ANALYSIS OF TWO TRIANGLE-BASED MULTI-SURFACE REGISTRATION ALGORITHMS OF IRREGULAR POINT CLOUDS

M. Al-Durgham [a,*], I. Detchev [b], A. Habib [b]

[a] Dept. of Civil Engineering, University of Toronto, 35 St. George Street, Toronto, ON, M5S 1A4, Canada - mohannad.al.durgham@utoronto.ca
[b] Dept. of Geomatics Engineering, University of Calgary, 2500 University Drive, Calgary, AB, T2N 1N4, Canada - (i.detchev, ahabib)@ucalgary.ca

Commission V, WG V/3

KEY WORDS: LiDAR, Surface Registration, Iterative Closest Patch (ICPatch), Iterative Closest Projected Point (ICPP)

ABSTRACT:

The registration of multiple surface point clouds into a common reference frame is a well addressed topic, and the Iterative Closest Point (ICP) is - perhaps - the most used method when registering laser scans due to their irregular nature. In this paper, we examine the proposed Iterative Closest Projected Point (ICPP) algorithm for the simultaneous registration of multiple point clouds. First, a point to triangular patch (i.e. closest three points) match is established by checking if the point falls within the triangular dipyramid, which has the three triangular patch points as a base and a user-chosen normal distance as the height to establish the two peaks. Then, the point is projected onto the patch surface, and its projection is then used as a match for the original point. It is also shown through empirical experimentation that the Delaunay triangles are not a requirement for establishing matches. In fact, Delaunay triangles in some scenarios may force blunders into the final solution, while using the closest three points leads to avoiding some undesired erroneous points. In addition, we review the algorithm by which the ICPP is inspired, namely, the Iterative Closest Patch (ICPatch); where conjugate point-patch pairs are extracted in the overlapping surface areas, and the transformation parameters between all neighbouring surfaces are estimated in a pairwise manner. Then, using the conjugate point-patch pairs, and applying the transformation parameters from the pairwise registration as initial approximations, the final surface transformation parameters are solved for simultaneously. Finally, we evaluate the assumptions made and examine the performance of the new algorithm against the ICPatch.

1. INTRODUCTION

There currently exist many ICP variants, which have various target functions and objectives. Rusinkiewicz & Levoy (2001), Bae & Lichti (2008), and Besl & McKay (1992) are a few examples of these variants. ICP like matching algorithms vary in the way their primitives are defined. The basic matching algorithms are mainly comprised of a point-to-point matching procedure; this method is sometimes desired due to the lack of pre-processing steps, the high convergence rate, and the speed of the algorithm. An example of such level could be found in the earliest work shown in Besl & McKay (1992). However, note that due to the false underlying assumption of point-to-point correspondence in the case of irregular point clouds, the final transformations may be slightly biased (Shan & Toth, 2008).

Another representation of primitives is found where one point cloud is maintained in its original shape, while the other point set is converted into a higher order primitive (e.g., triangles, planes, surface normals, and higher order surfaces). In this representation, the mathematical model used is more complex and the expected execution time is longer; however, the expected surface registration accuracy is higher. Beinat et al. (2006), and Boström et al. (2008) are two examples of this type of primitive representation.

An even more complex representation is to use object primitives. An example for the use of such primitives is found in Rabbani et al. (2007). In their work, point clouds are first segmented in order to extract useful objects (e.g., cylinders). Afterwards, a model fitting step takes place, followed by a correspondence specification step. Then, a check is performed to ensure that the information collected so far provide sufficient constraints. Finally, the sum of the squares of the orthogonal distances between the points and the extracted object are minimized, and the optimal registration is achieved.

In the aforementioned brief literature review, the authors do not attempt to review all the existing matching algorithms, but to shed light on the three levels of primitive variation: (1) point-to-point, (2) point-to-surface, and (3) point-to-object primitives. In the work presented here, both, the ICPatch and the ICPP belong to the second category of algorithms. However, the ICPP could also be seen as a first category algorithm due to the introduction of artificial projected points used for the registration as will be seen in the methodology section.

The focus of this paper is to present and compare two triangular patch based registration methods, namely - the Iterative Closest Patch (ICPatch) and the Iterative Closest Projected Point (ICPP). In the previous work shown in Habib et al. (2010), the authors introduced the ICPatch method for the registration of multiple point clouds into a common reference frame. The ICPatch method could be summarized in two main steps. First, a pairwise registration is run between every two overlapping point clouds in order to solve for their relative transformation parameters. Note that for each point cloud pair a Delaunay triangulation must be performed over one of the two point

[*] Corresponding author.

clouds, while the other one is kept in its original format. All possible point-to-closest triangular patch pairs are identified within the dataset. Second, the point-to-patch matches together with the relative transformation parameters from the pairwise registration step are used to compute the global transformation parameters of each point cloud, and thus perform the multi-surface registration process in a network mode.

Inhere, the ICPatch method is briefly reviewed. Then, the new one-step (multi-surface) registration ICPP algorithm is presented, starting with the underlying assumptions and the mathematical model. Next, a sample dataset is used to evaluate the performance of the aforementioned methods. The paper is then concluded with final remarks and a discussion of future work.

2. METHODOLOGY

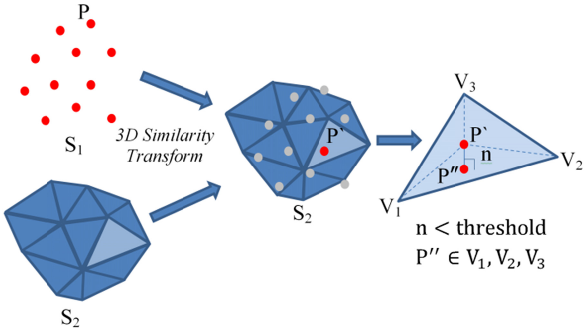
In this section, we first start by quickly reviewing the ICPatch method, then, we present the new ICPP method. The motivation behind both methods is the argument that a point-to-point correspondence between two or more irregular point clouds is an invalid assumption. The user simply cannot identify common points among the point sets. Therefore, one must consider higher order primitives to compensate for the discrete nature of the data. Triangles are considered since they are the most primitive areal features.

2.1 The Iterative Closest Patch

This algorithm has two main steps (for comprehensive details, please refer to Habib et al. (2010)). First, a pairwise registration is run between every two overlapping point clouds in order to solve for their relative transformation parameters. Second, the point-to-patch matches together with the relative transformation parameters from the pairwise registration step are used to compute the global transformation parameters of each point cloud; the following subsections elaborate more on these two steps.

2.1.1 Pairwise surface registration

Given two overlapping pairs, a Delaunay triangulation is performed over one point set $S_2 = (\boldsymbol{X}_2, \boldsymbol{Y}_2, \boldsymbol{Z}_2)$ to establish the triangle primitives, while maintaining the other point cloud $S_1 = (\boldsymbol{X}_1, \boldsymbol{Y}_1, \boldsymbol{Z}_1)$ in its original form. The point $\boldsymbol{p}$ in S_1 and a triangle (V_1, V_2, V_3) in S_2 are transformed into a common frame as shown in Figure 1. It is important to note that the Delaunay based Triangulated Irregular Network (TIN) patches are an acceptable primitive only in the cases when the TIN model represents the true physical surface of the reconstructed object or the scene of interest (Habib et al., 2010). This means that the surface models have to have point density high enough that no triangles are built across what would be a breakline. If this is true, then it can be assumed that point-to-patch correspondence between the overlapping surface models does exist (see Figure 1). In order to deal with cases where the TIN does not represent the physical surface (e.g. sparse areas in the point clouds), a threshold is implemented in the matching strategy to prevent undesired triangles from being used (thus resulting into a triangular-prism looking search space). So, if it is safe to assume that point $\boldsymbol{p}$ in surface one (S_1) corresponds to the triangular patch with vertices V_1, V_2, and V_3 in surface two (S_2), then this point should coincide with the patch after applying the proper initial transformation parameters. The transformed point $\boldsymbol{p}'$ and the triangle vertices comprise a tetrahedron volume. The target function is to minimize this tetrahedron volume. An alternative way to establish the similarity measure for the point-patch pair is achieved by modifying the weight matrix in the least squares adjustment. Basically, the weight matrix is relaxed in the direction parallel to the patch surface. Hence, any vertex of the patch could be used as a conjugate feature in the matching process. Refer to Habib et al. (2010) for more details about the weight constraints.



(a) (b)

Figure 1: Surface model representation (a), and criteria for accepting correspondence between conjugate primitives (b) (Habib et al., 2010)

2.1.2 Multi-surface registration

One way of achieving multi-surface registration is to choose the reference frame of one of the surface models as the common one, and then to sequentially register the rest of the surface models in the pairwise manner described before. However, the first and the last surface models might exhibit incompatibility due to errors propagated through the sequential registration process. This will be similar to a closed-loop traverse in surveying, where the constraint that the first and last point coincide, has not been used. To avoid such an incompatibility, the multiple-surface registration has to be performed simultaneously, i.e. in a network mode. This procedure can be viewed as an extension to the pairwise registration. In particular, the pairwise registration procedure is used to accumulate a list of the corresponding point-patch pairs and estimate the transformation parameters between any two overlapping surface models. The multiple-surface registration, on the other hand, uses these corresponding point-patch pairs, and applies the transformation parameters from the pairwise surface registration as initial approximates, to simultaneously solve for all the final surface transformation parameters. This is done in a least-squares adjustment, where each surface is iteratively transformed to a common reference frame until the sum of the squared normal distances between the conjugate point-patch pairs is minimized. This procedure is highly nonlinear, so that is why the initial approximates from the pairwise surface registration are necessary. It is also important to note that the transformation parameters for one of the surface models are kept fixed in order to define the datum for the final surface model.

2.2 The Iterative Closest Projected Point

Let us first consider a pairwise registration scenario, where the second point cloud $S_2 = (\boldsymbol{X}_2, \boldsymbol{Y}_2, \boldsymbol{Z}_2)$ is transformed to the reference frame of the first point cloud $S_1 = (\boldsymbol{X}_1, \boldsymbol{Y}_1, \boldsymbol{Z}_1)$. In this case, the transformation mathematical model takes the following form:

$$S_1 = \boldsymbol{T}_2^1 + \boldsymbol{R}_2^1 S_2 \tag{1}$$

where $\boldsymbol{T}_2^1$, is the translation vector between S_2 and S_1
$\boldsymbol{R}_2^1$, is the rotation matrix between S_2 and S_1

and when registering S_1 and S_2 onto a predefined reference, the relationship takes the following shape:

$$\boldsymbol{T}_1^r + \boldsymbol{R}_1^r S_1 = \boldsymbol{T}_2^r + \boldsymbol{R}_2^r S_2 \qquad (2)$$

where $\boldsymbol{T}_1^r$, is the translation vector between S_1 and a predefined reference frame
$\boldsymbol{R}_1^r$, is the rotation matrix between S_1 and a predefined reference frame
$\boldsymbol{T}_2^r$, is the translation vector between S_2 and a predefined reference frame
$\boldsymbol{R}_2^r$, is the rotation matrix between S_2 and a predefined reference frame

This formulation could be rearranged by transforming all the elements into the right side of the equation:

$$0 = \boldsymbol{T}_2^r + \boldsymbol{R}_2^r S_2 - (\boldsymbol{T}_1^r + \boldsymbol{R}_1^r S_1) \qquad (3)$$

Equation (3), establishes the mathematical relationship between two conjugate points in the aforementioned point clouds. Before searching for a match between the two clouds, a point $(\boldsymbol{p}_0)$ in S_1 must be transformed to the point cloud S_2 reference frame using the initial approximations provided:

$$\boldsymbol{p}_t = \boldsymbol{R}_2^{r\prime}(\boldsymbol{T}_1^r - \boldsymbol{T}_2^r) + \boldsymbol{R}_2^{r\prime}\boldsymbol{R}_1^r\boldsymbol{p}_0 \qquad (4)$$

where $\boldsymbol{R}_2^{r\prime}$ is the transpose of $\boldsymbol{R}_2^r$
$\boldsymbol{p}_t$ is $\boldsymbol{p}_0$ transformed to the S_2 reference frame.

Next, one finds the closest three points $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3)$ in S_2 to $\boldsymbol{p}_t$, these three points establish the triangular base of a tetrahedron. Note that these three points do not necessarily construct a Delaunay triangle. In fact, avoiding the Delaunay triangles can be beneficial as shown in Figure 2. For example, blunders can result in undesired triangles in the dataset as shown in Figure 2 (b), thus leading to incorrect point-to-patch matches in some cases. On the other hand, using the nearest neighbour for finding the closest three points completely ignores erroneous points falling beyond a certain threshold as shown Figure 2(a).

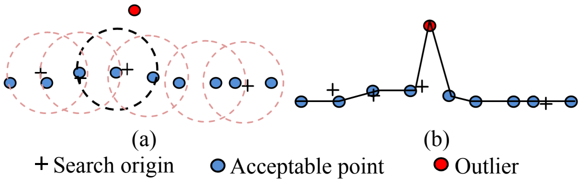


Figure 2: Side view example of a blunder ignored when using nearest neighbours (a) and included when using Delaunay triangles (b) for the same point set

A tetrahedron search space is then established to examine the validity of the matched pair $(\boldsymbol{p}_t\,\&\,\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3)$. Given a search space threshold with a value of N, one can extrude the centroid $(\boldsymbol{p}_c)$ of $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3)$ by N to point $\boldsymbol{p}_4$ using equation (5), thus completing a tetrahedron. In fact, note that there are two possible directions for the extrusion of centroid $\boldsymbol{p}_c$, thus resulting in two tetrahedrons $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4)$, and $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_5)$, or a triangular dipyramid. Since point $\boldsymbol{p}_t$ could fall in either tetrahedron, the matching condition becomes $\boldsymbol{p}_t \in \mathrm{V}(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4, \boldsymbol{p}_5)$; where V(...) is the convex volume of the shape specified by the given vertices. Figure 3 demonstrates one of the resulting tetrahedrons. The set $(\boldsymbol{p}_t\,\&\,\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3)$ is accepted if $\boldsymbol{p}_t \in \mathrm{V}\,(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4)$ or $\boldsymbol{p}_t \in$ dipyramid.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{p4\ or\ p5} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{pc} \pm \frac{N}{\sqrt{a^2+b^2+c^2}} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \qquad (5)$$

where x, y, z are the Cartesian coordinates of a point
a, b, c are the plane parameters derived from $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3)$

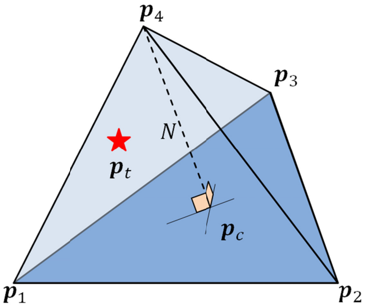


Figure 3: The resulting tetrahedron $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4)$, and the matched point $\boldsymbol{p}_t$

The underlying principle behind using tetrahedrons as opposed to using constant normal distance is to mimic a realistic interpolation scenario. In other words, the closer the matched points to the vertices, the smaller the matching threshold should be. Figure 4(a) shows a 2D cartoon of a constant search space, while Figure 4(b) is a 2D representation of the modified search space proposed.

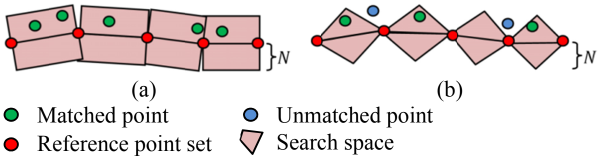


Figure 4: A constant search space (a) and the revised search space (b)

Using the point $\boldsymbol{p}_t$, the tetrahedron shown in Figure 3 could be split into four different tetrahedrons as shown in Figure 5; namely - $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_t)$, $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_4, \boldsymbol{p}_t)$, $(\boldsymbol{p}_1, \boldsymbol{p}_3, \boldsymbol{p}_4, \boldsymbol{p}_t)$, and $(\boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4, \boldsymbol{p}_t)$.

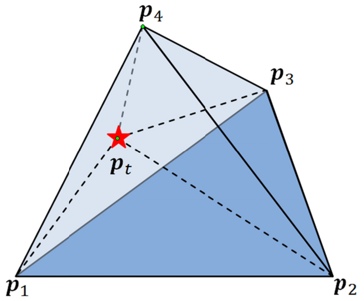


Figure 5: Four tetrahedrons resulting from splitting the original volume with $\boldsymbol{p}_t$

Next, one could evaluate if $\boldsymbol{p}_t$ falls inside the tetrahedron by examining the signs of the determinants resulting from each tetrahedron as suggested in Eq. (6). The signs of determinants in Eq. (7) indicate the side (i.e., negative or positive) at which the point $\boldsymbol{p}_t$ fall with respect to the tetrahedron faces. The point $\boldsymbol{p}_t$ is inside the tetrahedron only if it falls on the same side (having the same determinant sign) of the tetrahedron faces (Schneider & Eberly, 2002).

$$\begin{aligned} &if \begin{pmatrix} sign(D_1) = sign(D_2) = sign(D_3) = \\ sign(D_4) = sign(D_5) \end{pmatrix} \\ &\qquad \boldsymbol{p}_t \; is\ inside\ tetrahedron \\ &else \\ &\qquad \boldsymbol{p}_t \; is\ outside\ tetrahedron \end{aligned} \tag{6}$$

where, D_1 to D_5 are the following determinants:

$$D_1 = \begin{vmatrix} x_{p1} & y_{p1} & z_{p1} & 1 \\ x_{p2} & y_{p2} & z_{p2} & 1 \\ x_{p3} & y_{p3} & z_{p3} & 1 \\ x_{p4} & y_{p4} & z_{p4} & 1 \end{vmatrix}, \quad D_4 = \begin{vmatrix} x_{p1} & y_{p1} & z_{p1} & 1 \\ x_{p2} & y_{p2} & z_{p2} & 1 \\ x_{pt} & y_{pt} & z_{pt} & 1 \\ x_{p4} & y_{p4} & z_{p4} & 1 \end{vmatrix}$$

$$D_2 = \begin{vmatrix} x_{pt} & y_{pt} & z_{pt} & 1 \\ x_{p2} & y_{p2} & z_{p2} & 1 \\ x_{p3} & y_{p3} & z_{p3} & 1 \\ x_{p4} & y_{p4} & z_{p4} & 1 \end{vmatrix}, \quad D_5 = \begin{vmatrix} x_{p1} & y_{p1} & z_{p1} & 1 \\ x_{p2} & y_{p2} & z_{p2} & 1 \\ x_{p3} & y_{p3} & z_{p3} & 1 \\ x_{pt} & y_{pt} & z_{pt} & 1 \end{vmatrix} \tag{7}$$

$$D_3 = \begin{vmatrix} x_{p1} & y_{p1} & z_{p1} & 1 \\ x_{pt} & y_{pt} & z_{pt} & 1 \\ x_{p3} & y_{p3} & z_{p3} & 1 \\ x_{p4} & y_{p4} & z_{p4} & 1 \end{vmatrix}$$

The projection of $\boldsymbol{p}_t$ into the plane described by the parameters (a, b, c, d) $\boldsymbol{p}_p$ is given by:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\boldsymbol{p}_p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\boldsymbol{p}_t} - \frac{a x_{p_t} + b y_{p_t} + c z_{p_t} + d}{a^2 + b^2 + c^2} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{8}$$

where $\quad a, b, c, d$ are the plane parameters derived from $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3)$

Finally, the point $\boldsymbol{p}_t$ and its projection $\boldsymbol{p}_p$ constitute an accurate point pair between the two irregular point clouds. Similarly, in the case where more than two surfaces exist, one could establish a collection of points and their matching projections. Then, the conventional ICP algorithm could be used to register the point clouds into a common reference frame. Note that all the possible matches between the point clouds are considered and used at once to solve for the transformation parameters. Equation (9) gives the basis of our problem formulation. The left side of the equation is the observations, while the right side is used to construct the Jacobian matrix with respect to the unknowns. Note that, both, Horn's method using quaternions and nonlinear least squares with Euler angles could be used using this formulation (Vosselman & Maas, 2010). However, one should be aware of possible point cloud rotations that might lead to gimbal lock scenarios (Shoemake, 1985) when using Euler rotations.

$$0 = \boldsymbol{T}_2^r + \boldsymbol{R}_2^r \boldsymbol{p}_p - (\boldsymbol{T}_1^r + \boldsymbol{R}_1^r \boldsymbol{p}_0) \tag{9}$$

Due to the strict condition proposed in this algorithm (i.e., the dipyramid search space), one could expect a lower convergence rate of the ICPP. Thus, the following steps are suggested to speed up the alignment of the point clouds:

1. Random sampling of one of the datasets (using 10% of the points) is used for two purposes: (a) to speed up each individual iteration, and (b) to reduce the chances of converging into local minima.

2. A conventional ICP method is performed in the first few iterations with relaxed parameters.

3. The ICPP is performed using all the points in the last a few iterations to fine tune the transformation parameters into the final solution.

3. DATA USED

The ICPP method for performing multi-surface registration was tested using a dataset of a scoliotic torso mannequin. The approximate dimensions of the torso mannequin were: 60 cm, 40 cm, and 25 cm in height, width, and depth. The dataset consisted of four point clouds modelling four partial areas of the torso surface (i.e. the front, the back, and the two sides). The point clouds were derived using photogrammetric means with a system having multiple cameras and projectors (Detchev, 2010). Figure 6 shows the torso mannequin during the data collection process. The pattern projected was used to create artificial texture in order to allow for the automatic matching process. The resulting X, Y, Z coordinates for the four point clouds had sub-millimetre precision (approximately 0.3 mm) and the average overlap between the neighbouring clouds was roughly 30%. Table 1 shows the number of points generated for each cloud, together with the required initial approximations for the global transformation parameters. The initial approximations were derived by roughly evaluating the position and orientation of the central camera used in the point cloud generation. Note that S_1 was used as the reference point set, so its translational and rotational parameters are zeros.

Point set number	No. of points	X_0 (mm)	Y_0 (mm)	Z_0 (mm)	ω (°)	ϕ (°)	κ (°)
S_1 (ref.)	20,915	0	0	0	0	0	0
S_2	17,222	0	300	-300	90	0	0
S_3	21,878	0	600	0	180	0	0
S_4	17,833	0	300	300	270	0	0

Table 1: Transformation parameter approximations used for the initial point cloud alignment

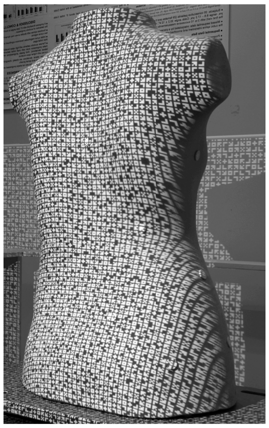
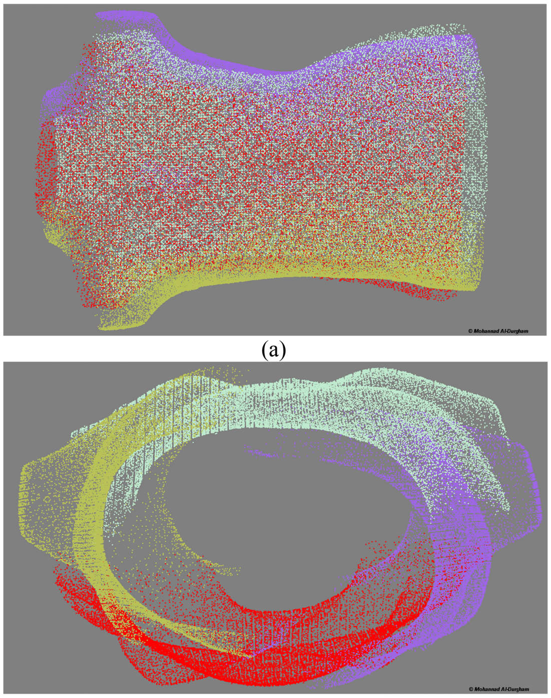


Figure 6: The torso model used (Detchev, 2010).

4. RESULTS AND DISCUSSION

In this section, the ICPP and the ICPatch algorithms were applied over the dataset described using the initial approximations provided. The following graph shows top and side views of the torso model before the adjustment. The initial approximations used provide minimal overlap for the matching process to start successfully.



(a)

(b)

Figure 7: Side (a) and top (b) views of the Torso point clouds before performing the ICPP

Table 2(a) and Table 2(b) show the final transformation parameters resulting from the ICPP and the ICPatch methods. The transformation parameters are comparable, with maximum discrepancy of up to 0.24 mm in the translation vectors, and up to 3' of angular variation.

	X_0 (mm)	Y_0 (mm)	Z_0 (mm)	ω (°)	ϕ (°)	κ (°)
S_2	-26.42	291.44	-265.04	89.846	-2.830	-5.560
	±0.009	±0.020	±0.016	±0.0046	±0.0017	±0.0017
S_3	-11.07	557.08	26.59	180.062	-8.365	-2.670
	±0.012	±0.019	±0.023	±0.0034	±0.0023	±0.0017
S_4	15.12	267.57	290.58	270.067	-5.526	2.858
	±0.014	±0.035	±0.015	±0.0046	±0.0017	±0.0017

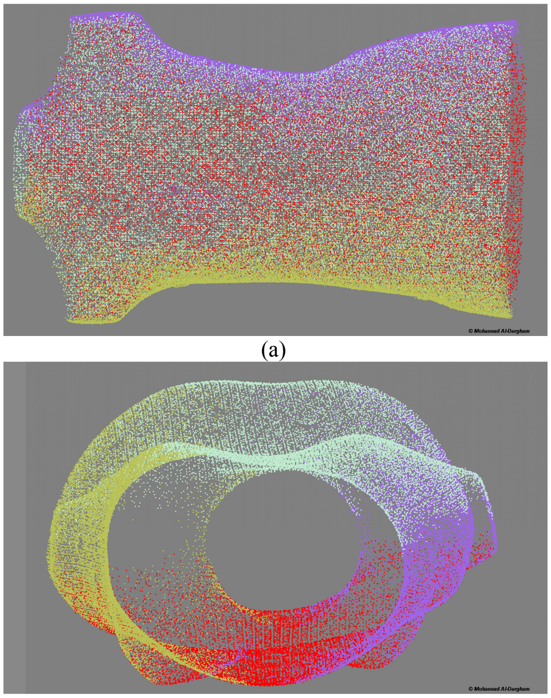
(a)

	X_0 (mm)	Y_0 (mm)	Z_0 (mm)	ω (°)	ϕ (°)	κ (°)
S_2	-26.47	291.52	-265.08	89.858	-2.816	-5.543
	±0.027	±0.043	±0.024	±0.0070	±0.0040	±0.0030
S_3	-11.31	556.91	26.93	180.111	-8.362	-2.688
	±0.029	±0.031	±0.049	±0.0080	±0.0040	±0.0040
S_4	15.06	267.36	290.71	270.089	-5.544	2.876
	±0.024	±0.053	±0.028	±0.0090	±0.0030	±0.0030

(b)

Table 2: Final transformation parameters for the ICPP (a) and the ICPatch (b) algorithms

One should also emphasize that these changes are justified due to existing correlations between the translation vectors and the rotation angles. For example, the largest correlation of 86% in the ICPP is observed between Z_0 and k of S_2, and 90% between the above parameters in the ICPatch. In addition, note that the standard deviations of the estimated parameters in the ICPatch case are double the values of those in the ICPP. This could be attributed to the more relaxed search space in the ICPatch.



(a)

(b)

Figure 8: Side (a) and top (b) views of the Torso point clouds after performing the ICPP

A more practical approach to compare both methods is achieved by comparing the resulting adjusted point sets. The Root Mean Square Error (RMSE) of the coordinate differences between the transformed point clouds from both registration procedures is shown in Table 3. Also, side (a) and top (b) views of the torso point clouds after running the ICPP are shown in Figure 8. Both point sets coincide within 0.25 mm RMSE, which is within the accuracy range of both solutions.

	$RMSE\ x$ (mm)	$RMSE\ y$ (mm)	$RMSE\ z$ (mm)	$RMSE$ (mm)
S_2	0.147	0.091	0.128	0.215
S_3	0.156	0.106	0.135	0.232
S_4	0.131	0.056	0.054	0.152

Table 3: RMSE of the transformed point clouds from both registration procedures

A systematic trend can be observed when the two point sets are subtracted from each other. Basically, the coordinate differences are evaluated between a point in the ICPP results and its conjugate point in the ICPatch results. This systematic difference could be attributed to: (1) the difference in the target functions between the two methods, (2) the variance in the number of matches and their nature (i.e., 26,147 Delaunay triangles found in the ICPatch vs. 9,958 nearest neighbour triangles in the ICPP), and (3) the shape of the search space (i.e., triangular prism vs. a triangular dipyramid). These conclusions were also verified via simulations, where the same reported variances were observed and in all cases, they remained within the surface roughness range.

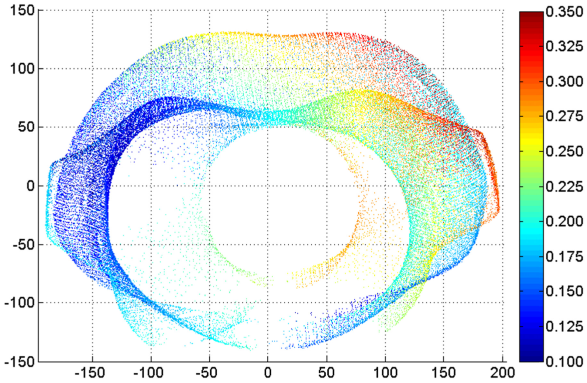


Figure 9: Top view of the torso, colour coded to reflect the coordinate differences (*mm*) between regenerated point clouds from both registration procedures

In Table 4, we present the main differences between the ICPatch and the ICPP algorithms in terms of the primitives used, the target function, the number of steps needed in a multi-surface registration scenario, and other properties.

Comparison	**ICPatch (coplanarity vs. modified weight)**	**ICPP**
Primitives	point to patch or point to point	point to point
Target function	minimize volume or normal distance	minimize normal distance
Registration	two step	one step
Matching space	triangular prism	triangular dipyramid
Triangles used	Delaunay	closest three points
Established matches	more matches (26,147 pairs)	less matches (9,958 pairs)
Convergence rate	fast, as a result of the pairwise step	slightly slower, due to dipyramid check

Table 4: Comparison between the ICPatch and the ICPP algorithms

5. CONCLUDING REMARKS

This paper presented a new approach for performing multi-surface registration of overlapping point clouds, namely, the Iterative Closest Projected Point (ICPP). In addition, the ICPP is compared against the ICPatch algorithm in terms of primitives, performance, number of matches established, etc. Both methods were found to agree within the range of the data noise (0.3*mm*). The main cause of variation is related to the triangle and search space definition. While the ICPatch uses Delaunay triangles, the ICPP uses the closest three points for the triangle definition. Although the closest three points yield less matches, it is favoured over Delaunay triangles as it reduces the number of possible mismatches, and eliminates the need for constructing Voronoi cells. In addition, a triangular dipyramid search space is used in the ICPP while the ICPatch uses triangular prism search space.

Future work will focus on evaluating the local point accuracy for automated estimation of the normal distance threshold. More datasets with varying overlap and point density will also be examined.

ACKNOWLEDGEMENT

The authors are grateful to Prof. Brenda McCabe, Department of Civil Engineering, University of Toronto for supporting this work. Also, sincere thanks go to the GEOIDE NCE network (project: PIV_SII72) and NSERC Discovery and Strategic Project Grants for their financial support of this research.

REFERENCES

Bae, K. & Lichti, D., 2008. A method for automated registration of unorganised point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1), pp.36-54.

Beinat, A., Crosilla, F. & Sepic, F., 2006. Automatic Morphological Prealignment and Global Hybrid Automatic Morphological Pre-Alignment and Global Hybrid. *ISPRS Journal of Photogrammetry and Remote Sensing*, 36(5). Online.

Besl, P. & McKay, N., 1992. A method for registration of 3-D shapes. *Institute of Elecrical and Electronics Engineers (IEEE) Transactions on Pattern Analysis and Machine Intelligence*, 10(3), pp.145-55.

Boström, G., Gonçalves, J. & Sequeira, V., 2008. Controlled 3D Data Fusion using Error-bounds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1), pp.55-67.

Detchev, I., 2010. *Implementation of a Close Range Photogrammetric System for 3D Reconstruction of a Scoliotic Torso*. Calgary: UCGE Report 20316.

Habib, A., Detchev, I. & Bang, K., 2010. A Comparative Analysis of Two Approaches for Multiple-Surface Registration of Irregular Point Clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information*, XXXVIII(Part 1).

Habib, A., Kersting, A., Bang, K.I. & Lee, D., 2010. Alternative Methodologies for the Internal Quality Control of Parallel LiDAR Strips. *IEEE Transactions on Geoscience and Remote Sensing*, 48(1), pp.221-36.

Rabbani, T., Dijkman, S., Heuvel, F. & Vosselman, G., 2007. An integrated approach for modelling and global registration of point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(6), pp.355-70.

Rusinkiewicz, S. & Levoy, M., 2001. Efficient Variants of the ICP Algorithm. In *3-D Digital Imaging and Modeling*., 2001.

Schneider, P. & Eberly, D., 2002. *Geometric Tools for Computer Graphics*. 1st ed. San Francisco: Morgan Kaufmann.

Shan, J. & Toth, C., 2008. *Topographic Laser Ranging and Scanning: Principles and Processing*. 1st ed. CRC Press.

Shoemake, K., 1985. Animating rotation with quaternion curves. *ACM SIGGraph*, 19(3), pp.245-54.

Vosselman, G. & Maas, H.-G., 2010. *Airborne and Terrestrial Laser Scanning*. 1st ed. CRC Press. pp. 83-85.