However, when a multi-view registration concerned, we need to rebuild kd-trees of the multiple surfaces at every iteration of registration.

Point-to-plane registration is another common technique. It searches the intersection on the destination surface from the normal vector of the source point. As shown in the Fig. 1(b), the destination control point $\boldsymbol{q}'$ is the projection of $\boldsymbol{p}$ onto the tangent plane at $\boldsymbol{q}$ which is the intersection from the normal of $\boldsymbol{p}$. One of the previous investigations is done by Chen and Medioni (Chen and Medioni, 1992). To find control points, they use a root searching technique similar to Newton–Raphson technique in an orthographic coordinate system. Bergevin et al. (1996) also use a similar technique with that of Chen and Medioni. They employ an image-based control point searching technique to reduce searching time.

Point-to-projection approach is known to be a fast registration technique. As shown in the Fig. 1(c), this approach determines a point $\boldsymbol{q}$ which is the conjugate of a source point $\boldsymbol{p}$, by forward-projecting $\boldsymbol{p}$ from the point of view of the destination $\boldsymbol{O}_Q$. In order to determine the projection point, $\boldsymbol{p}$ is first backward-projected to a 2D point $\boldsymbol{p}_Q$ on the range image plane of the destination surface, and then $\boldsymbol{p}_Q$ is forward-projected to the destination surface to get $\boldsymbol{q}$. This algorithm is very fast because it does not include any searching step to find the correspondence. However, one of its disadvantages is that the result of registration is not as accurate as those of the others (Rusinkiewicz and Levoy, 2001).

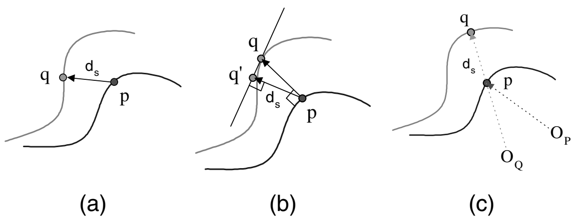

Fig. 1. Three common registration techniques: (a) *point-to-point*, (b) *point-to-plane*, (c) *point-to-projection*.

3. Contractive projection point (CPP) technique

3.1. Combining point-to-plane and point-to-projection techniques

In this section, we propose a novel *point-to-plane* registration technique by combining the fast searching property of the *point-to-projection* technique. Suppose there are two partial surfaces $\mathscr{S}_P$ and $\mathscr{S}_Q$ as shown in Fig. 2. They are assumed to be acquired from two different views P and Q of an object and coarsely registered to a common (world) coordinate system through transformation matrices $\boldsymbol{T}_P$ and $\boldsymbol{T}_Q$, respectively. Let us also suppose there is a source control point $\boldsymbol{p}_0$ on $\mathscr{S}_P$. Then the problem in the *point-to-plane* registration is finding the intersection point $\boldsymbol{q}_s$ on $\mathscr{S}_Q$ as shown in the figure. The point $\boldsymbol{q}_s$ is an intersection on $\mathscr{S}_Q$ by the normal vector $\hat{\boldsymbol{p}}$ of $\boldsymbol{p}_0$.

One of the typical methods of searching the intersection is first finding a triangle (when the surface consists of triangles) which is intersected by $\hat{\boldsymbol{p}}$. Then the intersection point $\boldsymbol{q}_s$ can be interpolated by three vertices on the triangle. However, this is a computationally expensive task and it may take several minutes to find all conjugates of

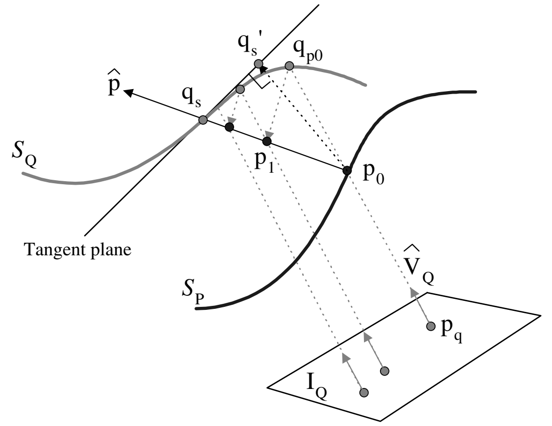

Fig. 2. Finding the intersecting point $\boldsymbol{q}_s$ from $\boldsymbol{p}_0$ by the proposed algorithm.

hundreds of control points. On the contrary, the *point-to-projection* technique can find a matching destination point from a source point within a millisecond on a typical personal computer, because it directly employs the parametric surface of the destination range image.

We consider a new *point-to-plane* registration technique by employing advantages from both techniques. The main idea is using *iterative point-to-projections* to determine the intersection on the destination surface. Let us back-project $\boldsymbol{p}_0$ in Fig. 2 to a 2D image point

$$\boldsymbol{p}_q = \boldsymbol{M}_Q \boldsymbol{T}_Q^{-1} \boldsymbol{p}_0, \tag{2}$$

where $\boldsymbol{M}_Q$ is the perspective projection matrix of view Q to the image plane I_Q, and $\boldsymbol{T}_Q$ is the transformation matrix from the camera coordinate system of view Q to the world coordinate system. Then let us forward-project $\boldsymbol{p}_q$ to a new 3D point $\boldsymbol{q}_{p0}$. Forward-projection $\boldsymbol{q}_{p0}$ is the range of the parametric surface at the destination image point $\boldsymbol{p}_q$. The point $\boldsymbol{q}_{p0}$ is computed by interpolating a grid of destination image plane and transformed back to the world coordinate system. Now let us consider another projection of $\boldsymbol{q}_{p0}$ to the normal vector $\hat{\boldsymbol{p}}$ at $\boldsymbol{p}_0$. Then we obtain a new 3D point $\boldsymbol{p}_1$, such that

$$\begin{aligned} \alpha &= (\boldsymbol{q}_{p0} - \boldsymbol{p}_0) \cdot \hat{\boldsymbol{p}}, \\ \boldsymbol{p}_1 &= \boldsymbol{p}_0 + \alpha \hat{\boldsymbol{p}}. \end{aligned} \tag{3}$$

If we iterate the same projections above using the new control point $\boldsymbol{p}_1$, then we get the next source point $\boldsymbol{p}_2$, and so on. If there is an intersection on $\mathscr{S}_Q$ by $\hat{\boldsymbol{p}}$, then a point $\boldsymbol{q}_{pi}$ (or $\boldsymbol{p}_i$) for the ith projection will converge to $\boldsymbol{q}_s$ when i goes to infinity, such that

$$\lim_{i\to\infty} \|\boldsymbol{p}_i - \boldsymbol{q}_{pi}\| \to 0. \tag{4}$$

However in real situation, only small number of projections can make a convergence measure

$$\epsilon_c = \|\boldsymbol{p}_i - \boldsymbol{q}_{pi}\| \tag{5}$$

become close to zero. If we find all convergence points $\{\boldsymbol{q}_s\}$ for all corresponding source $\{\boldsymbol{p}_0\}$, we can find all projection points $\{\boldsymbol{q}'_s\}$ on tangent planes at $\{\boldsymbol{q}_s\}$ and use them as control point sets in the *point-to-plane* registration.

3.2. Contraction mapping property of CPP

In order to show the validity of the proposed algorithm, we show contraction mapping properties of the algorithm. When a source control point converges to an intersection, the proposed technique shows contraction mapping properties such that the convergence measure ϵ_c becomes close to zero after a couple of normal projections. The definition of the contraction mapping is as follows.

Definition 3.1. Let $(\boldsymbol{X}, d)$ be a metric space and $f : X \to X$. We say a *contraction mapping*, if there is a real number $k, 0 \leqslant k < 1$, such that

$$d(f(x), f(y)) \leqslant kd(x, y)$$

for all x and y in X.

Let us consider a 2D coordinate system as shown in Fig. 3. When there are two different view points P and Q, the new axes consist of the viewing vector $\hat{\boldsymbol{V}}_Q$ of Q and the normal vector $\hat{\boldsymbol{p}}$ of the source control point $\boldsymbol{p}_0$. And the point $\boldsymbol{p}_0$ becomes the origin of the coordinate system. Now the surface $\mathscr{S}_Q$ becomes a contour on the 2D plane, which is the intersection of the surface with the 2D plane. As shown in the figure, the proposed algorithm projects $\boldsymbol{p}_i$ to a new point $\boldsymbol{p}_{i+1}$, iteratively until the convergence measure in Eq. (5) becomes very small.

Practically speaking, distance from $\boldsymbol{p}_0$ to the contour is very small, because two surfaces are assumed to be registered coarsely at the beginning. Therefore we can consider that all forward-projecting lines from the view origin of Q to all control point $\boldsymbol{p}_i$ are almost parallel. Then the following

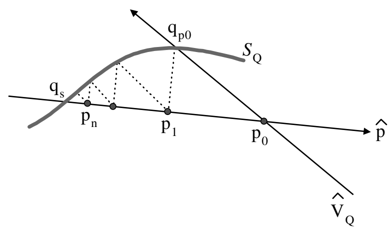

Fig. 3. Contraction mapping property of a new 2D coordinate system.

theorem shows contraction mapping properties of the proposed algorithm.

Theorem 3.1

(1) *A surface* $\mathscr{S}_Q$ *is a monotonic function with respect to* $\hat{\mathbf{v}}_{\mathbf{Q}}$.
(2) *A vector product* $\hat{\mathbf{v}}_{\mathbf{Q}} \cdot \hat{\boldsymbol{p}}_{\mathbf{0}} < 0$, *because we only consider a source point which is seen from the view Q. Therefore there is always a projection of* $\boldsymbol{q}_{pi}$ *to* $\hat{\boldsymbol{p}}_{\mathbf{0}}$.
(3) *If there is an intersection* $\boldsymbol{q}_s$, *a function* $f : \boldsymbol{p}_i \rightarrow \boldsymbol{p}_{i+1}$ *(or* $\boldsymbol{q}_{pi} \rightarrow \boldsymbol{q}_{p_{i+1}}$*) is a contraction mapping when* $\|\boldsymbol{p}_{i+1}, \boldsymbol{p}_{i+2}\| < k\|\boldsymbol{p}_i, \boldsymbol{p}_{i+1}\|$, *where* $0 \leqslant k < 1$.
(4) *Then, there is a 3D point* $\boldsymbol{q}_s$ *on the surface such that* $f(\boldsymbol{q}_s) = \boldsymbol{q}_s$.

3.3. Convergence condition

In the ideal situation, the CPP algorithm always converges to a convergence point. However, in a real situation, it may diverge or enter to a non-convergence cycle. Therefore we need to find out the convergence condition according to the contraction coefficient k. Three possible cases are shown in the Fig. 4. In the Fig. 4(a), a source control point is converging and the coefficient is $0 \leqslant k < 1$. In this case, we can use the convergence point as a matching control point of its conjugate.

However, as shown in the Fig. 4(b), the control point could diverge if the coefficient is $k > 1$. This could happen when the tangent normal at a point on the surface $\mathscr{S}_Q$ has a high angle with respect to $\hat{\boldsymbol{V}}_{\boldsymbol{Q}}$. However, we reduce the effect of diverging control points by discarding a destination point if the convergence measure ϵ_c is greater than an initial measure $\|\boldsymbol{q}_{p0} - \boldsymbol{p}_0\|$. The last case is when the mapping enters into a non-convergence cycle and $k = 0$, as shown in the Fig. 4(c). This case could happen when a segment of the destination curve is overlapping with the projection line from $\boldsymbol{q}_{p_i}$ to $\boldsymbol{p}_{i+1}$. We also remove this kind of points by checking the error measure for 3 or 4 iterations. If a measured value repeats for 3 or 4 iterations, we consider the mapping enters a non-convergence cycle and quit the mapping.

3.4. CPP algorithm

The proposed CPP algorithm can be easily implemented by a recursive searching program. The following function is a pseudo code of part of the algorithm. The function searches a 3D point $\boldsymbol{q}_{p1}$, the intersection of a control point $\boldsymbol{p}_1$ on the destination surface *Dest*. N_c is the maximum normal projections and D_c is threshold for the convergence error ϵ_c.

```
Vec3d Recursive Projection(Vec3d&p1, Vec3d&p̂, Vec3d&qp1, Mesh * Dest, int cnt)
{     if (cnt > Nc) return p1;
      pq = Mq p1;
      flag = OnObject(pq, p1, Dest, qp1);
      if (! flag) return NULL;
      α = (qp1 − p1) · p̂;
      p2 = p1 + αp̂;
      ε'c = ||p2 − qp1||;
      if (ε'c < Dc) return p2;
      return Recursive Projection(p2, p̂, qp1, Dest, cnt + 1);
}
```

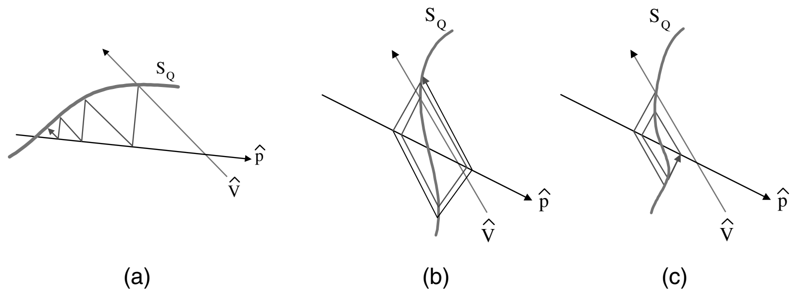

Fig. 4. Three cases of projection: (a) convergence ($0 \leqslant k < 1$), (b) divergence ($k > 1$), (c) infinity loop ($k = 1$).

After finding all matching control points, we use two control point sets $\{\boldsymbol{p}_0\}$ and $\{\boldsymbol{q}'_s\}$ for estimating a rigid transformation $\boldsymbol{T}$ between two surfaces. The transformation is computed by a SVD (Singular Value Decomposition) technique iteratively until two surfaces converge (Arun et al., 1987).

4. Experimental results

4.1. Test objects

We test our algorithm for three objects. The test objects are shown in the Fig. 5. The *wave* object is a pair of synthetic 3D sinusoidal surfaces. Two surfaces have 10 degree of phase difference in the *XY* range image plane and 10 mm of translation along the *Z* axis. Their height is 50 mm and we add random noise to every image point with maximum 12% of the height. The second object *angel* is a pair of range images obtained from a laser range finder. This object is one of the models from the Ohio State University's Range Image Database. The two images has 20° of rotation angle. For the third object *potatohead* in Fig. 5(c), we test both pair-wise and multi-view registration. We obtain eight range images of the object. Range images of the object have some erroneous points on their surfaces. The number of points and triangles on the test objects are shown in the Table 1. They are the number of points and triangles on the surface of the first view, but the numbers of the others are also similar.

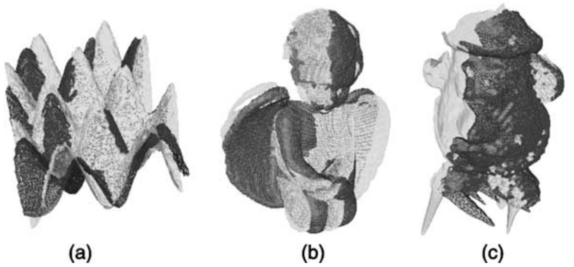

Fig. 5. Point clouds models of test objects: (a) wave, (b) angel, (c) potatohead.

Table 1
Test objects

Objects	Wave	Angel	Potatohead
Number of points (view 0)	22 500	14 089	13 827
Number of triangles (view 0)	44 402	27 213	27 144

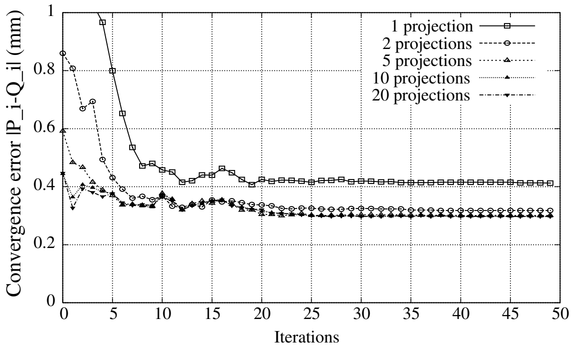

Fig. 6. Comparison of convergence error with respect to different number of projections. (*wave* object).

4.2. Registration error with respect to projection numbers

In order to decide an appropriate number of normal projections—from $\boldsymbol{q}_{pi}$ to $\boldsymbol{p}_{i+1}$, we apply the CPP technique to a noiseless *wave* object to plot convergence error ϵ_c with respect to different numbers of projections. In Fig. 6, the convergence error is plotted for 50 iterations of registration. At each iteration, RMS errors of all control points are plotted, for which we use about 300 pairs. As shown in the figure, the convergence error decreases if the number of projections increases. However, its decrease rate is saturated after 5 or 6 projections. Therefore, we use 5 normal projections for all following experiments.

4.3. Pair-wise registration

We compare our algorithm with the *point-to-projection* and the ICP algorithm. We also use a kd-tree searching technique for ICP. We test all three techniques in the same registration condition on a Pentium III 1 GHz personal computer. As far as the *wave* object is concerned, we know the real conjugate points between two range surfaces.

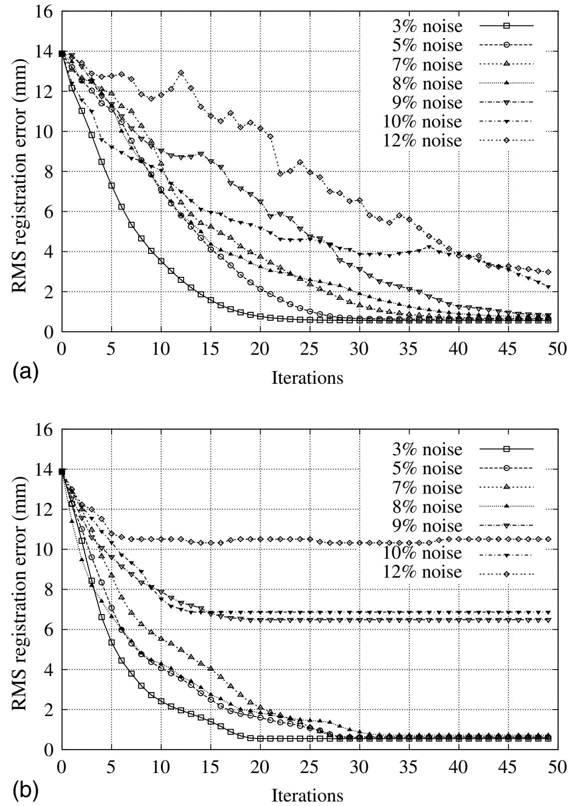

Fig. 7. Registration error of *wave* object: (a) CPP, (b) ICP.

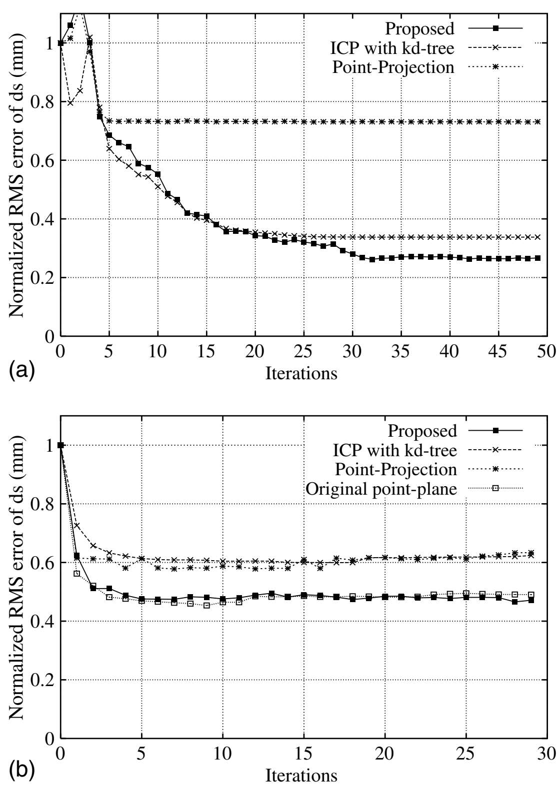

Fig. 8. RMS error results of (a) angel, (b) potatohead.

Therefore, we measure RMS distance error between two ground truth point sets. For the other objects, we measure d_s distance shown in the Fig. 1. However, the distance error in each approach cannot be compared directly because they use different error metric. To fairly compare them, we normalize each error by using its initial error. Because the initial condition of all approaches is the same each other, we normalize the RMS errors to watch the decrease rate of them.

Registration results of CPP and ICP for *wave* object are plotted in Fig. 7. We plot the results for 50 iterations, and for different noise rate on the object's surface. Two algorithms converge well with low noise rate, however the ICP algorithm fails to converge when noise rate is more than 9%. The results of the other objects are shown in the Fig. 8. Fig. 8(a) shows the results for the object

Table 2
Registration error and processing time after 50 iterations ($N_c = 5$, $D_c = 0.1$ mm, and d_s is normalized)

Method		Point-projection	ICP with kd-tree	Proposed CPP
Wave with	Ground truth (mm)	–	0.63	0.63
5% noise	Time/Itr (ms)	–	42.8	43.8
Wave with	Ground truth (mm)	–	6.86	2.25
10% noise	Time/Itr (ms)	–	44.0	42.8
Angel	d_s (mm)	0.73	0.34	0.27
	Time/Itr (ms)	30.8	37.4	36.2
Potatohead	d_s (mm)	0.63	0.63	0.47
	Time/Itr (ms)	26.4	30.6	30.8

angel. The *point-to-projection* result shows a bad registration result. The proposed technique shows better convergence rate than other techniques. Fig. 8(b) is the results for *potatohead*. This result only shows the registration of the first pair of images. In this figure, we also plot the result of original point-to-plane technique, which use a brute force searching to find matching control points. The result of the proposed technique is almost the same as that of the original technique.

Table 2 shows the results of registration error and processing time after 50 iterations. As shown in the previous figure, our CPP technique has the smallest registration error. As far as the computation time is concerned, the point-to-projection technique is faster than the other approaches as we expected. The proposed technique and the point-to-point technique show similar results but they depend on types of objects. This is because we need some computations for searching intersection points as shown in the CPP pseudo code.

4.4. Number of convergence

As described in an earlier section, we have three types of convergence—Convergence, Divergence, and Infinity loop—in searching of intersections. After some experiments, we find that CPP searching converges well in most cases to intersections on destination surface. However, sometimes it diverges or enters to an infinity loop. Even though we have some number of non-convergence points, we discard them and use only convergent control points for deriving transformation matrix. The numbers of converging and non-converging points are different for every iteration. However, Table 3 shows average numbers of them for 50 iterations on each object.

Table 3
Average number of convergence and non-convergence control points

Object	Convergence	Divergence	Infinity loop
Wave (5% noise)	92.5	6.2	1.3
Angel	220.5	52.7	2.6
Potatohead	202.1	32.9	3.2

4.5. Multi-view registration

Multi-view registration is a more difficult problem than a pair-wise registration, because registration error should be evenly distributed on all overlapping surfaces. We regard the first view

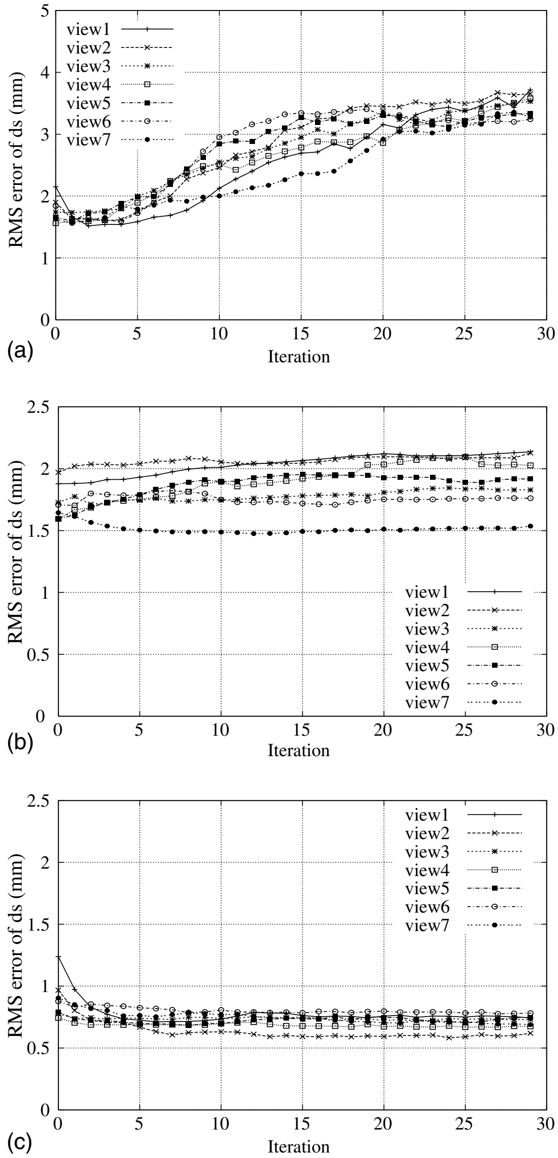

Fig. 9. Results of multi-view registration for *potatohead* after 30 iterations. (a) Point-to-projection, (b) ICP with kd-tree, (c) proposed (CPP).

Table 4
Multi-view registration error and processing time after 50 iterations. ($N_c = 5$, $D_c = 0.1$ mm)

Method	Point-projection	ICP with kd-tree	Proposed
RMS error d_s (mm)	3.45	1.88	0.74
Time (s)	9.61	46.57	12.63
Time/Itr (s)	0.192	0.931	0.253

of the *potatohead* object as a reference view and register all the other views to the reference view. In order to find the matching control points from a source view, we search on its neighboring view's surfaces. For example, if the ith view is concerned, we search the matching control points on its neighborhoods, $(i+1)$th view and $(i-1)$th view. After finding all matching points for every view point, except the first view, we compute rigid transformations and refine all surfaces simultaneously.

We test a multi-view registration for the 8 views of the *potatohead* object. Fig. 9 plots their results. In this figure we do not normalize the error, because the results are obvious to compare. Both the *point-to-projection* and the ICP techniques show bad registration results. Both of them fail to register the multi-view images. In contrast, the CPP algorithm register all range images well. Table 4 shows registration error and processing time for this experiment. The ICP with kd-tree technique take more time, because it needs to refine kd-trees of all surface (except the reference view) for every iteration. Table 4 shows that our CPP technique has advantages over ICP and *point-to-projection* techniques for registration of multiple range images.

5. Conclusions

We address a registration refinement problem and present an accurate and fast *point-to-(tangent)plane* registration technique. In order to find an intersection point on a destination surface, we project a source control point to the destination surface, re-project the projection point to the normal vector of the source point. We show that iterative projections of the projected destination point to the normal vector converge to the intersection point. By assuming the destination surface to be a monotonic function in a new 2D coordinate system, we show contraction mapping properties of our *contractive projection point* technique. In an alternative technique, finding convergence point could be implemented by another method such as the Newton–Raphson method. Instead of projecting the destination control to the vector normal, the tangent vector at the destination point can be used to find the intersection with the vector normal. Experimental results show that our approach is very accurate and fast for both pair-wise registration and multi-view registration problems.

References

Arun, K., Huang, T., Blostein, S., 1987. Least-squares fitting of two 3D point sets. IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (5).

Benjemaa, R., Schmitt, F., 1999. Fast global registration of 3D sampled surface using a multi-z-buffer technique. Image and Vision Computing 17, 113–123.

Bergevin, R., Soucy, M., Gagnon, H., Laurendeau, D., 1996. Toward a general multi-view registration technique. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (5).

Bernadini, F., Martin, I.M., Rushmeier, H., 2001. High-quality texture reconstruction from multiple scans. IEEE Transactions on Visualization and Computer Graphics 7 (4), 318–332.

Besl, P.J., McKay, N.D., 1992. A method of registration of 3D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2), 239–256.

Blais, G., Levine, M., 1995. Registering multiview range data to create 3D computer object. IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (8).

Campbell, R.J., Flynn, P.J., 2001. A survey of free-form object representation and recognition techniques. Computer Vision and Image Understanding 81, 166–210.

Chen, Y., Medioni, G., 1992. Object modeling by registration of multiple range images. Image and Vision Computing 10 (3), 145–155.

Gagnon, H., Soucy, M., Bergevin, R., Laurendeau, D., 1994. Registration of multiple range views for automatic 3D model building. IEEE Computer Vision and Pattern Recognition, 581–586.

Johnson, A., Kang, S., 1997. Registration and integration of textured 3D data. IEEE Conference on Recent Advances in 3-D Digital Imaging and Modeling.

Lensch, H., Heidrich, W., Seidel, H., 2000. Automated texture registration and stitching for real world models. In:

Proceedings of Pacific Graphics 2000, IEEE Computer Society Press, pp. 317–327.

Naylor, A., Sell, G., 1982. Linear operator theory in engineering and science. Springer-Verlag, New York.

Neugebauer, P., 1997. Geometrical cloning 3D objects via simultaneous registration of multiview range images. IEEE Conference on Shape Modeling and Applications, 130–139.

Nishino, K., Ikeuchi, K., 2002. Robust simultaneous registration of multiple range images. In: The 5th Asian Conference on Computer Vision (ACCV2002), pp. 23–25.

Pulli, K., 1999. Multiview registration for large data sets. In: Proceedings of 2nd International Conference on 3D Digital Imaging and Modeling.

Rusinkiewicz, S., Levoy, M., 2001. Efficient variants of the ICP algorithm, In: Proceedings of 3D Digital Imaging and Modeling, pp. 145–152.

Weik, S., 1997. Registration of 3-D partial surface models using luminance and depth information. In: Proceedings International Conference on Recent Advances in 3-D Digital Imaging and Modeling, pp. 93–100.