A new refinement method for registration of range images based on segmented data *

B. Kverh and A. Leonardis, Ljubljana

AMS classification: 68U10 image processing, 68D17 computer aided design

Abstract

We present a new method for registration of range images, which is based on the results we obtain from the segmentation process. We need two range images segmented into regions, each of them modeled by a paramteric model and the approximation of the transformation between the two range images. Then two sets of corresponding points, one from each range image, are chosen and the transformation between them is computed to further refine the initial approximation of the transformation. The novelty is how we obtain the a corresponding points for the original set of points from the range image. Namely, to obtain them we project set of points from the first range image onto geometric parametric models that were recovered in the second range image and viceversa. This way we obtain two sets of corresponding points. Then we compute the transformation between the two sets. Few iterations are required to improve the initial approximation of the transformation. The results have shown a significant improvement in precision of the registration in comparison with traditional approaches.

Keywords: registration, segmentation, range images, reverse engineering, CAD models.

[*] This work was supported by a grant from the Ministry of Science and Technology of Republic of Slovenia (Project J2-0414-1539-98).

1 Introduction and motivation

Registration of range images is considered an important part of any vision system that requires data acquisition of parts that are not visible from a single viewpoint. There is no 3D sensor system available on the market that would obtain 3D data of the complete surface by a single scan. Thus, to extract complete information about the surface of an object we need to scan it from several viewpoints and then merge the data together. This requires that we compute the transformation between data sets obtained from different viewpoints. A rough approximation of the transformation can be obtained by involvement of a human operator, from positioning devices, e.g., a turntable, or automatically by detecting corresponding features in both sets. For some tasks, like reverse engineering, the accuracy of this approximate transformation has to be further improved by one of the refinement methods. In this paper we present a new refinement method which is based on segmented data, i.e., descriptions obtained by Recover-and-Select paradigm [8, 9, 10, 11]. Refinement methods usually compute the transformation either between two sets of points [1, 15] or between a set of points and a set of some surface elements, e.g., triangles [14]. For each point from the first set, these methods find a corresponding point or surface element in the second set. Transformation between two sets of corresponding points is then computed by methods usually based on least squares. However, due to scanner sampling it is impossible to find the exact corresponding point or surface element for each point in the first set. This situation is illustrated in Fig. 1, which clearly shows that if the refinement is performed in this way, small errors will affect the computed transformation. Therefore we propose a novel approach which alleviates this problem.

Registration of all range images is achieved by repeatedly registering pairs of range images. In each step, 3D points from both range images are first segmented into descriptions by the Recover-and-Select paradigm. This gives us geometric parametric models which approximate each segmented region. Let us assume that we have an initial approximation of the transformation. Then, instead of computing the transformation between the two sets of 3D points scanned by the sensor, we can use one set of points and their corresponding *projections* to geometric parametric models that were obtained for the other set of points and viceversa. This process can be iterated until the transformation error reaches a predefined value or just for some predefined number of iterations. In this way we cancel the effect of scanner sampling

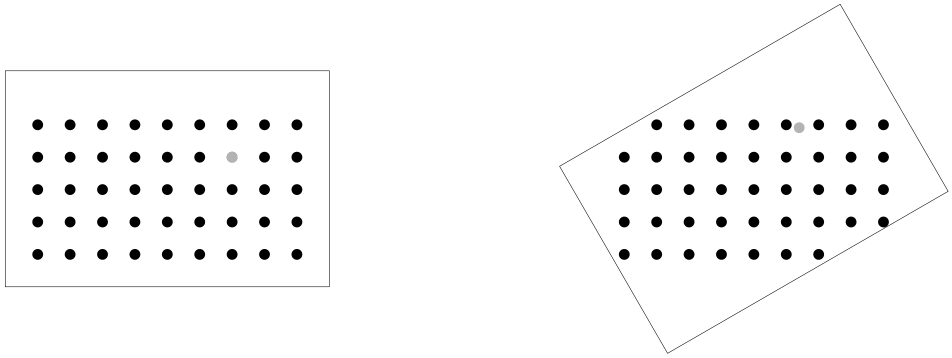

Figure 1: An example of two corresponding points. Gray point in the right image is not captured by the scanner in the second view.

(since the projection of a point onto a geometric model really corresponds with that point). Moreover, the point that corresponds to a particular point from one set, does not need to be captured by the scanner since it is computed from the segmented data.

The rest of the paper is organized as follows: in section 2 we introduce Recover-and-Select segmentation process, in section 3 we present several algorithms for improving the initial transformation, among them also our Projection Iterative Closest Point (PICP) algorithm. Sections 4 and 5 contain experimental results and discussion, respectively. Geometric parametric models and projections of points onto the models are presented in the appendix.

2 Segmentation of range images

Our method for refinement of the transformation between two partially overlapping range images assumes that both of them are segmented. We use the Recover-and-Select paradigm [8, 9, 10, 11], where the set of 3D points is segmented into regions. Each region though has its own geometric parametric model that fits best to the data in the region. The region and the model fitted to it form a description. To express the fit quality of model M to the region R, we use Eq. 1, known as description error,

$$Err = \frac{1}{|R|} \sum_{\mathbf{x} \in R} d(\mathbf{x}, M). \tag{1}$$

Function $d(\mathbf{x}, M)$ calculates distance of point $\mathbf{x}$ to the model M.

The Recover-and-Select paradigm can currently use the following geometric models: planes, spheres, cylinders, cones, and tori. But it can be easily extended in such a way that it recovers other geometric parametric models as well.

Here we will briefly review the segmentation procedure. For details, the reader is referred to [8, 9, 10, 11]. The Recover-and-Select paradigm starts with placing seed descriptions with very small regions on the range data. Then it performs few steps of description growing followed by description selection. This is performed until all remaining descriptions are fully grown. Description growing step consists of searching for compatible points in the neighbourhood of description region, joining these points to the region and fitting new description model. The compatibility criterion is usually based on the euclidian distance of point to the current description model. The description selection is performed since description regions start to overlap and many of them become redundant. The Recover-and-Select paradigm is based on principals of natural selection, which guarantees robustness in segmentation. This is necessary, since our refinement method strongly relies on segmentation and model fitting results.

An example of segmented data is shown in Fig. 2. Different gray levels denote different regions. Geometric parametric models in descriptions shown in Fig. 2 are planes (descriptions 2, 3, 4, 6, 7, 9 and 11), cylinder (description 1) and cone (description 5). In our transformation refinement method we use these models to compute projections of points to their surfaces.

3 Transformation refinement algorithms

3.1 ICP algorithm

The most popular algorithm for refining the initial transformation approximation between two sets of range data is so called ICP (Iterative Closest Point) algorithm [1]. For each feature (usually a point) in the first data set it finds the closest feature in the second data set under current transformation.

7
3
1
2
6
11
5
9
4

Figure 2: Generated intensity image of an object and its range image segmentation. Each description is labeled with a number.

These feature pairs are then used to compute the transformation refinement that minimizes the sum of distances between corresponding features.

The ICP algorithm in its simplest implementation computes the transformation between two sets of points X and Y from different coordinate systems. A point $\mathbf{y} \in Y$ is the corresponding point to a point $\mathbf{x} \in X$ if the condition 2 is satisfied.

$$\mathbf{y} = \arg\min_{\mathbf{t} \in Y} d(\mathbf{x}, \mathbf{t}). \tag{2}$$

Function d represents a measure of distance between the two points. The ICP algorithm then computes the transformation that minimizes the sum of distances between transformed points from set X and their corresponding points from the set Y. Several iterations of the ICP algorithm are usually performed to satisfy some convergence criterion, which is usually the average distance between the corresponding points or difference between this average distance in two successive iterations.

3.2 Improvements of the standard ICP algorithm

The convergence of the ICP algorithm occurs only if one of the data sets is the subset of the other data set. Otherwise incorrect correspondences occur for the points in one set, that are not included in the other data set. The first straightforward improvement is to introduce a threshold t_d on the distance

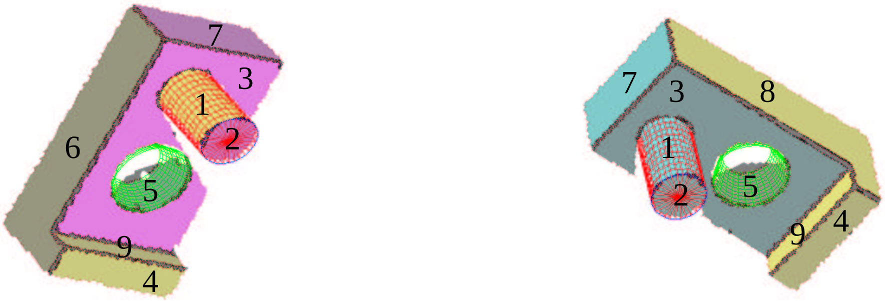

Figure 3: Two segmented range images. Corresponding descriptions have same indices in both images.

between corresponding points [2, 16]. Two points $\mathbf{x} \in X$ and $\mathbf{y} \in Y$ now correspond to each other if both conditions 2 and 3 are satisfied.

$$d(\mathbf{x}, \mathbf{y}) < t_d. \tag{3}$$

The condition 3 cuts out some incorrect correspondences generated by points not present in both data sets, but requires that a threshold value is set. This value primarily depends on the scanner resolution and the precision of the initial transformation approximation.

Another improvement was presented in [15]. The authors introduced additional condition 4 in order to reduce the number of incorrect correspondences between points.

$$\mathbf{x} = \arg\min_{\mathbf{t} \in X} d(\mathbf{t}, \mathbf{y}). \tag{4}$$

This condition guarantees, that if points $\mathbf{x} \in X$ and $\mathbf{y} \in Y$ correspond to each other, $\mathbf{x}$ is the closest point from X to $\mathbf{y}$ and viceversa. This approach was called ICRP (Iterative Closest Reciprocal Point) algorithm. Another possible improvement is described in [3]. Here from each point in the first data set a line is constructed that is parallel to a normal vector at the point. The closest point to this line in the second data set is then found and a tangent plane through it is used as surface approximation. The projection of the point from the first data set to the tangent plane is then used as the corresponding point.

This approach is similar to ours although it might be less robust to the noise because its surface approximation is based on local features only. Other improvements include modelling of sensor inaccuracies [4] and combining point locations with surface normals in the distance metric d [5]. Principle curvatures and other surface properties have also been used in determination of closest point [6, 7] .

One of the drawbacks of the ICP algorithm is that it is time consuming. Its time complexity is $O(n_1 n_2)$, where n_i is the number of points in data set i. Things do not change even if we use some of the improvements mentioned before. Time required for the finding closest point among n points is $O(n)$, unless we use a special tree structure ($k-d$ tree), which brings finding closest point time complexity down to $O(\log n)$. This means that ICP algorithm and its derivates have complexity $O(n_1 \log n_2)$ in the best case. On the other hand, it will be shown that our PICP algorithm has time complexity $O(n_1 + n_2)$ without using any special data structures.

3.3 PICP algorithm

We call our new transformation refinement method Projection Iterative Closest Point (PICP) algorithm, because it is similar to ICP algorithm in iterations performed after the correspondence has been obtained. PICP algorithm was developed to avoid the sampling effect. Using information obtained by segmentation stage and the initial approximation of the transformation between viewpoints from which the point sets have been scanned, we are able to determine pairs of corresponding descriptions. Description D_1 from the first set of points, consisting of region R_1 and model M_1, corresponds to description D_2 from the second set of points, consisting of region R_2 and model M_2 if conditions 5, 6 and 7 are satisfied.

$$\frac{1}{|R_2|} \sum_{\mathbf{x} \in T(R_2)} d(\mathbf{x}, M_1) < d_{tr}, \tag{5}$$

$$\frac{1}{|R_1|} \sum_{\mathbf{y} \in R_1} d(\mathbf{y}, M_2) < d_{tr}, \tag{6}$$

$$\exists \mathbf{x} \in R_1 \wedge \exists \mathbf{y} \in T(R_2) : d(\mathbf{x}, \mathbf{y}) < d_{tr}. \tag{7}$$

T denotes the initial approximation of the transformation, function d denotes a measure of distance between two points, or between a point and geometrical parametrical model, while d_{tr} represents a threshold value.

We obtain the initial approximation of the transformation automatically, so that each triple of descriptions from one set of points is matched to each triple of descriptions from the other set and the transformation between the two triples of descriptions is computed. The transformation that maximizes number of pairs of corresponding descriptions is chosen as the initial approximation. Characteristic features of each model, like normal vector of the plane, axis of rotation of cylinder, cone and tori, and center of the sphere are used to compute this initial approximation, which yields a good estimate of the real transformation.

For each pair of corresponding descriptions conditions 5 and 6 assure that points from the region of one description are close enough to the geometric model of the other description after the initial approximation of transformation has been applied to one of the sets of points. Condition 7 requires that points from regions of both descriptions are close enough, otherwise we have no information if descriptions really describe the same surface. In Fig. 3 two different segmented range images are shown.

When the pairs of corresponding descriptions are identified, PICP algorithm starts to construct sets X and Y. It starts with X and Y both empty sets. For each pair of the corresponding descriptions it adds points from the

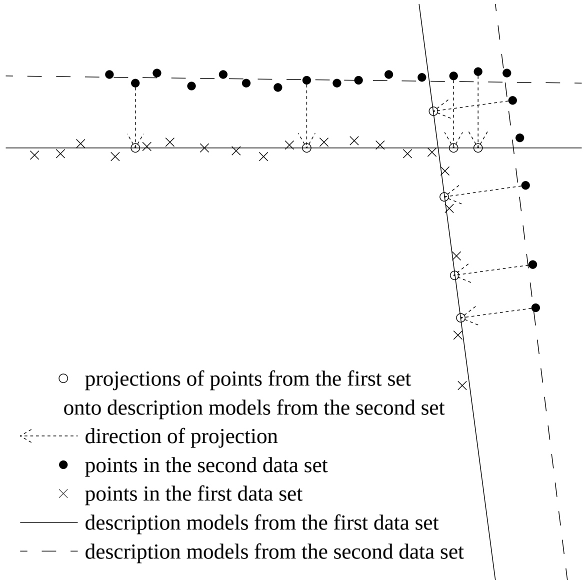
○ projections of points from the first set
onto description models from the second set
direction of projection
● points in the second data set
× points in the first data set
description models from the first data set
description models from the second data set

Figure 4: Projection of points from the first set of data to geometric models from the second set of data

region of description from the first range image to set X and projections of these points onto geometric model of description from the second range image to set Y. Also, points from the region of descriptions from the second range image are added to Y and their projections onto geometric model of descriptions from the first range image are added to X. We can always discard points and their projections which are too far apart from each other. In this way we can get rid of some noisy data and thus improve the precision of the estimated transformation. Figure 4 shows how to obtain the projection of a point from X to some geometric model of the description from second set of data. After all pairs of corresponding descriptions have been taken into consideration, sets X and Y are completed. PICP algorithm then uses a method based on quaternions [13] to compute the parameters of the transformation. The obtained transformation is used as the approximation of the transformation in the next iteration of PICP algorithm. Our experiments showed that with 20 to 50 PICP iterations we obtain significant improvement over other refinement methods in terms of accuracy. The whole method is outlined in algorithm 1. We use the following notation in algorithm 1. R_i and M_i denote the region and the model of description D_i, while N is the number of PICP iterations. In our experiments we used $N = 50$. We also use term list for X and Y since it is important to have a point and its corresponding counterpart in the same position in the list.

Since finding the closest (or corresponding) point on the surface of one data set for one point from the other data set requires time $O(1)$ (all we have to do is to project a point from one data set to the corresponding description in the other data set which provides the surface approximation), the overall time complexity of one PICP iteration is $O(n_1 + n_2)$.

4 Experimental results

We used the PICP algorithm in a reverse engineering application [9]. We scanned an object from four viewpoints and obtained four range images and segmented them with the Recover-and-Select paradigm (images RI_1, RI_2, RI_3, RI_4 shown in Fig. 6, while the corresponding generated intensity images are shown in Fig. 5. We registered images RI_1 and RI_2 and merged them together. In this way we obtained range image RI_{12}. Then we registered images RI_{12} and RI_3 and merged them to obtain the range image RI_{123}. Finally we registered images RI_{123} and RI_4 and merged them to obtain the

Algorithm 1 Outline of the PICP method

```
obtain initial approximation of the transformation $T$
transform second range image and its descriptions using $T$
for i = 1 to N do
  let $X$ and $Y$ be empty lists
  for each description $D_1 = (R_1, M_1)$ from the first range image do
    if there is corresponding description $D_2 = (R_2, M_2)$ from the second range image for $D_1$ then
      for each point $\mathbf{p}$ from $R_1$ do
        find its projection $\mathbf{p}^*$ onto $M_2$
        append $\mathbf{p}$ to $X$ and $\mathbf{p}^*$ to $Y$
      end for
    end if
  end for
  for each description $D_2 = (R_2, M_2)$ from the second range image do
    if there is corresponding description $D_1 = (R_1, M_1)$ from the first range image for $D_2$ then
      for each point $\mathbf{p}$ from $R_2$ do
        find its projection $\mathbf{p}^*$ onto $M_1$
        append $\mathbf{p}$ to $Y$ and $\mathbf{p}^*$ to $X$
      end for
    end if
  end for
  calculate new transformation $T$ between $X$ and $Y$
  transform second range image and its descriptions using $T$
end for
```

final image RI_{1234}. When merging two segmented range images to obtain a new range image, we do the following:

1. if description D_1 from the first segmented range image does not have a corresponding description in the second segmented range image, we insert D_1 into the new range image,
2. if description D_2 from second segmented range image does not have a corresponding description in the first segmented range image, we insert D_2 into the new range image,
3. if descriptions D_1 and D_2 from different range images correspond and

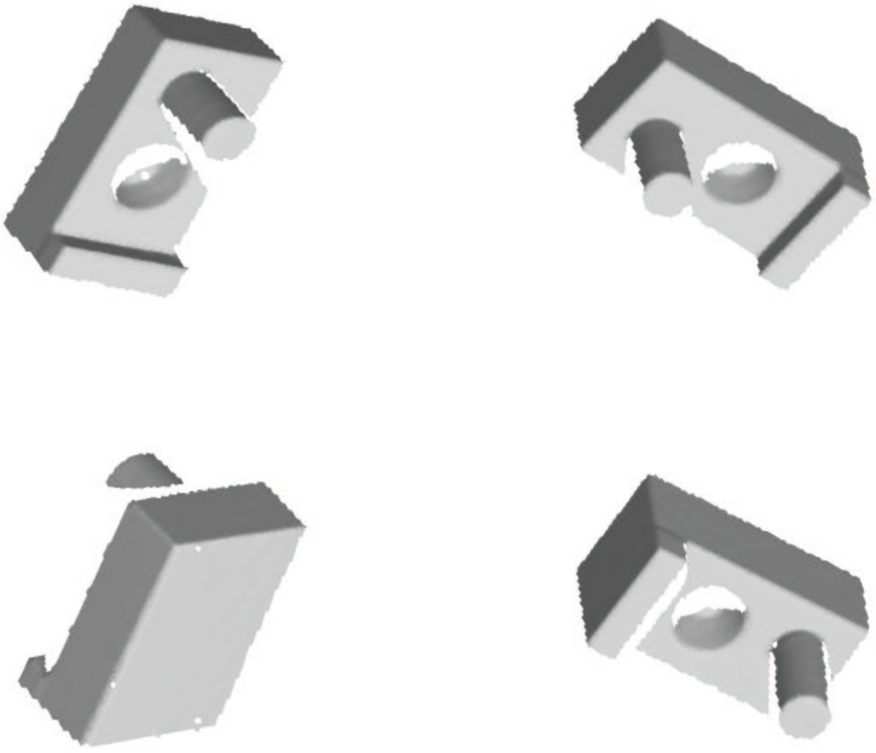

Figure 5: Generated intensity images of the same object taken from different viewpoints

> if R_1 is region of D_1 and R_2 is region of D_2, we form a new region $R = R_1 \bigcup R_2$ and fit new model M to region R. We insert the new description containing R and M into the new range image.

Let us consider for a moment the segmented range image which was generated by merging together two range images. If some description D was formed by merging together two corresponding descriptions D_1 and D_2, it is obvious that if the registration stage did not find a good transformation between the two range images, description error (Eq. 1) of D is significantly greater than description error of D_1 and D_2. Similarly, if the transformation found by the registration stage is accurate, description error of D should be

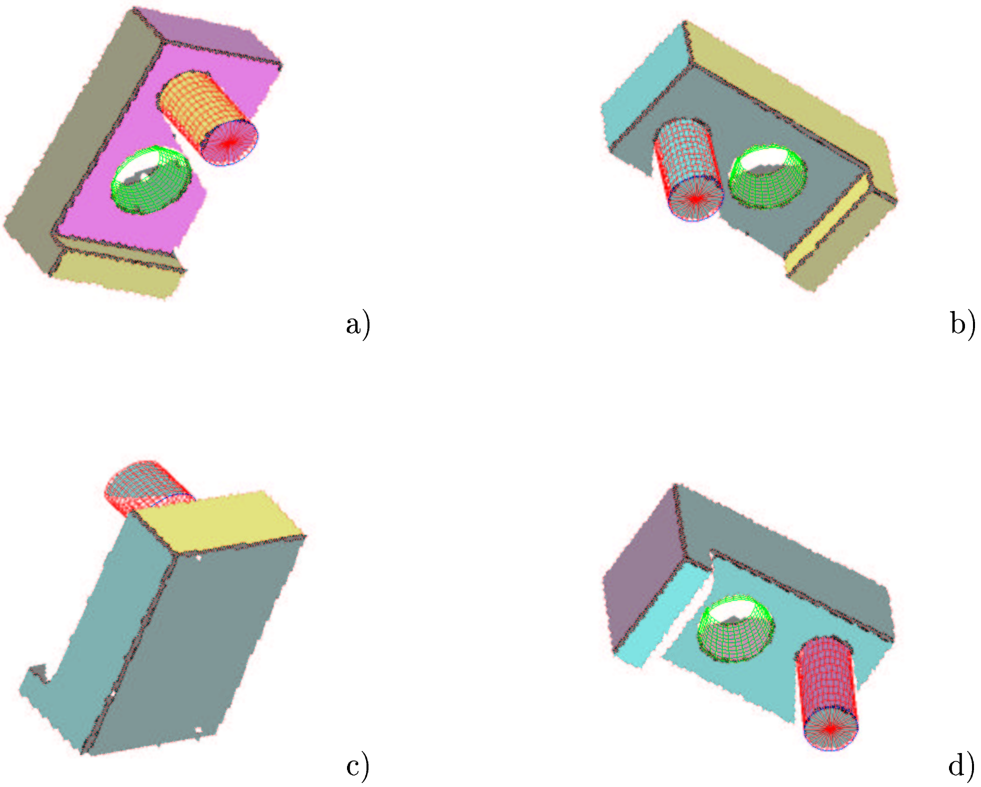

Figure 6: Segmented range images a) RI_1, b) RI_2, c) RI_3 and d) RI_4

similar to description errors of D_1 and D_2. Table 1 shows description errors for three descriptions shown in Fig. 7. Table 2 shows description errors after registration and merging without any refinement, while Table 3 shows description error after registration and merging with 50 PICP iterations performed. Descriptions that were merged from two corresponding descriptions when PICP algorithm was used, were not much worse in terms of description errors. This means that PICP algorithm significantly improves the accuracy of the registration. Note that when we used merged images RI_{123} and RI_4 without PICP refinement, the application did not recognize cone descriptions as corresponding (they did not match the conditions 5 and 6). This means that in this way we obtained two descriptions for the same object surface which is obviously a wrong result. This shows that refinement of the trans-

7
3
1
2
6
11
5
9
4

Figure 7: Generated intensity image of an object and its range image segmentation. Each description is labeled with a number.

Description	RI_1	RI_2	RI_3	RI_4
cylinder (index 1 in Fig. 7)	0.1023	0.1048	0.0548	0.1133
cone (index 5 in Fig. 7)	0.1654	0.1421	/	0.1269
planar (index 3 in Fig. 7)	0.0965	0.1172	/	0.1220

Table 1: Errors of three descriptions in four different viewpoints

formation is necessary in tasks which require precise estimation of object surfaces.

We define the transformation error as the average sum of distance between points from description regions and their projections to the models of corresponding descriptions. We analysed how the average distance of the points from region of description from the first data set to the model of the corresponding description from the second data set changes during PICP algorithm iterations. We have also compared PICP algorithm against some of the other refinement algorithms, mentioned in section 3.2, in terms of errors of the estimated transformations. Transformation errors when using different refinement algorithms on the data sets from Fig. 8 are summarized in Table 4.

The results that we obtained during our experiments show, that the refinement of initial approximation of the transformation is necessary to obtain more precise registration of range images. Our PICP algorithm proved to be

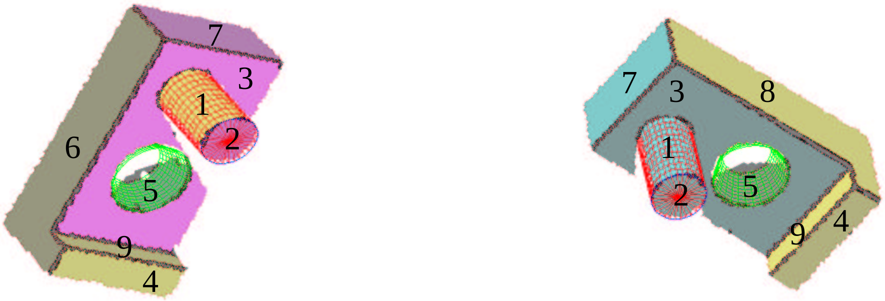

Figure 8: Two segmented range images for our experiments. Corresponding descriptions have same indices in both images.

Description	RI_{12}	RI_{123}	RI_{1234}
cylinder (index 1 in Fig. 7)	0.1739	0.3836	0.6636
cone (index 5 in Fig. 7)	0.2863	0.2863	0.2863
planar (index 3 in Fig. 7)	0.4513	0.4513	0.6813

Table 2: Errors of three descriptions after merging without PICP refinement

more precise than any other refinement algorithm we have tested (an example of comparison results is shown in Tab. 4). Moreover, PICP algorithm is less time consuming, although the time performance wasn't the primary issue of our work.

In Figs. 9 we show two examples of creating CAD models of objects with surfaces of regular geometry as possible application of our PICP algorithm. The left image shows a generated intensity image of the object, while the right one shows the computed boundary representation of the object. In the first example where we had symmetric object with only planar surfaces, sometimes the initial approximation of the transformation at registering had to be obtained manually. The use of human operator or positioning devices is often needed at registration of very simple and symmetric objects.

Description	RI_{12}	RI_{123}	RI_{1234}
cylinder (index 1 in Fig. 7)	0.1392	0.2793	0.2617
cone (index 5 in Fig. 7)	0.1877	0.1877	0.1799
planar (index 3 in Fig. 7)	0.1099	0.1099	0.1146

Table 3: Errors of three descriptions after merging with 50 PICP refinement iterations

	PICP	ICP	ICRP	Algorithm in [3]
initial approximation	0.7561	0.7561	0.7561	0.7561
after 10 iterations	0.2985	0.3498	0.3491	0.3378
after 20 iterations	0.2475	0.3510	0.3538	0.3308
after 30 iterations	0.2243	0.3572	0.3585	0.3257
after 40 iterations	0.2111	0.3566	0.3550	0.3207
after 50 iterations	0.2018	0.3621	0.3612	0.3155

Table 4: Comparison of three refinement algorithms in terms of description errors

5 Discussion

We presented a new method, PICP algorithm, for registering the range images. It is used to refine the initial approximation of the transformation between range images that were previously segmented. PICP's performance is good in terms of accuracy and required computational time although required preprocessing in terms of segmentation can be time consuming. But in tasks like reverse engineering, segmentation into object surfaces is a common step, so it may not be appropriate to think of segmentation as a preprocessing step for PICP algorithm. PICP has some drawbacks as well. The most significant is, that it cannot be used directly for objects with arbitrary free-form surfaces since our approach requires a segmentation into appropriate models which is currently limited to planes, spheres, cylinders, cones, and tori. One could use our PICP approach with arbitrary free-form surfaces for projection of points onto local models in a similar way as in [3]. Our future work will be directed towards including other types of models and automatic creation of CAD models out of segmented and registered range data. We are already able to create CAD models out of simple objects with planar,

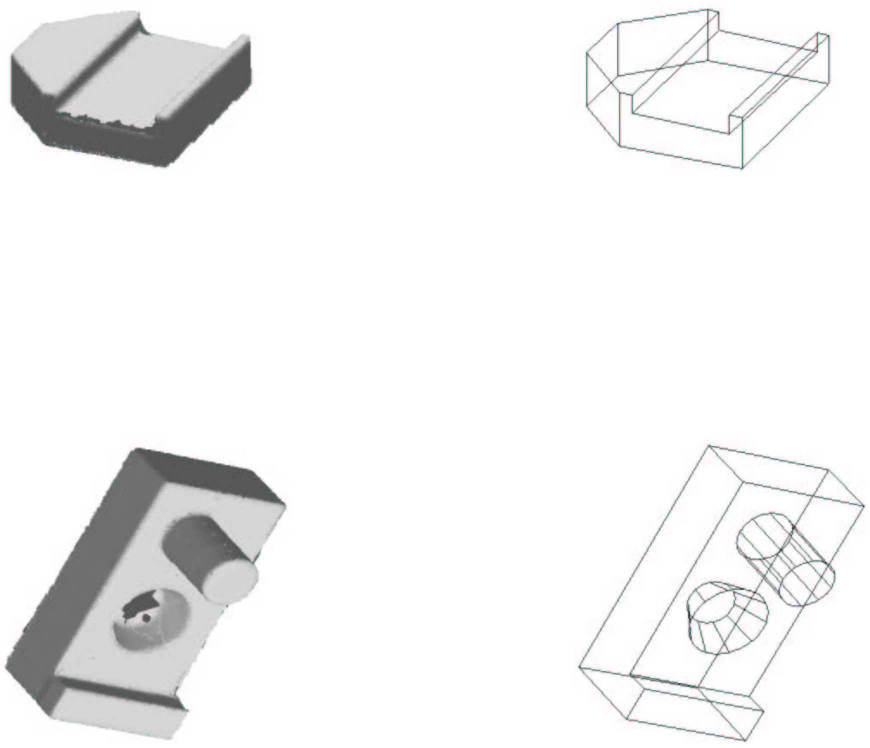

Figure 9: Generated intensity images of merged range data obtained from several views on the left and corresponding boundary representations of CAD model on the right

cylindrical and conical surfaces like the one used in our experimental results, but we want to extend this approach to other types.

A Projections of points onto the geometric parametric models

In this section we present geometric parametric models used in the Recover-and-Select based segmentation stage. Also, we show how to compute the coordinates of a point that has been projected onto a particular model. Our method requires that we have to be able to compute a projection of a point onto each model type used.

A.1 Projection of a point onto a plane

Planes are represented by parameters a, b, c, d and consist of points $\mathbf{t} = (x, y, z)$ that satisfy equation $ax + by + cz + d = 0$. The projection ($\mathbf{t_{proj}}$) of a point onto a plane can be computed as shown in Eq. 8.

$$\begin{aligned} \mathbf{t_{proj}} &= \mathbf{t} - d_1\mathbf{n}, \qquad (8) \\ \text{where } d_1 &= \mathbf{nt} + d, \\ \mathbf{n} &= (a, b, c) \qquad \text{and} \\ a^2 + b^2 + c^2 &= 1. \end{aligned}$$

A.2 Projection of a point onto a sphere

Spheres are represented by parameters a, b, c, r and consist of points $\mathbf{t} = (x, y, z)$ that satisfy equation $(x - a)^2 + (y - b)^2 + (z - c)^2 - r^2 = 0$. The projection ($\mathbf{t_{proj}}$) of a point onto a sphere can be computed as shown in Eq. 9.

$$\begin{aligned} \mathbf{t_{proj}} &= \mathbf{t} - d_2(\mathbf{t} - \mathbf{c}), \qquad (9) \\ \text{where } d_2 &= \frac{d_1 - r}{d_1}, \\ \mathbf{c} &= (a, b, c) \qquad \text{and} \\ d_2 &= ||\mathbf{t} - \mathbf{c}|| \,. \end{aligned}$$

A.3 Projection of a point onto a cylinder

Cylinders are represented by parameters $\rho, \phi, \theta, \alpha, k$ [12]. Parametrisation of a cylinder is shown in Fig. 10. Values ϕ and θ are used to compute vector $\mathbf{n}$ ($\mathbf{n} = (\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta)$), which is parallel to the normal vector of a tangent plane to a cylinder surface and orthogonal to a vector on the cylinder axis ($\mathbf{a}$). The projection ($\mathbf{t_{proj}}$) of a point to a cylinder can be computed as shown in Eq. 10.

$$\mathbf{t_{proj}} = \mathbf{t} - (\|\mathbf{t} - \mathbf{v}\| - 1/k)\mathbf{r}, \qquad (10)$$

$$
\begin{aligned}
\text{where} \quad \mathbf{r} &= \frac{\mathbf{t}-\mathbf{v}}{\|\mathbf{t}-\mathbf{v}\|} \qquad \text{and} \\
\mathbf{v} &= (\rho + 1/k)\mathbf{n} + ((\mathbf{t} - (\rho + 1/k)\mathbf{n}) \cdot \mathbf{a})\mathbf{a}.
\end{aligned}
$$

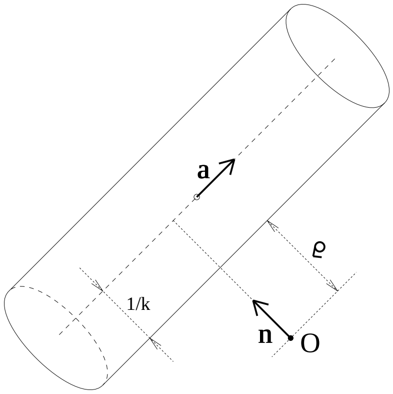

Figure 10: Parametrisation of a cylinder

A.4 Projection of a point onto a cone

Parametrisation of a cone is similar to the parametrisation of a cylinder. A cone has parameters $\rho, \phi, \theta, \sigma, \tau, k$ [12]. The parametrisation of a cone is shown in Fig. 11. The difference is that here vectors $\mathbf{n}$ and $\mathbf{a}$ are not necessarily orthogonal. Vector $\mathbf{n}$ is computed in the same way as for the cylinder, while $\mathbf{a}$ has components $(\cos\sigma\sin\tau, \sin\sigma\sin\tau, \cos\tau)$[12]. The projection ($\mathbf{t_{proj}}$) of a point to a cone can be computed as shown in Eq. 11.

$$
\begin{aligned}
\mathbf{t_{proj}} &= \mathbf{t} - (\|\mathbf{c} - z\mathbf{a}\| - d)\mathbf{r} \qquad (11) \\
\text{where} \quad \mathbf{c} &= \mathbf{t} - (\rho + 1/k)\mathbf{n}, \\
z &= \mathbf{a} \cdot \mathbf{c}, \\
\alpha &= \arccos(\mathbf{a} \cdot \mathbf{n}), \\
d &= \frac{1}{k \sin\alpha} - z \tan\alpha \quad \text{and} \\
\mathbf{r} &= \frac{\mathbf{c} - z\mathbf{a}}{\|\mathbf{c} - z\mathbf{a}\|.}
\end{aligned}
$$

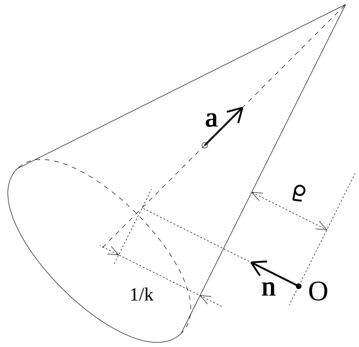

Figure 11: Parametrisation of a cone

A.5 Projection of a point onto a torus

A torus has parameters $\rho, \phi, \theta, \sigma, \tau, k, s$ [12]. Parameters $\phi, \theta, \sigma, \tau$ denote vectors $\mathbf{n}$ and $\mathbf{a}$, while both radii of a torus can be computed with $\frac{1}{k}$ and $\frac{1}{s}$. Parametrisation of a torus is shown on a torus cross-section in Fig. 12. The projection ($\mathbf{t_{proj}}$) of a point to a torus can be computed as shown in Eq. 12.

$$\mathbf{t_{proj}} = \mathbf{t} - (\|\mathbf{t} - \mathbf{c} - \mathbf{v}\| - 1/k)\mathbf{r}, \tag{12}$$

$$\begin{aligned}
\text{where } \mathbf{c} &= (\rho + 1/s)\mathbf{n} - |1/s - 1/k| \cos(\mathbf{a} \cdot \mathbf{n})\mathbf{a}, \\
\mathbf{p} &= \frac{(\mathbf{a} \times (\mathbf{t} - \mathbf{c})) \times \mathbf{a}}{\|(\mathbf{a} \times (\mathbf{t} - \mathbf{c})) \times \mathbf{a}\|}, \\
\mathbf{v} &= |1/s - 1/k| \sin(\mathbf{a} \cdot \mathbf{n})\mathbf{p} \quad \text{and} \\
\mathbf{r} &= \frac{\mathbf{t} - \mathbf{c} - \mathbf{v}}{\|\mathbf{t} - \mathbf{c} - \mathbf{v}\|}.
\end{aligned}$$

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on PAMI*, 14(2):239–256, February 1992.

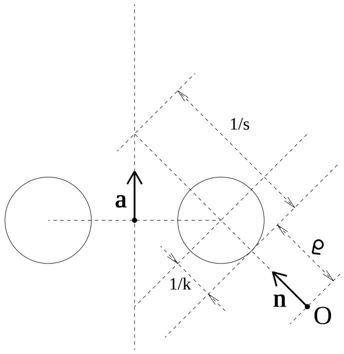

Figure 12: Parametrisation of a torus

[2] G. Blais and M. D. Levine. Registering multiview range data to create 3D computer objects. *IEEE Transactions on PAMI*, 17:820–824, 1995.

[3] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10:145–155, 1992.

[4] C. Dorai, J. Weng, and A. K. Jain. Optimal registration of multiple range views. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, pages 569–571, 1994.

[5] D. Eggert, A. W. Fitzgibbon, and R. B. Fisher. Simultaneous registration of multiple range views for use in reverse engineering of CAD models. *Computer Vision and Image Understanding*, 69(3):253–272, 1998.

[6] J. Feldmar and N. Ayache. Rigid and affine registration of smooth surfaces using differential properties. In *Proceedings of the 3rd European Conference on Computer Vision*, pages 397–406, 1994.

[7] J. Feldmar, G. Malandain, J. Declerck, and N.Ayache. Extension of the ICP algorithm to non-rigid intensity-based registration of 3D volumes. In *Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 84–93, 1996.

[8] A. Jaklič. *Construction of CAD models from range images.* PhD thesis, Faculty for Computer and Information Science, University of Ljubljana, 1997.

[9] B. Kverh. Segmentation and registration of range images as a tool for modelling objects. MSc. thesis, Faculty for Computer and Information Science, University of Ljubljana, 1998.

[10] B. Kverh and A. Leonardis. Using recover-and-select paradigm for simultaneous recovery of planes, second order surfaces and spheres from triangulated data. In *Proceedings of the Computer Vision Winter Workshop CVWW'98*, pages 26–37, February 1998.

[11] A. Leonardis, A. Gupta, and R. Bajcsy. Segmentation of range images as the search for geometric parametric models. *International Journal of Computer Vision*, 14:253–277, 1995.

[12] G. Lukács, A. D. Marshall, and R. R. Martin. Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. In *ECCV98 - Proceedings of the 5th European Conference on Computer Vision*, volume I, pages 671–686, June 1998.

[13] A. D. Marshall and R. R. Martin. *Computer Vision, Models and Inspection*, volume 4 of *World Scientific Series in Robotics and Automated Systems.* World Scientific, September 1991.

[14] T. Masuda, K. Sakaue, and N. Yokoya. Registration and integration of multiple range images for 3-D model construction. In *Proceedings of the 13th International Conference on Pattern Recognition*, pages 879–883, August 1996.

[15] T. Pajdla, V. Hlaváč, and D. Večerka. Semi-automatic building of complete 3D models from range images. In R. R. Martin and T. Varady, editors, *RECCAD Deliverable Documents 2 and 3 Copernicus Project No. 1068 Part II, Report on Basic Geometry and Geometry Model Creation with further contributions on Data Acquisition and advanced material on Merging and Applications*, GML 1997/5, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1997.

[16] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13:119–152, 1994.

Dr. Bojan Kverh and Prof. Dr. Aleš Leonardis
Faculty of Computer and Information Science
University of Ljubljana
Tržaška 25, Ljubljana, Slovenia.
e-mail:{bojank,alesl}@fri.uni-lj.si