UNIVERSITY OF CAPE TOWN
DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours
Final Paper 2021

1. **Title: Orchard Registration using Robust Point Matching**

2. **Author: Leo Chen**

2.1 Project Abbreviation: TREEREG

2.2 Supervisor(s): Patrick Marais
Deshen Moodley

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	10
System Development and Implementation	*0*	*20*	15
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	15
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	**80**

Orchard Registration using Robust Point Matching

Leo Chen
CHNLEO007@myuct.ac.za

3. ABSTRACT

This paper tries to implement robust point matching for orchard registration given detected tree polygon features. Tree polygons will change or disappear over time, thus complicates the problem. Robust point matching is an optimization-based approach which iteratively estimate transformation parameters and correspondence. It shows that RPM can handle different translations and rotations. However, when facing outliers, the registration accuracy will decrease.

4. INTRODUCTION

Agriculture is an important factor for a country's economy. One of the modern practices for agriculture is precision agriculture (PA), which is a farming management concept based on observing, measuring and responding to inter and intra-field variability in crops. It aims to improve the efficiency and productivity of agriculture. Unmanned Aerial Vehicles (UAVs) are commonly used for this purpose. Inspection of tree's growth is an important activity to manage farming. Currently, this is done manually with farmers walking through orchards. This paper focuses on a particular part of PA which is to align orthomosaic orchard images of a same orchard at different times to monitor trees' growth and manage it efficiently.

The main problem of this project is how to align two orthomosaic orchard images automatically when there were taken at different times. The provided features for registration are detected tree polygons as shown in Figure 1. The difficulty comes from two aspects: 1. With trees' growth (trees being removed, replanted, or just grow) over time, the detected tree polygons' shape and location will change. 2. Local deformation occurred during orthomosaic generation.

For the first aspect, for two different dates of the same orchard, there might be different numbers of trees, different layout of the orchard or changes in shape of some trees (as shown in Figure 2). Trees can also overlap with each other.

Figure 1: Detected Tree Polygons for Orchard 1

For the second aspect, when automatically generating orthomosaic images, a step of stitching is involved for stitching multiple parts of the same scene. Local deformation can be introduced during this stage, which further complicated the registration problem.

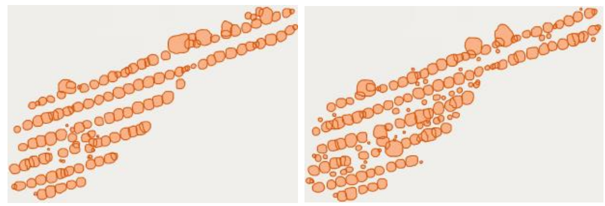


Figure 2: Detected Tree Polygons for Orchard 1 at Different Dates

Along with the detected tree polygons, there is a confidence level associated with each tree polygon to indicate how likely the detected tree polygon is a tree (or an error detection) and a height map to indicate the height for each tree. However, the detected height is not reliable so it cannot be used. This project is run on Windows 10 operating system with the processor of Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz and memory of 8GB RAM.

In this paper, the problem is reframed as matching tree polygons by representing those as points, where each point for a polygon is the centroid of that polygon. The resulting problem is to align one point cloud with another point cloud.

The approach taken in this paper is Robust Point Matching [1] which is an approach that iteratively estimate transformation parameters and correspondence at the same time.

The outline of the rest of the paper is as following: section 4 covers previous work that has been done on this area, section 5 introduces the method, section 6 presents the results and a discussion about it and section 7 presents conclusions for this paper.

5. Related Work

Image registration approaches are categorized into two categories: area-based and feature-based image registration [4]. For area-based methods, windows of predefined size or the entire image are used to calculate the similarity between two images. Window pairs with the highest similarities are matched and the transformation model is estimated from those matches.

Feature-based image registration approaches have two important stages: feature detection and feature matching. Most of the feature-based image registration approaches rely on SIFT or its variations focus on detecting unique features. It uses scale-space to detect feature points (which makes them scale invariant) and localize those feature points to get more consistent feature points [3]. Orientation is then assigned to each feature point to achieve rotation invariant. Feature points are represented as special feature descriptors so that matching feature points can be easily achieved. With unique features matching can be easily done. Other approaches focus on feature matching, where given features that are not quite unique, match them accordingly. Point cloud registration is one such category of method where it focuses on feature matching instead of feature detection.

Registering two point clouds can be done in mainly two ways: 1. By estimating the correspondence of points between two point clouds. A typical way for doing this is feature based method, which groups points into lines, curves or other higher level shapes such as in [5] and [6]. Then a matching is done for that. Graph based approach also fall under this category. It has the advantage of invariant to affine transformation, but it is very time inefficient. There were many methods proposed for graph matching such as [13]. 2. By estimating the transformation parameters to align the sensed point cloud to the reference point cloud. A classic method for this type is the method of moment [7]. There are other methods for this type such as Hough Transform [8] and tree searches [9].

These methods established for registering two point clouds solely by estimating the correspondence or by estimating the transformation parameters. The result for those approaches were not desirable. However, by estimating transformation parameters and correspondence in an iterative approach can result in an accurate point cloud registration. The simplest method is Iterative Closest Point (ICP) [10]. It is easy to implement however can easily fall into local minima. A probabilistic method which uses Gaussian mixture model is Coherent Point Drift (CPD) [11], which can register two point clouds with good accuracy but it does not handle outliers very well. The proposed method in this paper is also under this category. A different probabilistic strategy proposed in [12] is shape context. A point is defined by its relative positions to other points. However given the multitemporal aspect of the provided data and most of points are in a line, this method would not fit properly.

There are other novel approaches which uses deep neural network to learn a robust feature correspondence search [14] [15]. Due to the limited amount of data and time, this category of method was not chosen.

Point cloud registration methods can also be categorized into another three categories: optimization-based, feature-learning and end-to-end learning [2]. Robust point matching falls under the category of optimization-based approaches. Methods such as ICP and CPD are also under this category.

6. Design and Implementation

6.1 Method

6.1.1 Data Preprocessing

The input data of the proposed method is a set of point sets with each point set represents the boundary of the detected polygon. Each point set is used to calculate a centroid to represent the polygon. These points are in global positioning coordinate (GPS) system. Therefore, it needs to be transformed into normal coordinates. All points are scaled up by 100,000 and are translated to the origin by the amount of the point cloud center. All polygons with less than 50% confidence level are filtered out.

6.1.2 Robust Point Matching

Given two 2D point sets $\{X_j\}$ and $\{Y_k\}$, related by an affine transformation $\{A, t\}$ where A is the matrix product of scale, rotation and shear, and t is the translation. With points can be noisy – it might not be an actual tree, or the centroid calculated for the polygon can be slightly misaligned compared to its true centroid. A set of correspondences are defined in an *assignment matrix* m as:

$$m_{jk} = \begin{cases} 1 & \text{if point } X_j \text{ corresponds to point } Y_k, \\ 0 & \text{otherwise.} \end{cases}$$

The problem can then be defined as finding the affine transformation $\{A, t\}$ that best relates $\{X_j\}$ and $\{Y_k\}$. This approach estimates the transformation and the correspondence at the same time. This naturally arise from minimizing a cost function E:

$$E_{2D}(m, t, A) = \sum_{j=1}^{J} \sum_{k=1}^{K} m_{jk} \| X_j - t - AY_k \|^2 + g(A) - \alpha \sum_{j=1}^{J} \sum_{k=1}^{K} m_{jk}$$

subject to $\forall j \sum_{k=1}^{K} m_{jk} \leq 1$ and $\forall k \sum_{j=1}^{J} m_{jk} \leq 1$, $\forall jk\ m_{jk} \in \{0,1\}$ and

$$g(A) = \gamma(a^2 + b^2 + c^2).$$

A composes of four parameters $\{\alpha, \theta, b, c\}$ which are scale, rotation and two components of shear:

$$A = s(a)R(\Theta)Sh_1(b)Sh_2(c)$$

where

$$s(a) = \begin{pmatrix} e^a & 0 \\ 0 & e^a \end{pmatrix}, \qquad Sh_1(b) = \begin{pmatrix} e^b & 0 \\ 0 & e^{-b} \end{pmatrix},$$

$$Sh_2(c) = \begin{pmatrix} \cosh(c) & \sinh(c) \\ \sinh(c) & \cosh(c) \end{pmatrix}.$$

In this paper, since only rigid transformation is considered, scale and rotation parameters can be omitted.

The RPM algorithm optimizes the cost function using the *softassign* algorithm. It associates a value m_{jk} with a variable Q_{jk} where

$$Q_{jk} = -\left(\left\|x_j - t - AY_k\right\|^2 - \alpha\right) = -\partial E_{2D}/\partial m_{jk}.$$

The goal then is to maximize $\sum_{j=1}^{J} \sum_{k=1}^{K} m_{jk} Q_{jk}$. This discrete problem can be turned into a continuous problem by using deterministic annealing where it introduces a control parameter $\beta > 0$ then m_{jk} would be:

$$m_{jk} = \frac{exp(\beta Q_{jk})}{\sum_{j=1}^{J} \sum_{k=1}^{K} exp\ (\beta Q_{jk})}$$

This is known as the *softmax*. As $\beta \to \infty$, m_{jk} corresponds to the maximum will tend to 1 while others tend to 0.

The resulting algorithm would be as following:

Algorithm 1 Robust Point Matching For Orchard Registration

Initialize θ, t, a, b, c to 0, β to β_0, m_{jk} to $(1+\epsilon)$, γ to γ_0
while $\beta \leq \beta_f$ **do**
  while m_{jk} not converge or iteration $\leq \mathrm{I}_0$ **do**
    Update correspondence by $softassign$
    Normalize m_{jk} using Sinkhorn's method
    Update pose parameters by coordinate descent
  end while
  $\beta \leftarrow \beta_r \beta$, $\gamma \leftarrow \gamma / \beta$
end while

When updating correspondence, update $m_{jk} = exp(\beta Q_{jk})$, then normalize the matrix with Sinkhorn's method. The resulting matrix would be:

$$m_{jk} = \frac{exp(\beta Q_{jk})}{\sum_{j=1}^{J} \sum_{k=1}^{K} exp\ (\beta Q_{jk})}$$

Note that Sinkhorn's algorithm is used to normalize the matrix in order to satisfy the doubly-stochastic constraints. Since constraints are inequality, it can be addressed by introducing slack variables where

$$\forall j \sum_{k=1}^{K} m_{jk} \leq 1 \to \forall j \sum_{k=1}^{K+1} m_{jk} = 1$$

.

7. Results and Discussion

7.1 Evaluation

The RPM is evaluated from two aspects: 1. Whether it can accurately register two polygon clouds under different translation and rotation. 2. Whether it is robust under different amount of noise and outliers.

The measuring metric that is used is the mean square error between two point clouds between the closest points. A base MSE is calculated before deregistering the point clouds and compared it against the MSE for RPM registration.

First aspect of testing are applying different translations and rotations to find out RPM's accuracy. The translation ranges from small amount of translation (-10 ~ 10) to large amount of translation (-100 ~ 100). The rotations that were applied are below 30 degrees, reason for that is because according to Aerobotics there will not be a lot of rotations and translations difference. Thus, the choice of 30.

Second aspect of testing is robustness. The approach taken here is to eliminate/filter unreliable polygons (under a certain confidence level) which would result in missing points and missing correspondences since only polygons with confidence level above a certain amount would be used for testing. Figure 3 shows Orchard one under different confidence threshold.

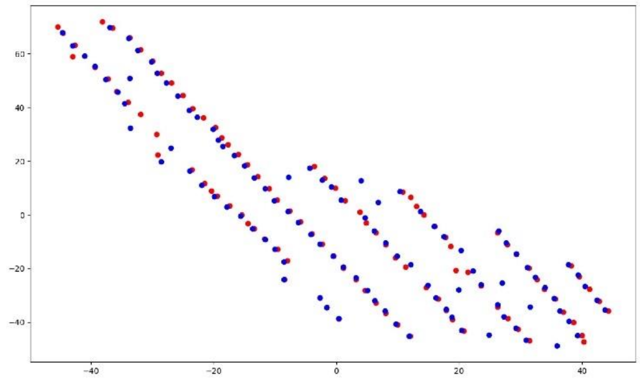
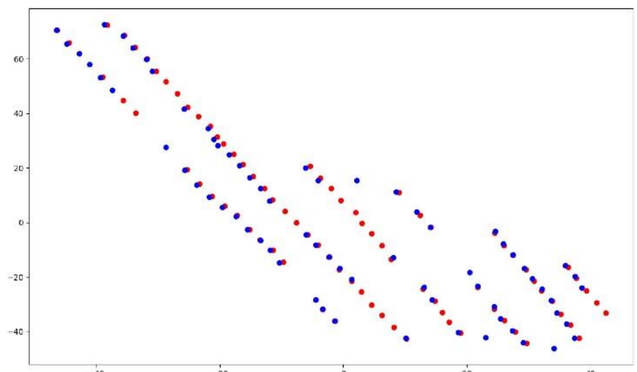


Figure 8: Point clouds with 50% threshold (Top) and point clouds with 90% threshold (Bottom) for Orchard 1



Reason for using different confidence levels to test the robustness of RPM is because it is more suitable for the problem domain.

Two types of orchards are chosen for testing which orchard 1's polygons are sparser while orchard 3 has denser polygons distribution as shown in Figure 9.

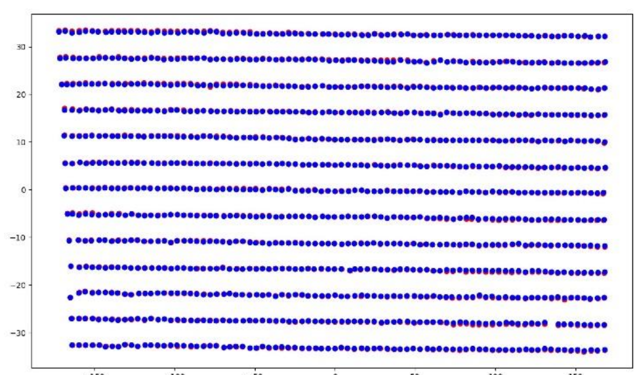


Figure 9: Orchard 3

Red point cloud is the source point cloud and it is registered to the blue point cloud (reference point cloud).

The MSE validation will be complemented by human inspection.

7.2 Results and Discussion

Figure 10 shows the difference between RPM MSE and preregistered MSE under different confidence level. Orange line represents Orchard 3 and blue line represents Orchard 1. The corresponding data is shown in Figure 11.

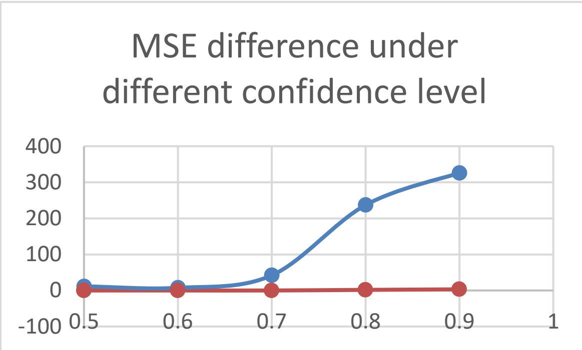


Figure 10: MSE difference under different confidence level

Orchard	Confidence Level	MSE Difference
Orchard 1	0.5	11.46
	0.6	7.35
	0.7	41.95
	0.8	236.95
	0.9	325.79
Orchard 3	0.5	0
	0.6	0
	0.7	0
	0.8	1.73
	0.9	3.18

Table 1: Data for Figure 10

As shown from Figure 10 and table 1, as the confidence level increases and thus resulting in more outliers (points with missing correspondence in the other point cloud), the MSE difference increases. It is not influential for orchard 3 but for orchard 1 it has a serious impact. Reason for that is because orchard 1 has sparser distribution of polygons and many of the polygons have low confidence level. On the other hand orchard 3's polygons have much higher confidences which results in less outliers. Table 2 and 3 shows the number of polygons before and after filtering.

Table 2: Number of data points before and after

Orchard1	Date 1	Date 1 (After filtering)	Date 2	Date 2 (After filtering)	Filtering Threshold
	143	101	166	101	0.5
	143	97	166	96	0.6
	143	92	166	88	0.7
	143	89	166	78	0.8
	143	83	166	70	0.9

filtering under different confidence level for orchard 1

Orchard3	Date 1	Date 1 (After filtering)	Date 2	Date 2 (After filtering)	Filtering Threshold
	1160	1076	1158	1076	0.5
	1160	1075	1158	1075	0.6
	1160	1074	1158	1075	0.7
	1160	1072	1158	1069	0.8
	1160	1057	1158	1054	0.9

Table 3: Number of data points before and after filtering under different confidence level for orchard 3

The result is confirmed by visual inspection as shown in Figure 11 and 12.

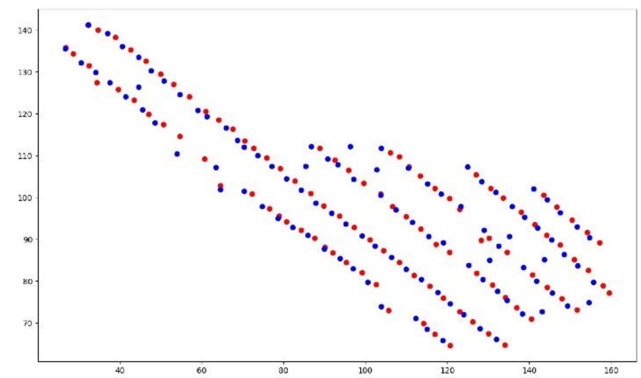
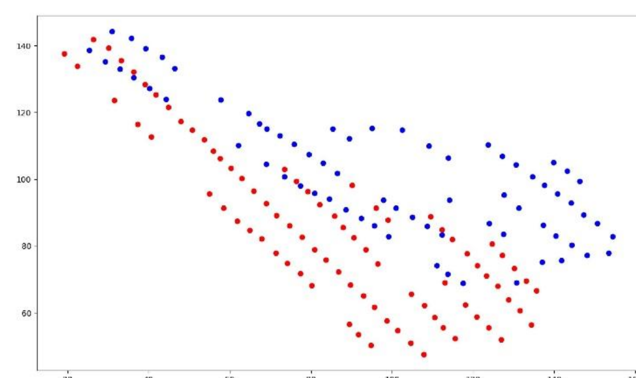


Figure 11: Registration using RPM for 50% confidence level (Top) and 90% confidence level (Bottom) for orchard 1



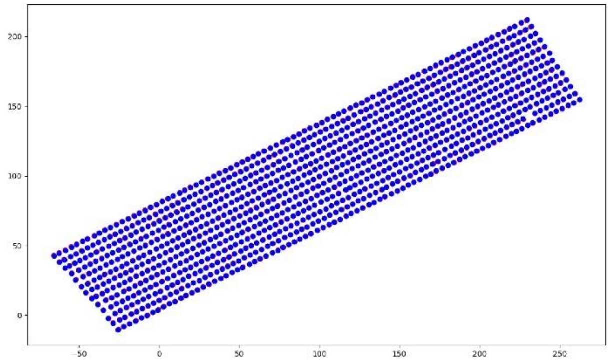
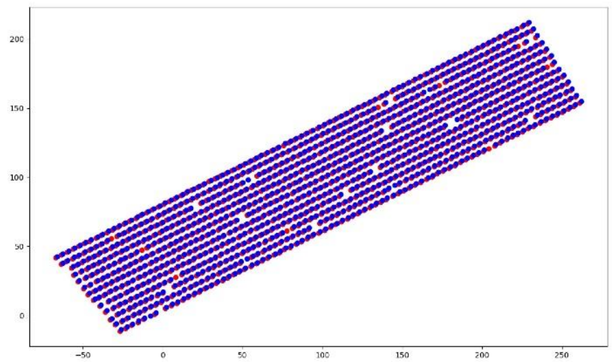


Figure 12: Registration using RPM for 50% confidence level (Top) and 90% confidence level (Bottom) for orchard 3



The result for MSE difference under different range of transformation is shown in Figure 13 and Table 4.

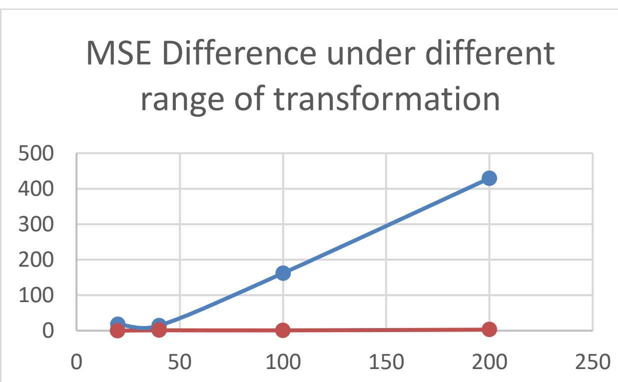


Figure 13: MSE difference under different range of transformation

Similar to confidence level test, orchard 3 has more accuracy than orchard 1 under different range of transformation where the larger the transformation, the less accurate the registration is.

The times taken for RPM to register orchards are rather slow. For orchard 1, it takes about 20 – 30 seconds to register 2 orchards, whereas for orchard 2 it can take up 20 – 40 minutes. In some cases, the run time can be more than an hour. The reason for that is because it involves manipulating a $n * n$ matrix. Therefore, the time complexity is at least $O(n^2)$. With points more than 1000 such as orchard 3, each iteration will execute more than 1 million operations.

Orchard	Translation Range	MSE Difference
Orchard 1	20	17.89
	40	14.41
	100	161.95
	200	429.25
Orchard 3	20	0
	40	1.09
	100	0.89
	200	2.93

Table 4: Data for Figure 13

Parameters chosen are as following: $\alpha = 0.03$, $\beta = 0.00091$, $\beta_r = 1.075$, $\beta_f = 0.2$. Inner iteration threshold (matrix normalization) is 30 and outer iteration threshold (the whole estimation algorithm) is 4. It is possible to increase the number of iteration threshold which will result in more accurate registration. However, after testing usually 4 can give a rather good registration and more iteration will increases the time taken.

8. Conclusions

RPM can register data points under different translation and rotation accurately. However, when facing some outliers, it will decrease the accuracy with the result of some misalignment. With a significant number of outliers, the registration will be very inaccurate.

The range of transformation can also have an effect on registration accuracy, with larger transformation the less accurate the registration would be.

Since the requirement from Aerobotics is that only small amount of transformations need to be considered. And since the smaller the transformations the more accurate the registration is. Thus the RPM algorithm can fulfil the need for orchard registration.

The algorithm has a time complexity of $O(n^2)$. When data points are a lot, it can take up to one hour to register two images.

In this paper, only rigid transformation is considered. However due to the parametric framework of RPM, it can be easily extended to all affine transformations. There are also some extensions to the RPM framework such as TPS-RPM (Thin plate spline), it can handle not just affine transformation but also thin plate spline transformation.

9. ACKNOWLEDGMENTS

We would like to thank A/Prof. Patrick Marais and A/Prof. Deshen Moodley for all their guidance and supervision throughout the research project. We would also like to thank Aerobotics for providing of the orchard data used in the project.

10. REFERENCES

[1] Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, Eric Mjolsness. 1998. New algorithms for 2D and 3D point matching: pose estimation and correspondence, Pattern Recognition, Volume 31, Issue 8, Pages 1019-1031. https://doi.org/10.1016/S0031-3203(98)80010-1.

[2] Xiaoshui Huang, Guofeng Mei, Jian Zhang, Rana Abbas. 2021. A comprehensive survey on point cloud registration. DOI:: arXiv:2103.02690

[3] Tony Lindeberg. 2012. Scale invariant feature transform. Vol. 7, no. 5. p. 10491. DOI: https://doi.org/10.4249/scholarpedia.10491

[4] Barbara Zitová and Jan Flusser. 2003. Image registration methods: a survey. Image and Vision Computing, (Oct. 2003), Volume 21, Issue 11, 977-1000. DOI: https://doi.org/10.1016/S0262-8856(03)00137-9

[5] Dimitris Metaxas, Eunyoung Koh, and Norman I Badler. 1997. Multi-Level Shape Representation Using Global Deformations and Locally Adaptive Finite Elements. International Journal of Computer Vision 25, 49–61. https://doi.org/10.1023/A:1007929702347

[6] Stan Sclaroff and Alex Pentland. 1995. Modal matching for correspondence and recognition, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 6, pp. 545-561. doi: 10.1109/34.387502.

[7] Lyndon S. Hibbard, Richard A. Hawkins. 1998. Objective image alignment for three-dimensional reconstruction of digital autoradiograms, Journal of Neuroscience Methods, Volume 26, Issue 1, Pages 55-74. https://doi.org/10.1016/0165-0270(88)90129-X.

[8] D.H. Ballard. 1981. Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition, Volume 13, Issue 2, Pages 111-122. https://doi.org/10.1016/0031-3203(81)90009-1.

[9] Henry Baird. 1985. Model-Based Image Matching Using Location.

[10] Shaoyi Du, Nanning Zheng, Shihui Ying, Jianyi Liu. 2010. Affine iterative closest point algorithm for point set registration, Pattern Recognition Letters, Volume 31, Issue 9, Pages 791-799. https://doi.org/10.1016/j.patrec.2010.01.020.

[11] Andriy Myronenko and Xubo Song. 2010. Point Set Registration: Coherent Point Drift, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 12, pp. 2262-2275. doi: 10.1109/TPAMI.2010.46.

[12] Serge J. Belongie, Jitendra Malik and J. Puzicha. 2010. Shape matching and object recognition using shape contexts. 2010 3rd International Conference on Computer Science and Information Technology, 9, 471-474.

[13] Ron Zass and Amnon Shashua. 2008. Probabilistic graph and hypergraph matching, 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1-8, doi: 10.1109/CVPR.2008.4587500.

[14] Haowen Deng, Tolga Birdal and Slobodan Ilic. 2018. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 195-205

[15] Zan Gojcic, Caifa Zhou, Jan D. Wegner and Andreas Wieser. 2018. The Perfect Match: 3D Point Cloud Matching with Smoothed Densities. arXiv:1811.06879