A Robust Registration Method using Huber ICP and Low Rank and Sparse Decomposition

Qiaochu Zhao[*], Xian-Hua Han[*], Yen-Wei Chen[*]
[*]Graduate School of Information Science and Engineering, Ritsumeikan University, Shiga, Japan
E-mail: gr0221ef@ed.ritsumei.ac.jp

***Abstract*—This paper proposes a robust registration and alignment framework for multiple point clouds using low rank and sparse decomposition. A coarse registration phase utilizing Huber-ICP is firstly performed to roughly align all the point clouds to a same location, and then sparse and low rank decomposition is applied to extract the low rank subspace of all the point clouds, which is expected to be outlier and loss data free. Finally, a fine registration procedure can be carried out between each point clouds from this low rank space to not only a more accurate registration result but also a more precise correspondence. Robustness of our method for outliers contained in point clouds is verified through manufactured data and it also shows that an effective result can still be achieved even when some points in the cloud are lost.**

***Keywords:* point cloud registration, sparse and low rank decomposition, Huber ICP**

I. INTRODUCTION

Point cloud registration and alignment are a necessary step for giving the required data to user when a same scene is taken at different times, from different viewpoints and/or by different sensors. This sort of problem has become an essential issue in fields such as robotic, computer vision, medical image processing and pattern recognition and so on.

Many approaches in literature have been proposed [1][2]. In [3] Singular Value Decomposition (SVD) is utilized to estimate transformation parameters under a given correspondence. Principal Component Analysis can also be used as a coarse registration method by simply aligning eigenvectors of two point clouds[4]. In[5], by considering point clouds as data generated by different Gaussian distributions, transformation parameters can be inferred by a standard EM procedure.

Among all of the proposed method, iterative closest point (ICP) [6] proposed by Berls in 1992, due to its advantage of simplicity and relatively fast performance[7], is most wildly used and plays as a basis role in other registration method. However, because of the jittering nature of an object or the mechanical property of sensors, noise or outliers can always present in real point clouds, in some occasions, even part of the point clouds could be lost for some reasons like occlusion. Therefore, the ICP algorithm may have a deteriorate performance in such application. To improve the performance of ICP, an extensive work has also been explored[7]. By adopting a more robust metric when estimating the correspondence, Huber ICP[9], point-to-plane ICP [9] and generalized ICP [10] are all shown to outperform the original one.

Specifically, Huber ICP, by using a robust kernel function, has been proven to be relatively efficient and have good performance even for sets of point clouds with outliers or noise. However, Huber ICP cannot give the accurate point correspondence for the outlier points and also is unable to restore the lost points in one set even other sets have the corresponding lost points. Therefore, this study proposes a robust registration and alignment framework by taking advantage of the robustness of Huber ICP as pre-process phase and following low rank and sparse decomposition for removing outliers and restoring lost data. With the proposed framework, the main advantages are two-folded: (1) Proper correspondences between clouds points can still be obtained even outliers are presented. (2) Point cloud with lost data can be, to some extent, recovered.

This paper is organized as follows. In Sec.2.1, a brief introduction of ICP will be given, while in Sec. 2.2, the Huber-ICP and its relation with conventional ICP will be clarified. A brief review of low rank and sparse decomposition will be presented in Sec. 2.3 and the proposed approach will be described in detail in Sec. 3. In Sec. 4 we will illustrate some results of our method and conclusion will be drawn in Sec. 5.

II. RELATED WORK

Given two (or more) point clouds obtained from different scenes or views, basically the aim of point cloud registration methods is trying to find out the most appropriate transformation that can achieve the minimum distance between corresponding points.

In this section, we will introduce several previous work related to the proposed method. Iterative Closest Point Algorithm as the most popular approach of point correspondence problem will be briefly reviewed firstly. Then its relative robust variant, Huber-ICP will be presented. Finally, some details and properties of low rank and sparse decomposition will be described.

A. Iterative Closest Point Algorithm [6]

Given two point clouds named reference and moving point clouds, ICP first estimates correspondence between two point clouds utilizing the nearest neighbor approach, which means assign the nearest point in moving point cloud to reference point cloud as correspondence point. Then, infer transformation parameters to minimize the Euclidean distance of correspondent point pairs. Some method such as SVD can be used as an efficient method in this phase. Next, apply the estimated parameters to transform the moving point cloud to get close to the location of the reference point cloud. Iteratively repeating these three steps and a refined registration can be finally achieved. Fig.1 illustrates the conception of ICP algorithm.

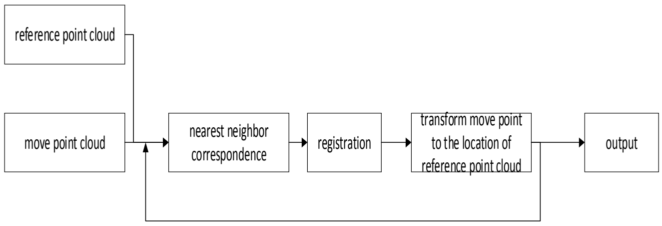


Fig.1. Flow of ICP algorithm

For its promised performance over variety data sets and relatively acceptable speed, ICP has become one of the most popular approaches of registration problem; however, since the original ICP does not distinguish between inliers and outliers, it also suffers a poor alignment [11]. Thereby, Huber ICP is proposed to improve its robustness.

B. *Huber ICP algorithm [8]*

Huber ICP follows the same procedure with the conventional ICP, which includes iteratively three steps: (1) nearest neighbor correspondence, (2) registration to estimate transformation parameters, (3) transform the moving point cloud to the location of the reference. In order to estimate transformation parameters in registration step, ICP minimize the summed Euclidean distance (i.e. L2 norm) of the correspondence points from the two point clouds, and it is well known that L2 norm is sensitive to outliers since the difference are squared. On the other hand, L1 norm is known to be robust towards outliers since it only takes the absolute sum of each residual. Thus, it will be ideal to take L1 norm to formulate the cost function, but its non-differentiable nature leads to an intricate optimization problem. As a compromise between L1 and L2 norm, a Huber loss function[12] was proposed for formulating the cost function of registration. The definition of Huber loss function is shown as follows:

$$f(n) = \begin{cases} n^2/2 & n < |k| \\ kn - k^2/2 & n \geq |k| \end{cases} \quad (1)$$

The basic idea is, when residual is small than a given threshold *k*, the function gives a quadratic cost, while the residual become larger which often associate with an outlier, the function behaves like a linear function to suppress the cost. Through this modification, outlier's contribution to the total cost can be limited. It is also notable that the Huber loss function maintained differentiable where *n* equals to *k*, which gives itself a fine property to many optimizers.

Andrew employed Levenberg-Marquardt (LM) [8] algorithm, which is a standard iterative non-linear optimization procedure, to optimize the cost function and it surprisingly turns out that the method do not have a significant loss of speed.

Here, we will give a brief reviewing of LM algorithm[13].

The basic idea of Levenberg-Marguardt algorithm is a combination of gradient descent method and Gauss-Newton iteration

Given a quadratic cost function:

$$f(x) = \sum_{i=1} r_i^2(x) \quad (2)$$

Where the residual of ith data point $r_i(x)$ is distance between its nearest point and itself.

Gradient descent method simply updates the position of x along the direction of f 's gradient.

$$x_{i+1} = x_i - \lambda \nabla f \quad (3)$$

Where ∇f represents the gradient of cost function f and λ controls the step size of optimization procedure.

It grantees that value of f will decreased while iteration, however, it suffers a different converge speed around optimal value.

On the other hand, Gauss-Newton method determined search direction according to Taylor expansion of gradient of f. Specifically, a Taylor series around the current position x_0 is

$$\nabla f(x) = \nabla f(x_0) + (x - x_0)^T \nabla^2 f(x_0) + h.o.t \quad (4)$$

Neglect the high order terms, set $\nabla f(x)$ to be 0 and solve the optimal *x* will give us

$$x_{i+1} = x_i - (\nabla^2 f(x_i))^{-1} \nabla f(x_i) \quad (5)$$

Where x_0 is substituted by x_i, and x is substituted by x_{i+1} to represent iteration.

Since the second order of $\nabla^2 f(x)$ is also considered, this method benefits an advantage of rapid convergence, while suffers sensitivity to initial location too.

Therefore, Levenberg combined advantages of two methods, giving the following update function:

$$x_{i+1} = x_i - (\nabla^2 f(x_i) + \lambda I)^{-1} \nabla f(x_i) \quad (6)$$

Marquardt further improved the update strategy by controlling the step-size λ according to second order of residual function $\nabla^2 f(x)$, which leads to the following function:

$$x_{i+1} = x_i - (\nabla^2 f(x_i) + \lambda diag(\nabla^2 f(x_i)))^{-1} \nabla f(x_i) \quad (7)$$

Eq (7) implies a large step in the direction with low curvature (i.e. an almost flat terrain) and a small in the direction with high curvature (i.e. a steep incline).

A comparison of conventional ICP and Huber ICP is illustrated in Fig.2. The red points are reference point cloud and blue point is point cloud containing outliers. For ease of explanation, here for simplicity, we just move some points up as the outliers. It can be seen that the performance of Huber ICP surpasses that of conventional ICP when outliers are contained in the point clouds.

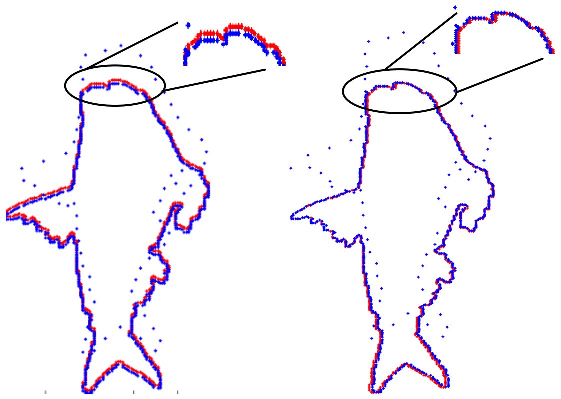


Fig.2 Comparison of the registration results by the conventional ICP (a) and Huber ICP (b).It can be seen that the performance of Huber ICP surpasses that of conventional ICP when outliers are contained in the point clouds.

C. Low Rank and Sparse decomposition

Low rank and sparse decomposition is a kind of matrix decomposition, which is able to decompose an observation matrix contained noise to an underlying low-rank matrix and a sparse error matrix [14].

Fig.3 illustrates the conception of low rank and sparse decomposition

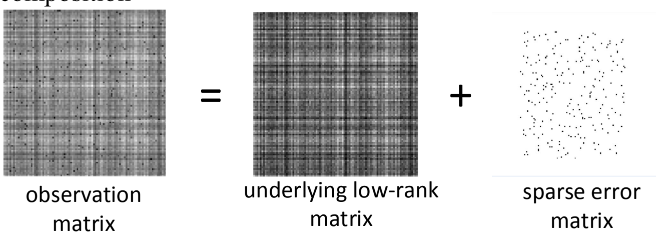


Fig. 3 Conception of low rank and sparse decomposition

Considering a matrix of corrupted observations X, straightforwardly low rank and sparse decomposition is to solve the following regularized rank minimization problem:

$$\min.\operatorname{rank}(\mathbf{D})+\lambda\,\|\mathbf{E}\|_0 \qquad \text{s.t. } \mathbf{X}=\mathbf{D}+\mathbf{E} \qquad (8)$$

Where D is the decomposed low rank matrix and E is the sparse one. $\operatorname{rank}(\cdot)$ as the name suggest is the rank of a matrix, and $\|\cdot\|_0$ indicates L0 norm, number of non-zero entries of a matrix. Since the non-differentiable property of rank function, solve this problem is intractable. A reasonable approximate solution is to optimize the following function(9).

$$\min.\|\mathbf{D}\|_*+\lambda\,\|\mathbf{E}\|_1 \qquad \text{s.t. } \mathbf{X}=\mathbf{D}+\mathbf{E} \qquad (9)$$

Where $\|\cdot\|_*$ signifies the nuclear norm (i.e. sum of eigen values) which can be viewed as an estimate of matrix's rank [15]; meanwhile L1 norm is also a good estimation of a matrix's sparsity [16].

The above problem is posed as a convex optimization problem, and can be solved by many standard approaches. Among all of them, the so-called "inexact augmented Lagrange multiplier (inexact ALM)" [17] also named of alternating direction method, for its efficiency, is most wildly used. It solves the primitive problem by solving the following conjugate Lagrange function:

$$L(\mathbf{A},\mathbf{E},\mathbf{Y},u)=\|\mathbf{A}\|_*+\langle\mathbf{Y},\mathbf{X}-\mathbf{A}-\mathbf{E}\rangle+\frac{u}{2}\|\mathbf{X}-\mathbf{A}-\mathbf{E}\|_F^2 \qquad (10)$$

Where **Y** and u are Lagrange multipliers.

Flow chart of this algorithm is illustrated in Fig.4.

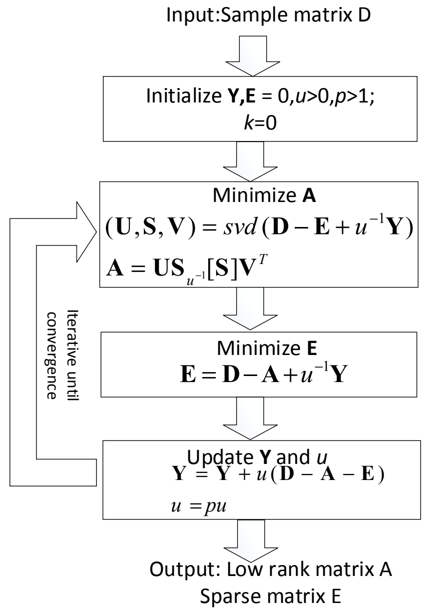


Fig. 4 Flow chart of inexact ALM

In this paper, it is notable that, the order of each sample makes sense, to recover the underlying clean model, all the data should be sorted to a same order in advance.

Since the recovered underlying low rank matrix of all the observation data is a good estimation of the real point cloud model, our proposed method can be considered to have the ability of correcting the corrupted points or, to some extent, recovering the lost points contained in some point clouds.

Additionally, low rank and sparse decomposition is potentially assumed all the observations are derived from a unique underlying low rank space. In this aspect, recently an extension of low rank and sparse decomposition which is named low rank representation [18] has demonstrated its power in many applications. As an opposite, this approach does not assume a unique underlying subspace rather recovering a multiple subspaces. In this paper, we are aiming to apply registration for multiple point clouds of a same object, and only assume these point clouds are corrupted by outliers or have some points lost. Therefore, low rank and sparse decomposition has already perfectly matched our assumption.

Point clouds corrupted by more complex kind of noise such as point cloud specific noise or a combination of many sorts of noise needs further study.

III. THE PROPOSED METHOD

In this section, we will describe our proposed method in details, and explain the role of each phase played.

A. The Proposed framework

The essential motivation of our method is to recover a noise and outlier free subspace using low rank and sparse decomposition (LRSD) over all the point clouds. However, as mentioned above, the precondition for using LRSD is that all the point data have to be put in a certain order (point corresponding) as well as do not have too much variants within point clouds themselves (rough registration). Therefore, preprocessing for roughly aligning the available point clouds is required. To this end, the Huber ICP is just satisfied with this requires and its robustness can give an initial alignment as good as possible for the corrupted point clouds.

Specifically, by reforming each point cloud as a column vector, we joint all these column vectors to form the data matrix which low rank and decomposition will be performed on. For example, with a 2 dimension cloud of 200 points, we reform it to a column vector by simply arrange all the points in the order of their x, y coordinates; that is to say a vector with size of 400 will be reformed. It should be noted that in order to form a matrix, the point number in each point cloud should be identical which is not often satisfied in a real application; therefore, a size unifying procedure is necessary. We will describe this phase later.

Then, we adopt low rank and sparse decomposition to decompose the generated data matrix to a low rank matrix and a sparse matrix, where point information contained in low rank matrix can be considered as the real object representation, and on the contrast, sparse matrix can be considered as where noise contained in each point cloud. Fig.5 illustrated the conception of this process.

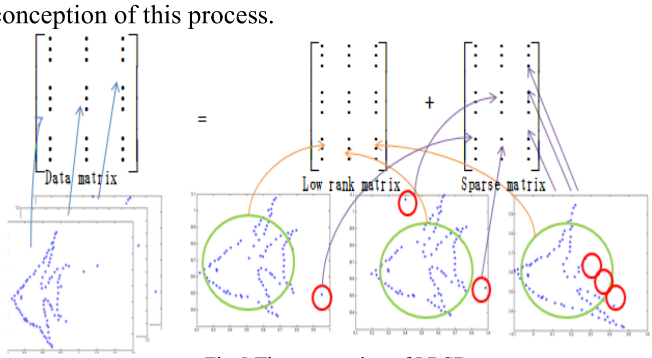


Fig.5 The conception of LRSD

Due to no prior knowledge about how each point cloud approached to the real object model, we just simply take the average of all the point clouds from low rank matrix as the real object model for reference. Next, with the obtained reference, we can implement point correspondence once again using Huber ICP, where the point cloud data from low rank matrix are used rather than the original point cloud data. Since outliers or lost data contained in the original point clouds have been removed or recovered already, a relatively correct correspondence of outlier or the lost data can be achieved. Furthermore, more accurate transformation parameters also can be estimated with more correct correspondence. Finally, noise or outliers contained in the sparse matrix are added back to point clouds according to the corresponding order for restoring the original point cloud. The above procedure is performed iteratively until reaching convergence. Fig. 6 illustrates the flowchart of the above mentioned procedure.

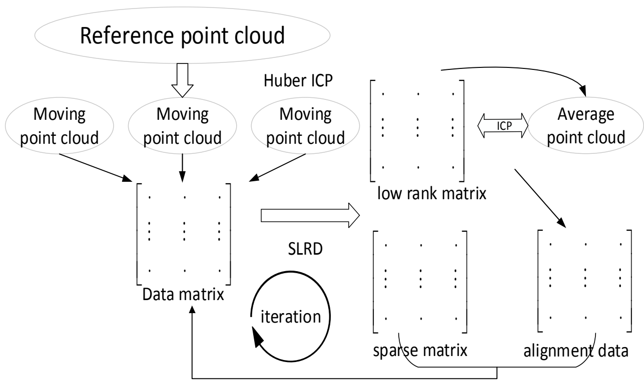


Fig.6. Framework of the proposed method

B. Some issues in implementation

Unifying point numbers in different clouds: As previously mentioned, in order to perform low rank and sparse decomposition, a size unifying operation of cloud point is required, which means point numbers of all clouds should be the same. This purpose may be achieved by clipping or padding operation. Apparently clipping excess data will lead to a loss of point cloud information, while padding operation maybe introduce extra noise to point cloud. In our experiments, we neither clipping nor padding, but down sampled the normal point cloud to the length of incomplete point cloud so as to avoid loss or introduce extra information.

Strategy of selecting parameter λ: λ in Eq. (9) for controlling trade-off between low-rank and sparse terms, has great affect on the final results. According to Eq. (9) it is clear that low rank part extracted with larger λ have more tolerance for noise and versa vice. Our experiment reveals that λ should not be too small as it will be failed to extract a meaningful low-rank part, nor to be too large as it will allows more corrupted point be contained in the extracted model. Fig.7 shows the results when λ is too small, proper or too large.

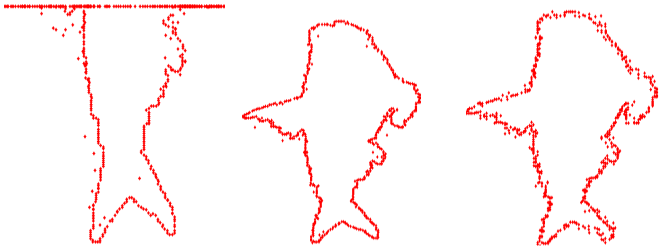

(a) (b) (c)

Fig. 7 The extracted low rank part with different λ
(a) λ =0.022,(b) λ =0.05, (c) λ =200

In practice, the parameter λ is generally chosen based on prior knowledge of data, such as data's noise ratio etc. While in most cases there is no additional information about noise ratio of a data set. In such a situation it is reasonable to assume that λ is dependent on the size of data matrix. Then, according to [19] we give the following estimation of λ, and it has been proven to give promised performances in different applications:

$$\lambda = \frac{2}{\sqrt{\max(m,n)}} \tag{11}$$

Where m and n are row and column numbers of the data matrix **X**, respectively.

IV. Experiment

In this section we verify the effectiveness of our proposed method on datasets of both 2D and 3D. We conduct a comparison with traditional ICP and Huber ICP qualitatively and quantitatively, the result demonstrates a promising performance of our method for accurate correspondence of outliers as well as recovery of lost data.

A. Data Sets

For 2D data, we use the SQUID fish contour database[1] from the university of Surrey UK. This dataset contains 1100 2D fish contours with the number of points varies from 500 to 3000.

For 3D dataset, the well known Stanford 3D Scanning Repository[2] is used to validate our proposed method. Several scanned 3D models are stored in this dataset; each of them contains points ranging from 2 thousand to 1.4 billion. In this paper we only used Dragon and Bunny models as our experimental data, which have 36,000 and 566,000 points, respectively. The samples from both 2D and 3D dataset are illustrated in Fig.8 and Fig.9.

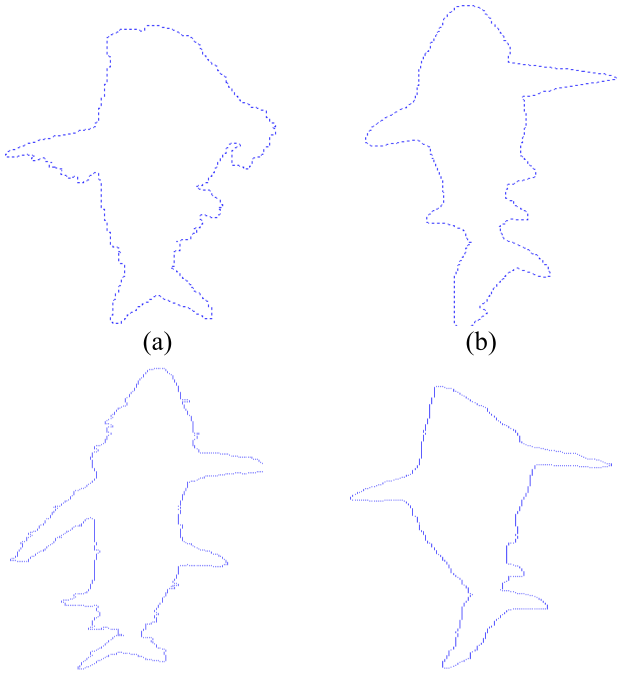
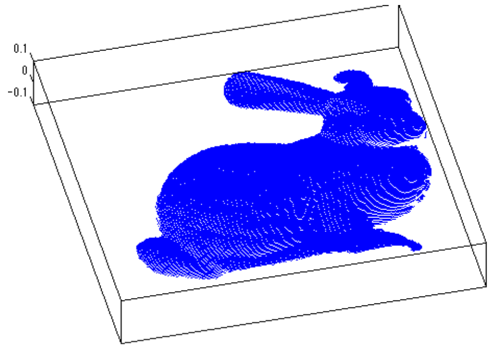
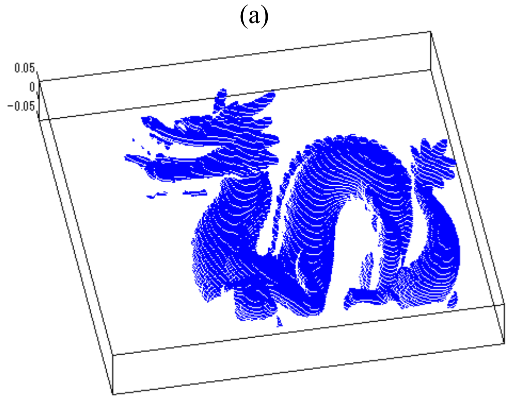

(a) (b) (c) (d)

Fig.8 2D dataset: some samples in SQUID



(a)



(b)

Fig.9 3D dataset: (a) Bunny and (b) Dragon

To verify the effectiveness of our proposed method, we synthetically generate an arbitrarily affine transformation to transform different point clouds since the original datasets are prepared for validating translation registration; we only applied a rotation with arbitrarily angle in the range of

[1] Please refer to http://www.ee.surrey.ac.uk/CVSSP/demos/css/demo.html
[2] Please refer to http://graphics.stanford.edu/data/3Dscanrep/

-30° to 30° to each point cloud for 5 times and create the multiple point clouds of the same object. Then outliers or occlusion are manually included to a part of point clouds, and the detailed step is as follows:

Outlier inclusion: we add outliers with different levels by varying outliers' distances above 0.2σ to the real point coordinates in the clouds, where σ is the variation of all point in the cloud.

Lost point inclusion: we also want to validate the effectiveness of our method when some point clouds are incomplete. This situation could arise when object is only partially observed because of occlusions. To simulate this situation, we randomly select a ball region B overlapped with the cloud and remove all of the points within B.

B. Experiment on 2D data set and numerical analysis

Firstly, we validate our proposed method on dataset containing outliers. Fig.10 illustrates some outlier-contained experimental data from a same object.

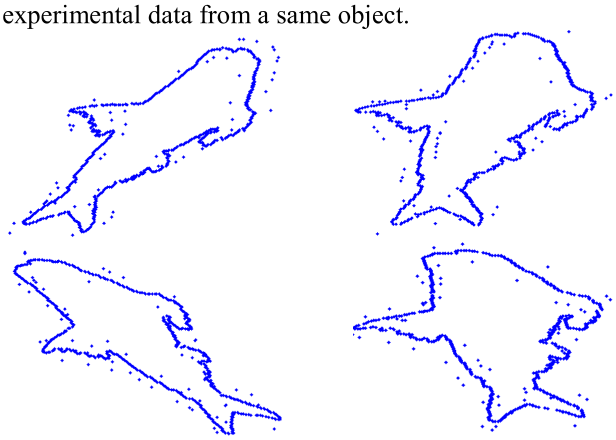
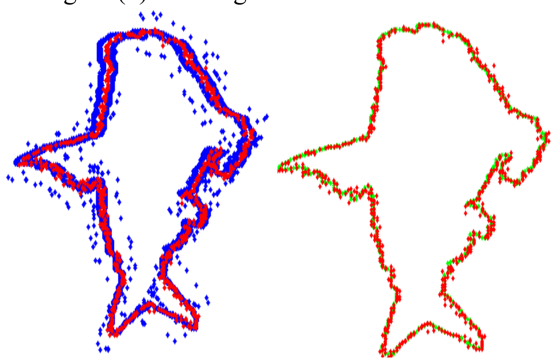


Fig.10 Examples of multiple point clouds containing outliers from a same object

The low-rank part extracted from these point clouds is illustrated in Fig.11. The red points are the recovered model from the decomposed low rank part and blue points in Fig. 11(a) are the multiple point clouds after registration; the green points in Fig. 11(b) are the ground-truth model.

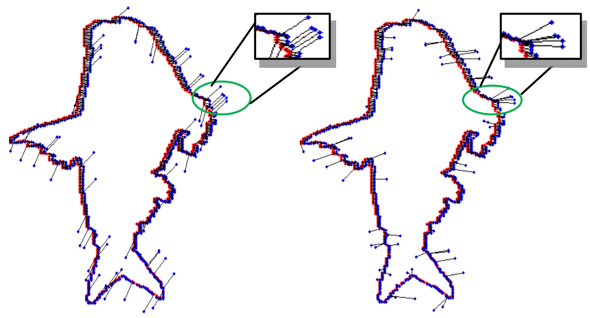


(a) (b)

Fig.11 The recovered point model (red), multiple point clouds after registration (blue in (a)), the ground-truth model (green in (b))

From Fig. 11(a), it can been seen that the recovered model from the low rank part don't contain so many outliers contained in the input point clouds; although different point clouds themselves contain scattered outliers, the corresponding recovered model from the low rank part is almost completely clean. On the other hand, from Fig. 11(b), it is intuitive that the recovered model from the low rank part is a very good approximation of the ground-truth model, and thus can be used as the reference of registration in the following step.

Fig.12(a) illustrates the point correspondence through our method, and(b) a comparison result through the traditional ICP. According to Fig.12, it can be seen that the traditional ICP just simply finds the nearest neighbor of a point as its corresponding point, thus leads to a many-to-one relation; while in our method, since a "clean" model can be accurately estimated firstly, a relatively correct correspondence relationship can be obtained.



(a) (b)

Fig.12 (a)Point correspondence using the proposed method and (b)the traditional ICP. Red points are the reference point cloud, blue points are moving point cloud, and black lines represent correspondence.

The numeric experiment is also conduct to verify the proposed method.

First we illustrate the comparison of registration error of our method with conventional ICP and Huber ICP in Fig.13.

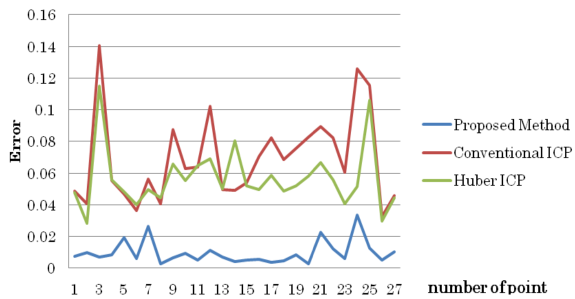


Fig.13 comparison of registration error of different methods

We can see that with cloud point varies, our method illustrate a constant lower error compare to other methods.

Next, an investigation with respect to correspondence precision of our method and the other two methods is illustrated in Fig.14.

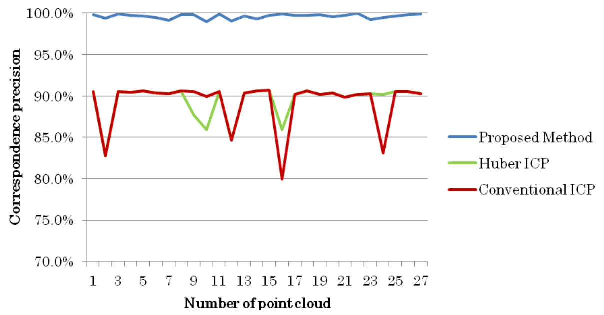


Fig.14 comparison of correspondence precision of different methods

For simplicity of 2D data, our method gives an almost perfect performance on correspondence precision, while the other two methods show a close and relatively inferior result.

Fig.15 and Fig.16 illustrates registration error and correspondence precision with respect to varies outlier raio respectively.

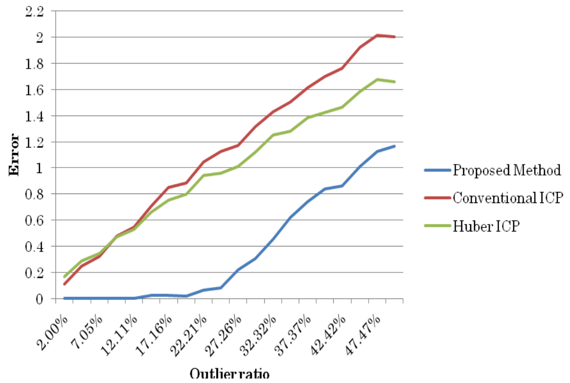


Fig.15 Error with respect to varies outlier ratio

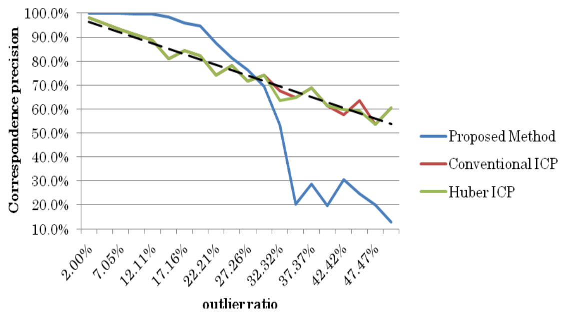


Fig.16 Correspondence precision with respect to varies outlier ratio

The dash line in second figure indicats the trend of traditional method, according to which we can figure that when the outlier ratio is lower than about 30% our method is overmatch to traditional method, while when outlier ratio is too large the low rank part extracted by our method can be meaningless, consequently leads to an inferior performance.

Moreover, we also verified the performance of the error with respect to different λ in our method and illustrated the result in Fig.17.

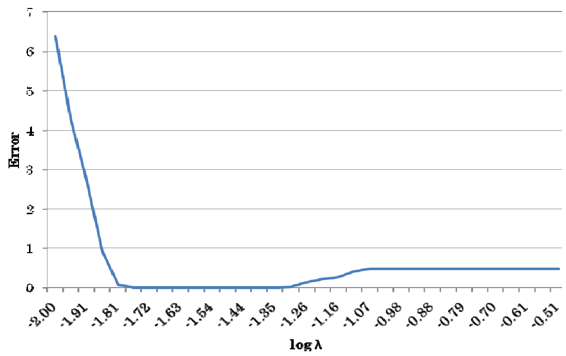


Fig.17 error with respect to varies λ

According to the figure, we can observe an obvious valley of error is attained when λ is at its proper value, which is consistent with the previous discussion of λ.

At last, we verified the performance of our method when the amount of multiple point clouds is varied. The result is illustrated in Fig.18.

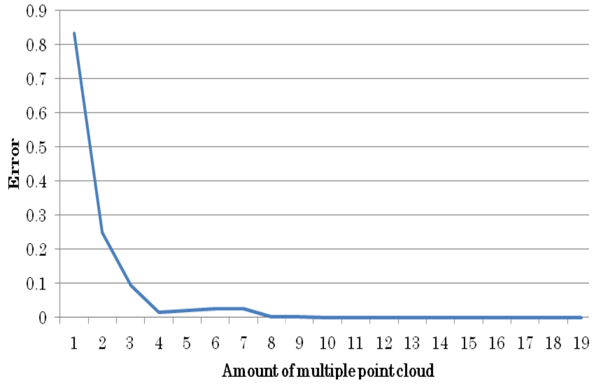
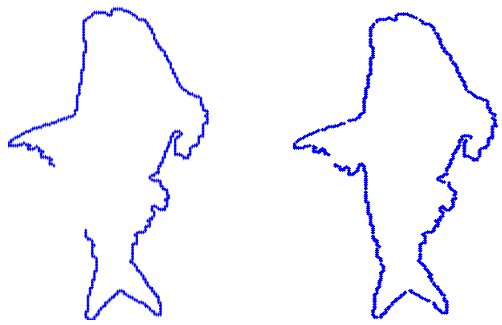


Fig.18 Error with respect to varies amount of multiple point cloud in SLRD

The more point cloud used, the higher precision our method can be achieved. Critically, the knee of the figure tells us around 4 or 5 point clouds is enough to recover all the information of the original point cloud.

We demonstrate the recovery performance of lost points in incomplete point cloud in Fig. 19, which shows that the lost points can be, to some extent, recovered by our proposed method.

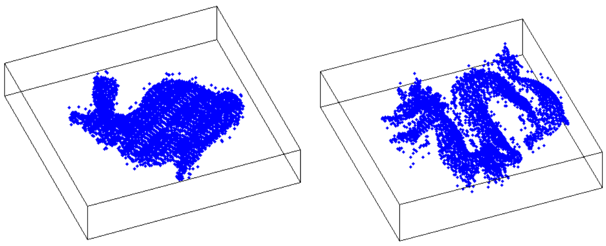


(a) (b)

Fig. 19 example of incomplete point cloud(a) and recovered point cloud(b)

C. *Experiment on 3D data set*

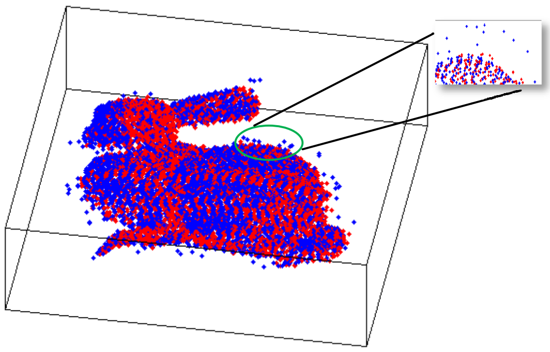
In this section, we will illustrate the effectiveness of our proposed method applying to a 3D data set. The experiment settings are similar to for 2D cases, only we explored the situation where point cloud contained outliers. Fig.20 illustrates some examples used in our experiment.



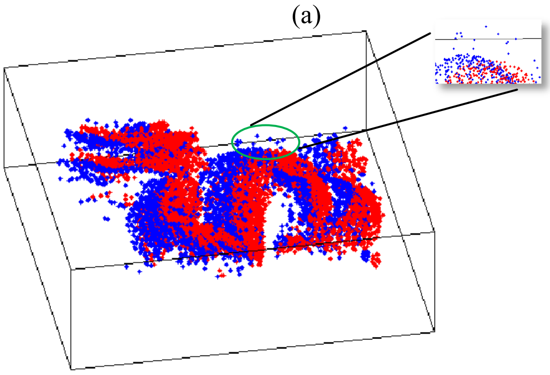
(a) (b)

Fig. 20 Examples of Bunny (a) and Dragon (b) with 10% outliers used in experiment

The extracted low rank part and one of point cloud containing outliers are showed in Fig. 21 simultaneously. Red points are the extracted low rank model and blue points are one of the outliers containing point cloud, we can observe that barely an outlier is contained in extracted point cloud.



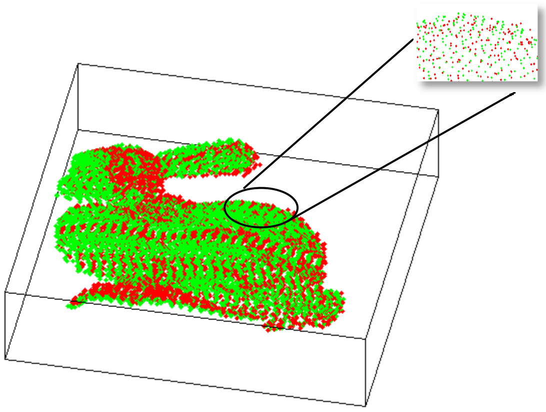
(a)



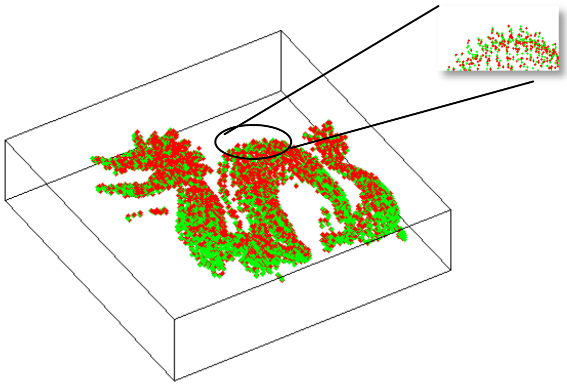
(b)

Fig. 21 Extracted point model (red) and some noise contained point cloud (blue)

In Fig. 22 we present a comparison of extracted low rank part and the ground truth model. It can be observed that two models are almost coinciding.



(a)



(b)

Fig. 22 Extracted point model (red) and ground truth point cloud (green)

Based on these figures we can conclude that in 3D situation, the extracted low rank model still have a strong robustness against outliers, therefore can provide a solid reference for registration.

V. CONCLUSIONS

In this paper we proposed a robust multiple point clouds registration method by taking advantage of Huber ICP and Sparse and Low rank decomposition. Experiment shows that our proposed can achieve promising performance for point correspondence in both 2D and 3D cases. Furthermore, a rather correct correspondence can still be obtained even some outliers are contained in the point clouds, and for the point cloud which is not complete, our method can also, to some extent, recover the lost data.

ACKNOWLEDGMENT

This work was supported by the Grant-in Aid for Scientific Research from the Japanese Ministry for Education, Scientific Culture and Sports under the Grant No. 15H01130, No.15K00253 and No.25280044, in part by the MEXT Support Program for the Strategic Research Foundation at Private Universities (2013-2017).

References

[1] Bellekens, Ben, et al. "A Survey of Rigid 3D Pointcloud Registration Algorithms." *Fourth International Conference on Ambient Computing, Applications, Services and Technologies*. IARA, 2014.

[2] Brown, Lisa Gottesfeld. "A survey of image registration techniques." ACM computing surveys (CSUR) 24.4 (1992): 325-376.

[3] S. Marden and J. Guivant, “Improving the Performance of ICP for RealTime Applications using an Approximate Nearest Neighbour Search,” 2012, pp. 3–5.

[4] Yambor, W. S., Draper, B. A., & Beveridge, J. R. (2002). “Analyzing PCA-based face recognition algorithms: Eigenvector selection and distance measures.” In H. Christensen & J. Phillips (Eds.), Empirical evaluation methods in computer vision (pp. 1–14). Singapore: World Scientific Press.

[5] Bing Jian; Vemuri, B.C., "Robust Point Set Registration Using Gaussian Mixture Models," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.33, no.8, pp.1633,1645, Aug. 2011

[6] Besl, P.J.; McKay, Neil D., "A method for registration of 3-D shapes," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.14, no.2, pp.239,256, Feb 1992

[7] Rusinkiewicz, Szymon, and Marc Levoy. "Efficient variants of the ICP algorithm." 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on. IEEE, 2001.

[8] Fitzgibbon, Andrew W. "Robust registration of 2D and 3D point sets." Image and Vision Computing 21.13 (2003): 1145-1153.

[9] Y. Chen, G. Medioni. ”Object Modeling by Registration of Multiple Range Images,” Proc. of the 1992 IEEE Intl. Conf. on Robotics and Automation, pp. 2724-2729, 1991.

[10] Segal, Aleksandr, Dirk Haehnel, and Sebastian Thrun. "Generalized-ICP."Robotics: Science and Systems. Vol. 2. No. 4. 2009.

[11] Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. 2013. Sparse iterative closest point. In*Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*(SGP '13). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 113-123.

[12] P. J. Huber, "Robust estimation of a location parameter." The Annals of Mathematical Statistics 35.1 (1964): 73-101.

[13] Moré, Jorge J. "The Levenberg-Marquardt algorithm: implementation and theory." *Numerical analysis*. Springer Berlin Heidelberg, 1978. 105-116.

[14] Chandrasekaran, Venkat, et al. "Rank-sparsity incoherence for matrix decomposition." SIAM Journal on Optimization 21.2 (2011): 572-596.

[15] Recht, Benjamin, Maryam Fazel, and Pablo A. Parrilo. "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization." SIAM review 52.3 (2010): 471-501.

[16] Candes, Emmanuel J., Michael B. Wakin, and Stephen P. Boyd. "Enhancing sparsity by reweighted ℓ 1 minimization." Journal of Fourier analysis and applications 14.5-6 (2008): 877-905.

[17] Lin, Zhouchen, Minming Chen, and Yi Ma. "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices." arXiv preprint arXiv:1009.5055 (2010).

[18] Liu, Guangcan, et al. "Robust recovery of subspace structures by low-rank representation." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.1 (2013): 171-184.

[19] Candès E J, Li X, Ma Y, et al. Robust principal component analysis?[J]. Journal of the ACM (JACM), 2011, 58(3): 11.