T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ

3D OBJECT RECOGNITION BY USING 3D POINT CLOUDS

Master of Science Thesis

YUSUF GÖKHAN YAVUZ

İSTANBUL,2013

T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

3D OBJECT RECOGNITION BY USING 3D POINT CLOUDS

Master of Science Thesis

YUSUF GÖKHAN YAVUZ

Supervisor: Asst. Prof. Dr. Övgü ÖZTÜRK

İSTANBUL,2013

T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

Title of the Master's Thesis : 3D Object Recognition By Using 3D Point Clouds

Name/Last Name of the Student : Yusuf Gökhan Yavuz

Date of Thesis Defense : 03/09/2013

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Asst. Prof. F. Tunç BOZBURA

Acting Director

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Committee Members:

Asst. Prof. Dr. Övgü ÖZTÜRK :

Asst. Prof. Dr. Devrim ÜNAY :

Asst. Prof. Dr. Egemen ÖZDEN

ACKNOWLEDGMENTS

I would never have been able to finish my master thesis without the guidance of supervisor Asst. Prof. Dr. Övgü ÖZTÜRK, help from friends, and support from my family and wife also my colleague.

I would like to express my deepest gratitude to my advisor, Asst. Prof. Dr. Övgü ÖZTÜRK, for her excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research. I would also like to thank my thesis patners Bahtiyar Kaba, Hüseyin İnan and Mustafa Tosun.

I would like to thank Metin Göğüş and Emre Begen, who as a good friend, was always willing to help and give his best suggestions. Many thanks to my colleague Akın Söylemez, Alper Gündüz Hayati Demirbaş, Harun Markopcuoğlu, and other friends for suggesting me. My research would not have been possible without their helps.

I would also like to thank my parents and parents in law, two sisters, and my cat. They were always supporting me and encouraging me with their best wishes. I also want to thank my relatives to show patience to me for not visiting them. Finally, I would like to thank my wife, Aliye YAVUZ. She was always there cheering me up and stood by me through the good times and bad.

Yusuf Gökhan YAVUZ

3 September, 2013

ÖZET

3 BOYUTLU NOKTA BULUTU KULLANARAK 3 BOYUTLU NESNE TANIMA

Yusuf Gökhan Yavuz

Bilgisayar Mühendisliği

Danışman: Yrd. Doç. Dr. Övgü ÖZTÜRK

Eylül 2013, 55 Sayfa

Akıllı sistemlerin en önemli parçalarından biride nesne tanıma sistemleridir. Nesne tanıma bir çok gelişmeye değişik alan ve endüstrilerde öncülük etmiştir. Örneğin askeri, sağlık, ulusal güvenlik, bankacılık, kültürel miraslar, … vb. Bu çalışmada ICP kullanarak kültürel bir 3 boyutlu nesne üzerinde yine bu 3 boyutlu nesne ait bir nesneyi tanımaya çalıştık. ICP methodu 3 boyulu bir nesne ile kesilmiş olan 3 boyutlu nokta bulutu ile minimum uzaklıkları karşılaştırarak çalışır. Daha sonrasında tekrarlı olarak dönme ve öteleme matrislerini bularak aralarındaki uzaklığı iki nesne en yakın eşleşmesine kadar devam eder. 3 boyutlu nesne tanımada ICP kullanmanın en büyük avantajı hızlı olması ve parçalı eşleştirme yapabiliyor olmasıdır. ICP'yi 4 farklı nokta bulutu ve 3 boyutlu nesne üzerinde değerlendirmeye tabi tutulup efektifliği gösterilmiştir.

Anahtar Kelmeler: Nesne Tanıma, 3 Boyutlu Eşleştirme, Nokta Bulutu, ICP

ABSTRACT

3D OBJECT RECOGNITION BY USING 3D POINT CLOUDS

Yusuf Gökhan Yavuz

Computer Engineering

Supervisor: Asst. Prof. Dr. Övgü ÖZTÜRK

September 2013, 55 Pages

Object recognition systems are one of the most important part of the intelligent systems. Object recognition has led many developments on different areas and industrials such as medical, military, national security, banking, cultural heritages, ... etc.. In this study, we work on a cultural object to recognize a 3D object in a 3D point clouds by using ICP. This method based on comparing minimal distances between points on the 3D object and points on the sliced parts of 3D point clouds. Afterwards we iteratively calculate rotation and translation matrix for minimizing the distances between two compared object until we find the two closest matching. One of the most important advantage of using ICP for 3D object recognition is getting fast results and partial matching. We evaluate ICP on four different point clouds and 3D objects, and show that it effectively work on different type of point clouds and 3D objects.

Keywords: Object Recognition, 3D Matching, Point Cloud, ICP

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The rapid evolution in technology (hardware and software) gives huge oppurnities to not only reasearchers or company also regular users to acquire, create and manipulate 3D models. The 3D modelling tools such as MAYA, 3ds Max, AutoCAD and sculpting systems, make it easier to users to create 3D models directly on the computers. On the other hand the 3D digitizing tools for digitizing 3D objects from the real world, and include 3D scanner machines (Leica, LiDAR, Microsoft Kinect …), registration from range data, automatic modelling from multi-view video, etc. Thus increasing of 3D models both on the internet and standalone leads robust and efficient technique for searching, finding or recognizing 3D models or 3D objects in a data set.

3D model search could be design by using a textual based which can be done by giving user's desired name to models which identifies the semantic meaning of the desired model or class of models. However this method is not useful for model searching. For example an user might give objects' name as "Ferrari" but another user might use "Luxury Car" for the same model. So if a common user who wants to find "Car" could not find either "Ferrari" or "Luxury Car". It is also impossible to use same annotating for every user.

Currently, popular approaches in object recognition focus on two trends: one is the appearance-based methods (Murase and Nayar, 1995; Fergus et al., 2006) and the model-based methods (Gardner and Lawton,1996; Romdhani et al., 2002).

Figure 1.1 Different photos of same car and different angle of light source (Tingbo Hou,Sen Wang et al.,2011)

In appearance-based methods, objects are typically represented by a group of feature vectors. A set of positive and negative examples of classifier spanning on the principle conponent analysis subspace or feature subspace is accepted to train. In practice, technical issues occur from appearance variation due to different pose and lightings. Model-based methods require a set of 3D models to provide geometric constraints. When object place is determined, usage of 3D models is to solve the problem of next matching. However, it stands on two basic assumptions: first, the 3D model can precisely fit to the input images. Second, pose estimation is accurate enough. To estimate appearance of objects, global and local clues have been used to simulate texture of the 3D model. Despite the progress, it still has limited success in illumination variations, since illumination conditions can dramatically affect appearances as shown in Figure 1.1

Also in 3D model object recognition, there are many challenges which are listed in the following.

(1) Automation: When request a 3D model, the 3D model should be automatically find in 3D data.

(2) Efficiency: 3D object recognition should be efficient. 3D model object recognition should be fast especially in small data.

(3) Scope: 3D object recognition should work well in various kinds of 3D models.

(4) Strongness: 3D object recognition should be developed against geometric processing, such as similarity transformation (translation, rotation and scaling), connectivity changes (remeshing,sub-division and simplification), model degeneracy (missing, wrongly oriented, intersecting, disjoint and overlapping polygons), random noise, smoothing, deformation, and posture changing, etc.

(5) Discrimination: 3D object recognition should be sensitive to preserve important distinctions through 3D models.

2 3D OBJECT RECOGNITION

Object recognition systems are deeply rooted component that are built into modern intelligent systems. Research on object recognition algorithms has led to advances in factory and office automation through the creation of optical character recognition systems, assembly-line industrial inspection systems, as well as chip defect identification systems. It has also led to significant advances in medical imaging, defence and biometrics.

First apperance of recognition systems also appeared in biomedical research for the chromosome recognition task (G. Gallus, et al.1968,1974). Even though people did not understand importance of the, its importance became clearer later. Recognition technologies are also successfully used in the food industry (e.g., for the automated classification of agricultural products (A. Jimenez, et al., 2000), the electronics and machinery industry (for automated assembly and industrial inspection purposes (E.N. Malamas, E.G.M. Petrakis, M. Zervakis, et al. 2003), and the pharmaceuticalindustry (for the classification of tablets and capsules) (M. Ejiri, 2007). Many of the models used for representing objects are also effectively employed by the medical imaging autorithy for the robust segmentation of anatomical structures such as the brain and the heart ventricles (T. McInerney, D. Terzopoulos, 1996) (A. Andreopoulos, J.K. Tsotsos, 2008). Handwritten character recognition systems are also employed in mail sorting machines as well as for the digitization and automated indexing of documents (O.D. Trier, A.K. Jain, T. Taxt, 1996) (S. Mori, H. Nishida, H. Yamada,1999). Furthermore, traffic monitoring and license plate recognition systems are also successfully used (K. Takahashi, T. Kitamura, et al, 1996) (C.-N. Anagnostopoulos, I. Anagnostopoulos, et al. 2008) as are monetary bill recognition systems for use with ATMs (M. Ejiri, 2007). Biometric vision-based systems for fingerprint recognition (D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, 2009), iris pattern recognition (K.W. Bowyer, K. Hollingsworth, P.J. Flynn, 2008), as well as finger-vein and palm-vein patterns (C.-L. Lin, K.-C. Fan, 2004) (N. Miura, A. Nagasaka, 2005) have also gained acceptance by the law enforcement community and are widely used.

Despite the success of recognition systems that are feautured for specific task, improved solutions to the more general problem of recognizing complex object classes. They are uncatchable under poorly controlled environments. Furthermore, there is no universal agreement on the definitions of various vision subtasks. Often experienced terms in the literature such as detection, localization, recognition, understanding, classification, categorization, verification and identification, are often ill defined, leading to confusion and ambiguities.

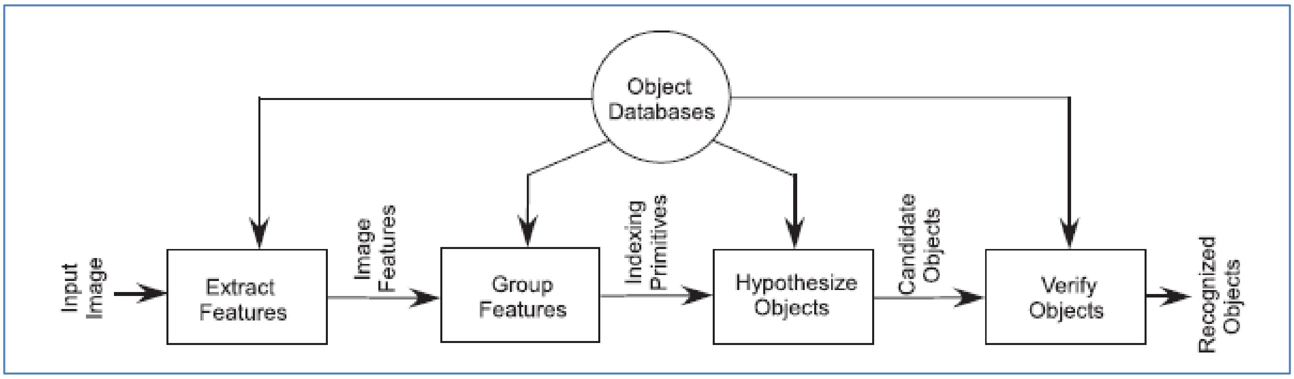


Figure 1.2 The components used in a typical object recognition system (J. Tsotsos ,1992) (S. Dickinson, 1999)

Purpose of this section is to give information about 3 main types of 3D object recognition method. These are RGB, RGB-D and Depth Data. However this is our categorization method on general categorization are recognition using volumetric parts, Automatic programming, Perceptual organization, Interpretation tree search, Geometric invariants, Qualitative 3-D shape-based recognition and deformable models, Function and context, Appearance based recognition, Local feature-based recognition and constellation methods, Grammars and related graph representations (Alexander Andreopoulos, John K. Tsotsos, 2013)

2.1 RGB

The earliest applications based on RGB were pattern recognition systems for character recognition in office automation related tasks (L.G. Roberts,1960) and (J.T. Tippett, D.A. Borkowitz, et al. 1965). Early work by Roberts in the 1960s (L.G. Roberts, 1963) first identified the need to match two-dimensional features extracted from images with the three-dimensional representations of objects. Other applications were built for chromosome recognition and the analysis of aerial images.

Given more details about choromosome recognition which is closely related to medical area let's explain what is choromosome and why it is important. A chromosome is an organized structure of DNA, protein, and RNA found in cells which contains all information about body. To make a visual examination of a chromosome image for various chromosome abnormalities, individual chromosome regions have to be determined in the subject image and classified into the distinct chromosome types image chromosome images for various chromosome abnormalities plays an important role in many clinical practices including treatment and prevention of genetic disorders, radiation dosimetry, toxicology, etc. Usually, the visual chromosome examination requires the following procedures (J.Graham and J.Piper 1994) .

1st Staining a set of chromosomes in a cell nucleus and capturing its image,

2ndDetermining individual chromosome regions in the subject image,

3rd Classifying the determined regions into the 24 distinct chromosome types (1, 2, . . . , 22, X, and Y).

With proper staining methods (e.g. G-banding method etc.),a characteristic series of light and dark bands appears along the longitudinal axis of a chromosome. The band appearance on a chromosome is called a band pattern, and it is unique to each type of chromosome. For determining and classifying the chromosome regions in an image, individual chromosome regions are extracted from the subject image, the longitudinal profile of intensity on each region is acquired as its band pattern, and the region is classified according to the band pattern. (J.Graham and J.Piper, 1994) (A. Carothers and J. Piper, 1994) (Q.Wu, Z. Liu et al. Castleman, 2005) (M. Moradi and S.K. Setarehdan,2006)

With proper staining methods (e.g. G-banding method etc.),a characteristic series of light and dark bands appears along the chromosome. The band appearance on a chromosome is called a band pattern, and it is unique to each type of chromosome. For determining and classifying the chromosome regions in an image, individual chromosome regions are substracted from the subject image, profile of intensity on each region is achieved as its band pattern, and the region is classified according to the band pattern. (J.Graham and J.Piper, 1994) (A. Carothers and J. Piper, 1994) (Q.Wu, Z. Liu et al. Castleman, 2005) (M. Moradi and S.K. Setarehdan,2006)

Later applications led to progress in pattern recognition, feature detection and segmentation but dealt with objects of a different type. These later approaches are closely related to modern 2D appearance based object recognition research.

Color provides powerful information for object recognition. A simple and effective recognition scheme is to represent and match images on the basis of color histograms as proposed by Swain and Ballard (M.J. Swain, D.H. Ballard, 1991). The work makes a very helpful in introducing color for object recognition. However, it has the drawback that when the illumination circumstances are not equal, the object recognition accuracy degrades significantly. This method is extended by Funt and Finlayson (B.V. Funt, G.D. Finlayson, 1995), based on the retinex theory of Land (E.H. Land, J.J.

McCann, 1971), to make the method Pattern recognition illumination independent by indexing on illumination fixed surface descriptors (color ratios) computed from neighboring points. However, it is assumed that neighboring points have the same surface normal. Therefore, the derived illumination fixed surface descriptors are negatively appected by rapid changes in surface orientation of the object (i.e. the geometry of the object). Healey and Slater (G. Healey, D. Slater, 1995) and Finlayson et al. (G.D. Finlayson, S.S. Chatterjee, B.V. Funt, 1996) use illumination fixed moments of color distributions for object recognition.

These methods are sensitive to stopped object and complicated as the moments are defined as an inseperable attiribute on the object as one. In global methods, in general, occluded parts will disturb recognition. Slater and Healey (D. Slater, G. Healey, 1996) to get over this problem by computing the color features from small object regions instead of the entire object.

The choice which color models to use does not only depend on their strongness against varying illumination across the scene (e.g. multiple light sources with different spectral power distributions), but also on their strongness against changes in surface orientation of the object (i.e. the geometry of the object), and on their strongness against object obturation and cluttering. Furthermore, the color models should be brief, differential and strong to noise.

2.2 RGB-D

RGB-D cameras such as Prime Sense, Microsoft Kinect, … etc. are an emerging trend of technologies that provide high quality synchronized depth and color data. Using active sensing techniques, robust depth estimation is able to be done real time. Microsoft Kinect, orginally designed for video games afterwards it turns into a depth camera that has made it into consumer applications. It is a huge success with comprehensive implications for real-world visual perception. One key area of depth camera usage is in object recognition, a fundamental problem in computer vision and robotics.

Figure 2.1 (left) RGB image and (right) depth information captured by an RGB-D camera. Recent systems can capture images at a resolution of up to 640x480 pixels at 30 frames per second. White pixels in the right image have no depth value, mostly due to occlusion, max distance, relative surface angle, or surface material. (Manuel Blum, Jost Tobias Springenberg, Jan W¨ulfing and Martin Riedmiller, 2012)

During the last decades a dozen of different feature extraction methods have been designed for object recognition tasks in computer vision community. These methods mostly use a fixed grid or extract from local image patches around detected interest points. The most important approaches of these are based on orientation histograms such as SIFT (David G. Lowe, 2004) and SURF (Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, 2006). These methods are really hard to design or implement to other enviroment however these methods was used too many application. A histogram based on generalizing feature called kernel description Figure 2.1 which give a general design pattern for local feature responses and give us additional information in a different way.

To learn low level feature from a data has solved by several different ways. The work on deep belief networks (Geoffrey E Hinton, 2007) and deep autoencoders (Dan C. Ciresan, Ueli Meier, Jonathan Masci, et al, 2011), (Quoc V Le, Jiquan Ngiam, Zhenghao Chen, et al. 2010) resulted in object recognition architectures that can achieve on several benchmarks. Sparseness of the learned feature can be representate such as sparse coding (Adam Coates, Andrew Y. Ng, and Serra Mall, 2011), and local coordinate coding (Kai Yu and T. Zhang, 2010), they have been successfully implement to object recognition duties. Also another interesting method which is unsupervised feature was developed by Coates et al. on learning (Adam Coates, Honglak Lee, and Andrew Y. 2011).

Flynn and Jain (P. Flynn and A. Jain, 1992) describe an approach for 3D to 3D object matching using invariant features indexing. Solid models of objects composed of cylinders, spheres, planes are used to determine corresponding triples $\{(M_1, S_1), (M_2, S_2), (M_3, S_3)\}$ where M_i represents a model surface and S_i represents a surface of corresponding scene. For each pair of extracted scene cylinders, spheres and planes, an invariant feature is defined and extracted. For example, for each pair of cylinders and planes the angle between the plane's normal and the cylinder's axis of symmetry is removed. Pairs or triples of such invariant features are used to access tables where each table entry contains a linked-list of all the database object models composed of the same invariant features. The table contains votes for each object. As a result the most voted object is recognized by system.

Hoiem et al. (D. Hoiem, A. A. Efros, and M. Hebert, 2006) use probabilistic estimates of 3D geometry of objects relative to other objects in the scene to make estimates of the similarity of the various object hypotheses. Given an exampe to the system, if a current hypothesis detects a person and a building in the scene or in the image, However the hypothesis assumes that a person taller than the building. Their approach can be united as a ‘‘wrapper’’ method around any object detector. Markov random fields (MRFs) are also a popular method for incorporating contextual information via spatial dependencies in the images (S. Z. Li. Markov, 2001). In more recent work, Kumar and Hebert (S. Kumar and M. Hebert, 2003) use Discriminative Random Fields (DRFs), an extension of MRFs, for merging similar scene interactions. The most important advantage of DRFs is their ability to flexible the conditional independence hypothesis of MRFs. A few researchers use the statistics of bags of localized features (edges, lines, local orientation, color, etc.) to determine the likely distribution of those features depending on the scene or current context (Torralba et al., 2003), (Wolf and Bileschi,2006), (Siagian and Itti, 2005).

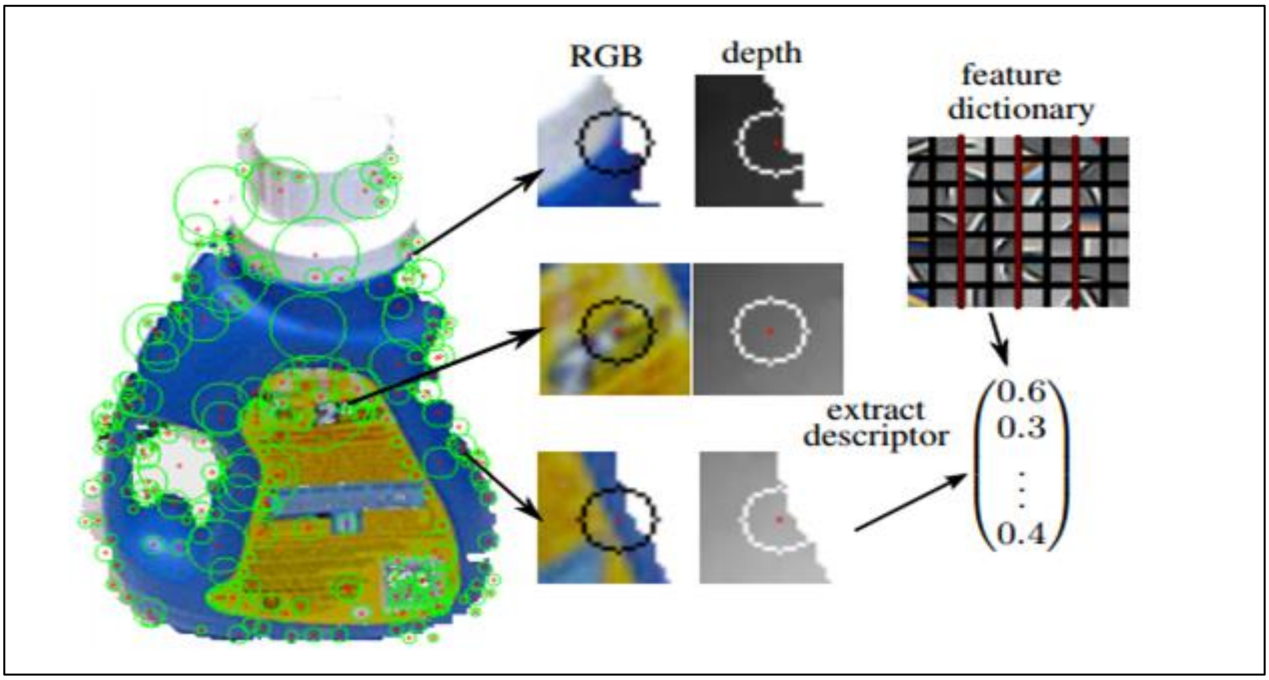


Figure 2.2 The general descriptor extraction procedure for the convolutional k-means desriptor. First, a set of interest points is detected in the input image. Around each interest point a 16x16 px area is extracted. To build the feature descriptor for an image point feature responses from image patches 6x6 px within this area are compared using the learned feature dictionary. (Manuel Blum, et al. 2012)

A Learned Feature (Manuel Blum, et al. 2012)'s method consider a specific recognition setting in which the objects are represented using high resolution RGB-D data and propose to extract a feature histogram descriptor combining information from all 4 channels. To make their approach scalable to high resolution images they adapt the standard setting used by Hessian based approaches and chose to extract their learned feature responses around interest points, effectively substituting the hand designed Hessian descriptors. The descriptor is built from features, which are learned via a k-means approach that is adapted from the previously mentioned work in (Adam Coates, Honglak Lee, and Andrew Y. , 2011) Figure 2.2 their work is similar to work on kernel descriptors (Liefeng Bo, Xiaofeng Ren, and Dieter Fox, 2011) in which a descriptor is built by comparing pixel orientations or color intensities. However, in contrast to this approach they did not explicitly design the used feature responses using pixel comparisons, but decided to learn a representative set of features which is then compared to the vicinity of the detected interest point.

2.3 Depth Data

Retrieval of data based on shape has been studied in several fields, including computer vision, computational geometry, mechanical CAD, and molecular biology.

3D shape retrieval methods can be roughly subdivided into three categories: (1) methods that first attempt to derive a high-level description (e.g., a skeleton) and then match those, (2) methods that compute a feature vector based on local or global statistics, and (3) miscellaneous methods.

Give an example of the first method might be the medial scaffolds (Ming-Ching Chang, Benjamin B. Kimia, 2011) This method typically a major branch in shape representation is the symmetry-based medial axis (MA) representation (K. Siddiqi, S. Pizer (Eds.), 2009) and (H. Blum, 1973). The MA is promising for shape recognition (T. Sebastian, P. Klein, B. Kimia, 2004) and (K. Siddiqi, J. Zhang, et al. 2008) in that (i) it organizes the shape information in a hierarchical, intrinsic graph-like structure (F. Leymarie, B. Kimia, 2007), which enables matching parts of deformed shapes naturally, and (ii) such information captured with the MA is complete in that a full shape reconstruction is always possible (P. Giblin, B. Kimia, 2003). Despite these advantages, the MA is generally sensitive to perturbation and difficult to model in the 3D case (D. Attali, J.-D. Boissonat, H. Edelsbrunner, 2004). Such issues have been recently addressed (F. Leymarie, B. Kimia, 2007). Medial Scaffold (MS) a hierarchical organization of the 3D MA into a hypergraph form (M.-C. Chang, B. Kimia, 2008) and a regularization framework of the MS to deal with the above barriers. The MA instabilities which induce sudden topological changes are formally classified as a set of transitions and thus can be regularized via a set of transforms (M.-C. Chang, B. Kimia, 2008) They proposed to match the regularized MS such as the ones shown in Figure 2.3 to estimate a global similarity between shapes.

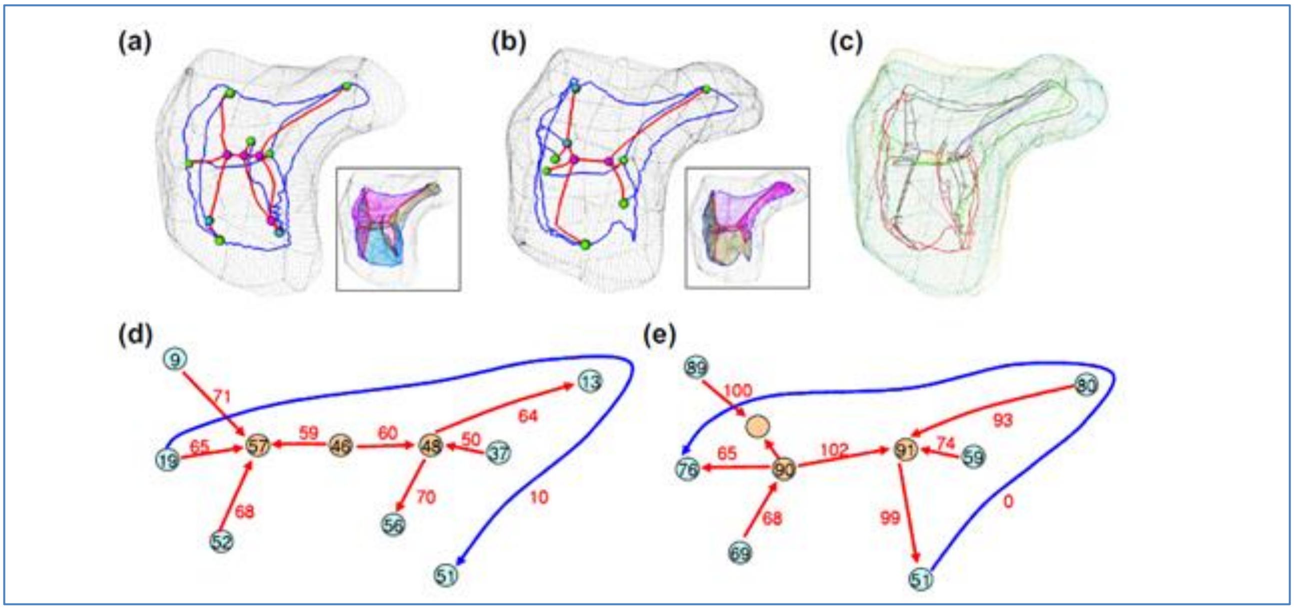


Figure 2.3 shows that the matching of the MS hypergraphs of two carpal bones in (a) and (b) is shown in (c). (d and e) show a manual correspondence, where the graph components are labeled with identification numbers. (Ming-Ching Chang, Benjamin B. Kimia, 2011)

Give an example of the second method might be Shape Distributions method. Main idea is to represent the marker of an object as a shape distribution sampled from a shape function that is measuring global geometric properties of an object. First duty for this approach is to decrease the shape matching problem to the comparison of probability distributions, which is simpler than traditional shape matching methods that require pose registration, feature correspondence, or model fitting. the diversity between sampled distributions of simple shape functions (e.g., the distance between two random points on a surface) provide a robust method for disjunctive between classes of objects Figure 2.4 and Figure 2.5 (e.g., cars versus airplanes) in a moderately sized database, despite the presence of optional translations, rotations, scales, mirrors, tessellations, simplifications, and model corruption. They can be estimated quickly, and thus the proposed method could be applied as a pre-classifier in an object recognition system or in an interactive content-based withdrawal application.

The shape functions are;

- A3: Measures the angle between three random points on the surface of a 3D model.
- D1: Measures the distance between a fixed point and one random point on the surface. We use the centroid of the boundary of the model as the fixed point.

- D2: Measures the distance between two random points on the surface.
- D3: Measures the square root of the area of the triangle between three random points on the surface.
- D4: Measures the cube root of the volume of the tetrahedron between four random points on the surface.

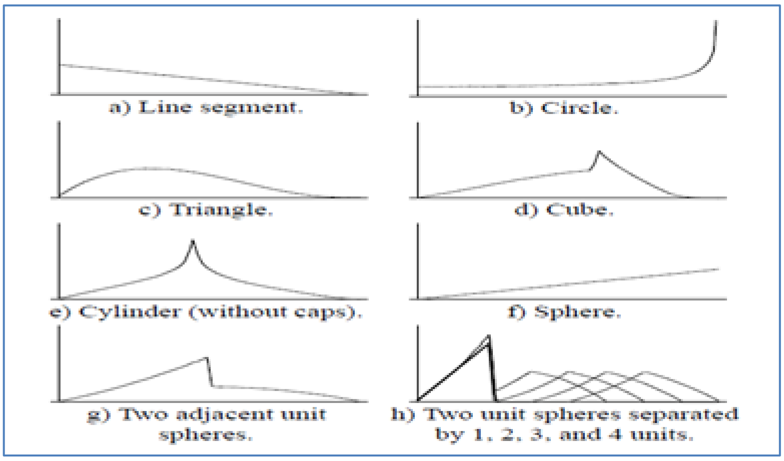


Figure 2.4 Example D2 shape distributions, in each plot, the horizontal axis represents distance, and the vertical axis represents the probability of that distance between two points on the surface. (Robert Osada, Thomas Funkhouser, et al., 2002)

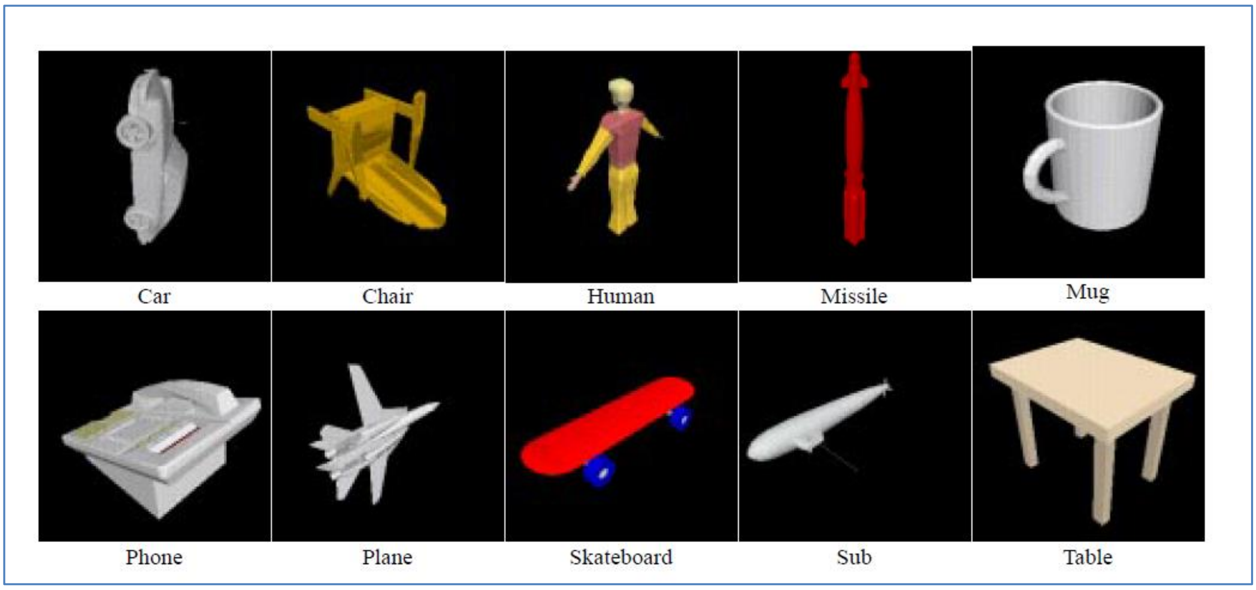


Figure 2.5 chosen object to test based on D2 shape distribution. (Robert Osada, Thomas Funkhouser, et al , 2002)

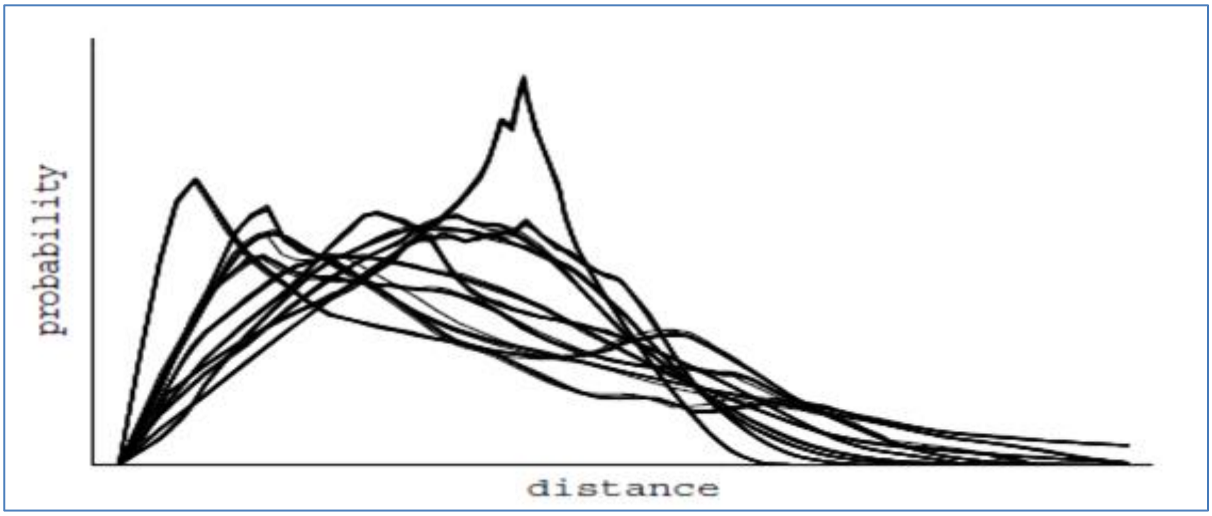


Figure 2.6 D2 shape distributions for seven variants of ten models in Figure 2.5 (Robert Osada, Thomas Funkhouser, et al., 2002)

To give a few examples of the last method might be Prioritized Feature Matching (Y. Li, N. Snavely), Discriminative Sketch-based (T. Shao, W. Xu, et al. 2011) ,Topology Matching method (M. Hilaga, Y.Shinagawa, et al.) , 3D Object Recognition in Range Images Using Visibility Context (E. Kim,2011) Signature-Based Method (S.R. Correa, L. G. Shapiro, M.Melia, 2001)

Each method has advantages and disadvantages for matching 3D objects. In this work we present our novel method which works based on ICP. The reason behind ICP is that more easier than other methods also much faster than others. It can be implement any kind of systems or 3D object to work on. Following chapters we show that ICP can be used not only aligning two 3D object also can be used for partial or complete 3D object matching and recoginizing. Also we evaluated the performance and robustness of ICP.

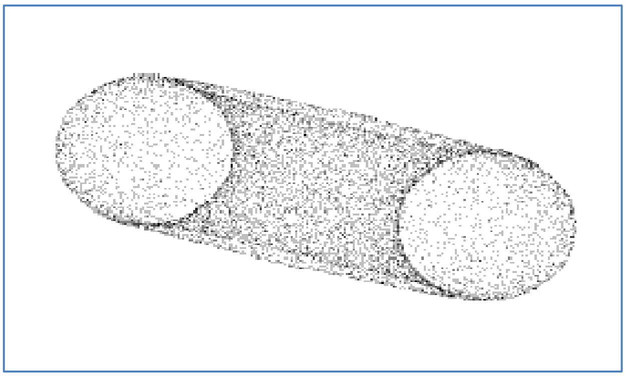
3 3D POINT CLOUD DATA

3.1 What is Point Cloud ?

A point cloud is a set of data points which can be on 2D,3D or more coordinate system. Figure 3.1 show an example of point cloud image. It also contains information about point’s face. For example in a 3D coordinate system, data points represent X, Y, Z and often are intended to represent the external surface of an object.

Nowdays point clouds can be easily created by 3D scanners such as Leica, LiDAR, Microsoft Kinect. There are many purpose to have a point cloud data for example to create 3D models for manufactured parts, cultural heritage, medical purposes, and a multitude of visualization, animation, rendering and mass customization applications.

However point clouds have information about objects. They are not usable for many application so point clouds can be converted to a 3D surface by using Delaunay triangulation, alpha shapes, and ball pivoting, build a network of triangles over the existing vertices of the point cloud, while other approaches convert the point cloud into a volumetric distance field and reconstruct the implicit surface so defined through a marching cubes algorithm.



,

Figure 3.1 A point cloud image of a torus.

3.2 Saltanat Gate

Dolmabahçe Palace was built by Sultan Abdulmecid (1839-1861) who was the thirty first Ottoman Sultan. The palace, whose construction commenced on June 13th, 1843, was brought into use on June 7th, 1856, upon completion of surrounding walls. The palace mainly consists of three parts, named as the Imperial Mabeyn (State Apartments), Muayede Salon (Ceremonial Hall) and the Imperial Harem. Saltanat Gate locates at Muayede Salon.

We used Leica laser scanner to get 3D points data of Saltanat Gate. This scanner able to scan 270 degree horizontly and 360 degree vertically and get up to 500,000 points/sec. To complete whole Saltanat Gate we put the scanner different locations. Figure 3.2 shows that where we located and scaned the gate.

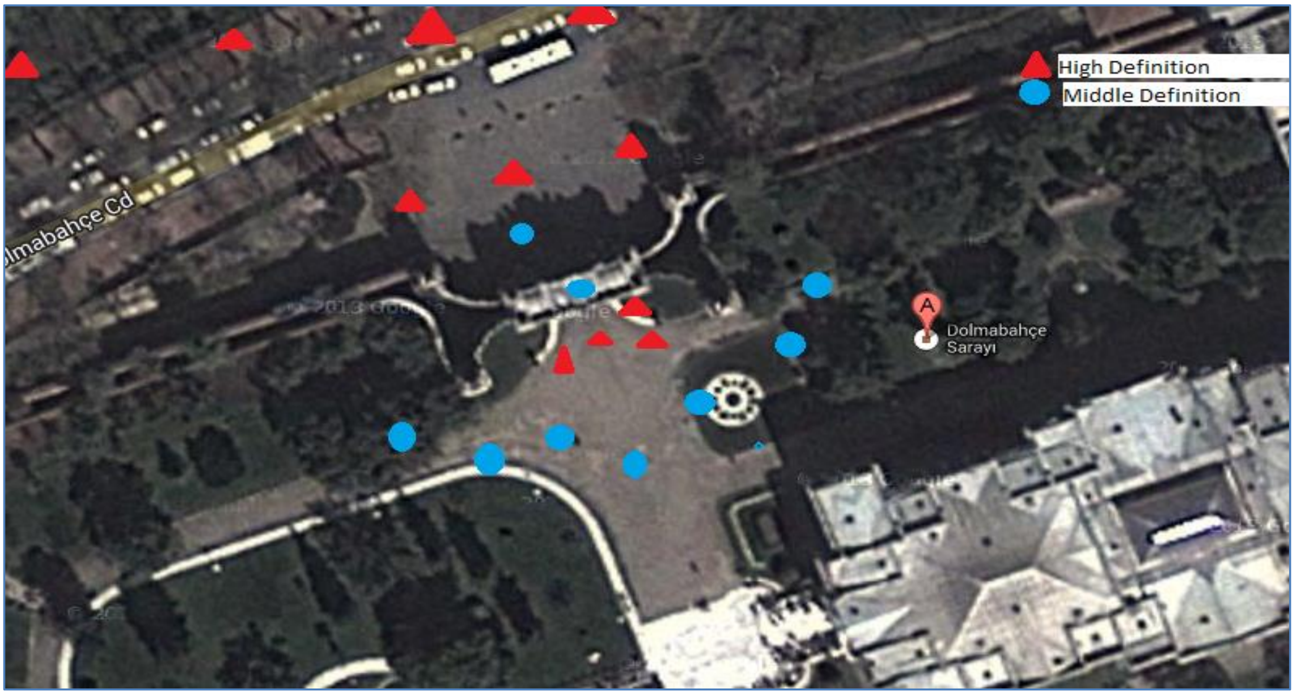


Figure 3.2 shows the location of scanning places and scanning rate.

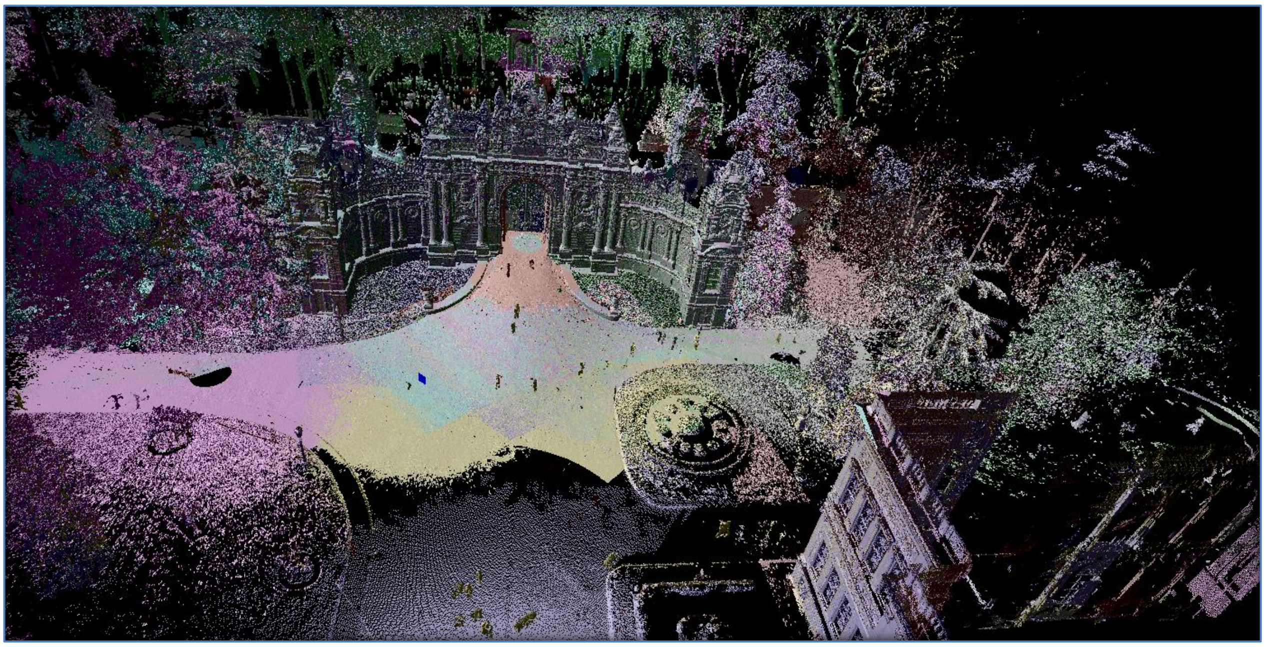


Figure 3.3 shows that how it looks like all scan completed and alligned.

After getting the 3D points data as ptx format we converted them into ply format by using Leica Cyclone software. Ply is a format for storing graphical objects that are described as a collection of polygons which is much more easy to handle and friendly format to other software such as MAYA, Blender, Meshlab.

Figure 3.4 A photo of Saltanat Gate

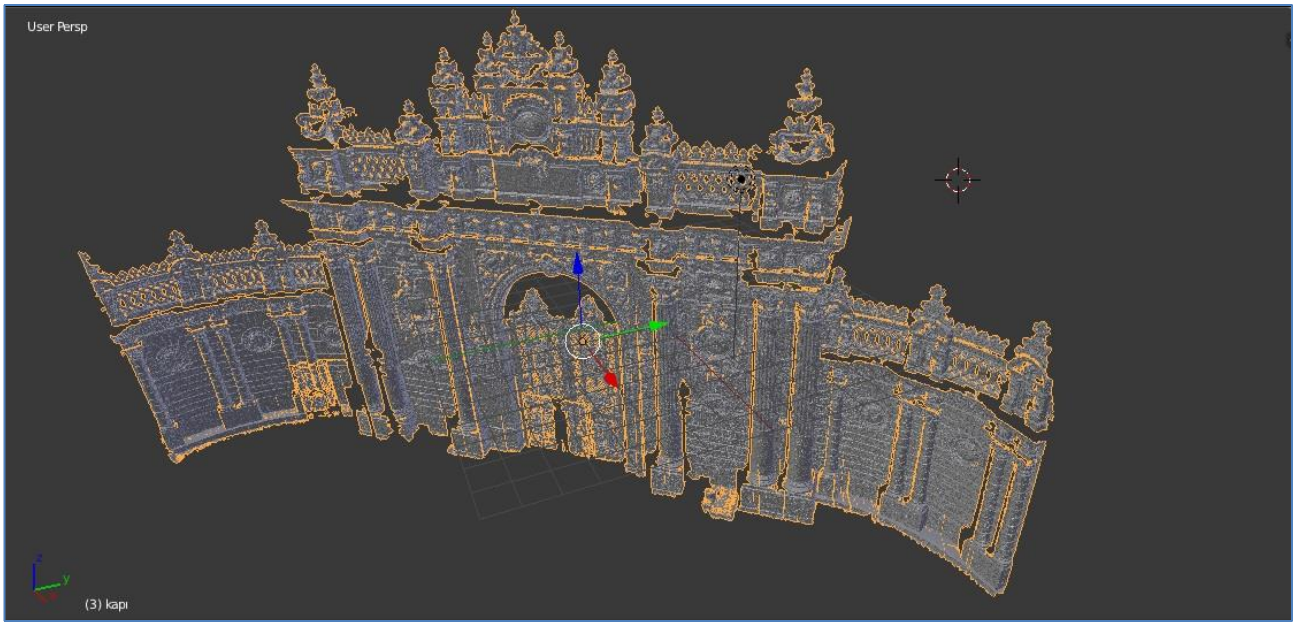


Figure 3.5 3D point clouds of Saltanat Gate which was scanned from just one position

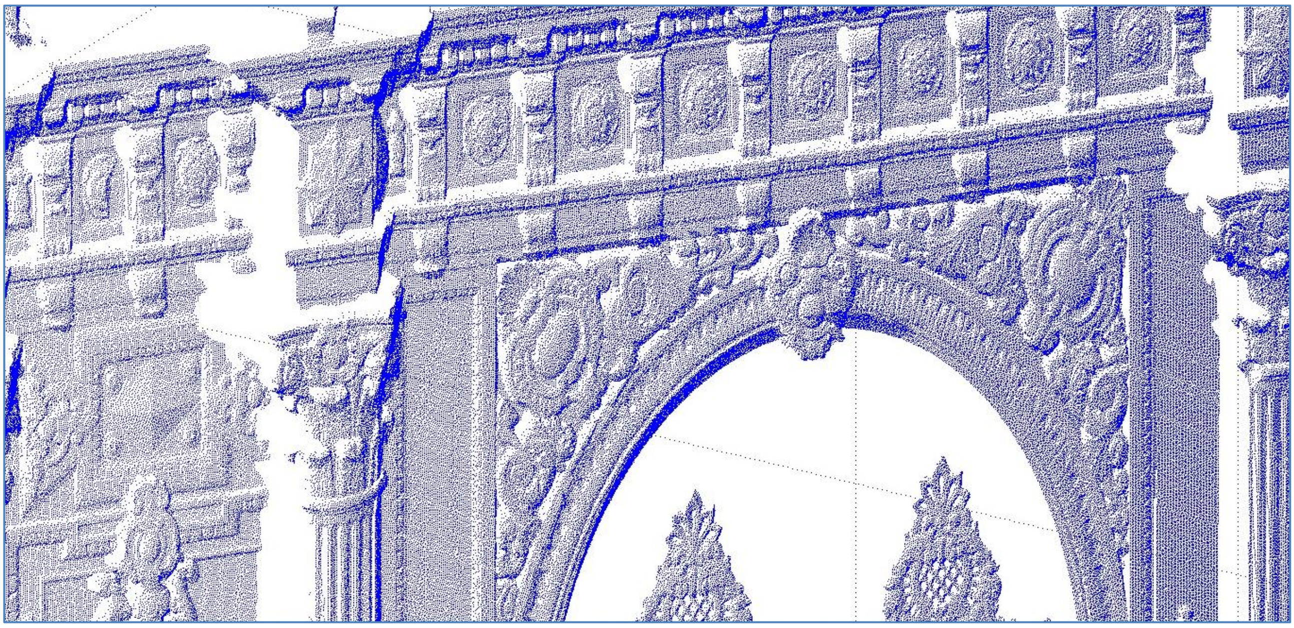


Figure 3.6 Close lookup 3D point clouds of Saltanat Gate which was scanned from just one position

4 3D OBJECT RECOGNITION BY USING ICP

Object recognition is the ability to perceive an object's physical properties (such as shape, colour and texture) and apply semantic attributes to the object, which includes the understanding of its use, previous experience with the object and how it relates to others. (Enns, J. T., 2004)

4.1 What is ICP ?

The ICP Algorithm was developed by Besl and McKay (P. Besl and N. McKay,1992) and is usually used to register two given point sets in a common coordinate system. The algorithm calculates iteratively the registration. In each iteration step, the algorithm selects the closest points as correspondences and calculates the transformation, i.e., rotation and translation (R,t), for minimizing the equation

$$E(R,t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \left\| m_i - \left(Rd_j + t\right) \right\|^2$$

where N_m and N_d, are the number of points in the model set M and data set D, respectively, and $w_{i,j}$ are the weights for a point match. The weights are assigned as follows: $w_{i,j}$= 1, if m_i is the closest point to d_j, $w_{i,j}$= 0 otherwise. Equation can be reduced to

$$E(R,t) \propto \frac{1}{N} \sum_{j=1}^{N} \left\| m_i - \left(Rd_j + t\right) \right\|^2$$

with

$$\mathrm{N} = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j}$$

,since the correspondence matrix can be represented by a vector v containing the point pairs, i.e.,

v = $\left(d_{1,} m_{f(d1)}\right), \left(d_{2,} m_{f(d2)}\right), \ldots, \left(d_{N_d,} m_{f\left(d_{N_d}\right)}\right)$, with $f(x)$ the search function returning the closest point. The assumption is that in the last iteration step the point correspondences, thus the vector of point pairs, are correct.

In each ICP iteration, the transformation can be calculated based on these four methods: A singular value decomposition based method of Arun et al.(K. S. Arun, T. S. Huang, and S. D. Blostein, 1987) a quaternion method of Horn (B. K. P. Horn. 1987), an algorithm using orthonormal matrices of Horn et al. (B.K. P. Horn, H. M. Hilden, et al., 1988) and a calculation based on dual quaternions of Walker et al. (M. W. Walker, L. Shao, and R. A. Volz, 1991). These algorithms show similar performance on noisy data (A. Lorusso, D. Eggert, and R. Fisher, 1995).

Arun et al. observed that "the computer time requirements for the SVD and (unit) quaternion algorithms are comparable" (B. K. P. Horn. 1987). In a paper not directly related to any of the methods, Zhang implemented both of the quaternion algorithms and found that "they yield exactly the same motion estimate" (B. K. P. Horn. 1987). Also, he found these two techniques to be more efficient than an iterative technique based on the extended Kalman filter that he developed. Finally, Walker et al. stated that "the two algorithms produce the same rotation errors for the translation errors, the DQ algorithm exhibits better performance than the SVD algorithm" (M. W. Walker, L. Shao, and R. A. Volz, 1991). The thorough and unbiased comparison presented here will clarify, extend (and even refute) some of these previous findings

4.2 Using ICP on 3D Data

ICP based on finding minimal distances between closest points and then rotates and transforms the data thus this methods takes long time work on huge data such as Salanat Gate which has 2770070 points. Instead of working huge data it is better of slice the data and then work on it. One of the big consideration is to divide 3D point clouds data. Because as we worked on ply file format we did not know how our looking object locates on 3D point cloud. It is not important as long as we slice the 3D point cloud correctly beacuse ICP is not affected by rotation or transformation but How can be sure of that we sliced correctly? First we decided to display our 3D scanned point cloud. Earlier ICP method did not give good results because we sliced it wrong. Figure 4.1 and Figure 4.2 shows that how we previously sliced 3D point cloud object that led us wrong object matching and object recognition.

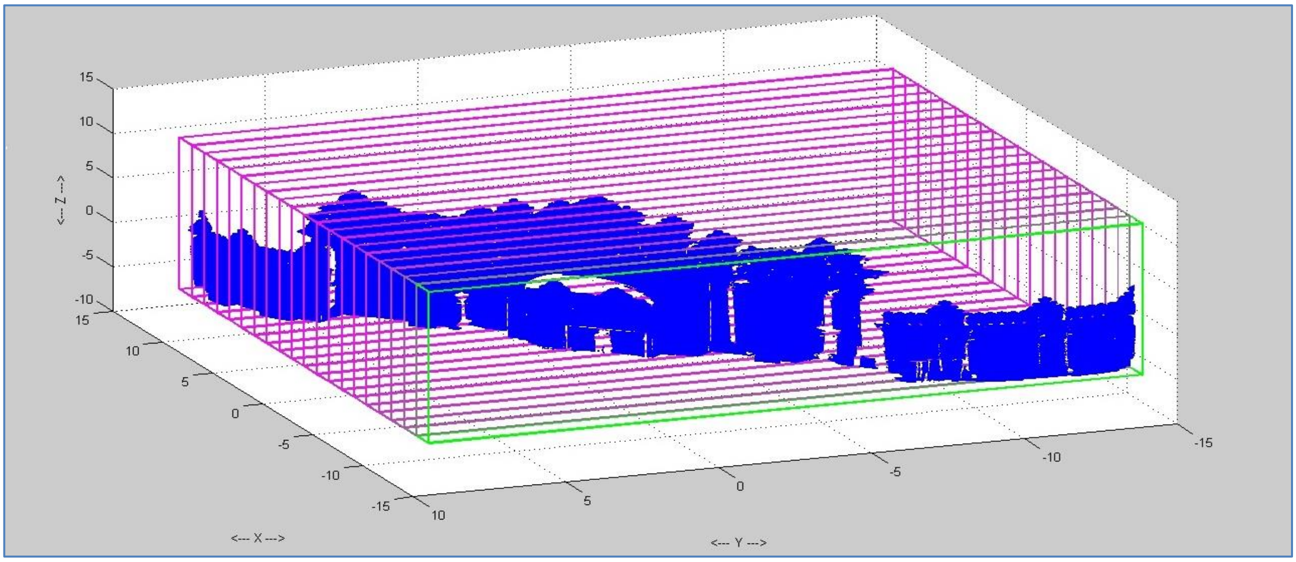


Figure 4.1 shows that how we sliced the object based on X axes

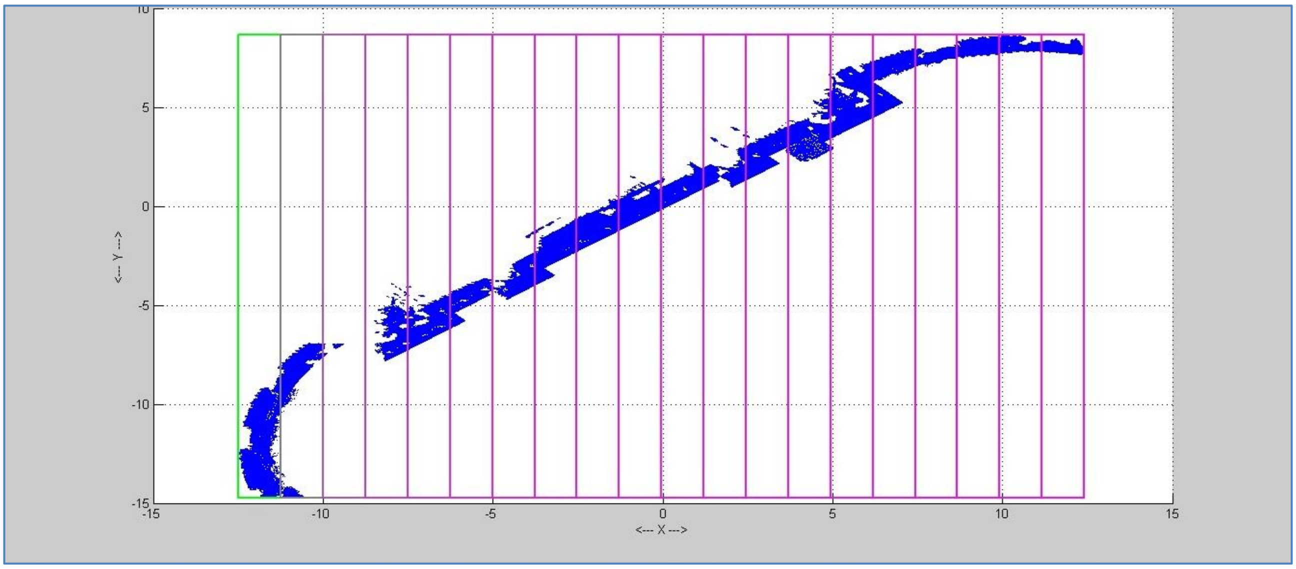


Figure 4.2 shows top view of how we sliced the object based on X axes

The 3D point cloud data must be rotated as same as desired object. There are too many methods to rotate the object. M. Chaouch, A. Verroust's (M. Chaouch, A. Verroust, 2008) method is one of best automated alligmanted method for desired results. It also possible to rotate and trasnform the object by using Meshlab, Maya, … etc. After rotating the object is ready to recognize for desired 3D object Figure 4.4. First step is dividing the 3D point cloud data as close as desired object. Main reason for dividing 3D point cloud to sub points clouds is to get good results and save the time. We diveded 3D point cloud data as 20 horizontly Figure 4.4 and 20 vertically Figure 4.7 such as 400 sub point clouds. To be safe side we decided to divide big as each sub point cloud at least 3 times bigger than the looking 3D object.

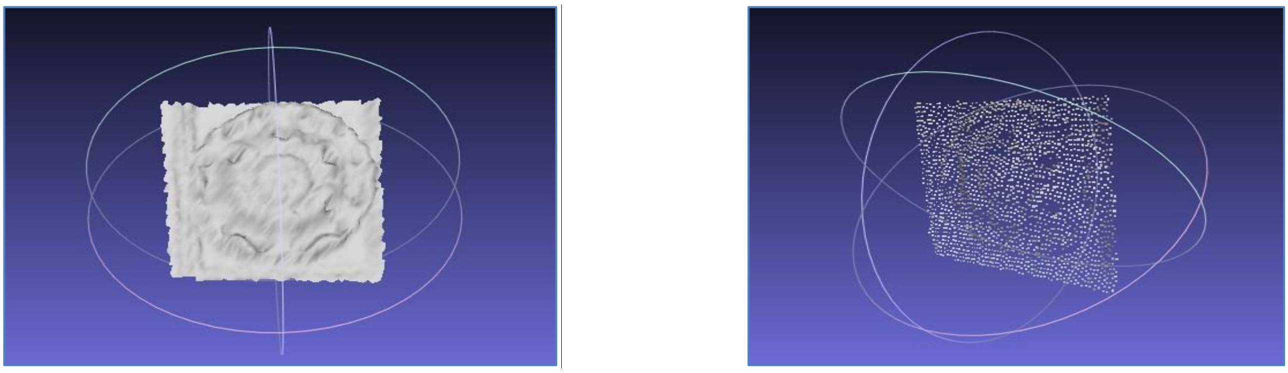


Figure 4.3 This object one part of Saltanat Gate which is desired object. This object as small as approximately 1/1200 of Saltanat Gate.

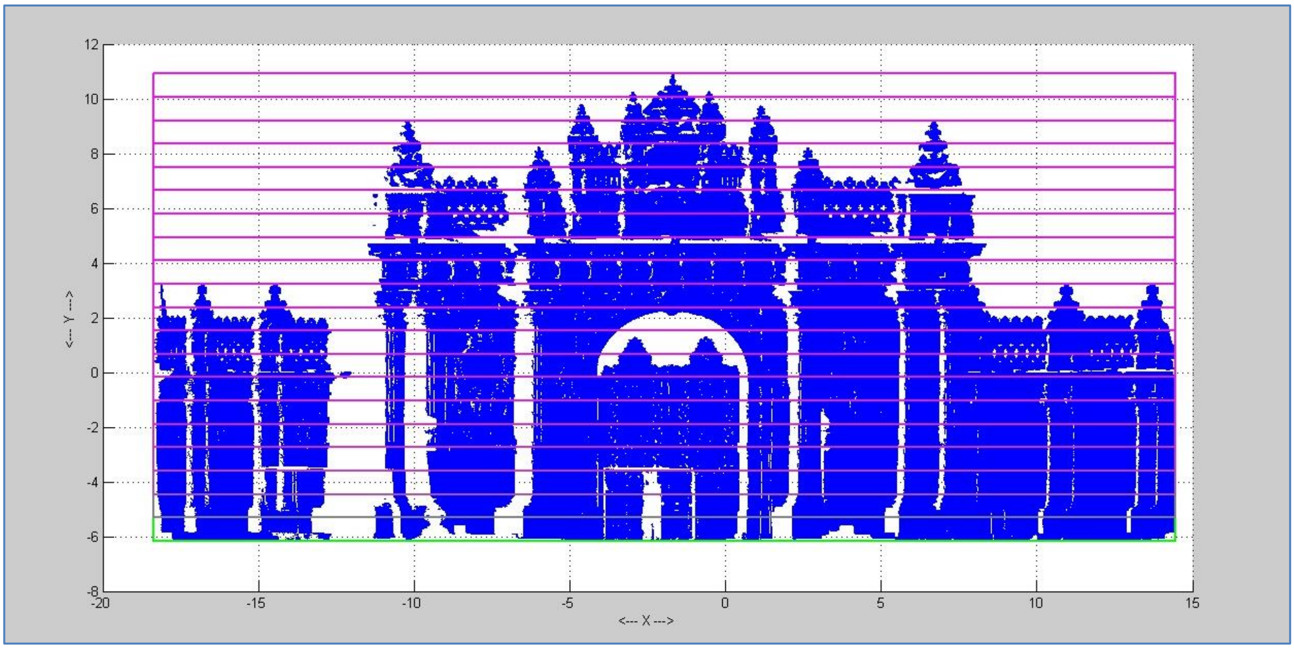


Figure 4.4 horizontly dividing into 20 sub point clouds.

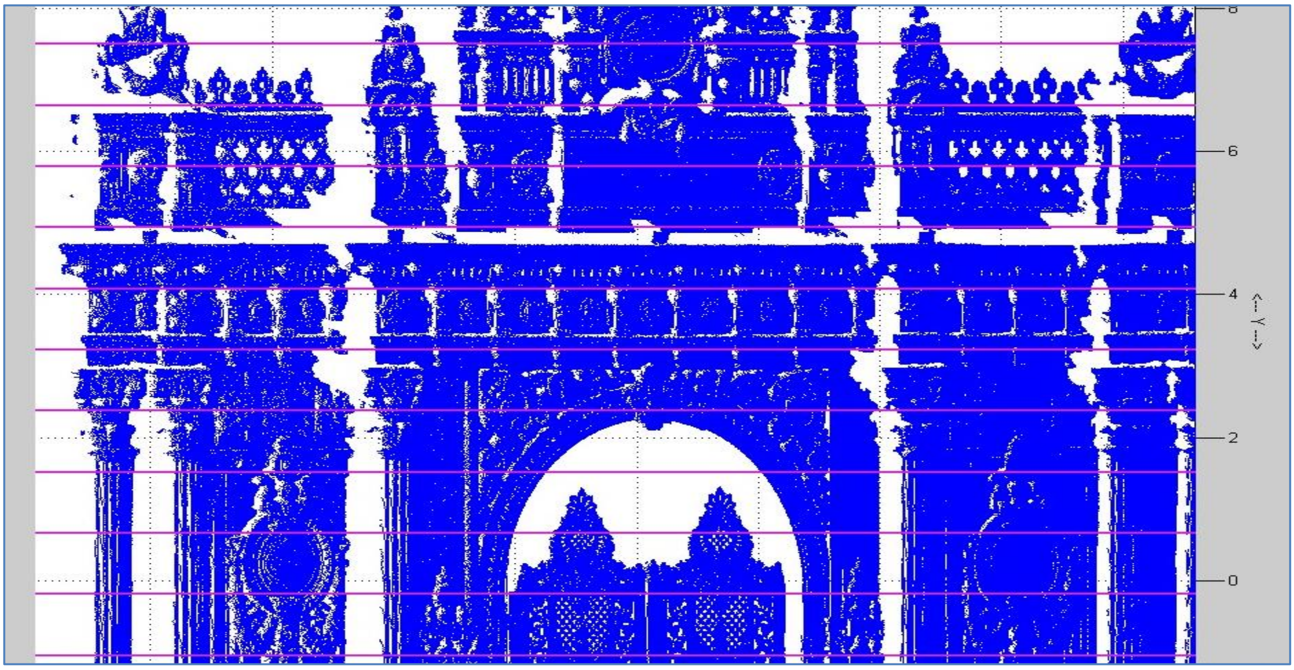


Figure 4.5 close look up of horizontly divided sub point clouds.

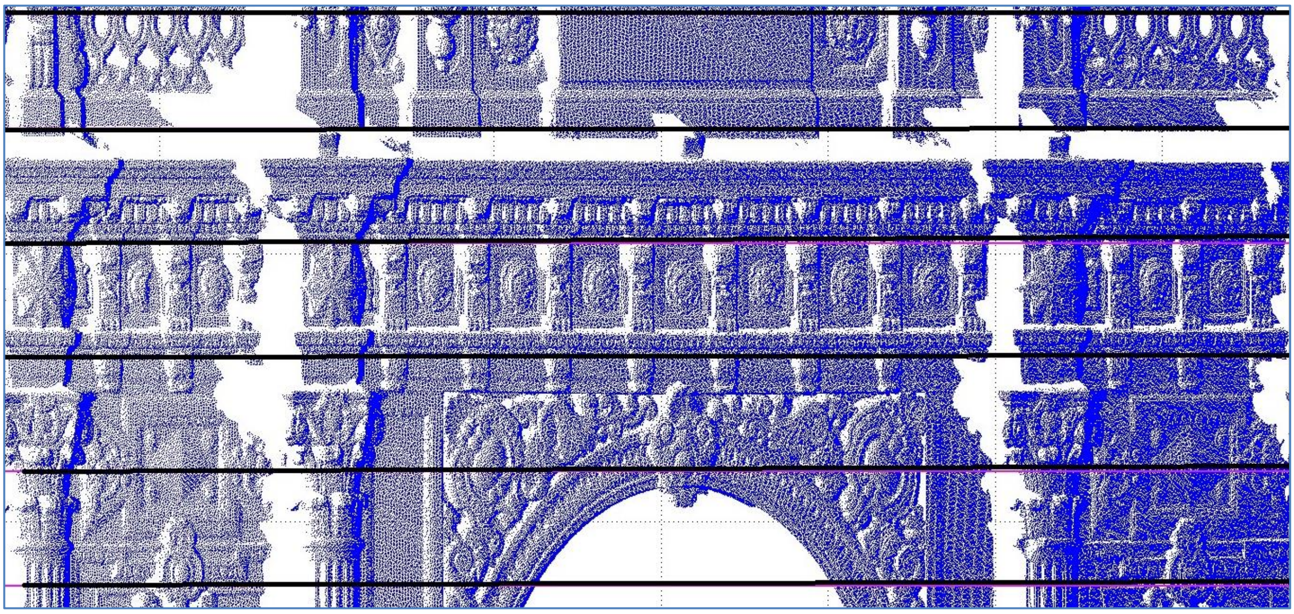


Figure 4.6 Checking desired object to make sure it is not divide by our method.

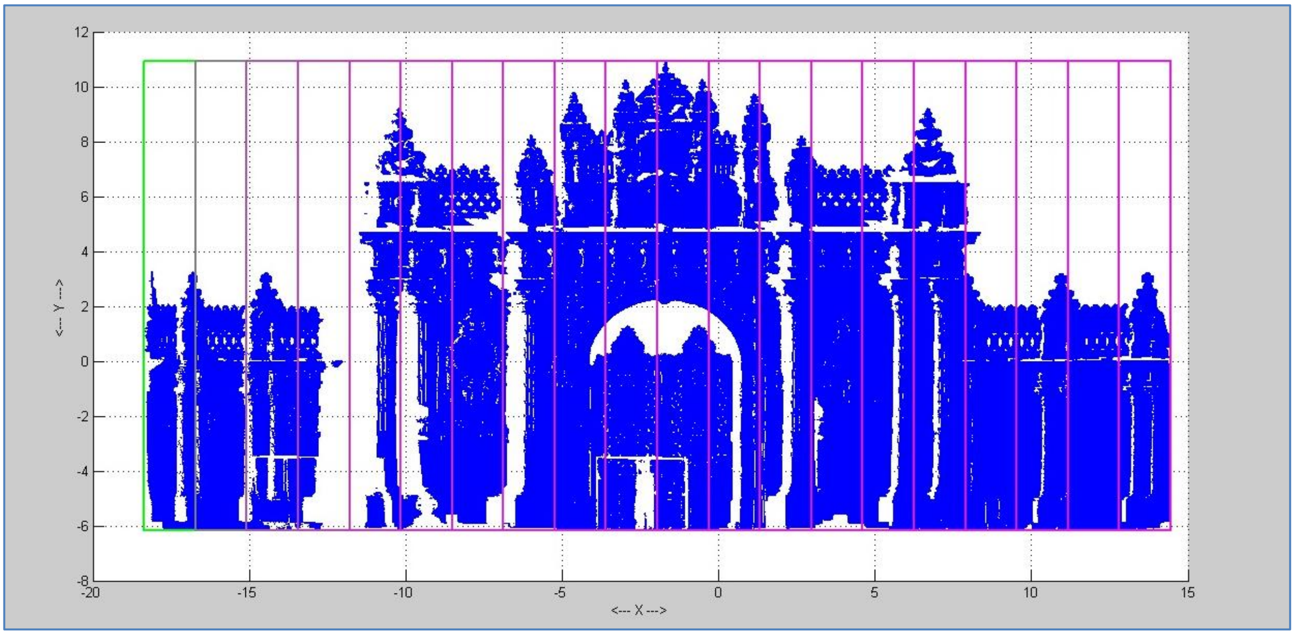


Figure 4.7 verticaly dividing into 20 sub point clouds.

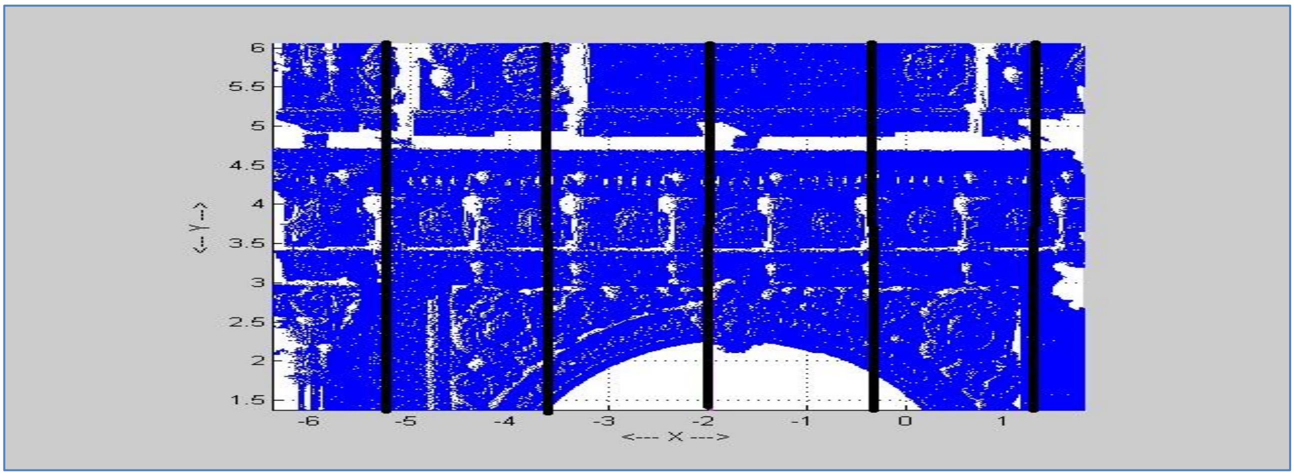


Figure 4.8 close look up of horizontly divided sub point clouds.

After slicing vertically and horizontly the data (Saltanat Gate) turn into small pieces close to desired object.

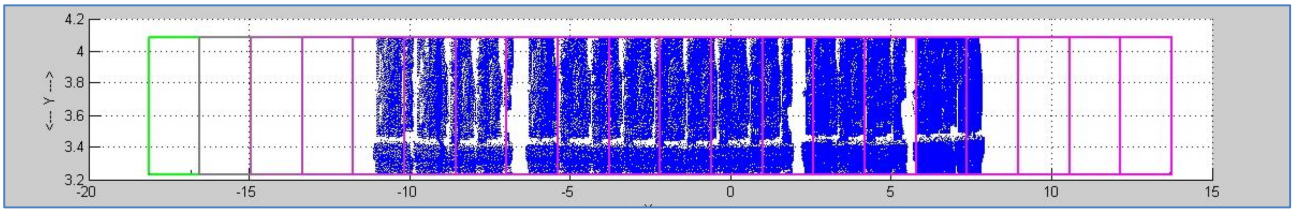


Figure 4.9 Now it is ready to apply ICP [Figure 4.4] on each small pieces.

The algorithm of our ICP is shown below;

Algorithm : ICP

Input: Reference 3D point cloud or 3D object P and input 3D point cloud or 3D object X

Output : Registration result of X and iteration number

Intialization: Set iteration number k=0, maximum iteration numeber to N, ICP convergence tolerance ţ and stoping threshold th_{ε} for the whole process, R^0 and t^0 are set by Principle Compenent Analysis, invariant feature point extraction threshold th_M and increment Δth_M

1st Find the corresponding between P and X with the transformation (R^k, t^k)

2ndCompute mean square error $\overline{d_k}$ of the corresponding points between P and X

3rd **While** ($\overline{d_k} > th_{\varepsilon}$ and k<N)

4th **While** ($\overline{d_k} > \overline{d_{k-1}} >$ ţ)

5th Find the corresponding between P and X with the transformation (R^k, t^k)

6th Apply transformation (R^k , t^k) to the (k-1)th input data X_{k-1} , then $X_k = R^k, X^{k-1} + t^k$

7th **end while**

8th Set thM=thM- Δth_M

9th **end while**

Figure 4.10 shows that how ICP algorithm works on our designed system.

After getting the result of distances for each pieces Table A.1. we choosed the closest top three divisions based on average incerasing rate of iteration. The reason of chosen top three increasing rated division is that every iteration two compared piece getting closer which means they have higher approach rate Table A.2. After selecting top three divisions now we compared which one getting closer to desired object. Figure 4.10

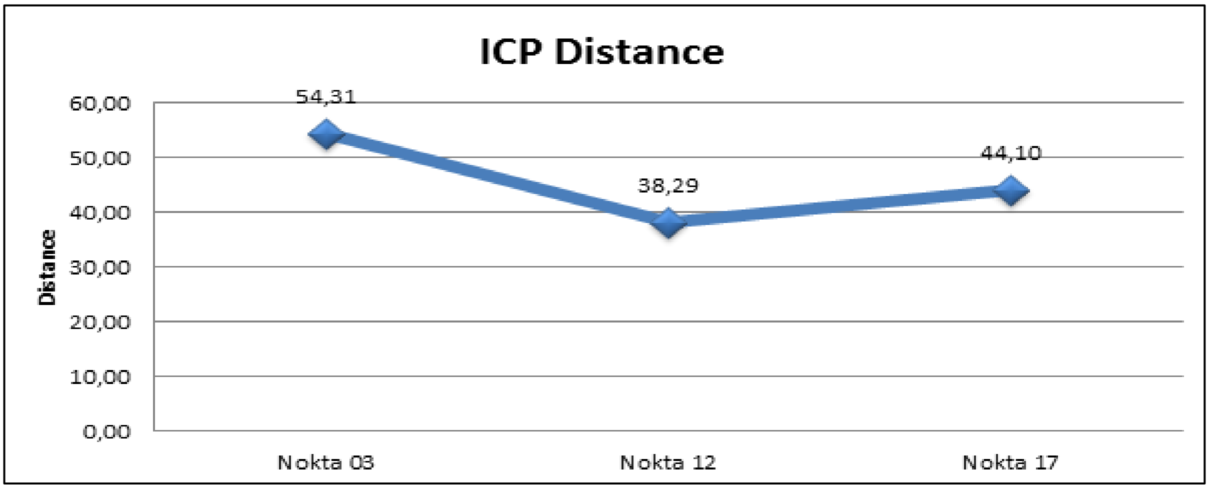


Figure 4.11. shows Nokta 12 division has minimum distance to desired object

Therefore we found best division to work on it. To see better result we sliced the divisions close as our compared object approximately one of third.

The result satisfies us because we did not remove noisy point on our Saltanat Gate or sliced same size as desired object also we used brute force method ICP (root mean square).However in recent years different strategies for point reduction,i.e., point selection, matching and weighting have been proposed and evaluated (] S. Rusinkiewicz and M. Levoy, 2001) Rusinkiewicz and Levoy propose a high speed ICP variant using a point-to-plane error metric (P. Neugebauer, 1997) and a projection-based method to generate point correspondences (] G. Blais and D. Levine, 1995). Furthermore they conclude that the other stages of the ICP process appear to have little effect of convergence rate, so that they choose the simplest ones, namely random sampling, constant weighting, and a distance threshold for rejecting point pairs (] S. Rusinkiewicz and M. Levoy. 2001).or Sparse Iterative Closest Point (S. Bouaziz, A. Tagliasacchi and M. Pauly,2013) which excludes outliers and missing parts of data on object and gives superior registration results when dealing with outliers and incomplete data. However our result good as developed ICPs. We get almost exact matching. Our desired object as close as 0.08133 to sliced part of Saltanat Gate and we got this result by trying 50 iteration in 6.1 second (After fifth iteration it is not necessary to iterate) Figure 4.12. By this way we found where desired object locates on Saltanat Gate. The part we worked on it locates 12nd horizontly sliced pieces 13rd vertically sliced pieces and first parts of this piece.

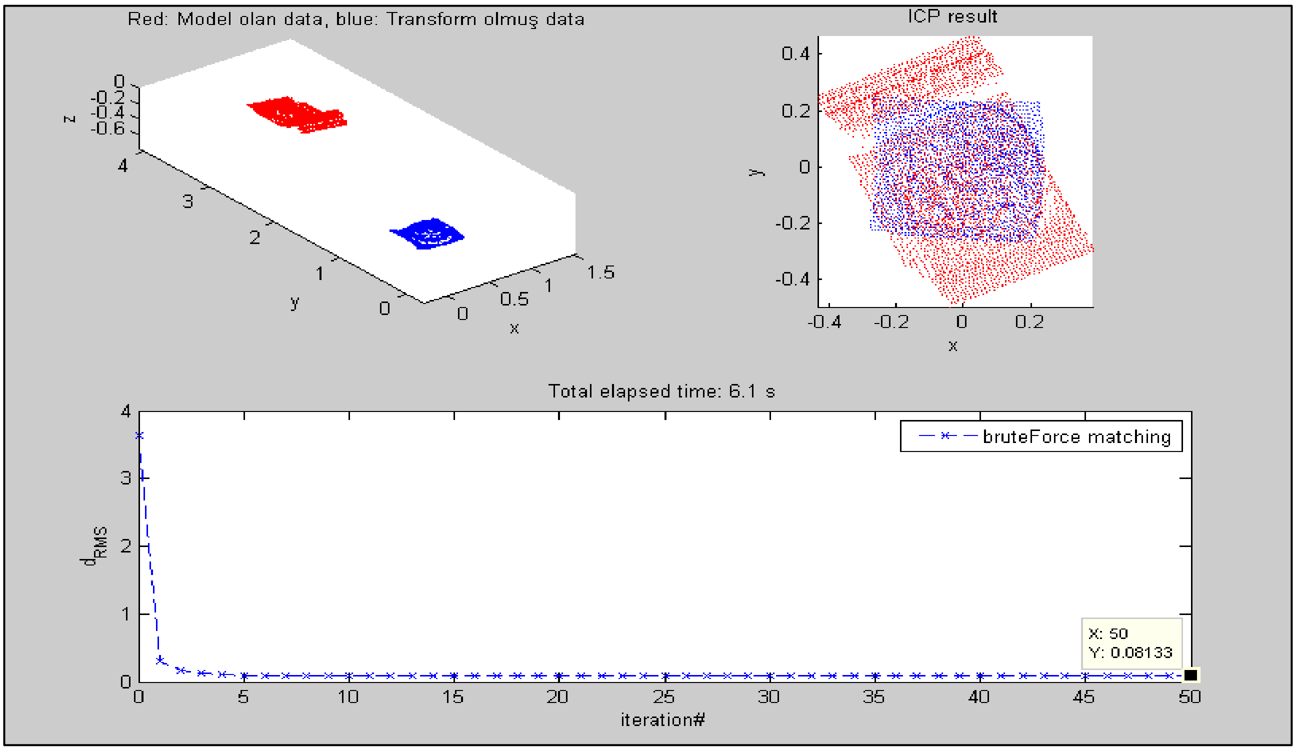


Figure 4.12 shows that iteration number, time, distance of two objects, locations of two objects before ICP apply on it also after it.

5 EXPERIMENTAL STUDY

For our experiments of ICP, we used a computer running on Windows8 64bit operating system, having 16GB of main memory and having Intel I7 3630QM 2.4 GHz processor. We also used Mathlab R2013 to implement our ICP and showing results, also MAYA, Meshlab, Leica Cyclone to handle the point cloud data.

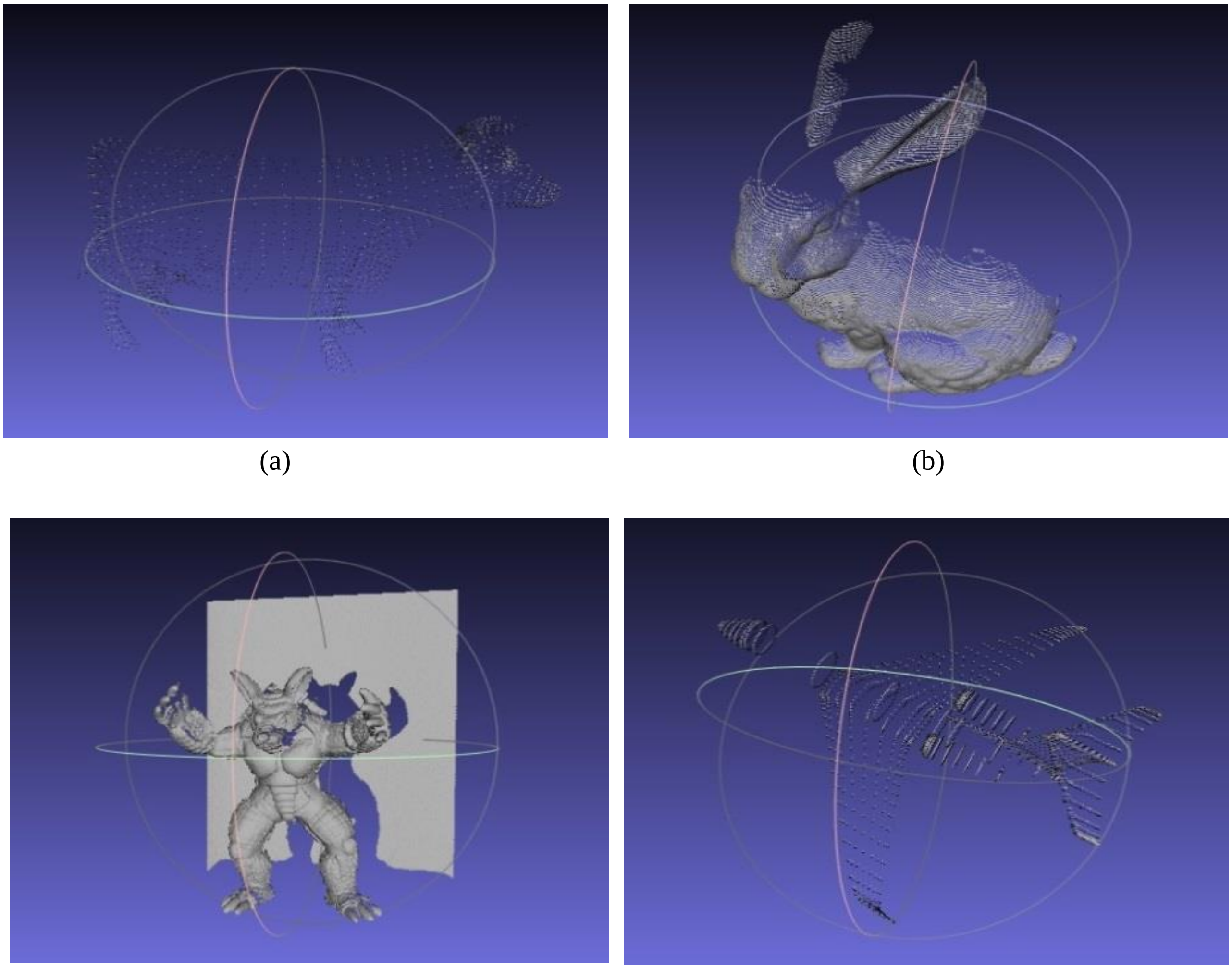
We evaluted our ICP method 4 different objects Figure 5.1 to show its effiency and applicability on different objects. Some objects are available in The Stanford 3D Scanning Repository. To give small information about object, every object has different size, vertex, face and scanned angle. Table 5.1 Some objects are incomplete which scanned form one angle others are completed.

Object Name	Number of Vertex	Number of Face	View	Completed Object
Armadillo	169016	335506	Front	Not
Bunny	40256	79312	Top	Not
Business Jet	6367	12702	Diagonal	Yes
Cow	2903	5804	Front	Yes

Table 5.1 shows properties of 3D objects

5.1 Efficiency

We started with chosing a 3D object which sliced from whole 3D object to recognize in the whole 3D object. Figure 5.1 shows which models was used to evalute our ICP to recognize 3D object.



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

Figure 5.1 shows 3D objects which was used to evalute our ICP method, (a) is a cow, (b) is a bunny, (c) is an Armadillo and (d) is a jet.

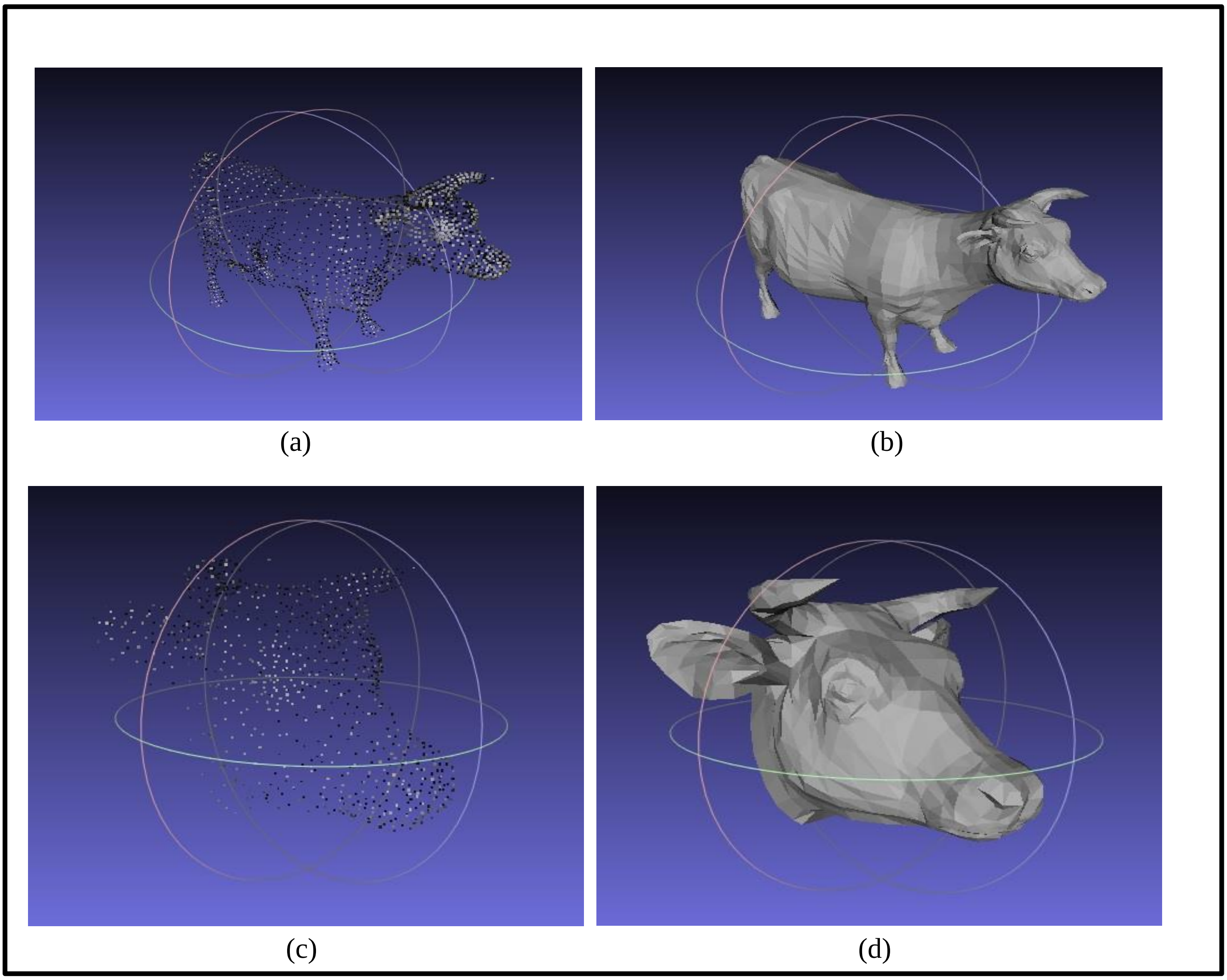


Figure 5.2 (a) is points view of whole 3D object. (b) is falted view of the 3D object. (c) is points view of part of the 3D object which is used for recognizing on the 3D object (d) is falted view of the 3D object.

As usual our approach we first divided the compeleted cow data as close as cow's head or desired 3D object.

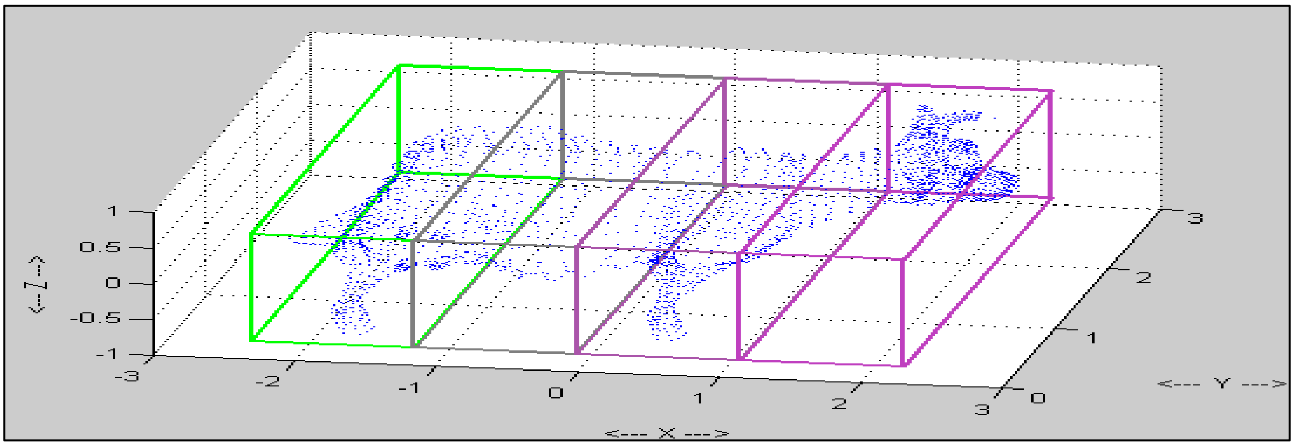


Figure 5.3 shows how looks like when we divided Cow as close as Cow's head. In this case we divided 4 pieces based on X axis.

Process time of recognizing the object is 6 seconds. The ICP is really fast and we have exact match on last division of cow data. Figure A.3 and Table 5.2 shows results of ICP

Division Number	Number of Iterate	Distance
Nokta1	83	19.5824
Nokta2	18	81.6433
Nokta3	62	37.2276
Nokta4	27	0.0001

Table 5.2 shows number of iteration and closest distances between two 3D objects for Cow.

Figure A.3. shows how our ICP method matches two 3D object. The figures are organizated by first location of two object and then location after ICP. The figures are consist of two sysmbol which are first object is represent by "o" symbol the second object represent by "x" symbol.

To challenge we chosed Bunny which is more complicated 3D object than Cow which uncompeleted object also has noisy points on 3D object. To push the limit we chosed bunny's head for being medium size object. We also rotated the bunny's head from orginal location to see result of ICP.

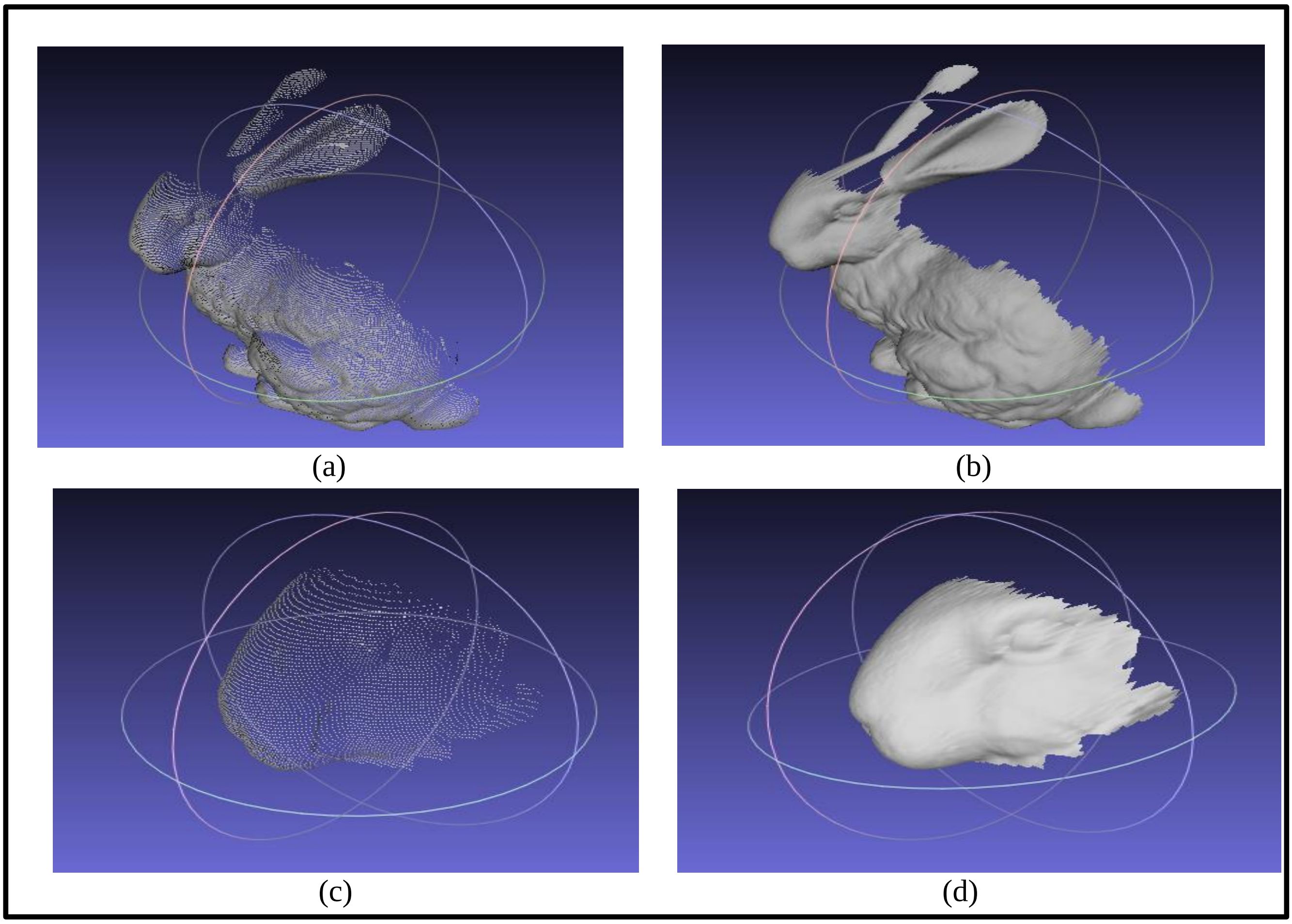


Figure 5.4 (a) is points view of whole 3D object. (b) is falted view of the 3D object. (c) is points view of part of the 3D object which is used for recognizing on the 3D object (d) is falted view of the 3D object.

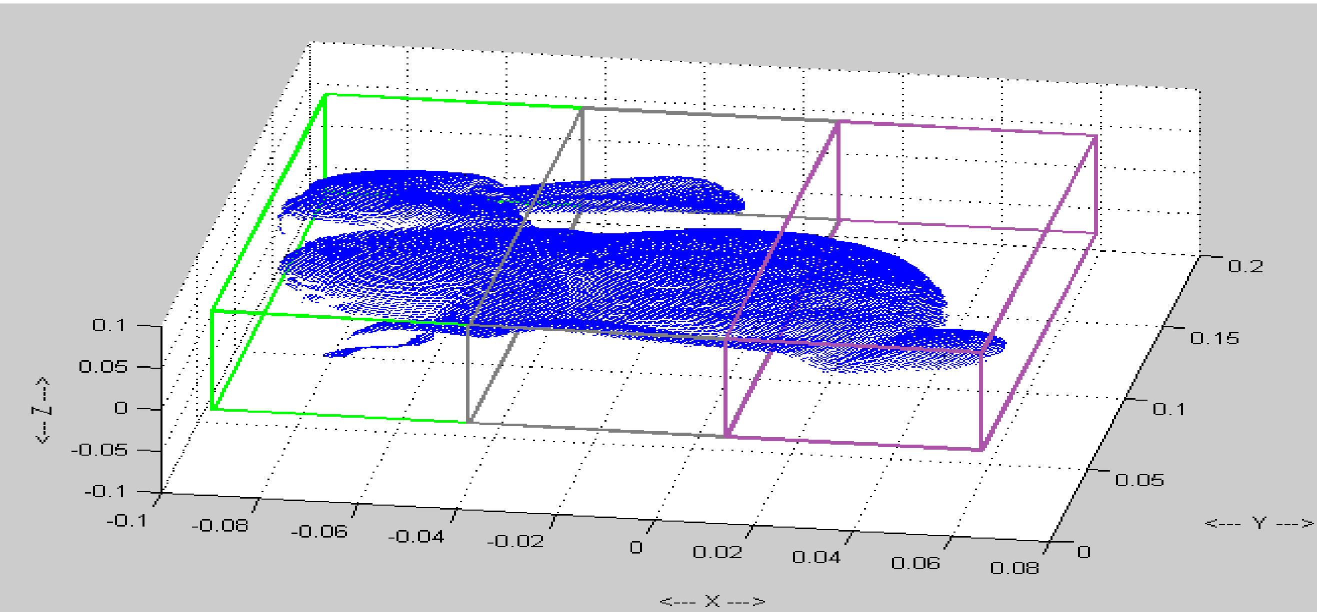


Figure 5.5 shows how looks like when we divided Bunny as close as Bunny’s head. In this case we divided 3 pieces based on X axis.After that we divided 2 pieces based on Y axis.

Process time of recognizing the object is 23 seconds. The ICP is really fast and we have exact match on second sub cloud division of bunny data. Figure A.6 and Table 5.5 shows results of ICP so according to bunny case rotating the object has minor effect on recognizing the object.

Next scenario is moving part of object and then try to recognize where the object belong. So we decided to choose Jet and make it interesting we purposely choosed the jet motor and we expect that we find two closest object matching.

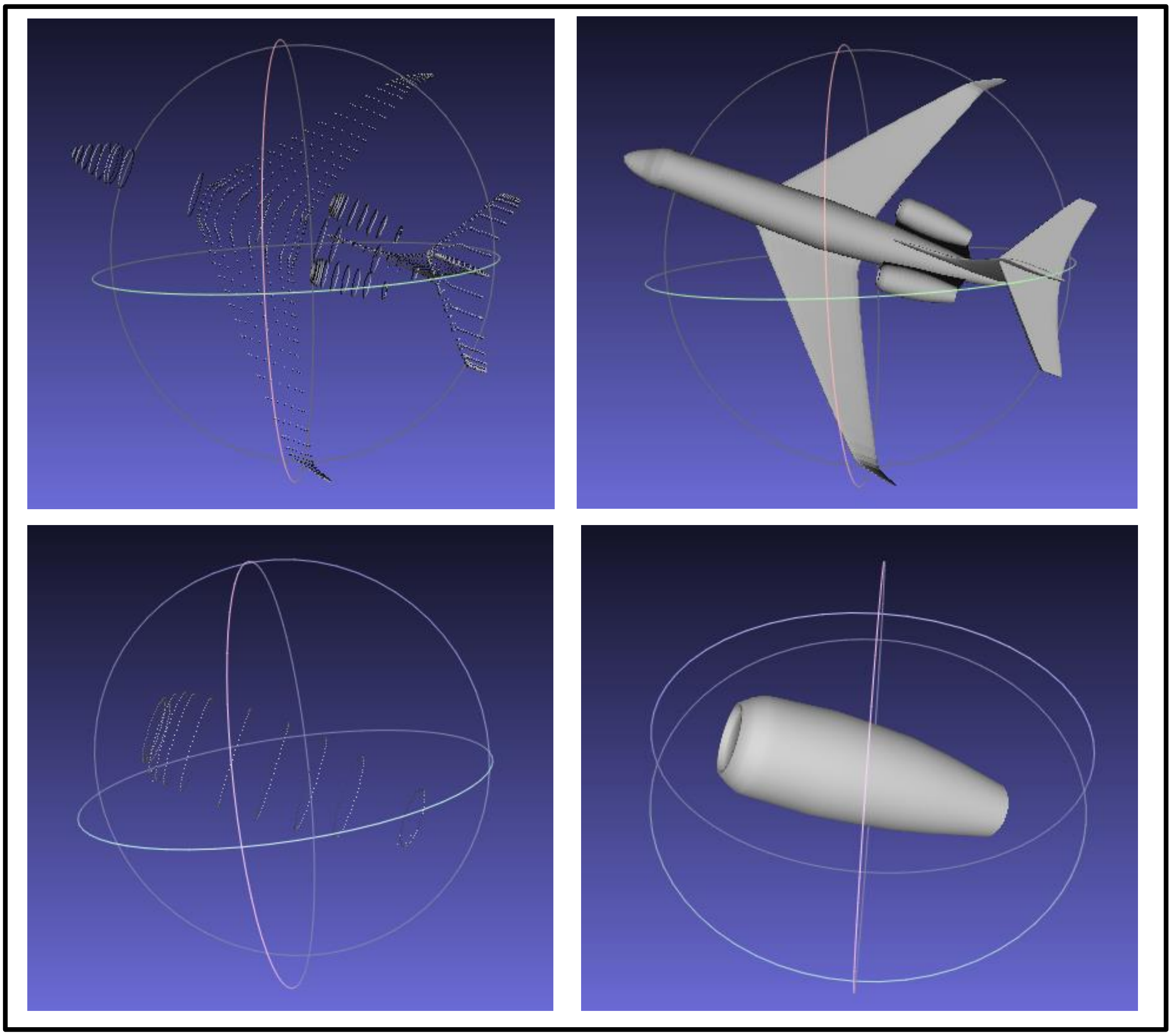


Figure 5.6 (a) is points view of whole 3D object. (b) is falted view of the 3D object. (c) is points view of part of the 3D object which is used for recognizing on the 3D object (d) is falted view of the 3D object.

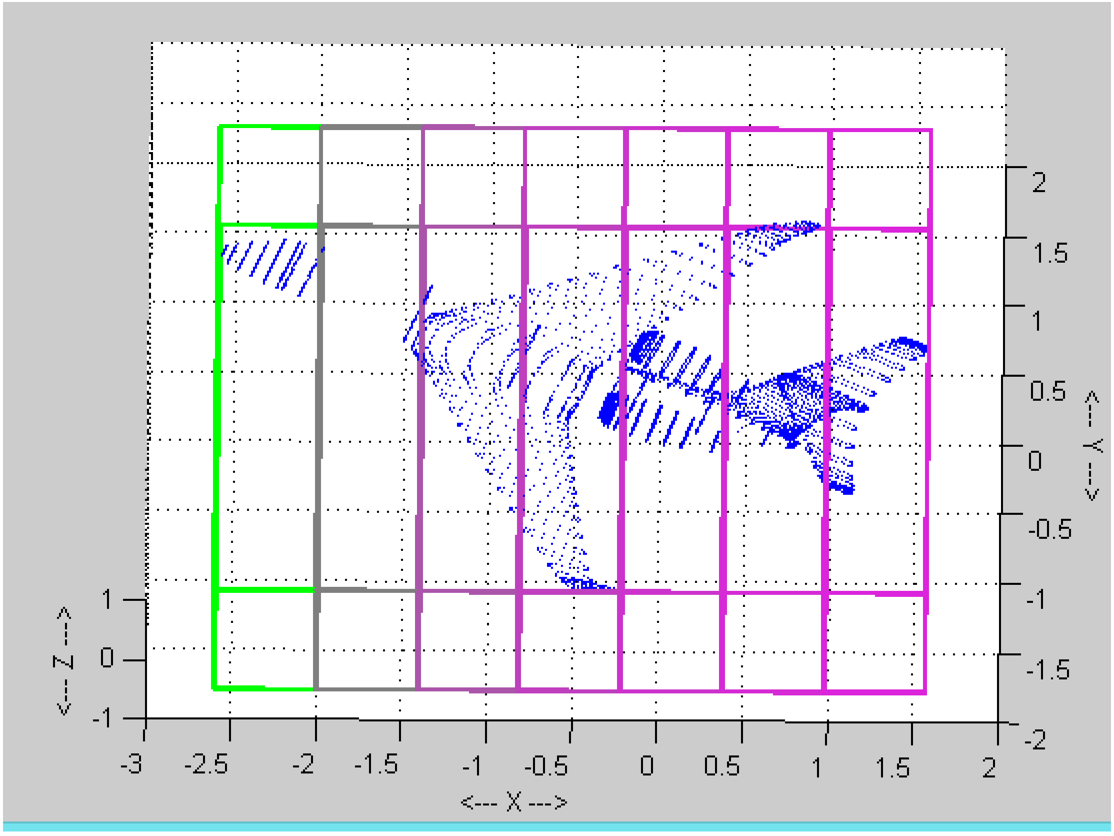


Figure 5.7 shows how looks like when we divided Jet as close as Jet's motor. In this case we divided 7 pieces based on X axis.After that we divided 5 pieces based on Y axis. The object itself is rotated.

Even though we had the smallest distance between two point clouds object shown in Table A.6 . It does not mean that we found the desired object. Because the noisy data always have fatal error on the result Figure A.7 and Figure A.8. To discard the noisy data affect on the results we have to slice the object small as the jet motor or have to find outliers on each sub cloud and exclude the outliers. However our ICP method reyls on simplest matching algorithm so we have to slice it as much as possible.Therefore we sliced 10 pieces based on Y axis and

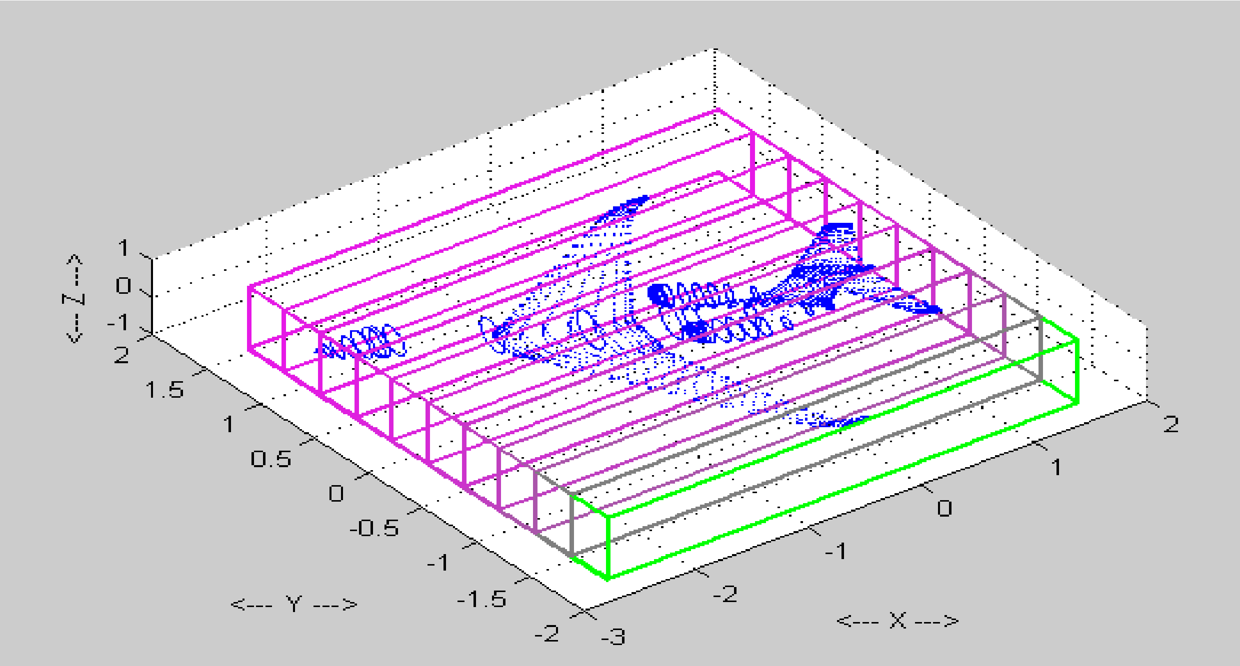
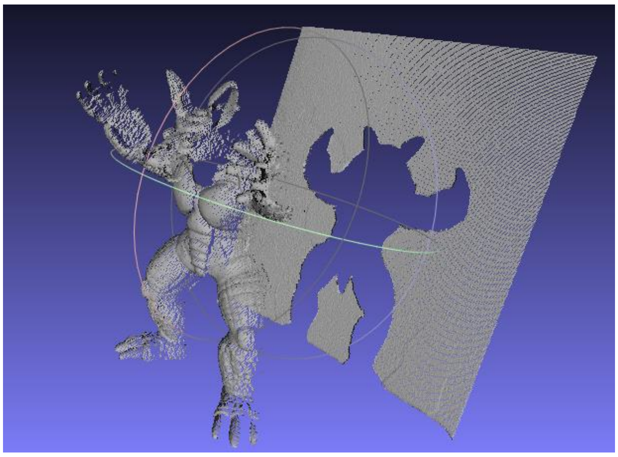


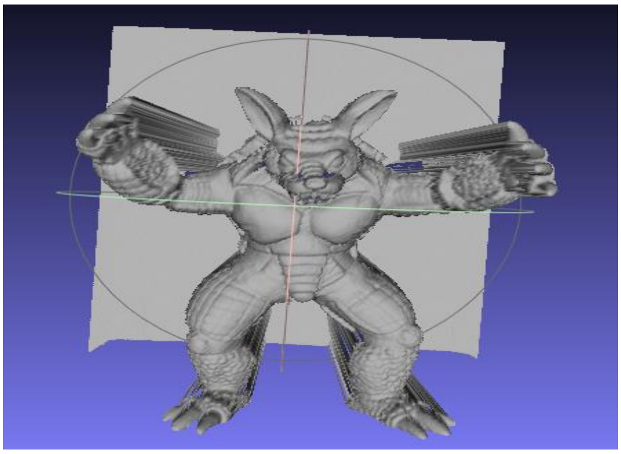
Figure 5.8 we divided 10 pieces based on y axis.After that we divided 7 pieces based on x axis.

Even though we got good result Table A.7 we could not get rid of noisy data affect. There we can understand our ICP method has weakness to noisy data.

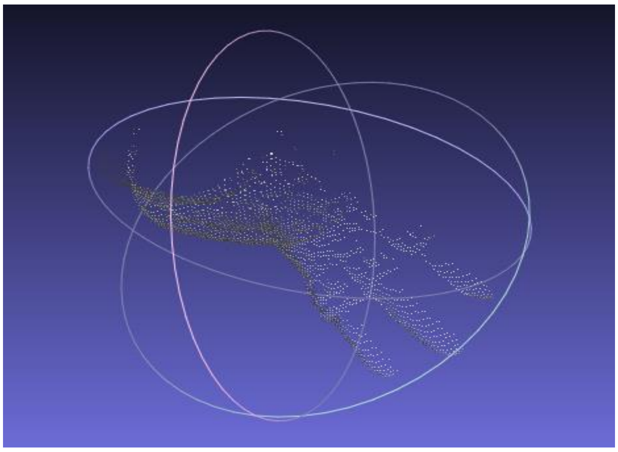
To challenge we chosed much more complicated 3D Object which uncompeleted object also has noisy points on 3D object which is Armadillo. To make it interesting we chosed very small part of the Object which is left leg. We also moved and rotated the leg from orginal direction.

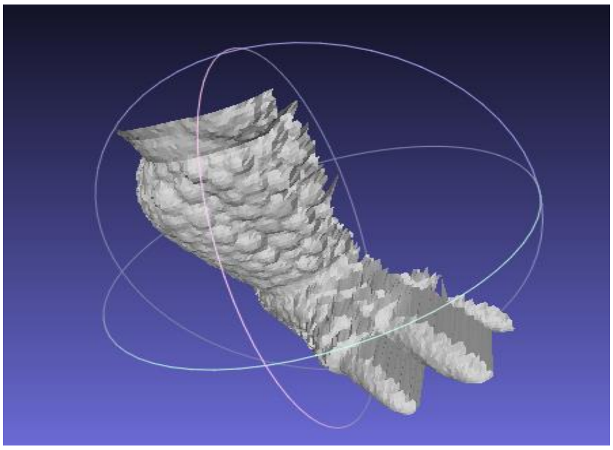


(a)



(b)





(c) (d)

Figure 5.9 (a) is points view of whole 3D object. (b) is falted view of the 3D object. (c) is points view of part of the 3D object which is used for recognizing on the 3D object (d) is falted view of the 3D object.

As usual our approach we first divided the compeleted Armadillo data as close as Armadillo's left leg or desired 3D object. Figure 5.5 also shows that there are too many noisy point on 3D object.

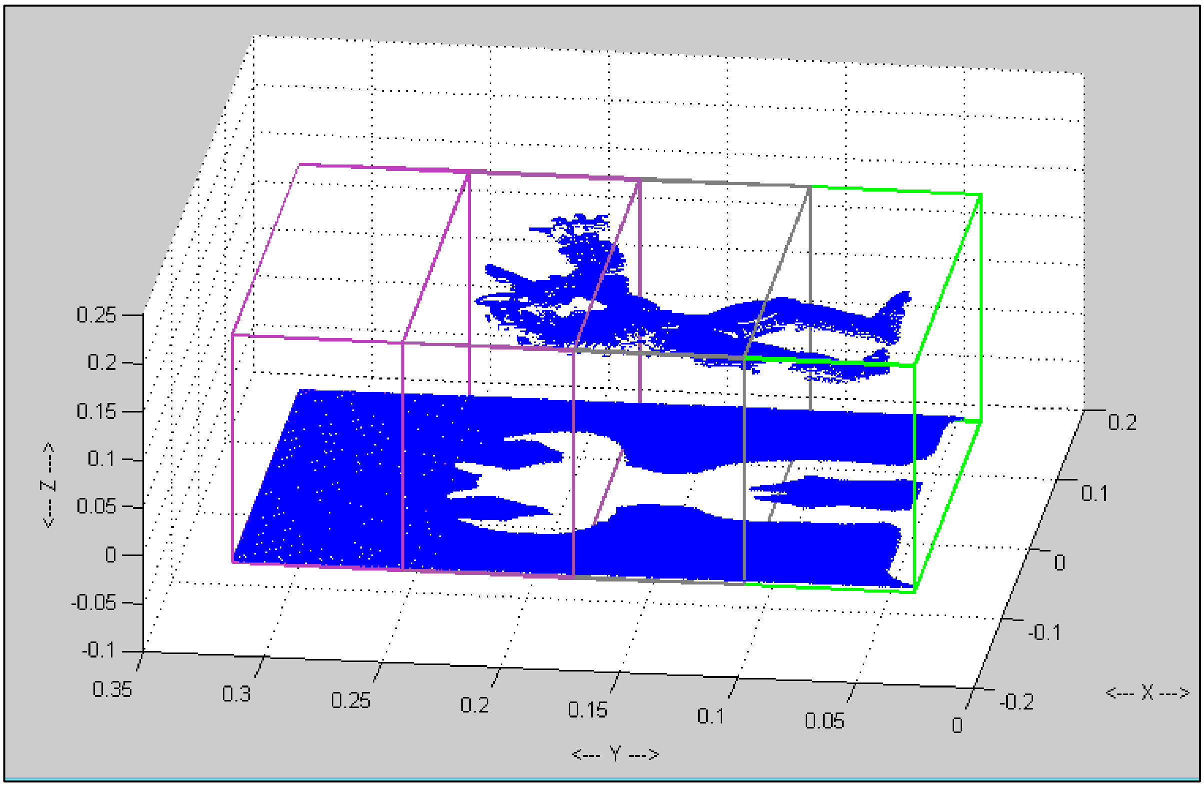


Figure 5.10 shows how looks like when we divided Armadillo as close as Armadillo's left leg. In this case we divided 4 pieces based on Y axis.

Lets see how it responses only based on slicing Y axis it is actually conflict our method however we want to show the results of it. Figure A.5. shows that size is really essential for our ICP method. According to Table 5.3. none of iteration gives close matching result.

Division Number	Number of Iterate	Distance
Nokta1	20	0,16307
Nokta2	23	0,35163
Nokta3	41	0,22765
Nokta4	439	0,64425

Table 5.3 shows number of iteration and closest distances between two 3D objects for Armadillo.

Slicing the object just base on Y axis is not enough so we have divided base on X axis. It is essential to consider X,Y,Z when you working on 3D object.

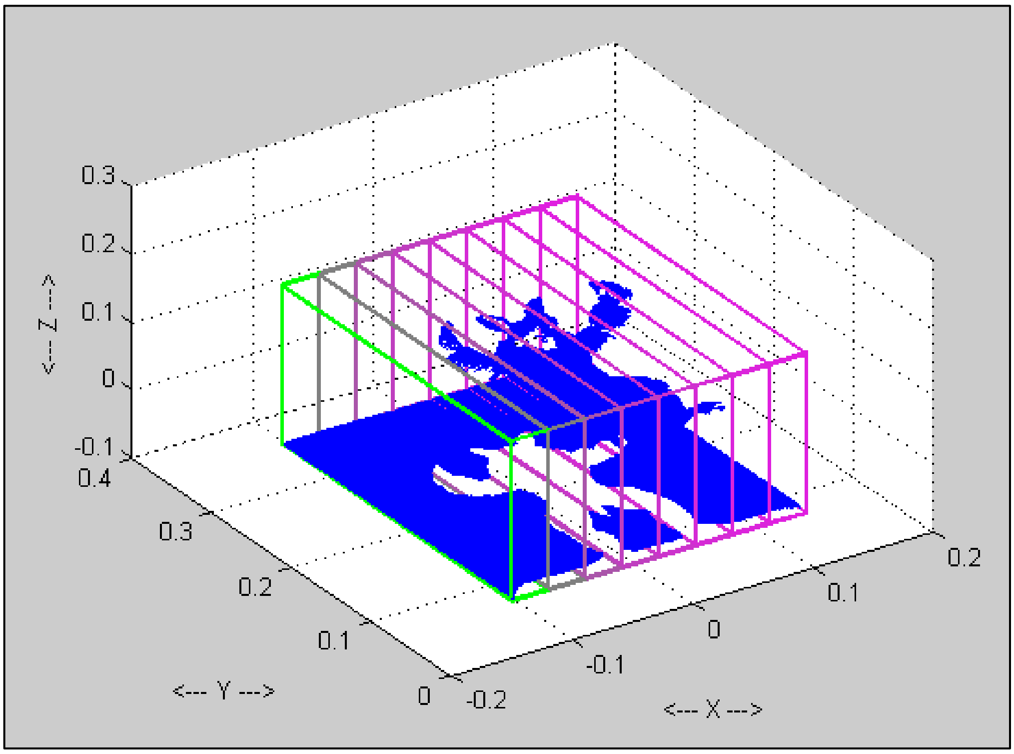


Figure 5.11 shows how looks like when we divided Armadillo as close as Armadillo's left leg. In this case we divided 8 pieces based on X axis.

We purposely sliced the whole 3D data which does not contain completed desired object in any sub cloud. We purposely put some part of desired object in a sub cloud other parts located other sub clouds. The reason behind it that is we want to evalute our approach of average increasing rate of iteration that have any connection to desired recognized object. According to Table A.4. we could understand the desired object locates one of them. We could recognize the object by increasing or decreasing number of sub cloud and then progress same as Saltanat Gate.

5.2 Robustness

ICP works based on closest point even so if the closest points match exactly to each other it does not mean that rest of points will be matched. Thus just using Euclidean distance on ICP is not enough. Adjusment is essential for ICP specially for the closest point. Also to choose which points are closed it takes too much time to decide and calculate for each point. This makes too much consumption of CPU, Ram.

Also the problem of registering point clouds with outliers including noises and missing data makes ICP less reliable.

6 CONCLUSION

ICP based on finding minimal distances between closest points and then rotates and transforms the data until two object got enough closed. As we work on ICP method we able to recoginize a part of object on completed object. ICP really works well on well defined 3D object which has no outliers or hole on 3D objects. ICP is affected by false matching, noise also size which is compared between two 3D objects. To get better result on ICP method, first make sure all outliers excluded from object also the object sliced close to compared object.

According to our study ICP works based on comparing distances between closest point so basically comparing each point to find which point is the closest makes our algoritm slow. For future work, ICP can be implement faster by finding new methods which reduce the time of comparing the closest points also can be focused on the stability and robustness of ICP. In addition, a better analysis of the effects of various kinds of noise and distortion would yield further insights into the best alignment algorithms for real-world, noisy scanned data. Algorithms that switch between variants, depending on the local error landscape and the probable presence of local minima, might also provide increased robustness.

REFERENCES

Murase, H. and Nayar, S. K. (1995). *Visual learning and recognition of 3-d objects from appearance*. IJCV, 14(1):5-24.

Gardner, W. F. and Lawton, D. T. (1996). *Interactive model-based vehicle tracking*. *TPAMI*, 18(11) pp.:1115-1121.

Enns, J. T. (2004). *The Thinking Eye, The Seeing Brain: Explorations in Visual Cognition*. New York: W. W. Norton & Company.

P. Besl and N. McKay. 1992, *A method for Registration of 3-D Shapes*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 14(2) pp:239 - 256

Tingbo Hou, Sen Wang and Hong Qinet, 2011, *3D Model-Driven Vehicle Matching and Recognition* pp. 1-2

G. Gallus, 1968 *Contour analysis in pattern recognition for human chromosome classification*, Appl Biomed Calcolo Electronico

G. Gallus, G. Regoliosi, 1974 *A decisional model of recognition applied to the chromosome boundaries*, Journal of Histochemistry & Cytochemistry 22

A. Jimenez, R. Ceres, J. Pons, 2000 *A survey of computer vision methods for locating fruits on trees*, IEEE Transactions of the ASABE 43 (6) 1911–1920.

E.N. Malamas, E.G.M. Petrakis, M. Zervakis, L. Petit, J-D. Legat, 2003,*A survey on industrial vision systems, applications and tools*, Image and Vision Computing 21 (2) 171–188.

M. Ejiri, 2007, *Machine vision in early days: Japan's pioneering contributions*, in Proc. 8th Asian Conference on Computer Vision (ACCV)

Alexander Andreopoulos, John K. Tsotsos 2013, *50 Years of object recognition: Directions forward* , Computer Vision and Image Understanding, 1-3

M. Ejiri, 2007, *Machine vision in early days: Japan's pioneering contributions*, in: Proc. 8th Asian Conference on Computer Vision (ACCV).

T. McInerney, D. Terzopoulos, 1996, *Deformable models in medical image analysis a survey*, Medical Image Analysis 1 (2) ,91–108.

A. Andreopoulos, J.K. Tsotsos, 2008, *Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI*, Medical Image Analysis 12 (3) 335–357.

O.D. Trier, A.K. Jain, T. Taxt, 1996, *Feature extraction methods for character recognition – a survey*, Pattern Recognition 29 (4) 641–662.

S. Mori, H. Nishida, H. Yamada, 1999, *Optical Character Recognition*, John Wiley and Sons,

K. Takahashi, T. Kitamura, M. Takatoo, Y. Kobayashi, Y. Satoh, 1996, *Traffic flow measuring system by image processing*, in: Proc. IAPR MVA, , pp. 245–248.

C.-N. Anagnostopoulos, I. Anagnostopoulos, I. Psoroulas, V. Loumos, E. Kayafas, 2008, *License plate recognition from still images and video sequences: a survey*, IEEE Transactions on Intelligent Transportation Systems 9 (3) 377–391.

D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, 2009**, Handbook of Fingerprint Recognition**, 2nd ed., Springer Publishing Company

K.W. Bowyer, K. Hollingsworth, P.J. Flynn, 2008, *Image understanding for iris biometrics: a survey*, Computer Vision and Image Understanding 110 (2) 281–307.

C.-L. Lin, K.-C. Fan, 2004, *Biometric verification using thermal images of palm-dorsa vein patterns*, IEEE Transactions on Circuits and Systems for Video Technology 14 (2) 199–213.

N. Miura, A. Nagasaka, 2005, *Extraction of finger-vein patterns using maximum curvature points in image profiles,* in: IAPR Conference on Machine Vision Applications

J. Tsotsos, 1992, **The Encyclopedia of Artificial Intelligence**, John Wiley and Sons, (Chapter: Image Understanding). pp. 641–663

S. Dickinson,1999, **What is Cognitive Science?**, Basil Blackwell Publishers, pp 172–207 (Chapter: Object Representation and Recognition).

L.G. Roberts, 1960, *Pattern recognition with an adaptive network*, in: Proc. IRE International Convention Record , pp. 66–70.

J.T. Tippett, D.A. Borkowitz, L.C. Clapp, C.J. Koester, A.J. Vanderburgh (Eds.), 1965. **Optical and Electro-Optical Information Processing**, MIT Press,

L.G. Roberts, 1963, Machine Perception of Three Dimensional Solids, Ph.D. thesis, Massachusetts Institute of Technology,

J.Graham and J.Piper, 1994, *Automatic karyotype analysis,* Meth. Mol. Biol., vol.29, pp.141–185.

A. Carothers and J. Piper, 1994, *Computer-aided classification of human chromosomes: a review*, Statistics Computing**,** vol.4, no.3, pp.161–171

Q.Wu, Z. Liu, T. Chen, Z. Xiong, and K.R. Castleman, 2005, *Subspace-based prototyping and classification of chromosome images*, IEEE Trans. Pattern Anal. Mach. Intell., vol.14, no.9, pp.1277–1287,

M. Moradi and S.K. Setarehdan,2006, *New features for automatic classification of human chromosomes: a feasibility study*, Pattern Recognit. Lett., vol.27, no.1, pp.19–28, M.J. Swain, D.H. Ballard, 1991, *Color indexing*, Int. J. Computer. Vision 7(1) 11 – 32

B.V. Funt, G.D. Finlayson, 1995, *Color constant color indexing*,IEEE Trans. PAMI 17(5) 522-529.

E.H. Land, J.J. McCann, 1971, *Lightness and retinex theory*, J. Opt. Soc. Am. 61 1-11.

G. Healey, D. Slater, 1995, *Global color constancy: recognition of objects by use of illumination invariant properties of color distributions*, J. Opt. Soc. Am. 11(11) 3003-3010.

G.D. Finlayson, S.S. Chatterjee, B.V. Funt, 1996, *Color angular indexing*, ECCV96, Vol. II, pp. 16-27.

D. Slater, G. Healey, 1996, *The illumination-invariant recognition of 3-D objects using local color invariants*, IEEE Trans. PAMI 18(2) 206-211.

David G. Lowe, 2004, *Distinctive Image Features* from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2): 91–110

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, 2006, *Speeded up robust features.* In Proceedings of the 9th European Conference on Computer Vision (ECCV),

Geoffrey E Hinton, 2007, *Learning multiple layers of representation. Trends in cognitive sciences*, 11:428–34,

Dan C. Ciresan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, and J¨urgen Schmidhuber, 2011, *High-Performance Neural Networks for Visual Object Classification. Technical report*, Dalle Molle Institute for Artificial Intelligence, Manno, Switzerland.

Quoc V Le, Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang Wei Koh, and Andrew Y. Ng. 2010 *Tiled convolutional neural networks. Advances in Neural Information Processing Systems*

Adam Coates, Andrew Y. Ng, and Serra Mall, 2011, *The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization*. In Proceedings of the 28th International Conference on Machine Learning (ICML)

Kai Yu and T. Zhang, 2010, *Improved local coordinate coding using local tangents*. In Proceedings of the 28th International Conference on Machine Learning (ICML)

Adam Coates, Honglak Lee, and Andrew Y. Ng. 2011, *An analysis of singlelayer networks in unsupervised feature learning*. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)

P. Flynn and A. Jain , 1992, *3D object recognition using invariant feature indexing of interpretation tables.* CVGIP, 55(2):119–129,

D. Hoiem, A. A. Efros, and M. Hebert, 2006, *Putting objects in perspective.* In Computer Vision and Pattern Recognition

S. Z. Li. Markov, 2001, *Random Field Modeling in Image Analysis*. Springer-Verlag

S. Kumar and M. Hebert, 2003, *Discriminative random fields: A discriminative framework for contextual interaction in classification*. In International Conference on Computer Vision.

L. Wolf and S. Bileschi, 2006, *A critical view of context.* International Journal of Computer Vision, 69(2):251–261.

C. Siagian and L. Itti. Gist, 2005, *A mobile robotics application of context-based vision in outdoor environment*. In Computer Vision and Pattern Recognition Workshops

Adam Coates, Honglak Lee, and Andrew Y. , 2011, An analysis of singlelayer networks in unsupervised feature learning. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)

Liefeng Bo, Xiaofeng Ren, and Dieter Fox, 2011, *Depth Kernel Descriptors for Object Recognition*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

Ming-Ching Chang, Benjamin B. Kimia, 2011, *Measuring 3D shape similarity by graph-based matching of the medial scaffolds* Computer Vision and Image Understanding 115 pp 707–720

K. Siddiqi, S. Pizer (Eds.) 2009, Medial Representations: Mathematics, Algorithms and Applications, Kluwer Academic Publishers

H. Blum, 1973, Biological shape and visual science, Journal of Theoretical Biology 38, 205–287.

T. Sebastian, P. Klein, B. Kimia, 2004, Recognition of shapes by editing their shock graphs, PAMI 26 551–571.

K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, S. Dickinson, 2008, Retrieving articulated 3-D models using medial surfaces, Machine Vision and Applications 19, 261–275.

F. Leymarie, B. Kimia, 2007, The medial scaffold of 3D unorganized point clouds, PAMI 29 (2), 313–330.

P. Giblin, B. Kimia, 2003, On the intrinsic reconstruction of shape from its symmetries, PAMI 25 (7), 895–911.

D. Attali, J.-D. Boissonat, H. Edelsbrunner, 2004, Stability and computation of the medial axis, in: Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration.

F. Leymarie, B. Kimia, 2007, The medial scaffold of 3D unorganized point clouds, PAMI 29 (2), 313–330.

M.-C. Chang, B. Kimia, 2008, Regularizing 3D medial axis using medial scaffold transforms, in: Proceedings of IEEE CVPR, pp. 1–8.

M.-C. Chang, B. Kimia, 2008, Regularizing 3D medial axis using medial scaffold transforms, in: Proceedings of IEEE CVPR, pp. 1–8.

(Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin, *Matching 3D Models with Shape Distributions*
(Yunpeng Li Noah Snavely Daniel P. Huttenlocher, *Location Recognition using Prioritized Feature Matching)*

T. Shao, W. Xu, K. Yin, J. Wang, K.Zhou, B. Guo 2002, *Discriminative Sketch-based 3D Model Retrieval via Robust Shape Matching*

S. R. Correay , L. G. Shapiroy and M.Meliaz, 2001 , *A New Signature-Based Method for Efficient 3-D Object Recognition*

Eunyoung Kim,2011, *A New Signature-Based Method for Efficient 3-D Object Recognition*, Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on pp 3800 - 3807

Enns, J. T. ,2004, *The Thinking Eye, The Seeing Brain: Explorations in Visual Cognition*. New York: W. W. Norton & Company.

K. S. Arun, T. S. Huang, and S. D. Blostein, 1987, *Least square fitting of two 3-d point sets.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(5):698 – 700.

B. K. P. Horn. 1987, *Closed–form solution of absolute orientation using unit quaternions*. Journal of the Optical Society of America A, 4(4):629 – 642.

B. K. P. Horn, H. M. Hilden, and Sh. Negahdaripour, 1988, *Closed–form solution of absolute orientation using orthonormal matrices.* Journal of the Optical Society of America A,5(7):1127 – 1135

A. Lorusso, D. Eggert, and R. Fisher, 1995, *A Comparison of Four Algorithms for Estimating 3D Rigid Transformations*.In Proceedings of the 4th British Machine Vision Conference (BMVC ’95), pages 237 – 246, Birmingham, England.

M. W. Walker, L. Shao, and R. A. Volz, 1991, *Estimating 3-d location parameters using dual number quaternions*. CVGIP: Image Understanding, 54:358 – 367.

S. Bouaziz, A. Tagliasacchi and M. Pauly, 2013, *Sparse Iterative Closest Point* , Eurographics Symposium on Geometry Processing 2013

M. Chaouch, A. Verroust, 2008, *Alignment of 3D models* , Graphical Models 71 (2009) 63–76

PrimeSense. http://www.primesense.com/.

Microsoft Kinect. http://www.xbox.com/en-us/kinect.

APPENDICES

Appendix A Experimental Results Details

Division Number	Sub Division	Number of Itterate	Distance	Distance/ Itterate
Nokta 01	01	62	22,8871	0,36915
Nokta 01	02	21	13,0481	0,62134
Nokta 01	03	115	16,6232	0,14455
Nokta 01	04	23	17,6079	0,76556
Nokta 01	05	78	17,3768	0,22278
Nokta 01	06	35	15,9987	0,45711
Nokta 01	07	23	16,3421	0,71053
Nokta 01	08	96	7,2019	0,07502
Nokta 01	09	86	11,54	0,13419
Nokta 01	10	32	12,7544	0,39858
Nokta 01	11	5	41,3471	8,26942
Nokta 01	12	13	24,8694	1,91303
Nokta 01	13	11	18,7759	1,70690
Nokta 01	14	29	23,1946	0,79981
Nokta 01	15	22	37,536	1,70618
Nokta 01	16	11	37,0735	3,37032
Nokta 01	17	10	29,176	2,91760
Nokta 01	18	35	14,2386	0,40682
Nokta 01	19	46	11,6235	0,25268
Nokta 01	20	6	59,4612	9,91020
Nokta 02	01	32	33,4214	1,04442
Nokta 02	02	7	50,7633	7,25190
Nokta 02	03	7	67,3246	9,61780
Nokta 02	04	4	82,5767	20,64418
Nokta 02	12	11	39,8309	3,62099
Nokta 02	13	6	59,5644	9,92740
Nokta 02	14	25	22,2091	0,88836
Nokta 02	15	72	23,547	0,32704
Nokta 02	16	41	4,7545	0,11596
Nokta 02	17	121	5,0016	0,04134
Nokta 02	18	66	3,4277	0,05193
Nokta 02	19	29	8,3755	0,28881
Nokta 02	20	55	2,1348	0,03881
Nokta 03	01	8	47,7578	5,96973
Nokta 03	02	7	49,7303	7,10433
Nokta 03	03	7	47,7408	6,82011
Nokta 03	04	4	62,5715	15,64288
Nokta 03	05	3	71,112	23,70400
Nokta 03	13	14	72,1399	5,15285
Nokta 03	14	13	46,6661	3,58970
Nokta 03	15	7	57,49	8,21286
Nokta 03	16	7	54,6397	7,80567
Nokta 03	17	6	53,2724	8,87873
Nokta 03	18	14	48,5276	3,46626
Nokta 03	19	13	47,6531	3,66562

Division Number	Sub Division	Number of Itterate	Distance	Distance/ Itterate
Nokta 03	20	14	46,777	3,34121
Nokta 04	01	10	30,7567	3,07567
Nokta 04	02	31	9,8316	0,31715
Nokta 04	03	30	13,6502	0,45501
Nokta 04	04	13	20,2248	1,55575
Nokta 04	05	208	4,0678	0,01956
Nokta 04	06	69	5,3979	0,07823
Nokta 04	07	13	13,4094	1,03149
Nokta 04	08	37	18,2655	0,49366
Nokta 04	09	45	22,5368	0,50082
Nokta 04	10	20	29,1431	1,45716
Nokta 04	11	111	14,8164	0,13348
Nokta 04	12	50	37,9914	0,75983
Nokta 04	13	34	33,2743	0,97866
Nokta 04	14	30	33,8461	1,12820
Nokta 04	15	12	51,063	4,25525
Nokta 04	16	8	72,8528	9,10660
Nokta 04	17	12	29,5439	2,46199
Nokta 04	18	11	42,7476	3,88615
Nokta 04	19	5	64,5111	12,90222
Nokta 04	20	16	50,1508	3,13443
Nokta 05	01	11	31,7601	2,88728
Nokta 05	02	26	13,6556	0,52522
Nokta 05	03	10	14,2066	1,42066
Nokta 05	04	26	13,7101	0,52731
Nokta 05	05	55	13,0263	0,23684
Nokta 05	06	47	18,7249	0,39840
Nokta 05	07	25	20,3267	0,81307
Nokta 05	08	11	13,6775	1,24341
Nokta 05	09	83	8,8816	0,10701
Nokta 05	10	20	16,3053	0,81527
Nokta 05	11	10	24,9708	2,49708
Nokta 05	12	48	18,2225	0,37964
Nokta 05	13	13	27,5625	2,12019
Nokta 05	14	88	7,1316	0,08104
Nokta 05	15	77	10,9878	0,14270
Nokta 05	16	43	9,8868	0,22993
Nokta 05	17	19	15,5359	0,81768
Nokta 05	18	12	52,8618	4,40515
Nokta 05	20	4	76,7105	19,17763
Nokta 06	01	6	44,4852	7,41420
Nokta 06	02	23	29,652	1,28922
Nokta 06	03	22	18,2256	0,82844
Nokta 06	04	19	37,9874	1,99934
Nokta 06	05	14	22,5919	1,61371
Nokta 06	06	21	36,6096	1,74331
Nokta 06	07	27	30,5466	1,13136
Nokta 06	08	9	23,9623	2,66248
Nokta 06	09	21	26,3844	1,25640
Nokta 06	10	22	25,1856	1,14480

Table A.1 Result of ICP on each sliced piece of Saltanat Gate

Division Number	Sub Division	Number of Itterate	Distance	Distance/ Itterate
Nokta 06	11	33	20,1643	0,61104
Nokta 06	12	28	17,178	0,61350
Nokta 06	13	39	25,7022	0,65903
Nokta 06	14	39	26,7201	0,68513
Nokta 06	15	9	28,4547	3,16163
Nokta 06	16	10	27,6326	2,76326
Nokta 06	17	56	30,8997	0,55178
Nokta 06	18	28	10,2485	0,36602
Nokta 06	19	7	22,7074	3,24391
Nokta 06	20	46	17,4223	0,37875
Nokta 07	01	7	33,1491	4,73559
Nokta 07	02	36	22,0542	0,61262
Nokta 07	03	66	12,6743	0,19203
Nokta 07	04	31	9,9431	0,32075
Nokta 07	05	13	15,7621	1,21247
Nokta 07	06	261	4,3376	0,01662
Nokta 07	07	29	5,3057	0,18296
Nokta 07	08	51	18,5082	0,36291
Nokta 07	09	56	31,2091	0,55731
Nokta 07	10	27	25,1628	0,93196
Nokta 07	11	164	7,4717	0,04556
Nokta 07	12	14	5,9808	0,42720
Nokta 07	13	328	6,5534	0,01998
Nokta 07	14	49	17,0347	0,34765
Nokta 07	19	59	39,4564	0,66875
Nokta 07	20	34	37,588	1,10553
Nokta 08	01	10	32,7869	3,27869
Nokta 08	02	35	16,0161	0,45760
Nokta 08	03	59	20,8621	0,35359
Nokta 08	04	41	28,6967	0,69992
Nokta 08	05	26	28,5592	1,09843
Nokta 08	06	22	9,3188	0,42358
Nokta 08	07	30	15,7901	0,52634
Nokta 08	08	71	20,4199	0,28760
Nokta 08	09	55	16,4647	0,29936
Nokta 08	10	37	14,9504	0,40406
Nokta 08	11	15	20,679	1,37860
Nokta 08	12	32	18,4585	0,57683
Nokta 08	13	11	19,1113	1,73739
Nokta 08	14	31	7,3975	0,23863
Nokta 08	15	59	4,6082	0,07811
Nokta 08	16	24	24,5869	1,02445
Nokta 08	17	4	67,4061	16,85153
Nokta 08	18	62	44,4044	0,71620
Nokta 08	19	10	49,4348	4,94348
Nokta 08	20	5	67,9544	13,59088
Nokta 09	01	12	34,4481	2,87068
Nokta 09	02	15	18,4038	1,22692
Nokta 09	03	36	15,4094	0,42804
Nokta 09	04	33	14,1613	0,42913
Nokta 09	05	49	22,1184	0,45140
Nokta 09	06	40	23,2909	0,58227
Nokta 09	07	37	22,8113	0,61652
Nokta 09	08	11	17,5895	1,59905
Nokta 09	09	8	17,999	2,24988
Nokta 09	10	120	10,9065	0,09089
Nokta 09	11	17	9,0687	0,53345
Nokta 09	12	27	16,1817	0,59932
Nokta 09	13	14	10,2554	0,73253
Nokta 09	14	91	13,7727	0,15135
Nokta 09	15	7	43,2929	6,18470
Nokta 09	16	5	60,0084	12,00168
Nokta 09	17	33	48,5238	1,47042
Nokta 09	18	8	50,9941	6,37426
Nokta 09	19	4	72,8376	18,20940
Nokta 09	20	6	74,0835	12,34725
Nokta 10	01	6	38,9415	6,49025
Nokta 10	02	6	26,4063	4,40105
Nokta 10	03	11	12,7351	1,15774
Nokta 10	04	26	15,4048	0,59249
Nokta 10	05	28	15,6773	0,55990
Nokta 10	06	14	15,5309	1,10935
Nokta 10	07	47	12,7361	0,27098
Nokta 10	08	35	7,2446	0,20699
Nokta 10	09	78	16,1022	0,20644
Nokta 10	10	57	10,6185	0,18629
Nokta 10	11	16	10,0244	0,62653
Nokta 10	12	23	7,9868	0,34725
Nokta 10	13	19	8,3088	0,43731
Nokta 10	14	96	8,4337	0,08785
Nokta 10	15	41	24,2902	0,59244
Nokta 10	16	13	31,022	2,38631
Nokta 10	17	101	10,8239	0,10717
Nokta 10	18	43	9,0552	0,21059
Nokta 10	19	25	20,8415	0,83366
Nokta 10	20	5	67,3768	13,47536
Nokta 11	10	15	11,4993	0,76662
Nokta 11	11	28	7,8821	0,28150
Nokta 11	12	76	16,3243	0,21479
Nokta 11	13	27	12,9316	0,47895
Nokta 11	14	21	12,2807	0,58480
Nokta 11	15	38	11,965	0,31487
Nokta 11	16	7	17,3842	2,48346
Nokta 11	17	64	13,1487	0,20545
Nokta 11	18	50	26,2734	0,52547
Nokta 11	19	64	32,1016	0,50159
Nokta 11	20	15	33,9957	2,26638
Nokta 12	01	7	34,812	4,97314
Nokta 12	02	19	20,7286	1,09098
Nokta 12	03	20	16,861	0,84305

Table A.1 Result of ICP on each sliced piece of Saltanat Gate

Division Number	Sub Division	Number of Itterate	Distance	Distance/ Itterate
Nokta 12	04	16	27,5884	1,72428
Nokta 12	05	8	32,5247	4,06559
Nokta 12	06	24	7,0325	0,29302
Nokta 12	07	23	26,7814	1,16441
Nokta 12	08	59	18,5672	0,31470
Nokta 12	09	25	32,8136	1,31254
Nokta 12	10	16	51,18	3,19875
Nokta 12	11	16	29,1058	1,81911
Nokta 12	12	9	12,234	1,35933
Nokta 12	13	37	13,2297	0,35756
Nokta 12	14	4	56,3289	14,08223
Nokta 12	15	8	68,1388	8,51735
Nokta 12	16	9	46,3927	5,15474
Nokta 12	17	6	57,1451	9,52418
Nokta 12	18	9	69,7645	7,75161
Nokta 12	19	5	75,3419	15,06838
Nokta 12	20	5	69,2621	13,85242
Nokta 13	01	7	39,9919	5,71313
Nokta 13	02	34	15,4437	0,45423
Nokta 13	03	34	15,1425	0,44537
Nokta 13	04	31	32,9834	1,06398
Nokta 13	05	23	17,3909	0,75613
Nokta 13	06	12	9,9909	0,83258
Nokta 13	07	22	29,0837	1,32199
Nokta 13	08	10	28,6639	2,86639
Nokta 13	09	14	40,04	2,86000
Nokta 13	10	23	28,5392	1,24083
Nokta 13	11	116	7,0846	0,06107
Nokta 13	12	108	9,6179	0,08905
Nokta 13	13	49	9,6023	0,19597
Nokta 13	14	53	7,0661	0,13332
Nokta 13	15	23	7,9525	0,34576
Nokta 13	16	90	8,4933	0,09437
Nokta 13	17	30	10,9382	0,36461
Nokta 13	18	39	8,4202	0,21590
Nokta 13	19	15	38,534	2,56893
Nokta 13	20	7	59,3009	8,47156
Nokta 14	01	17	17,7991	1,04701
Nokta 14	02	22	20,2293	0,91951
Nokta 14	03	26	23,5352	0,90520
Nokta 14	04	6	48,3895	8,06492
Nokta 14	05	25	18,9226	0,75690
Nokta 14	06	30	18,33	0,61100
Nokta 14	07	6	38,3826	6,39710
Nokta 14	08	23	4,5616	0,19833
Nokta 14	09	17	10,3307	0,60769
Nokta 14	10	37	29,0713	0,78571
Nokta 14	11	38	9,4915	0,24978
Nokta 14	12	36	30,1941	0,83873
Nokta 14	13	10	49,1123	4,91123

Division Number	Sub Division	Number of Itterate	Distance	Distance/ Itterate
Nokta 14	14	39	16,4614	0,42209
Nokta 14	15	19	27,2482	1,43412
Nokta 14	16	4	84,4238	21,10595
Nokta 14	17	6	47,9819	7,99698
Nokta 14	20	5	73,4253	14,68506
Nokta 15	01	9	40,5021	4,50023
Nokta 15	03	5	55,4626	11,09252
Nokta 15	04	24	14,9499	0,62291
Nokta 15	05	62	12,8183	0,20675
Nokta 15	06	49	4,122	0,08412
Nokta 15	07	82	3,7817	0,04612
Nokta 15	08	13	7,8447	0,60344
Nokta 15	09	51	24,1512	0,47355
Nokta 15	10	108	5,6041	0,05189
Nokta 15	11	77	16,4764	0,21398
Nokta 15	12	41	37,8913	0,92418
Nokta 15	13	39	33,0851	0,84834
Nokta 15	14	5	39,7706	7,95412
Nokta 15	15	16	19,7382	1,23364
Nokta 15	16	25	20,5744	0,82298
Nokta 15	17	18	34,9873	1,94374
Nokta 15	18	12	34,3745	2,86454
Nokta 15	19	18	37,3623	2,07568
Nokta 15	20	27	10,7807	0,39929
Nokta 16	01	8	31,7772	3,97215
Nokta 16	02	21	15,918	0,75800
Nokta 16	03	29	42,0049	1,44844
Nokta 16	04	50	5,6759	0,11352
Nokta 16	05	41	14,2552	0,34769
Nokta 16	06	18	29,5556	1,64198
Nokta 16	07	7	68,3231	9,76044
Nokta 16	08	32	45,2429	1,41384
Nokta 16	09	16	40,6281	2,53926
Nokta 16	10	8	57,852	7,23150
Nokta 16	11	4	79,3925	19,84813
Nokta 16	12	9	24,4264	2,71404
Nokta 16	13	10	45,9104	4,59104
Nokta 16	14	10	39,7609	3,97609
Nokta 16	15	5	54,5091	10,90182
Nokta 16	16	34	12,2092	0,35909
Nokta 16	17	37	3,9724	0,10736
Nokta 16	18	58	4,0699	0,07017
Nokta 16	19	57	4,218	0,07400
Nokta 16	20	14	17,2844	1,23460
Nokta 17	01	7	41,6452	5,94931
Nokta 17	02	13	36,884	2,83723
Nokta 17	03	6	53,7082	8,95137
Nokta 17	04	14	41,1712	2,94080
Nokta 17	05	31	39,224	1,26529
Nokta 17	06	23	37,7926	1,64316

Table A.1 Result of ICP on each sliced piece of Saltanat Gate

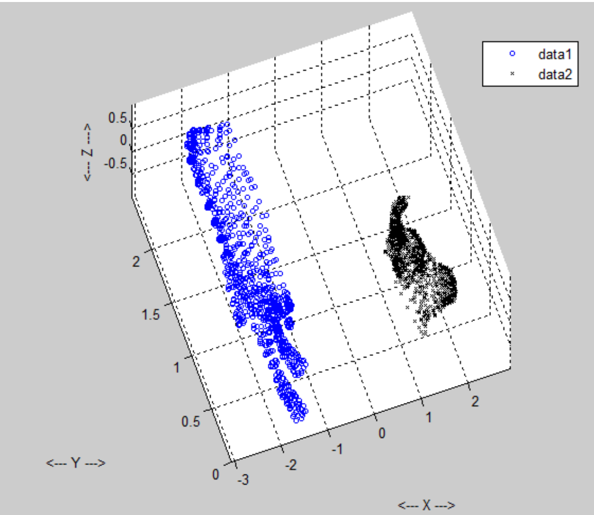
Division Number	Sub Division	Number of Itterate	Distance	Distance/ Itterate
Nokta 17	07	92	33,215	0,36103
Nokta 17	08	55	37,3078	0,67832
Nokta 17	09	7	27,1785	3,88264
Nokta 17	10	38	22,4999	0,59210
Nokta 17	11	66	15,3637	0,23278
Nokta 17	12	58	32,8723	0,56676
Nokta 17	13	22	32,3881	1,47219
Nokta 17	14	6	67,1475	11,19125
Nokta 17	15	4	84,2666	21,06665
Nokta 17	16	12	40,8114	3,40095
Nokta 17	17	11	41,752	3,79564
Nokta 17	18	5	67,2905	13,45810
Nokta 17	20	4	85,4703	21,36758
Nokta 18	01	6	70,0357	11,67262
Nokta 18	02	6	62,6117	10,43528
Nokta 18	03	11	40,0651	3,64228
Nokta 18	04	17	29,8769	1,75746
Nokta 18	05	63	32,7045	0,51912
Nokta 18	06	28	46,9388	1,67639
Nokta 18	07	19	66,2849	3,48868
Nokta 18	08	33	23,3055	0,70623
Nokta 18	09	69	18,8524	0,27322
Nokta 18	10	115	17,2	0,14957
Nokta 18	11	23	34,5562	1,50244
Nokta 18	12	74	31,7626	0,42922
Nokta 18	13	13	32,1285	2,47142
Nokta 18	14	18	38,278	2,12656
Nokta 18	15	23	24,1232	1,04883
Nokta 18	16	42	13,5471	0,32255
Nokta 18	17	117	13,4102	0,11462
Nokta 18	18	74	14,2615	0,19272
Nokta 18	19	62	21,3296	0,34403
Nokta 18	20	14	35,8972	2,56409
Nokta 19	01	32	59,3148	1,85359
Nokta 19	02	10	35,2285	3,52285
Nokta 19	03	44	54,4513	1,23753
Nokta 19	04	21	45,8153	2,18168
Nokta 19	05	29	50,431	1,73900
Nokta 19	06	19	46,7274	2,45934
Nokta 19	07	29	49,4168	1,70403
Nokta 19	08	35	23,2042	0,66298
Nokta 19	09	16	44,4486	2,77804
Nokta 19	10	15	32,0811	2,13874
Nokta 19	11	14	27,4674	1,96196
Nokta 19	12	12	33,6872	2,80727
Nokta 19	13	17	32,9903	1,94061
Nokta 19	14	13	45,9979	3,53830
Nokta 19	15	33	39,1402	1,18607
Nokta 19	16	17	38,0856	2,24033
Nokta 19	17	34	15,2986	0,44996

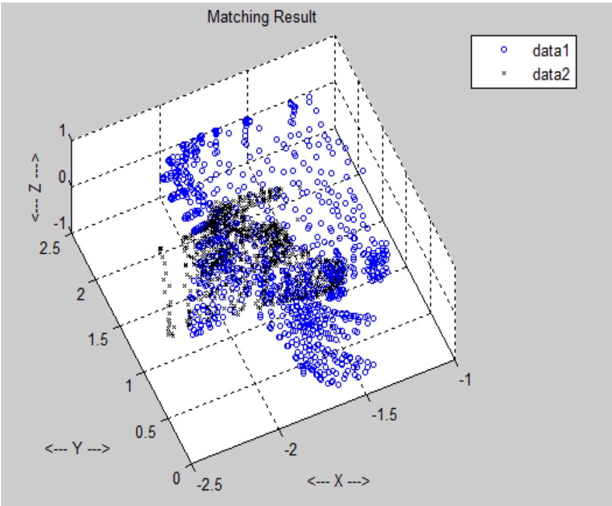
Division Number	Sub Division	Number of Itterate	Distance	Distance/ Itterate
Nokta 19	18	31	21,4388	0,69157
Nokta 19	19	20	19,973	0,99865
Nokta 19	20	20	63,352	3,16760
Nokta 20	01	9	56,756	6,30622
Nokta 20	02	25	42,536	1,70144
Nokta 20	03	87	41,0484	0,47182
Nokta 20	04	51	48,3249	0,94755
Nokta 20	05	20	44,5829	2,22915
Nokta 20	06	23	50,0715	2,17702
Nokta 20	07	22	44,3078	2,01399
Nokta 20	08	25	50,6942	2,02777
Nokta 20	09	11	48,4255	4,40232
Nokta 20	10	47	21,3081	0,45336
Nokta 20	11	27	21,4799	0,79555
Nokta 20	12	38	17,4964	0,46043
Nokta 20	13	108	19,0287	0,17619
Nokta 20	14	43	20,2275	0,47041
Nokta 20	15	23	48,798	2,12165
Nokta 20	16	15	34,7614	2,31743
Nokta 20	17	45	19,0667	0,42370
Nokta 20	18	99	21,6538	0,21873
Nokta 20	19	66	31,0665	0,47070
Nokta 20	20	37	53,4798	1,44540

Table A. 1. Result of ICP on each sliced piece of Saltanat Gate

Division Number	Average increasing rate of Itteration
Nokta 01	1,757587696
Nokta 02	4,142996078
Nokta 03	7,95030379
Nokta 04	2,386564687
Nokta 05	2,043446783
Nokta 06	1,705864889
Nokta 07	0,733741291
Nokta 08	2,448263806
Nokta 09	3,457456248
Nokta 10	1,714296902
Nokta 11	0,783988171
Nokta 12	4,823368766
Nokta 13	1,504758143
Nokta 14	3,996516516
Nokta 15	1,945368964
Nokta 16	3,655158228
Nokta 17	5,560692376
Nokta 18	2,271866436
Nokta 19	1,963003793
Nokta 20	1,581541639

Table A. 2. Average increasing rate of ICP on division of point cloud





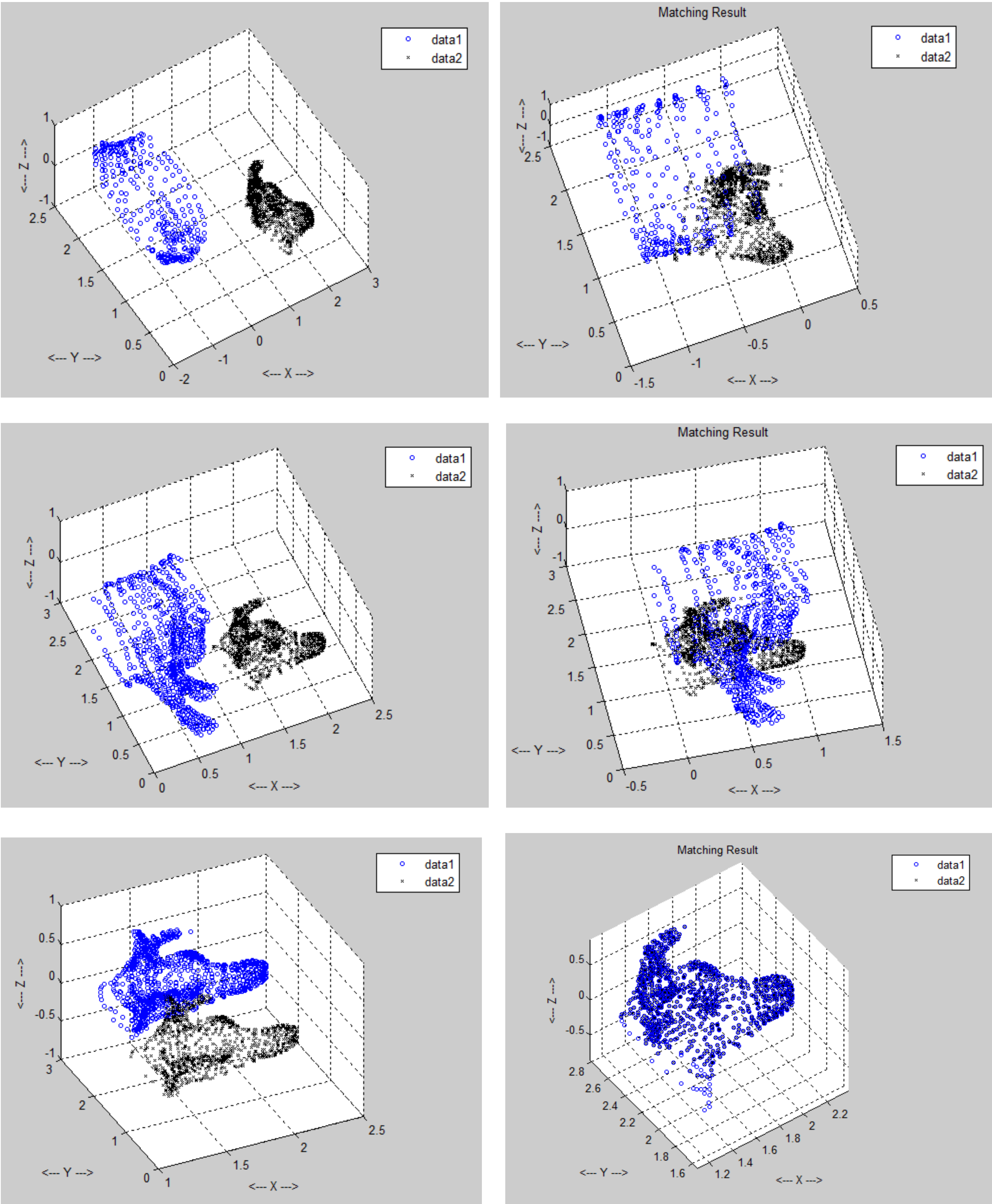


Figure A. 1 . ICP results on each sliced part of whole cow data. Last data shows exact match of two compared data

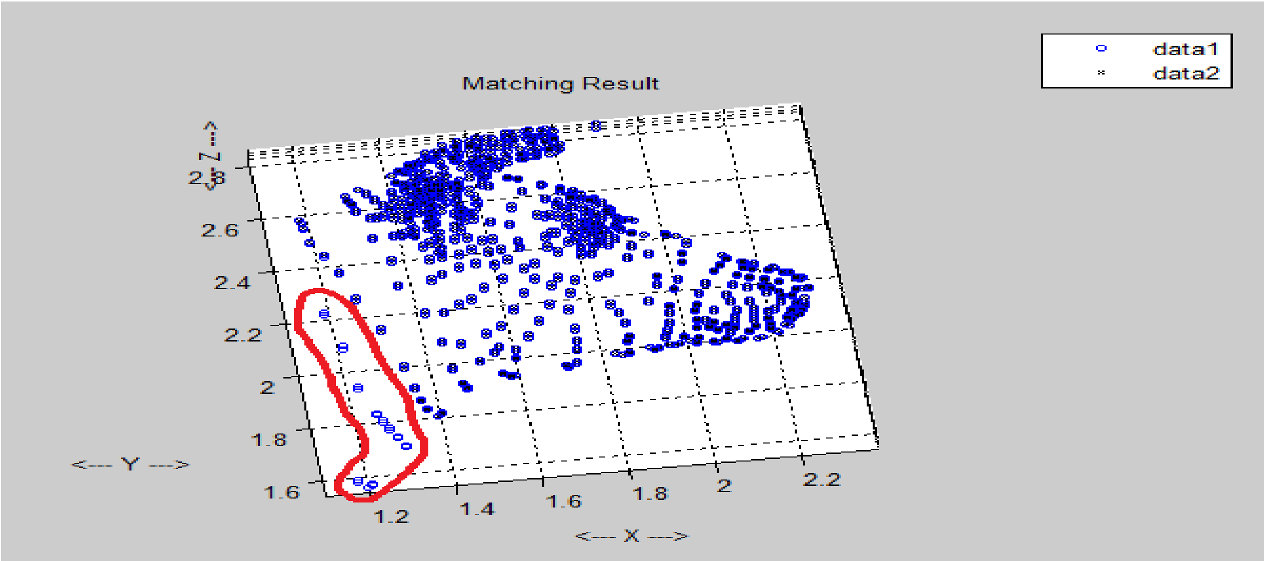
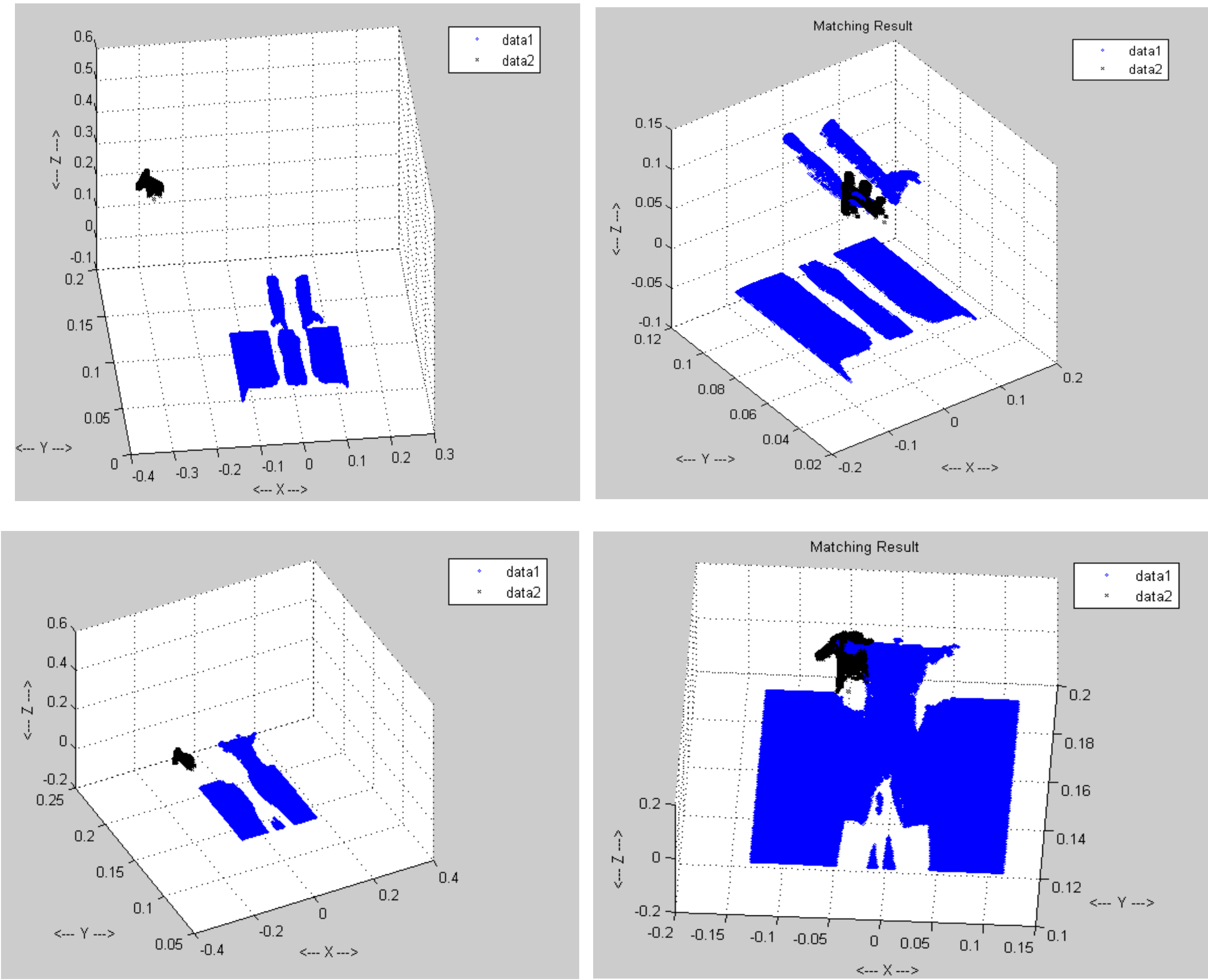


Figure A.2. Even a few points over sized of sliced object the recognizing result is still perfect.



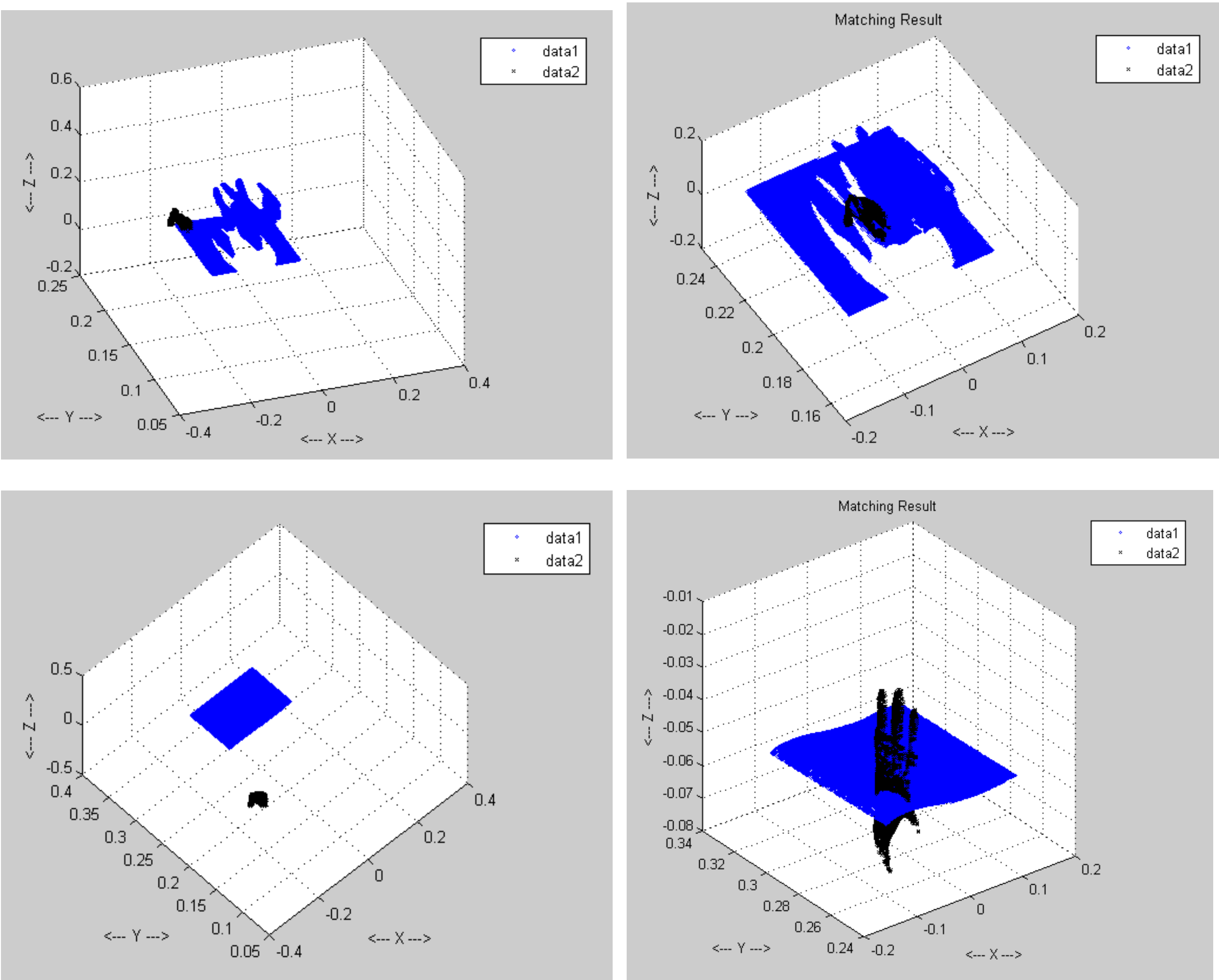


Figure A.3 shows exactly what we have expected from our wrong using ICP on different size.

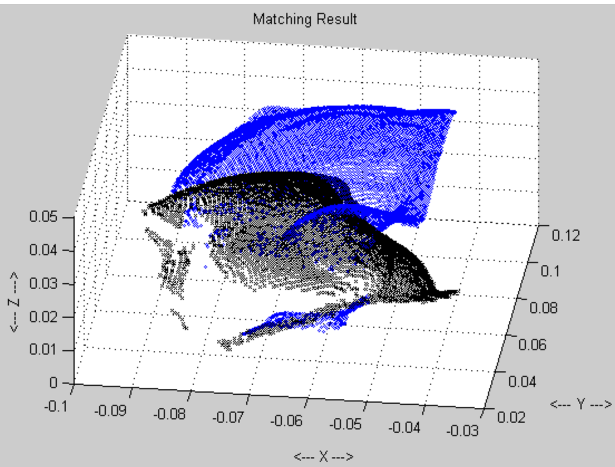
Division Number	Sub Division	Number of Iterate	Distance	Distance/Iterate
NoktaEks1	1	24	0,71573	0,029822
NoktaEks1	2	21	0,69125	0,032917
NoktaEks1	3	12	0,45523	0,037936
NoktaEks1	4	22	0,42535	0,019334
NoktaEks1	5	9	0,91931	0,102146
NoktaEks1	6	15	0,52805	0,035203
NoktaEks1	7	9	0,69845	0,077606
NoktaEks1	8	29	0,69036	0,023806
NoktaEks2	1	35	0,71903	0,020544
NoktaEks2	2	67	0,68698	0,010253
NoktaEks2	3	9	0,78953	0,087726
NoktaEks2	4	26	0,41247	0,015864
NoktaEks2	5	15	0,29016	0,019344

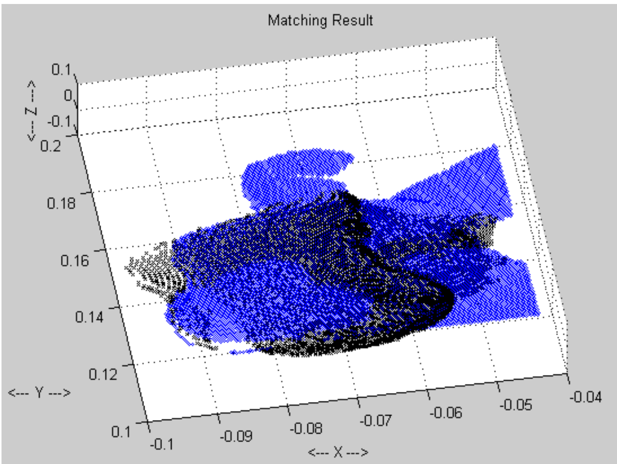
NoktaEks2	6	28	0,46642	0,016658
NoktaEks2	7	4	1,2706	0,31765
NoktaEks2	8	65	0,69327	0,010666
NoktaEks3	1	42	0,71721	0,017076
NoktaEks3	2	11	0,71603	0,065094
NoktaEks3	3	67	0,22601	0,003373
NoktaEks3	4	107	0,34947	0,003266
NoktaEks3	5	43	0,20661	0,004805
NoktaEks3	6	20	0,59288	0,029644
NoktaEks3	7	14	0,41084	0,029346
NoktaEks3	8	72	0,69427	0,009643
NoktaEks4	1	65	0,71942	0,011068
NoktaEks4	2	68	0,6937	0,010201
NoktaEks4	3	78	0,68166	0,008739
NoktaEks4	4	58	0,68568	0,011822
NoktaEks4	5	67	0,68867	0,010279
NoktaEks4	6	72	0,69917	0,009711
NoktaEks4	7	47	0,71218	0,015153
NoktaEks4	8	124	0,69813	0,00563

Table A.3. Result of ICP on each sliced piece of Armadillo

Division Number	Average Increasing rate of Iteration
NoktaEks1	0,044846079
NoktaEks2	0,06233806
NoktaEks3	0,020280833
NoktaEks4	0,010325371

Table A.4. Average increasing rate of ICP on division of point cloud





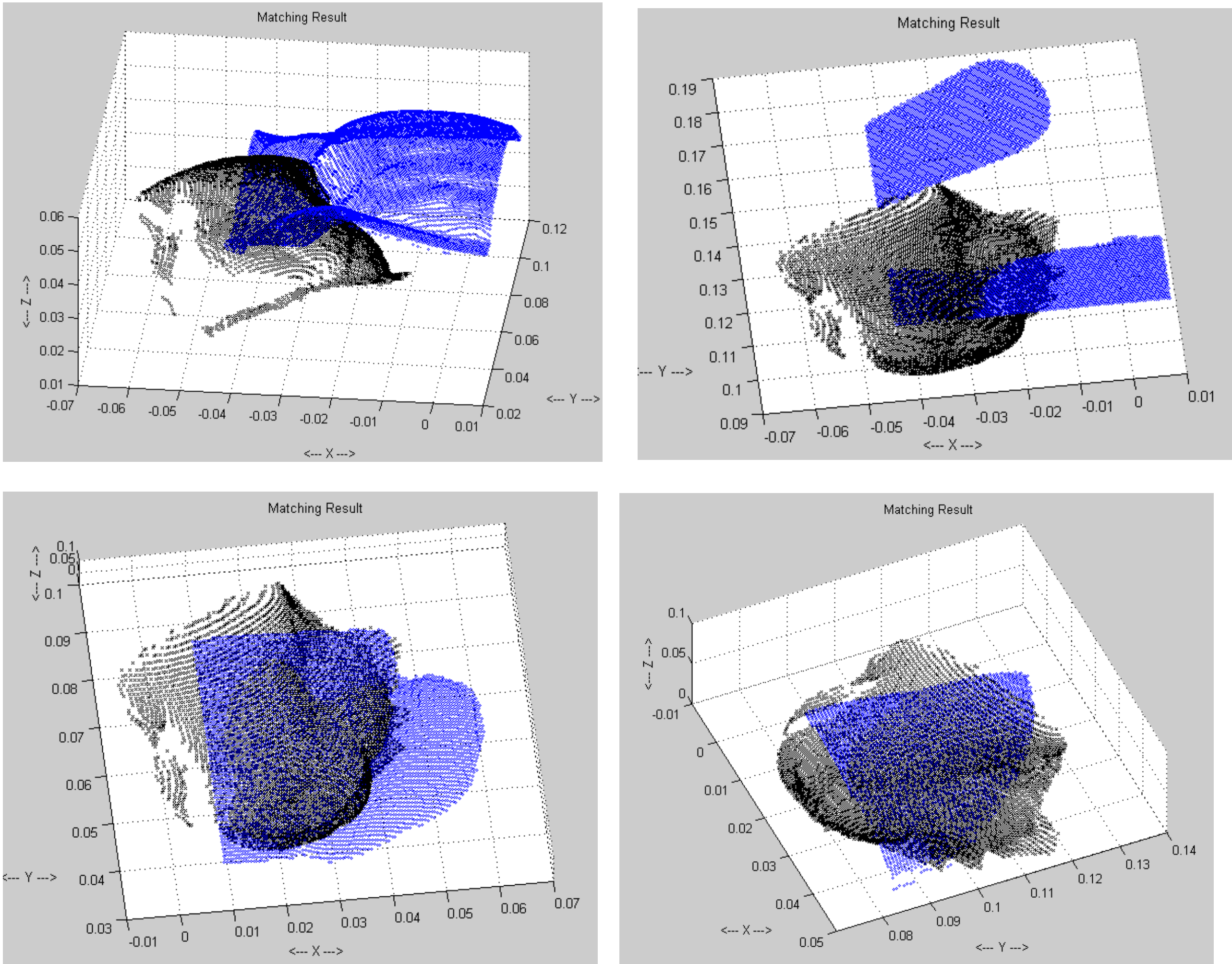


Figure A. 4 ICP results on each sliced part of whole bunny data. The second data shows exact match of two compared data

Division Number	Number of Iterate	Distance
NoktaEks1_1	31	0.54766
NoktaEks1_2	27	0.060251
NoktaEks2_1	20	0.64014
NoktaEks2_2	17	0.89446
NoktaEks3_1	17	0.29889
NoktaEks3_2	28	0.31615

Table A.5. Result of ICP on each sliced piece of Bunny

Division Number	Sub Division	Number of Iterate	Distance
NoktaEks1	1	19	35,4777
NoktaEks1	2	10	18,3836
NoktaEks1	3	13	8,9376
NoktaEks1	4	19	6,0178
NoktaEks1	5	15	8,1549
NoktaEks2	1	5	32,998
NoktaEks2	2	8	45,0732
NoktaEks2	3	8	58,7288
NoktaEks2	5	12	57,9235
NoktaEks3	1	8	15,5928
NoktaEks3	2	12	0,0001
NoktaEks3	3	22	7,9671
NoktaEks3	4	45	4,846
NoktaEks3	5	27	6,0626
NoktaEks4	1	26	6,1976
NoktaEks4	2	20	5,2937
NoktaEks4	3	10	4,4611
NoktaEks4	4	31	3,7144
NoktaEks4	5	40	4,8113
NoktaEks5	1	4	59,9731
NoktaEks5	3	90	3,0442
NoktaEks5	4	7	3,9279
NoktaEks5	5	19	4,7333
NoktaEks6	1	14	8,3267
NoktaEks6	2	26	7,3062
NoktaEks6	3	12	13,8357
NoktaEks6	5	21	6,8579
NoktaEks7	1	14	22,8334
NoktaEks7	2	3	37,4566
NoktaEks7	3	4	31,4038
NoktaEks7	4	6	17,5423
NoktaEks7	5	20	4,8417

Table A.6. Result of ICP on each sliced piece of Jet

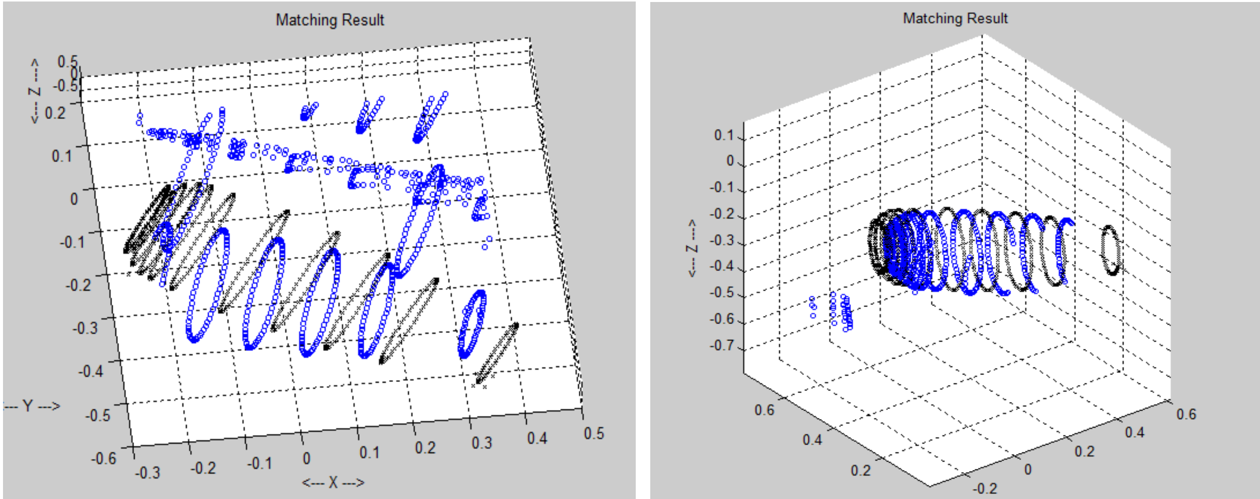


Figure A. 5. The effect of noisy data which leads mismatching of data.

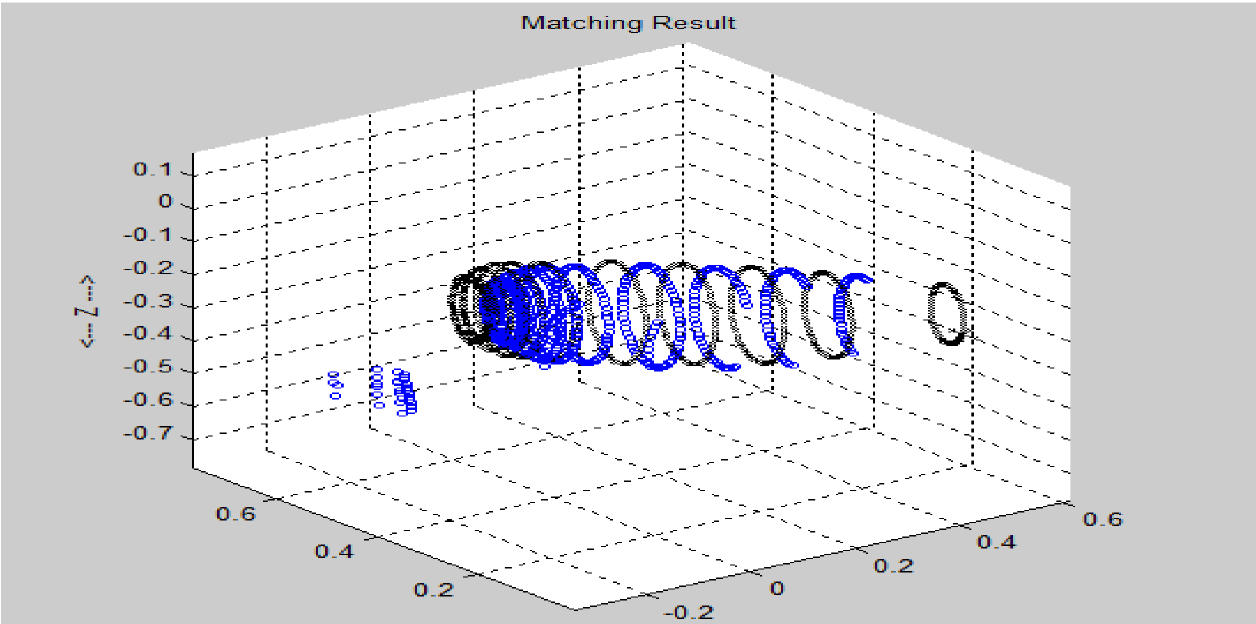


Figure A.6. The effect of noisy data gives us 0.001 distances between two 3D object however these data is related but the distance must be more then 0.001

Division Number	Sub Division	Number of Itteration	Distance
NoktaEks6	5	28	5,2136
NoktaEks7	3	24	7,5672
NoktaEks6	3	16	8,4053
NoktaEks6	6	16	8,5653
NoktaEks7	1	22	9,0063
NoktaEks7	2	21	9,4519
NoktaEks6	1	22	9,7354
NoktaEks6	2	18	9,7408
NoktaEks7	4	7	9,7565
NoktaEks9	7	12	10,1582
NoktaEks6	4	21	10,2044
NoktaEks9	6	15	10,5662
NoktaEks7	7	8	10,6337
NoktaEks8	4	18	11,3527
NoktaEks9	1	10	12,5562
NoktaEks4	7	9	12,6639
NoktaEks5	7	8	12,948
NoktaEks8	1	16	12,9528
NoktaEks5	5	14	13,2657
NoktaEks5	6	15	13,3671
NoktaEks8	6	14	13,7417
NoktaEks5	4	18	13,8565
NoktaEks8	5	10	14,0911
NoktaEks8	7	14	14,7835
NoktaEks5	2	10	15,1958
NoktaEks4	2	11	15,2875
NoktaEks3	1	8	15,9693
NoktaEks5	1	17	16,6658
NoktaEks6	7	8	16,8093
NoktaEks5	3	10	17,0731
NoktaEks4	1	7	17,7749
NoktaEks1	4	9	18,3886
NoktaEks2	5	13	22,0931
NoktaEks3	7	6	22,3463
NoktaEks1	3	8	23,4983
NoktaEks3	2	11	23,9245
NoktaEks2	3	11	24,8462
NoktaEks1	5	7	25,0073
NoktaEks2	4	12	26,9985
NoktaEks2	6	11	28,0941
NoktaEks9	5	8	29,3383
NoktaEks10	4	9	30,4031
NoktaEks1	2	5	31,1124
NoktaEks9	2	5	35,3107
NoktaEks2	2	5	36,5974
NoktaEks10	5	12	36,9751
NoktaEks10	3	8	36,9993
NoktaEks10	6	8	38,4633
NoktaEks10	7	6	39,7656
NoktaEks7	5	6	40,7629
NoktaEks10	2	4	42,8127
NoktaEks8	2	4	42,9103
NoktaEks2	7	7	46,5395
NoktaEks1	6	5	47,6285
NoktaEks1	1	4	47,7883
NoktaEks1	7	5	50,7857
NoktaEks2	1	3	52,8702
NoktaEks4	5	6	59,7372
NoktaEks8	3	5	60,2362
NoktaEks10	1	3	60,9434

Table A.7. Result of ICP on each sliced piece of Jet. Red colour dedicates bad recognizing Blue dedicates good recognizing