



Erdal Özbay
Ahmet Çınar

Fırat University, Elazığ-Turkey
erdalozbay@firat.edu.tr, acinar1972@gmail.com,

<http://dx.doi.org/10.12739/NWSA.2016.11.4.2A0104>

A METRIC BASED NOVEL CLASSIFICATION APPROACH TO KINECT POINT CLOUD DATA

ABSTRACT

Object classification in 3D point cloud data is an emerging topic attracting increasing research interest. Object detection is one of the most important challenges in computer vision. This paper proposes a novel method for the efficient detection of the real objects with respect to 3D point cloud data. The real object detection is performed using a technique based on mean shift clustering algorithm. The efficiency of the method is verified comparative official 3D data and real 3D point cloud data. We embed presented approach in a framework that combination of extracts shape and point cloud data metric information to improve the outcome of the classification stage. For this aim, classification of 3D point cloud data allows robust segmentation and feature descriptions into different objects by significantly reducing the error. Performed mean shift classification algorithm on the raw data and metric data classification with mean shift algorithm implementation are automatically compared to for evaluation the accuracy of the classification of metric classification algorithm. The results obtained metric classification algorithm and mean shift algorithm on automatic classification of simple planimetric object shapes with the surface of the point cloud show that proposed method is an efficient process.

Keywords: Classification, Point Cloud, Mean-Shift, Object Detection, Kinect

KINECT NOKTA BULUTU VERİSİNE METRİK TABANLI YENİ BİR SINIFLANDIRMA YAKLAŞIMI

ÖZ

3B nokta bulutu verileri ile nesne sınıflandırma araştırma alanında gelişmekte olan bir konudur. Nesne algılama bilgisayarlı görme çalışmalarında en önemli sorunlardan biridir. Bu yazıda 3B nokta bulutu verilerine göre gerçek nesnelerin etkin şekilde tespiti için yeni bir metriksel yöntem önermekteyiz. Gerçek nesnelerin algılanması ortalama kayma (mean-shift) sınıflandırma algoritmasına dayalı bir yöntemle yapılmaktadır. Yöntemin verimliliği karşılaştırmalı resmi 3B verileri ve gerçek 3B nokta bulutu verileri ile doğrulanmaktadır. Nokta bulutu verileri ve metrik bilgilerin kombinasyonu bir yazılım çerçevesinde uygulanarak sınıflandırma aşamasının sonuçlarını iyileştirilmektedir. Bu amaçla, 3B nokta bulutu verilerinin sınıflandırılması önemli ölçüde hata azaltılarak farklı nesnelere içine sağlam segmentasyon ve özellik çıkarımları değerlendirilmiştir. Ham veri üzerine uygulanan ortalama kayma algoritması ile metrik sınıflandırma gerçekleştirilmiş veri üzerine ortalama kayma algoritması uygulaması otomatik olarak karşılaştırılarak metrik sınıflandırma algoritmasının doğruluğu değerlendirilmektedir. Sonuçlar metrik sınıflandırma algoritmasının basit düzlemsel şekle sahip farklı nesnelerin otomatik olarak sınıflandırılmasına verimli bir süreçte sahip olduğu göstermektedir.

Anahtar Kelimeler: Sınıflandırma, Nokta Bulutu, Mean-Shift, Nesne tanıma, Kinect



1. INTRODUCTION

Object detection is the process of arranging for each object of the data obtained using a low cost sensor camera which located on the same floor, having no mutual data and producing the digital position information of distinctly separate objects from each other [1]. Object detection by subtracting the realization of digital information in the form of a 3D point cloud brings the very new field of study [2 and 3]. For this aim, low-cost range sensors and Kinect sensor devices can be used. Low-cost range sensors are an attractive alternative according to expensive laser scanners in application areas such as scene understanding, indoor mapping, robot localization and navigation, map registration, surveillance, robot manipulation, pattern recognition and image analysis,...,etc. [4]. One of the most usable devices and recent devices developments in sensing technology is Microsoft's Kinect sensor camera [5].

Kinect has originally designed to help the natural interaction in a computer game environment. However, the available properties of data captured by Kinect have attracted the attention of researchers working in the field of 3D modeling and mapping. The Kinect sensor captures depth data and color images together at a frame rate of approximately 30 frames per second [6]. Integration of depth data can be provided to capture in each frame by obtaining a point cloud containing approximately 300,000 points. This low-cost sensor, when compared to the other scanning techniques, are able to survey the 3D surface by acquiring a large amount of data in a shorter time and as often as required [7]. Low-cost sensor can obtain with the resulting point cloud scanning; basic measurement data, 2D or 3D graphical drawings, solid surface models, photometric images, 3D animation or 3D models dressed tissue. These laser scanners have practical and effective data collection techniques, both for researchers who will perform the measurement process and provide great convenience for using these measures [8].

The approach of a mesh on the object's surfaces (meshing) are usually done by obtaining reference data from objects that has a complicated exterior surfaces such as animal models, sculptures, buildings and art models with the aid of a laser scanner [9]. The basic method to distinguish geometrical based objects is done by extracting and using the data from a low cost laser scanner. This method is especially used in geometrical objects such as plane, cylinders, triangles, cones, spheres, ..., etc. When examining the studies done using the data gathered from a low-cost laser scanner etc., we can see that the geometrical based modeling is usually used to distinguish between objects [10]. In other study, the segmentation of the classification process of extracted point cloud from simple geometrical objects is performed [11]. The processing steps of the using methods are:

- Obtaining the rough data of the object's surface depth information,
- Metrical classification stage,
- Data normalization phase,
- Application of mean shift algorithm and processing the segmentation step.

In this context, a simple geometrical surfaces (square, sphere, triangle, plane) are automatically classified by acquiring 3 dimensional point cloud data by using a low-cost Kinect camera. Metrical classification system is used while extracting geometrical surfaces from point clouds. Subsequently, the mean shift is executed while a normalization process is applied. To assess the accuracy of

the mean shift algorithm's surface scanning, untreated raw data are applied and compared with normalized data. The paper proceeds with a description of the depth measurement principle, developed mathematical model and the calibration of depth data parameters in section 2. In section 3 and 4, metric classification process and normalization application is detailed. In section 5, mean shift algorithm implementation steps and verified through a number of experiments and the results are discussed in section 6. The paper concludes with some remarks in section 7.

2. RESEARCH SIGNIFICANCE

This paper proposes a novel method for the efficient detection of the real objects with respect to 3D point cloud data. The real object detection is performed using a technique based on mean shift clustering algorithm. The efficiency of the method is verified comparative official 3D data and real 3D point cloud data. We embed presented approach in a framework that combination of extracts shape and point cloud data metric information to improve the outcome of the classification stage. For this aim, classification of 3D point cloud data allows robust segmentation and feature descriptions into different objects by significantly reducing the error. Performed mean shift classification algorithm on the raw data and metric data classification with mean shift algorithm implementation are automatically compared to for evaluation the accuracy of the classification of metric classification algorithm. The results obtained metric classification algorithm and mean shift algorithm on automatic classification of simple planimetric object shapes with the surface of the point cloud show that proposed method is an efficient process.

3. DEVELOPED MATHEMATICAL MODEL AND CALIBRATION OF DEPTH DATA

3.1. Kinect Sensor Technology

The depth sensor is able to send back images as if it was an ordinary camera, but instead of color, the pixel value represents the distance to the point. As such, the sensor can be seen as a range, or 3D camera. The basic technique of the depth sensor is to emit an infrared light with an IR laser diode, and calculate the depth from the reflection of the light at different positions using a traditional IR-sensitive camera.

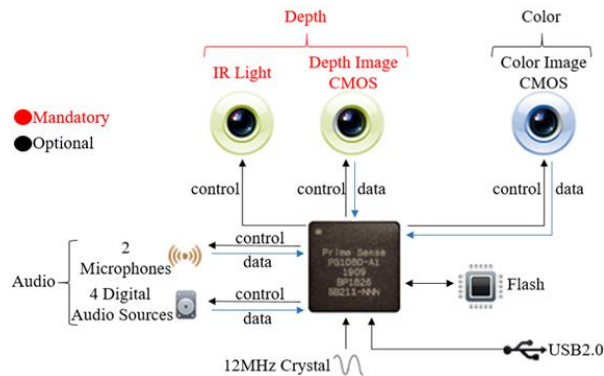


Figure 1. Kinect hardware specifications

The technology behind the Kinect sensor was originally developed by a company called PrimeSense, which released their version of an SDK to be used with the Kinect as part of the OpenNI organization [12]. The basic principle behind the Kinect depth sensor is the emission of

an IR pattern and simultaneously capturing the IR image with a traditional CMOS camera outfitted with an IR-pass filter, as show in figure 1. The image processor of the Kinect uses the relative positions of the dots in the pattern to calculate the depth displacement at each pixel position in the image [13]. It should be noted that the actual depth values are distances from the camera-laser plane rather than the distances from the sensor itself.

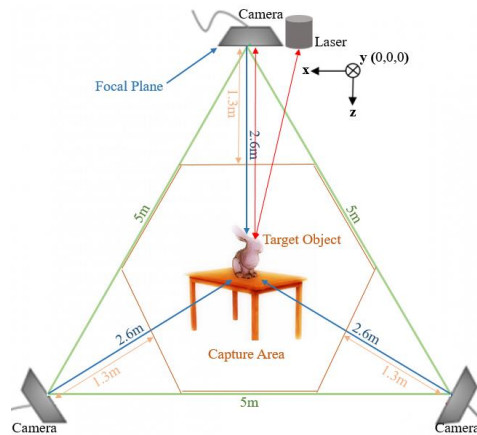


Figure 2. Presented Kinect capture system

Figure 2 describes how the Kinect depth to estimate ought to be interpreted. The depth is to mean that estimation of the distance from object to the camera-laser plane rather than the actual distance from the object to the sensor. As such, the depth sensor can be understood as a device which returns (x, y, z) coordinates of 3D objects. Kinect hardware specifications are shown in table 1.

Table 1. Kinect hardware features

Property	Value
Angular Field-of-View	57°horizontal, 43°vertical
Framerate	approximately 30Hz
Nominal spatial range	640 x 480 (VGA)
Nominal spatial resolution (2m)	3mm
Nominal depth range	0.8m - 3.5m
Nominal depth resolution (2m)	1cm
Device connection type	USB (+ external power)

3.2. Calibration of Depth Data

This section outlines the proposed framework for the depth measurement principle, developed mathematical model and the calibration of depth data parameters of the 3D real object data inside a scene [14]. The flow diagram sketching the algorithm pipeline is shown in figure 3.

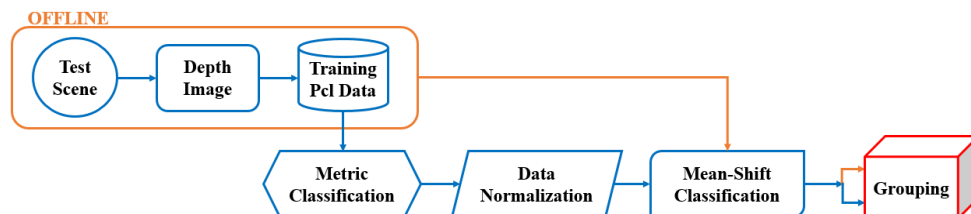


Figure 3. Flow diagram of the proposed 3D data classification algorithm

In this study, depth information helps create a set of point cloud from actual objects and metric classification algorithm is used to extract simple geometric surfaces. The needed normalization procedure to use the mean shift algorithm is conducted with Visual Studio .Net and using C# as a programming language. As for making point cloud data, point cloud library is used which is itself an open source library (PCL: Point Cloud Library) [15]. With PCL, filtering, object prediction, surface reconstruction, model creation, segmentation, registration and such algorithms can be performed. Java codes are used for the mean shift algorithm in which it is used for the extraction of simple geometrical surfaces from point clouds and is re-run by adapting to the data set used in the study. For each pixel the distance to the sensor camera can then be retrieved from the corresponding disparity as follows. Figure 4 illustrates the depth measurement from the speckle pattern.

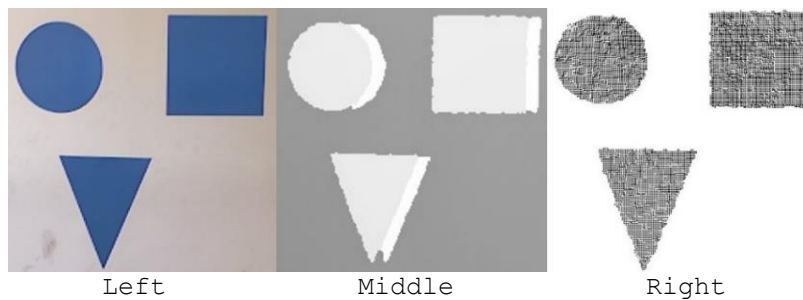


Figure 4. Left: infrared image of the pattern of speckles projected on the planimetric object, Middle: the resulting of depth image, Right: raw data of the simple planimetric objects

In calibration phase, the reflection time of the sensor light is calculated for each voxel by means of the depth sensor, and distance to the object is determined. From this viewpoint, distance between the camera and the objects for which model is to be created and their stance positions are important. As shown in the figure 5 sensor camera and objects are placed a mechanism to obtain more delicate and fixed images to prevent shifting in the coordinate points during uniting of images.

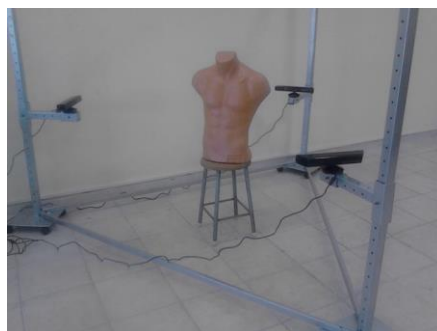


Figure 5. Developed mechanism of the camera and object placement

As mentioned, the calibration parameters involved in the geometric object model for the calculation of 3d coordinates from the raw image measurements include [14]; distance of the reference pattern (Z_p), Z_t denotes the distance (depth) of the point t in object, focal length of the infrared camera (L_f), base length (L_b), lens distortion coefficients (δx , δy) and principal point offsets (x_p , y_p).

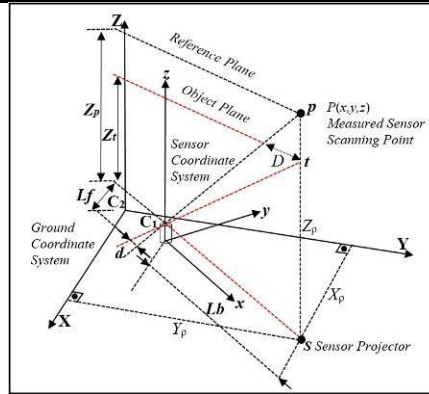


Figure 6. Developed low-cost sensor scanner measurements of depth-disparity relation on sensor and ground coordinate system

Figure 6 shows the distance of an object at t point from the sensor and measure of d for its unbalanced relation with the reference plane in geometrically. We consider a depth coordinate system C_2 with its origin at the perspective center of the low cost sensor camera C_1 . The X and Y axis perpendicular to the Z axis in the direction of the baseline L_b between the low cost camera center and the sensor projector, Z axis is orthogonal to the image plane towards the object. Let's say that an object is at a certain Z_p distance from a sensor on P point which is in a reference plane. If this object gets closer or far from the sensor camera, the object will change place along the x plane, located on the image plane. The measured irregularity image plane of the t point on an object plane is equal to d and changes place as much as D on an object plane.

From the similarity of triangles we have [16]:

$$\frac{D}{L_b} = \frac{Z_p - Z_t}{Z_p} \quad \text{and} \quad \frac{d}{L_f} = \frac{D}{Z_t} \quad (1)$$

Substituting D in equation (1) and expressing Z_t in terms of the other variables yields:

$$Z_t = \frac{Z_p \cdot L_f \cdot L_b}{L_f \cdot L_b + Z_p \cdot d} \quad (2)$$

In equation (2) the mathematical model of sensor and ground coordinate system can be determined by calibration for derivation of depth from the observed disparity provided that the constant parameters Z_p , L_f , and L_b . The planimetric objects with simple geometric shape coordinates of each point can be calculated from its image coordinates and the scale values:

$$X_t = -\frac{Z_t}{L_f}(x_t - x_p + \delta x), Y_t = -\frac{Z_t}{L_f}(y_t - y_p + \delta y) \quad (3)$$

In equation (3) X_t and Y_t are the image coordinates of a point (pcl), x_p and y_p are the coordinates of the principal point, and δx and δy are corrections for camera lens distortion. In here we assume that the image coordinate system is parallel to depth coordinate system and thus with the base line [16]. As a pixel size the resolution of the depth of the infrared sensor camera, determines the point density of the depth data on the X - Y coordinate plane (perpendicular to camera axis). By the way, since each depth image contains a constant 640x480 pixels the point density will decrease with increasing distance of the object surface from the sensor camera. The density of the points that a number of dots per unit area is proportional to the square distance from the sensor to the object plane. Therefore, the point density is



inversely proportional to the square distance from the sensor camera. The resolution of depth images are evaluated by the number of bits per pixel used to store the disparity measurements. The Kinect disparity measurements are gathered as 11-bit integers, where 1 bit is earmarked to mark the pixels for which no disparity is measured, so-called no data. In practice, it is not possible to stream the real measured disparities, usually due to bandwidth limitation. Alternately, the raw depth disparity values are normalized between 0-2047, and together streamed as 11 bit integers. Therefore the equation (2) has been renewed. d should be restored with $md' + n$ with d' the normalized disparity. m , n are the parameters of a linear normalization (nominally). Including these in equation (2) and converting it yields:

$$Z_t^{-1} = \left(\frac{m}{L_f \cdot L_b} \right) d^{-1} + \left(Z_p^{-1} + \frac{n}{L_f \cdot L_b} \right) \quad (4)$$

Equation (4) expresses a linear relation between the inverse depth of a point and its corresponding normalized disparity. By examining the normalized disparity for a number of object points at known distances to the sensor camera the coefficients of this linear relation can be measured in a least-squares form. However, the inclusion of the normalization parameters does not allow determining L_b and Z_p separately. Since depth stream is inversely proportional to disparity the resolution of depth is also inversely related to the levels of disparity. So, depth resolution rate is not constant and decreases with increasing distance to the sensor camera. For instance, approximately at a range of 1.5 meters one level of disparity corresponds to about 1 cm depth resolution, whereas at a range of 5 meters one disparity level corresponds to about 7 cm depth resolution. After calibration phase in order to make classification process the raw data of the object are obtained approximately 1.5 meters from the object-sensor camera distance.

4. METRIC CLASSIFICATION

The segmentation stage in geometrical based modeling is the most important stage modeling. In this stage, if a planimetric object has prominent geometrical features, it is separated with cloud point by a specified method. In the segmentation of the data from terrestrial laser scanners, data from the cloud point are grouped and classified by their geometrical characteristics and features. Nowadays, different types and methods of segmentation are used in cloud point's data [17]. Among these methods are the Hough transformation, mathematical morphology, area development, Ransac algorithm and metrical classification. In this study, the process of geometrical based point cloud classification segmentation is performed. In this context, surfaces having a simple geometrical shape (triangle, sphere, and square) with 3 dimensional point cloud data obtained by terrestrial laser scanner are classified automatically. Data set has been obtained in laboratory environment. For the extraction of geometrical surfaces within point clouds, a k -NN (k -Nearest Neighbor) based metrical classification algorithm is used [18]. In object detection, the k -NN algorithm is a non-parametric method for classification. The input consists of the k -closest training examples in the feature space. The output depends on whether k -NN is used for classification. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. Behind the object is simply assigned to the class of that single nearest neighbor [19]. To evaluate this algorithm's surface accuracy classification, mean shift classification algorithm has been applied over the applied metrical classification algorithm data [20]. The mean



shift algorithm is directly applied to the raw data with the metrical classification algorithm then evaluates the results of applied data. Generally Euclidean distance metric is used for continuous variables. Such as for text and speech classification another metric classification can be used for discrete variables, such as the overlap metric classification. k -NN has also been operated with correlation coefficients ingredient of gene expression microarray data [21]. The extract of simple geometric surface through the point cloud classification algorithm metric for each stage containing multiple planimetric objects was compiled [22]. A different classification procedure is provided for distinctively different geometrical objects. For this, the objects that are obtained by considering to calibration procedure shown above, has been worked on point cloud data set [23]. Z dimension of point cloud is not taken to process for the study with the fixed distance data. With using the metrical classification algorithm, we'll need to calculate every point of the point cloud to find out which point belongs to which object. Hence, one needs to know the point of this sequential port adjacent to each other. k -NN is used to determine the adjacent points [22]. According to the principle of the closest neighborhood, to determine the closest point to the point of operation, one needs to evaluate the Euclid distance of the neighboring point from the given point numbers.

Every point the objects point clouds parameters has been defined as $P(x, y, z)$ respectively. Every Z sized object has a stable, equal distance and because of their point cloud parameters being joint, it has been disabled. After the point parameters has been determined and set, plane with a specified threshold distance (t) are assigned to the point in the remaining planes. In every point cloud, the first reference data is accepted as the threshold value's first divergent points coordinates. To identify if any points remain within the threshold points, $P_i(x_i, y_i)$ and $P_{i+1}(x_{i+1}, y_{i+1})$ one has to calculate the Euclid distance between the points. The most important points that should be determined in this classification is to determine threshold distance (t). In this article, every t threshold value data set has been assigned in experimental. The threshold value distance (t) to a camera pointing at an object from approximately 1.5 meters is set in 6mm.

$$d(P_i, P_{i+1}) = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \leq t \quad (5)$$

```

-Begin
Step 1- $P_1(x_1, y_1)$  the first element of the input set is
assigned as a reference to the Object-1 ( $P_1$ ).
Step 2-input:  $D = \{(P_i, P_{i+1}), \dots, (P_{N-1}, P_N)\}$ 
Step 3- $P = (P_1, \dots, P_N)$  new instance to be classified
Step 4-calculate  $d(P_i, P_{i+1})$  each labelled instance ( $P_i, P_{i+1}$ )
Step 5-order  $d(P_i, P_{i+1})$  from lowest to highest, ( $i = 1, \dots, N$ )
Step 6-if  $d(P_i, P_{i+1}) \geq t$ 
As a result of inferences metric classification, it is also
included in existing object or considered as the first
reference point of a new object.
Step 7- (smallest  $x$   $O_1 < c_1 <$  biggest  $x$   $O_1$ ) &&  $(d(c_1, x)$ 
 $(P_i, P_{i+1}) \leq t$ ) included in existing object  $O_1$ .
-else
Step 8-the first reference point of a new object  $O_2$ .
Step 9-Assign to  $P$  the most frequent class in  $D_p^X$ 
-End
    
```

Figure 7. The pseudo code of the metric classification algorithm



The equation (5) shows with the Euclid distance calculation formula. The reason of why metric classification algorithm is used in conjunction with nearest neighbor classification. Because it is very simple and intuitive idea and relatively. Figure 7 depicts in pseudo code of metric classification.

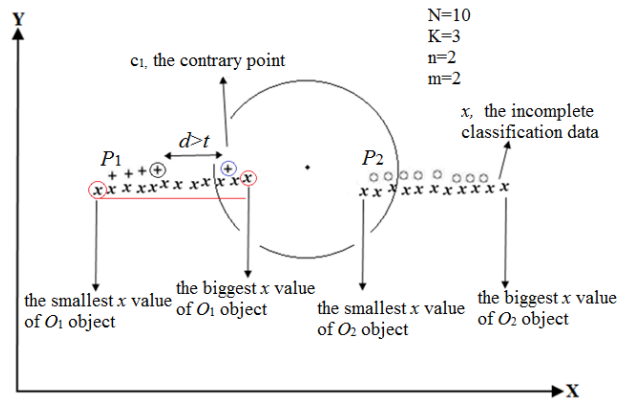


Figure 8. Example for the basic k-NN

In figure 8, m denotes the number of classes, n the number of predictor variables, and N the number of labelled cases. The metric classification process described as above $d(P_i, P_{i+1}) \geq t$, is explained as in which cases the pseudo code regarded point will be integrated in available object and in which point it will be used as a reference point to a new object as shown in figure 8. As seen in the figure 8, the contradictory point c_1 is consecutive order between the neighbors. If Euclid distance is exceeded 6mm, then candidate becomes the first reference point to a new object. Because $d > t$ or the $d > 6\text{mm}$ condition is provided, the metrical classification system will be considered. The inconsistent point c_1 which will be evaluated belongs to P_1 object which is controlled by the neighboring candidate spots ($d(c_1, x) \leq t$). Between these candidate spots at least one of them has to be a neighbor and c_1 point has to be between the maximum and minimum candidate coordinates. In some exceptional situations, the X dimension and Y dimension has been used in a similar logical procedures. Every data set has been classified and colored depending by their object numbers. RGB color codes are assigned as a value to the classified data so that the object point cloud can be distinguished.

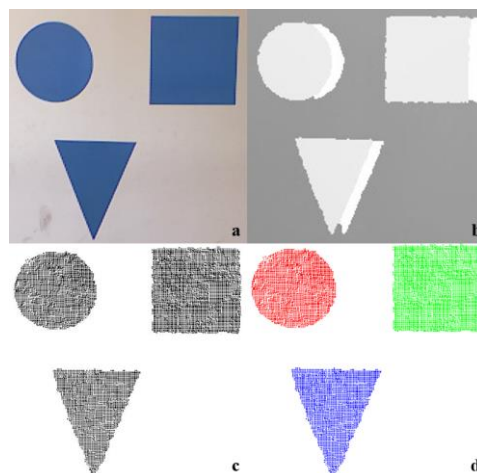


Figure 9. The classified data set of three different planimetric objects

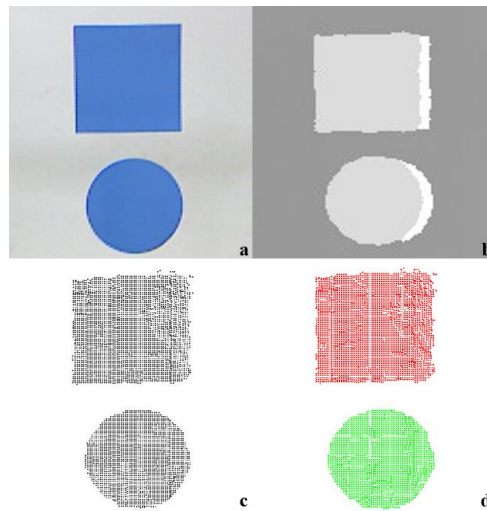


Figure 10. The classified data set of two different planimetric objects

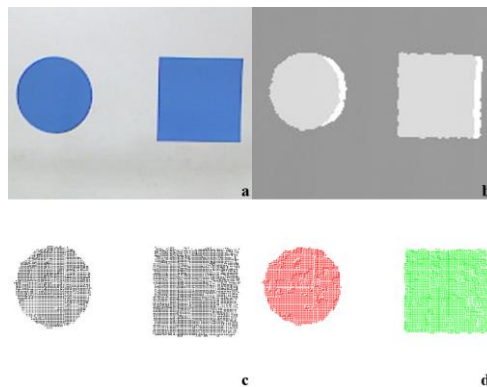
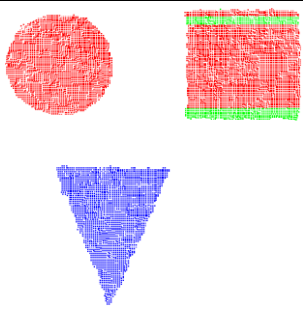
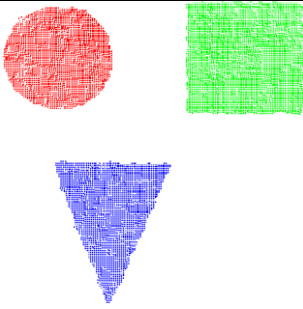
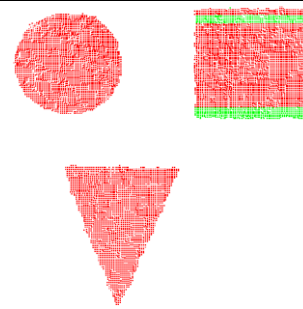
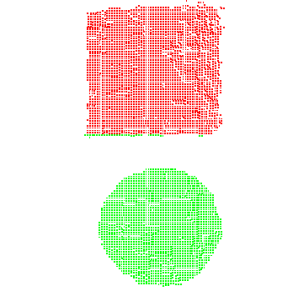
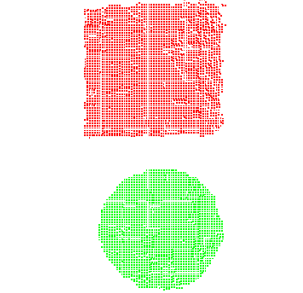
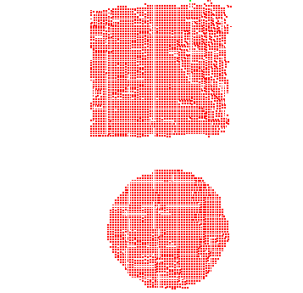
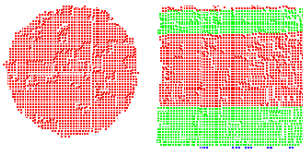
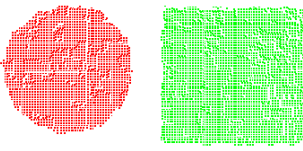
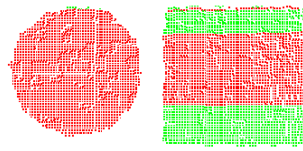


Figure 11. The classified data set of two different planimetric objects for the value of $t= 0.006m$. a) Calibration image of different objects, b) Depth images, c) The raw depth information of the objects (point cloud), d) Application of Metrical classification algorithm to classify data set of unbundled version of different objects

In Figure 9, the total point cloud data set: 9540, number of red dots: 2778 (circular object surface), number of green dots: 4324 (quadratic object surface), number of blue dots: 2438 (triangular object surface). In Figure 10 the total point cloud data set: 4879, number of red dots: 2938 (quadratic object surface), number of green dots: 1941 (circular object surface). In Figure 11 the total point cloud data set: 4871, number of red dots: 1934 (circular object surface), number of green dots: 2937 (quadratic object surface). Figures 9, 10 and 11 show the optimum classification results of three different data sets. The first set is belonging to three different objects, the second and third are the data sets belonging to two different objects. Different color code classification of planimetric objects into each data set are shown in Table 2.

Table 2. Classification results with different threshold values

Threshold/ Data Set	$t= 0,001m$ (error)	$t= 0,06m$ (optimum)	$t= 0,01m$ (error)
Circle, Square, Triangle			
Square, Circle			
Circle, Square			

While low-cost sensor cameras are using for scanning, because of the surface roughness of the object being scanned and the scattering properties of the surface of the laser beam, point clouds are created a thickness at the object surface [26]. For assignment of the points on the surface to the issued model, the threshold value of t must be determined accurately. Application results are shown in the table 2. Extracting object surface from point cloud with this metric classification algorithm, the most important parameter is t threshold distance need to be determining. In this context, the metric classification algorithm are applied to the data set with giving different values to t . threshold value t distances $t=0.001m$, $t=0.01m$, $t=0.005m$ and $t=0.007m$ are selected and applied to the data set. The results obtained from experiments, the best result is observed to be worth at $t=0.006m$. Thus, metric classification algorithm on the test data set of point cloud with the ultimate object of the planimetric t value classification is applied. In figure 9 (c), within the data set belonging to three different object, 9540 pieces number of points are available. These three object the data set is automatically classified such a short time 0.85×10^3ms in total. In the figure 9, the plane belonging to different objects in the same data set is classified as coloured with different colours. The number of points of each plane is shown depending on colour. As a result of above 9540 total points classification of cloud data sets, 2778 number of data are coloured with red point clouds, 4324 number of data are coloured with green point clouds and 2438 number of data coloured with blue point clouds. For displayed other data sets, metric classification process was performed using the algorithm separately. Similarly in figure 10, 4879 pieces are available within the data set belonging to two different objects. These two objects data set are classified in 0.44×10^3ms in

total. As a result of this classification, 2938 pieces of data are coloured with red point clouds, 1941 pieces of data coloured with green point clouds. Also in figure 11, 4871 pieces are available within the data set belongs to two different objects. These two objects data set are classified in 0.44×10^3 ms in total like the data set in figure 10. As a result of this classification, 1934 pieces are coloured with red point clouds, 2937 pieces are coloured with green point clouds.

5. NORMALIZATION

3D digital information obtained by Kinect sensor cameras is calibrated over a certain distance, each point on the surface of the object contains three-dimensional digital information [24]. Approximately 300,000 units of each digital data contains a positive or negative x and y information and a positive z information. z is a direct perpendicular distance information from the camera sensor surface of the object.

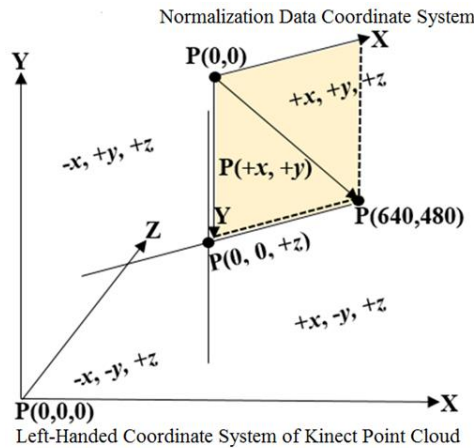


Figure 12. The developed normalization data coordinate system of the converted data normalization for the implementation of mean shift algorithm from left-handed coordinate system of the Kinect point cloud

In figure 12, normalized coordinate system and coordinate system which data obtained by Kinect camera are shown. Data relating to Kinect coordinate system should be normalized with mean shift algorithm in order to run with the Java environment. Belonging to the data obtained by Kinect, the x and y coordinates may have a negative value and after the comma it has 5 digits precision. In an example about 1.5m camera distance -18.14078, 32.36885, 151.6 x coordinate has a negative value. When completed the editing process of this data, the negative values x or y should be positive and z should be removed. After normalized the data related this coordinate values, it will be a positive integer value such as 319, 723. For this, algorithm is applied as shown below as the pseudo code in figure 13. This normalization algorithm for the mean shift classification algorithm using Kinect raw data is passed through metric classification process. In the point cloud's coordinate plane, without slip in position with each other and data are arranged. Thus, for applying mean shift algorithm point coordinate values in the entire data set have become positive and integer.



```

-Begin
Step 1- $\sum P_i(x_i, y_i, z_i)$  Remove z coordinate  $i=\{(P_1 \text{ to } P_N)\}=P_i(x_i, y_i)$ 
Step 2-find  $(sml\_x, sml\_y)$   $P_{sml}(x_{sml}, y_{sml})$   $sml=\{(P_1 \text{ to } P_N)\}$ 
    Find the smallest x and y values
     $\sum P_i(x_i, y_i)$ 
     $\left\{ \begin{array}{l} norm\_x = 200; \text{ if } (sml\_x < 0 \text{ } sml\_x \geq -10) \\ norm\_x = 300; \text{ if } (sml\_x < -10 \text{ } sml\_x \geq -20) \\ norm\_x = 400; \text{ if } (sml\_x < -20 \text{ } sml\_x \geq -30) \\ norm\_x = 500; \text{ if } (sml\_x < -30 \text{ } sml\_x \geq -40) \end{array} \right.$ 
     $i=\{(P_1 \text{ to } P_N)\}$ 
     $\left\{ \begin{array}{l} norm\_y = 200; \text{ if } (sml\_y < 0 \text{ } sml\_y \geq -10) \\ norm\_y = 300; \text{ if } (sml\_y < -10 \text{ } sml\_y \geq -20) \\ norm\_y = 400; \text{ if } (sml\_y < -20 \text{ } sml\_y \geq -30) \\ norm\_y = 500; \text{ if } (sml\_y < -30 \text{ } sml\_y \geq -40) \end{array} \right.$ 
     $i=\{(P_1 \text{ to } P_N)\}$ 
Step 3-int  $data\_x_i = (int)x_i * 10 + norm\_x_i$   $i=\{(P_1 \text{ to } P_N)\}$ 
Step 4-int  $data\_y_i = (int)y_i * 10 + norm\_y_i$   $i=\{(P_1 \text{ to } P_N)\}$ 
Step 5-write  $P_i(data\_x_i, data\_y_i)$   $i=\{(P_1 \text{ to } P_N)\}$ 
-End
    
```

Figure 13. Pseudo code of Normalization algorithm

6. APPLICATION OF METRIC BASED MEAN-SHIFT ALGORITHM

In this paper a simple iterative procedure is implemented in mean shift algorithm. The algorithm is that shifts each data point to the average of data points in its neighbourhood. This generalization makes some k-means like clustering algorithms its special cases. The mean shift algorithm is a statistical nonparametric clustering technique without having prior knowledge of the number of clusters, and does not constrain the shape of the clusters [25]. This algorithm aims to find out where the point spread is most intense. The algorithm was first proposed by Fukunaga and Hostetler and was later developed by the Comanicu. Given in equation (6), N data points x_i , $i = 1, \dots, N$ on a d -dimensional space R^d , the multivariate kernel density estimate obtained with kernel $K(x)$ and window radius h is as follows:

$$f(x) = \frac{1}{n \cdot h^d} \sum_{i=1}^N K\left(\frac{x-x_i}{h}\right) \tag{6}$$

In equation (6) where x_i are the input samples and $k(r)$ is the kernel function. h is the only parameter in the algorithm and is called the bandwidth. We calculate $f(x)$, and can find its local maxima using gradient ascent or some other optimization method. The trouble with "brute force" approach is for higher dimensions, it happens computationally prohibitive to evaluate $f(x)$ over the exact search space. Instead of mean shift uses a variant of what is known in the optimization literature as multiple restart gradient descent. Starting at some guess for a local maximum, y_k , which can be a random input data point x_1 , mean shift computes the gradient of the density estimate $f(x)$ at y_k and takes an uphill step in that direction. Traditional mean shift algorithm assesses to movement of the object's centre by calculating the average shift vector Δx . If we think that the reference point of the object's position \hat{x} and we want to calculate the new position of the object point \hat{x}^I , it will be $\hat{x}^I = \hat{x} + \Delta x$. At last, the mean shift vector of equation (7) is calculated as shown [26]:

$$\Delta x = \frac{\sum_i^N K(x_i - \hat{x}) \cdot w.(x_i) \cdot (x_i - \hat{x})}{\sum_i^N K(x_i - \hat{x}) \cdot w.(x_i)} \tag{7}$$

Mean shift is a procedure for positioning the maximum of a density function given discrete data sampled from equation (7). It is useful for determining the modes of density. It is an iterative technique and it starts with an initial estimate x . Let A kernel function $K(x_i - x)$ be given. This function identifies the weight of closer points for re-estimation of the mean. Generally a Gaussian



kernel on the distance to the current estimate is used, $K(x_i - x) = e^{-c \|x_i - x\|^2}$. The weighted mean of the density is identified with K . Kernel function K has a radially symmetrical structure and a bandwidth h . If model and colour of the two distribution functions of the candidate objects are the p and q , the weight value of x pixels are calculated as specified in equation (8):

$$w(x) = \sqrt{q(I(x)) / p(I(x))} \quad (8)$$

7. EXPERIMENTAL RESULTS

In this section, the proposed method of three results obtained metric classification algorithms running on an image array of different sets of data are given visually and numerically. The maximum number of iterations performed all experiments should be in the mean shift algorithm and threshold parameters are defined as 200 and 2 in order. Depending on the size of the data set programs that run on a Java environment from a graphic panel with a resolution of 640x480 and 1280x960 are visualized. Figure 14 is conducted in follow-up operations using the proposed method are shown the results obtained. As seen objects within each data set is provided by the classification separated significantly from one another. Classification criteria for success, before applying mean shift algorithm has been compared to depending on whether the metric is passing through the classification process. As a result, a single point of each data set is not even classified differently. So, that classification process has been completed successfully in 100% similarity. In order to verify the proposed algorithm, feature points classified exactly. Our algorithm can find classification for different objects data. Average similarities in feature points are shown in table 3. As a result common data set with the same number of points have been obtained in less time classified with the same number of data results.

In figures 14, before the implementation of the mean shift algorithm metric classification method proposing whether using the results or not obtained is shown. Data sets are subject to the metric classification algorithm is implemented in the first sequence of images edited by the normalized mean shift algorithm. In the second image sequences mean shift algorithm is directly run on the raw data. As a result, more uniform and rapid planimetric objects in the data passed from the metric process is classified in a clearly separated from each other. The mean shift algorithm is implemented without metric process is irregular and take place longer because it lacks the pre-classification process. Both ultimately result in the data is classified in a similar manner 100%. This means that the proposed metric classification method, in less time with the same success ratio shows that fulfill the desired classification process [27].

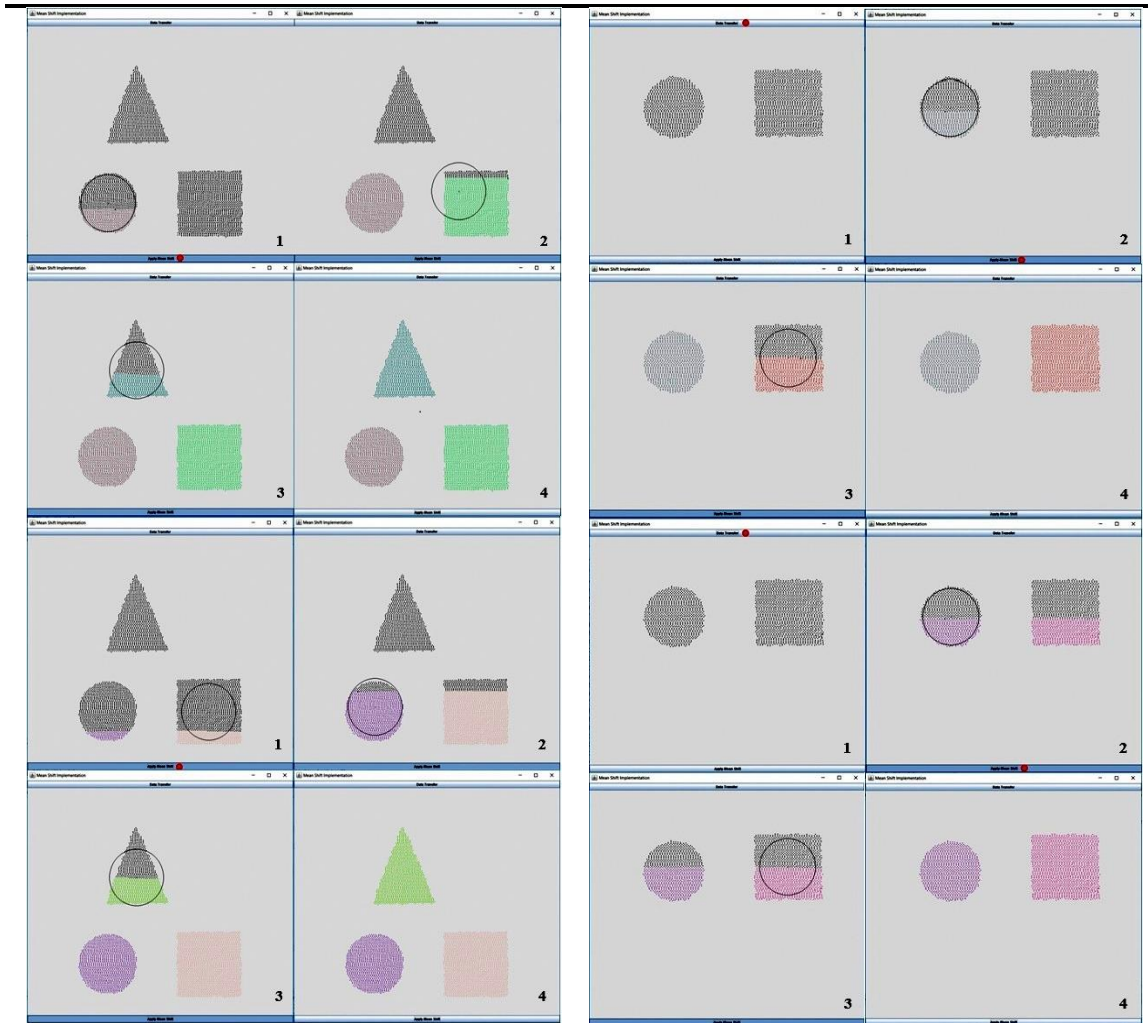


Figure 14. (Top 1-4) Metric classification process applied to planimetric data sets of the objects with mean shift classification. (Bottom 1-4) Mean shift classification of raw data not subjected to any processing

Table 3. Metric and Mean-Shift Algorithms Application Duration and Number of Classified Points

Data Sets	#of Point Cloud	# of Classified DataMetric & Mean-Shift			Spent time (millisecond)		
		Red	Green	Blue	Metric Classification	Mean-Shift Classification	
						Raw data	Proposed method
Circle, Square, Triangle	9540	2778	4324	2438	850	1112670	1110560
Circle, Square	4871	1934	2937	-	440	396630	394980
Square, Circle	4879	2938	1941	-	440	397280	395630



Mathematical results are given in table 3 obtained by the suggested method by operating alone administered on mean shift algorithm the image sequence. There are two applied data sets from real object classification algorithms metric and mean shift time and point numbers of each classification classified as shown in table 3. When the results are evaluated, coarse data directly applicable period of mean shift algorithm, applied to the data passed from the time the metric processing times mean shift algorithm seems to be more than the sum. This has been shown to be classified as a significant advantage in terms of time to implement the proposed method is also considering the same for both things and the number of data. Implementation are realized in PC which have 64-bit Windows 10 operating system, 8GB of RAM and Core i7 2.3 Ghz personal computer.

8. GENERAL CONCLUSIONS

In this study, Kinect sensor of the depth of planimetric object images obtained by the camera in a 3D point cloud classification procedure was carried out. While the classification process was used performing metric classification algorithm supported by mean shift sorting algorithm results are evaluated. Mean shift algorithm is a non-parametric method. In some cases in which the object to be classified and environment (background) color, so color probability distributions, it may be very similar. Using the color distribution information for the classification process in such cases will not give good results. To overcome this problem, the proposed method were carried out numerical processing point cloud obtained on the object. For this aim, the first step was performed a metric classification, then classification of mean shift method was applied. For a reliable classification process, on quantitative data that can be applied directly done and metric classification process is combined in the mean shift algorithm.

A metric classification algorithm used for fast and regularly execution of the objects to be classified and data were normalized. For mean shift algorithm the similarity coefficient was determined by a new similarity function. Finally, the classification process has been realized with the classical mean shift algorithm. The obtained numerical results demonstrate proposed method's superiority opposite of classical methods. The developed method has been proved that it gives better performance by applying it on real life taken image sequences. As a future work, we plan developing more accurate and robust detection of features. For better performance the number of parameters of deformable templates can be added. We can also inquiries a statistical classification of surface topology from 3d point cloud data. Also in further study, the classification process will focus on separation of data from a single object such as classification of the human body and limbs.

REFERENCES

1. Eskandari, A.R. and Kouchaki, Z., (2012). Regular Shapes Detection in Satellite Images, *Malaysian Journal of Computer Science*, vol:25, no:1, pp:56-66.
2. Lai, K. and Fox, D., (2010). Object Recognition in 3D Point Clouds Using Web Data and Domain Adaptation, *The International Journal of Robotics Research*, vol:29, no:8, pp:1019-1037.
3. Schnabel, R., Wahl, R., Klein, R., (2007). Efficient RANSAC for Point-Cloud Shape Detection, *Computer Graphics Forum*, vol:26, no: 2, pp:214-226.
4. Hao, W., Wang, Y., Ning, X., Zhao, M., Zhang, J., Shi, Z.,



-
- and Zhang, X., (2013). Automatic Building Extraction from Terrestrial Laser Scanning Data, *Advances in Electrical and Computer Engineering*, vol:13, no:3, pp:11-16.
5. Zhang, Z., (2012). Microsoft Kinect Sensor and Its Effect, *MultiMedia, IEEE*, vol:19, no:2, pp:4-12.
 6. Zhang, M., Zhang, Z., Chang, Y., Esche, S. K., Chassapis, C., (2015). Kinect-based Universal Range Sensor and its Application in Educational Laboratories, *International Journal of Online Engineering (IJOE)*, vol:11, no:2, pp:26-35.
 7. Khoshelham, K. and Elberink, S.O., (2012). Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications, *Sensors*, Vol:12, No:2, pp:1437-1454.
 8. Ozbay, E. and Cinar, A., (2013). 3D Reconstruction Technique With Kinect and Point Cloud Computing, in *Proc. 3rd World Conference on Information Technology*, vol:3, pp:1748-1754.
 9. Ozbay, E., Cinar, A., (2013). A Novel Approach to Smoothing on 3d Structured Adaptive Mesh of The Kinect-Based Models, in *Proc. 2th International Conference on Advanced Information Technologies and Applications*, pp:13-22.
 10. Atmosukarto, I. and Shapiro, L.G., (2008). A Learning Approach to 3D Object Representation for Classification, in *Proc. Structural, Syntactic, and Statistical Pattern Recognition*, Pp:267-276.
 11. Wei, X., Phung, S.L., and Bouzerdoum, A., (2014). Object segmentation and classification using 3-D range camera, *Journal of Visual Communication and Image Representation*, Vol:25, No:1, pp:74-85.
 12. Berger, K., Ruhl, K., Schroeder, Y., Bruemmer, C., Scholz, A., and Magnor, M.A., (2011). Markerless Motion Capture using multiple Color-Depth Sensors, In *VMV* pp:317-324.
 13. Zalevsky, Z., Shpunt, A., Malzels, A., and Garcia, J., (2013). Method and system for object reconstruction" Patent No 8,400,494, Washington Patent and Trademark Office.
 14. Smisek, J., Jancosek, M., Pajdla, T., (2013). 3D with Kinect, In *Consumer Depth Cameras for Computer Vision* Springer London, Pp:3-25.
 15. Rusu, R.B. and Cousins, S., (2011). 3D is here: Point Cloud Library (PCL), in *Proc. 2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1-4.
 16. Khoshelham, K., (2011). Accuracy Analysis Of Kinect Depth Data. *ISPRS workshop laser scanning*, vol:38, no:5, in *Proc. ISPRS Workshop Laser Scanning*.
 17. Tombari, F. and Stefano, L.D., (2011). 3d Data Segmentation by Local Classification and Markov Random Fields, in *Proc. International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pp:212-219.
 18. Weinberger, K.Q., Blitzer, J., and Saul, L.K., (2005). Distance Metric Learning for Large Margin Nearest Neighbor Classification. *The Journal of Machine Learning Research*, vol:10, pp:207-244.
 19. Altman, N.S., (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression, *The American Statistician*, vol:46, no:3, pp:175-185.
 20. Cheng, Y., (1995). Mean Shift, Mode Seeking, and Clustering, *Pattern Analysis and Machine Intelligence, IEEE Transactions*, Vol:17, no:8, pp:790-799.
 21. Jaskowiak, P.A. and Campello, R.J.G.B., (2011). Comparing Correlation Coefficients as Dissimilarity Measures for Cancer



-
- Classification in Gene Expression Data, in Proc. Brazilian Symposium on Bioinformatics, pp. 1-8.
22. Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., and Breitkopf, U., (2012). The ISPRS Benchmark on Urban Object Classification and 3d Building Reconstruction, *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, Vol:1, No:3, pp:293-298.
 23. Klasing, K., Wollherr, D., and Buss, M., (2008). A Clustering Method for Efficient Segmentation of 3d Laser Data, in Proc. IEEE International Conference on Robotics and Automation (ICRA), 2008, pp:4043-4048.
 24. Kim, T.K., Kee, S.C., and Kim, S.R., (2001). Real-time Normalization and Feature Extraction of 3D Face Data Using Curvature Characteristics, in Proc. 10th IEEE International Workshop on Robot and Human Interactive Communication, Pp:74-79.
 25. Hwang, J.P., Baek, J., Choi, B., and Kim, E., (2015). A Novel Part-Based Approach to Mean-Shift Algorithm for Visual Tracking, *International Journal of Control, Automation and Systems*, Vol:13, No:2, pp:443-453.
 26. Talu, M.F., Türkoğlu, İ., and Cebeci, M., (2010). Real-time Kernel Based Object Tracking Using Mean Shift, in Proc. 18th IEEE Signal Processing and Communications Applications (SIU), Pp:328-331.
 27. Yao, W., Hinz, S., and Stilla, U., (2009). Object Extraction Based on 3D-segmentation of Lidar Data by Combining Mean Shift with Normalized Cuts: Two Examples from Urban Areas, in Proc. IEEE Urban Remote Sensing Event, pp:1-6.