# Improved Interval Type-2 Fuzzy Subtractive Clustering for Obstacle Detection of Robot Vision from Stream of Depth Camera

Mau Uyen Nguyen, Long Thanh Ngo, Thanh Tinh Dao
*Department of Information Systems, Faculty of Information Technology*
*Le Quy Don Technical University, Hanoi, Vietnam*
*Email: nguyenitt2005@gmail.com, ngotlong@gmail.com, tinhdt@mta.edu.vn*

***Abstract*—Obstacle detection is a fundamental issue of robot navigation and there have been several proposed methods for this problem. In this paper, we propose a new approach to find out obstacles on Depth Camera streams. The proposed approach consists of three stages. First, preprocessing stage is for noise removal. Second, different depths in a frame are clustered based on the Interval Type-2 Fuzzy Subtractive Clustering algorithm. Third, the objects of interest are detected from the obtained clusters. Beside that, it gives an improvement in the Interval Type-2 Fuzzy Subtractive Clustering algorithm to reduce the time consuming. In theory, it is at least 3700 times better than the original one, and approximate 980100 in practice on our depth frames. The results conducted on frames demonstrate that the distance from the camera to objects retrieved is exact enough for indoor robot navigation problems.**

***Keywords*-Type-2 Fuzzy Sets, Subtractive Clustering, Obstacle Detection, Robot Navigation, Depth Camera.**

## I. Introduction

Obstacle detecting is the process to identify the objects in the viewable space of robots. An object contains information about the distance and the size of it to the robot - camera. Two important stages are mapping the depth space and identify the objects appeared. The map of depth is estimated by the multi-camera model as [11], [12]. Recently, Kinect cameras with RGB-D frames make an improvement in estimating depth. The paper [13] shows that, the point cloud of the properly calibrated Kinect sensor does not contain large systematic errors when compared with a laser scanning data, so it can be used in object detecting problems. Some researchers proposed methods to identify objects on RGB-D [13], [14]. In stereo camera models, objects are detected based on some techniques: blocking match, gradient method, feature matching, dynamic programming, intrinsic curves, graph cuts, nonlinear diffusion, belief propagation, correspondenceless methods mentioned in [11]. In the [14], the 2D chamfer distance matching algorithm is involved to locate the head of bodies, then the depth information is used to check the 3D model compared with the real head model. In stereo images, objects detecting is based on finding out features and matching them, so that in complex environment, they take long time and less exactly.

Subtractive clustering [1], [2] is an extension of the Mountain clustering methods by improving the Mountain function to calculate potential of becoming a cluster centroid for each data point based on the location of the data point with respect to the other data points. The Subtractive clustering algorithm only consider data points, not a grid points, which reduces the computational complexities and gives better distribution of cluster centroids. Kim et al. [5] has improved subtractive clustering algorithm by proposing a kernel-induced distance instead of the conventional distance when calculating the mountain value of data point. Meanwhile, type-2 fuzzy sets allow us to obtain desirable results in designing and managing uncertainty. Therefore, type-2 fuzzy sets have been studied and widely applied in many fields [7], [8], especially pattern recognitions. In [6], authors also proposed an extension of Subtractive clustering using interval type-2 fuzzy sets, called interval type-2 fuzzy subtractive clustering, that be able to manage uncertainty better.

In this paper we propose steps to identify obstacles based on Kinect depth frames: preprocessing depth frames, clustering distances to clusters, locating objects in each frame, combining separated objects, and measuring distances, size of objects. On preprocessing stage, the median filter with a window of size 5 x 5 is used to remove noise in depth frames. Note that, in the proposed approach, the background is captured in the depth frames so that it must be removed before processing the main tasks, approximating background algorithm is involved to remove it. Clustering algorithm is utilized to isolate points from an object to other objects' points. However, because the surface of some objects spreads in some clusters, so the combining task is involved to solve this issue. Because the number of clusters is not defined for each frame, Mountain clustering and Subtractive clustering are adopted to identify the number of clusters. Mountain clustering calculates a mountain function (density function) at every possible position in the data space [10], subtractive clustering is similar to mountain clustering, except that instead of calculating the density function at every possible position in the data space, it uses the positions of the data points to calculate the density function [3]. Based on mountain clustering and subtractive clustering and effective

978-1-4673-5119-5/12/$31.00 

of fuzzy theory, the Improved Interval Type-2 Fuzzy Subtractive Clustering is proposed to reduce the uncertainty of data and the processing time of obtaining clusters.

## II. PRELIMINARIES

### A. Type-2 Fuzzy Sets

A type-2 fuzzy set in $X$ is denoted $\tilde{A}$, and its membership grade of $x \in X$ is $\mu_{\tilde{A}}(x, u), u \in J_x \subseteq [0, 1]$, which is a type-1 fuzzy set in [0, 1]. The elements of domain of $\mu_{\tilde{A}}(x, u)$ are called primary memberships of $x$ in $\tilde{A}$ and memberships of primary memberships in $\mu_{\tilde{A}}(x, u)$ are called secondary memberships of $x$ in $\tilde{A}$.

*Definition 2.1:* A $type-2\ fuzzy\ set$, denoted $\tilde{A}$, is characterized by a type-2 membership function $\mu_{\tilde{A}}(x, u)$ where $x \in X$ and $u \in J_x \subseteq [0, 1]$, i.e.,

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \quad (1)$$

or

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u))/(x, u), J_x \subseteq [0, 1] \quad (2)$$

in which $0 \le \mu_{\tilde{A}}(x, u) \le 1$.

At each value of $x$, say $x = x'$, the 2-D plane whose axes are $u$ and $\mu_{\tilde{A}}(x', u)$ is called a $vertical\ slice$ of $\mu_{\tilde{A}}(x, u)$. A *secondary membership function* is a vertical slice of $\mu_{\tilde{A}}(x, u)$. It is $\mu_{\tilde{A}}(x = x', u)$ for $x \in X$ and $\forall u \in J_{x'} \subseteq [0, 1]$, i.e.

$$\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/u, J_{x'} \subseteq [0, 1] \quad (3)$$

in which $0 \le f_{x'}(u) \le 1$.

Type-2 fuzzy sets are called an interval type-2 fuzzy sets if the secondary membership function $f_{x'}(u) = 1\ \forall u \in J_x$ i.e. a type-2 fuzzy set are defined as follows:

*Definition 2.2:* An *interval type-2 fuzzy set* $\tilde{A}$ is characterized by an interval type-2 membership function $\mu_{\tilde{A}}(x, u) = 1$ where $x \in X$ and $u \in J_x \subseteq [0, 1]$, i.e.,

$$\tilde{A} = \{((x, u), 1) | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \quad (4)$$

Uncertainty of $\tilde{A}$, denoted FOU, is union of primary functions i.e. $FOU(\tilde{A}) = \bigcup_{x \in X} J_x$. Upper/lower bounds of membership function (UMF/LMF), denoted $\overline{\mu}_{\tilde{A}}(x)$ and $\underline{\mu}_{\tilde{A}}(x)$, of $\tilde{A}$ are two type-1 membership function and bounds of FOU.

### B. Subtractive Clustering Algorithm

There are some clustering techniques proposed as C-mean clustering, K-mean clustering, Mountain clustering, Subtractive clustering... C-mean, K-mean are based on the idea that number of clusters were identified, so they are not suitable for robot navigation problems - number of clusters changes based on frames and the environment. The Subtractive clustering is based on the modified of Mountain clustering so that, it shows the advanced in comparison with Mountain clustering [1]. Because of uncertainty of depth information in depth frames received from Kinect cameras, type-2 fuzzy sets are employed in the Subtractive clustering algorithm. The [16], [17] show the efficience of type-2 fuzzy sets in combining with algorithms.

In the expansion Subtractive clustering algorithm, the membership degree of a point in $k^{th}$ cluster centroid is defined as following formula:

$$\mu_{ik} = e^{-\frac{4}{r_a^2}(x_i - x_k)^{\frac{2}{m-1}}} \quad (5)$$

where the $x_k$ is the $k^{th}$ cluster centroid. According to the formula (5), membership value of a data point in $k^{th}$ cluster centroid depends on the position of $k^{th}$ cluster and the fuzziness parameter $m$. On the other hand, the position of the $k^{th}$ cluster also depends on the fuzziness parameter $m$. Thus, the fuzziness parameter $m$ is the most uncertainty element in the expanded Subtractive clustering algorithm. Therefore, to design and manage the uncertainty for fuzziness parameter $m$, we extend a pattern set to interval type-2 fuzzy sets using two fuzzifiers $m_1$ and $m_2$ which creates a footprint of uncertainty (FOU) for the fuzziness parameter $m$. Then the degree of membership of $k^{th}$ cluster centroid is defined as following formula:

$$\begin{cases} \overline{\mu}_{ik} = e^{-\frac{4}{r_a^2}(x_i - x_k)^{\frac{2}{m_1-1}}} \\ \underline{\mu}_{ik} = e^{-\frac{4}{r_a^2}(x_i - x_k)^{\frac{2}{m_2-1}}} \end{cases} \quad (6)$$

We have two density functions to calculate potential of each data point as follows:

$$\begin{cases} \overline{P}_i = \sum_{j=1}^{n} e^{-\frac{4}{r_a^2}(x_j - x_i)^{\frac{2}{m_1-1}}} \\ \underline{P}_i = \sum_{j=1}^{n} e^{-\frac{4}{r_a^2}(x_j - x_i)^{\frac{2}{m_2-1}}} \end{cases} \quad (7)$$

If the centroids are identified by the formula (7), we will have centroids $v_L$ and $v_R$. Thus, we will do type-reduction for centroids as bellows:

$$P_i = \frac{\overline{P}_i * m_1 + \underline{P}_i * m_2}{m_1 + m_2} \quad (8)$$

And when we identified $k^{th}$ cluster center, the density of all data points is revised by using following formula:

$$\begin{cases} \underline{P}_i^{sub} = P_k^* \sum_{j=1}^{n} e^{-\frac{4}{r_b^2} d_{\mathrm{ij}}^{\frac{2}{m_1-1}}} \\ \overline{P}_i^{sub} = P_k^* \sum_{j=1}^{n} e^{-\frac{4}{r_b^2} d_{\mathrm{ij}}^{\frac{2}{m_2-1}}} \\ P_i^{sub} = \frac{\underline{P}_i^{sub} * m_1 + \overline{P}_i^{sub} * m_2}{m_1 + m_2} \\ P_i = P_i - P_i^{sub} \end{cases} \quad (9)$$

**Algorithm 1** Interval type-2 fuzzy subtractive clustering

**Step 1**: Initialization, $r_a$, $\eta$ with $\eta = \frac{r_b}{r_a}$, $\overline{\varepsilon}$ and $\underline{\varepsilon}$, $m_1$ and $m_2$ $(1 < m_1 < m_2)$

**Step 2**: Calculating density for all data points with two fuzzifiers $m_1$ and $m_2$ by using formulas (7) and (8). Data point with the highest density is selected as the first cluster centroid: $P_k^* = \max_{i=1}^{n} P_i$ where $k = 1$ and $P_k^*$ is the density of the first cluster centroid.

**Step 3**: The density of all data points is revised by using formula (9).

**Step 4**: Identification of the next cluster centroids are as similar as SC.

**Step 5**: Output the results of clustering.

## III. Obstacle Detection from Stream of Depth Camera

Based on the stream of depth frames, the problem of obstacle detection is addressed in this paper. Each pixel on frames is the value of distance from a point of the obstacle to the camera. The mechanism for detecting obstacles consists of three main steps: preprocessing depth frames - removing noise and removing background, clustering distances to clusters - subtractive clustering, identifying objects - locating objects in each frame, combining separated objects, and measuring distances of objects. We show the detail of techniques in removing background and subtractive clustering bellow.

### A. Background removal

In the real distance frames - $I_d$, the background is captured, calculated, and treated as objects. It is better to remove the background before clustering. Assuming that $B$ is the background, it is a flat space with some low edges, which robot may get over. $I_y$ is the Y coordinate frame corresponding to $I_d$, which is provided by the Kinect SDK function based on $I_d$. Assuming that, the lowest pixels in the distance image belong to the background. On each column of the distance image, the $L_i$ is an approximate line to emulate the trend of background based on $I_y$. Each pixel in the column $i$ - $p_{ji}$, it belongs to background if its distance to the $L_i$ - $d(p_{ji}, L_i)$, is less than or equal $\sigma_h$ predefined.

$$B = \{p_{ji}, d(p_{ji}, L_i) \leq \sigma_h, \forall p_{ji} \in I_y\} \tag{10}$$

### B. Improved Interval Type-2 Fuzzy Subtractive Clustering

In the real problems, the *algorithm 1* takes a long time for $480 \times 640$ frames, so it must be optimized to reduce time consuming. Assuming that, $N$ is the number of points in a frame, for example, $N$ is 307200 in the $480 \times 640$ frames. The (7) and (8) show that, the time for calculating $P_i$ is $\bigcirc(N^2)$.

Because processing time depends on $N$, thus it must be found out the way to reduce the number of elements involved. The first idea is decreasing the size of frames, consequently the number of pixels will be less than 307200. In this case, some information in frames may be lost. We try other approaches, that use the original size of frame to maintain the information contained. Pixels in a frame present the distance between the camera and the target, so some pixels, that have the same value, are calculated again in the formula (7). It is a waste of time for recalculating for each pixel in a group having the same value. The Fig. (1)

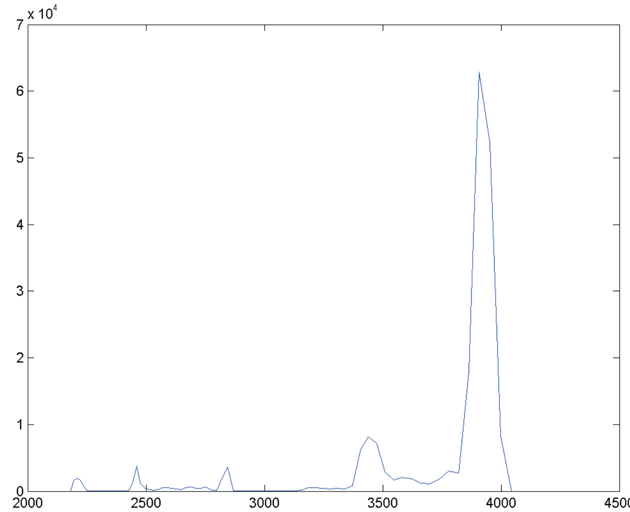



Figure 1: Histogram of frame

shows that, a large number of pixels takes the same value. Assuming that, $N_g$ is the number of values appeared in a frame. Instead of involving the formula (7) and (8) over the number of points $N$, it can be calculated on the number of values $N_g$. It reduces the size of input from $N$ to $N_g$. But in this case, the algorithm does not show the concentration on the number of items in each group. Considering two pixels $p_i$ and $p_j$, which take the same value, the formula (6) gives $\overline{\mu}_{ik} = \overline{\mu}_{jk}$ and $\underline{\mu}_{ik} = \underline{\mu}_{jk}$, so in the formula (7), instead of calculating $p_i$ and $p_j$, it can be replaced by $2 * \overline{\mu}_{ik}$ and $2 * \underline{\mu}_{ik}$. Extending this idea, $x_i$ is the value of each group and the $w_i$ is defined as the number of pixels in the group $i$. And another case is $p_i$ and $p_k$ equal in value, consequently, $\underline{\mu}_{ik} = \overline{\mu}_{ik} = 1$. The formula (7) is rewritten as follows:

$$\begin{cases} \overline{P}_i = \sum_{j=1}^{n} w_j e^{-\frac{4}{r_a^2}(x_j - x_i)^{\frac{2}{m_1 - 1}}} + w_i \\ \underline{P}_i = \sum_{j=1}^{n} w_j e^{-\frac{4}{r_a^2}(x_j - x_i)^{\frac{2}{m_2 - 1}}} + w_i \end{cases} \tag{11}$$

In the formula (7) number of $P_i$ is $N$, and in the formula (11) number of $P_i$ is $N_g$, in theory $N \geq N_g$. In the Fig. (2a): $N = 640 \times 480 = 307200$, the real statistic on Fig. (2a)

(a) Original depth frame

(b) Real frame

Figure 2: Depth frame and color frame

shows that $N_g = 308$, thus $\frac{N}{N_g} \approx 990$. Consequently, the use of the formula (11) is $990 \times 990 = 980100$ times faster than the use of the formula (7). It is not certain that, all frames will have the same number of groups as mentioned, but in depth frames, the range is 0-5 m, the maximum number of groups is 5000. The new approach reduces at least 3700 times of time processing. In the indoor robot navigation problems, the precision needed is only the Centimetre, so theoretical the number of groups is reduced by 10 times, the formula (11) gains 100 times faster. So that, in theory, this approach helps the formula (11) 370000 times faster. In the real Fig. (2a), the number of groups is reduced to 150. In this case, Mountain clustering can be considered because it only calculates over data space so that, it is better than Subtractive clustering. The Improved Subtractive clustering only calculates on number of data group, so the number of items calculated usually is less the size of data space, in the worst case their number of processing points are equal. In theory and in real scenarios, the Improved Subtractive clustering shows the advance in time consuming to Subtractive clustering and Mountain clustering.

### C. *Obstacle Detection for Robot Vision*

The distance frames inform the distance from points of objects to the robot, but they do not give information about objects, thus the robot is difficult to navigate. It must identify the objects to decide the trend to move. Depth frames are filtered by the median filter with a window of size 5 x 5. Background removal step is involved base on the equation 10. Using the *algorithm 1* with new approach mentioned above, frames' pixels are separated into clusters. On each cluster, the improved flood fill algorithm is used to find out regions. Avoiding regions, which is small in amount considering as noise, thus objects are detected. The combining objects stage is called to rebuild objects. Basing on coordinate of pixels in the object, the real distance from the camera to the object is recalculated by looking back to the original values from the distance frame. To detect the objects, the *algorithm 2* is proposed as following:

Base on the objects' information given by *algorithm 2*, the robot decides the path to travel.

**Algorithm 2** Obstacle detection

**Input**: Depth frame from the Kinect Camera $I_d$

**Output**: Objects in frame with real depths

**Step 1**: Implement 5x5-window median filter on $I_d$

**Step 2**: Calculate $I_y$ from $I_d$, remove background based on 10

**Step 3**: Call improved *Algorithm 1* on the depth frame

**Step 4**: If there are not clusters, which have not been calculated, go to *step 13*.

**Step 5**: Choose a cluster, mark as calculated.

**Step 6**: If there are not points, which are not processed, in the frame, go to *step 5*.

**Step 7**: Choose a seed point from the frame.

**Step 8**: Flood fill points that are the same region with the seed point, sum the real distance of points, count number of points in this region.

**Step 9**: If number of the region's point is greater than $C_{min}$, recognize it. Go to *step 6*.

**Step 10**: While there are objects, that have the points near by each others.

**Step 11**: If the distance from two points are less then $\delta_s$ mm then join.

**Step 12**: Loop to *step 10*.

**Step 13**: Out put the objects.

## IV. Experimental results

### A. *Experimental conditions*

To experiment the approach of detecting obstacles, we set some parameters for *algorithm 1*: $r_a = 0.04$, $\eta = 1.5$, $\bar{\varepsilon} = 0.5$, $\underline{\varepsilon} = 0.01$, $m_1 = 1.5$, $m_2 = 2.6$.
Parameters for *algorithm 2*: $C_{min} = 500$, $\delta_s = 25$, $\sigma_h = 20$
The *algorithm 2* is involved on 15 frames with different distance to the camera to show the moving of the robot.

### B. *Experimental output*

Involving *algorithm 2* on the first frame Fig. (3), including in that situation color frame Fig. (3a), depth frame Fig. (3b), histogram of depth Fig. (3c).

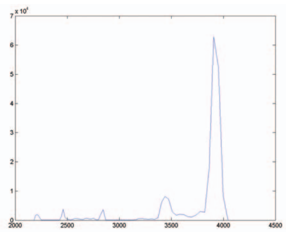
The frame is divided into 4 clusters as shown in Fig. (4).

Base on those clusters, improved flood fill algorithm is performed and objects are detected. There are 166 objects detected. After combining and filter, 20 obstacles are figured out in that frame. Combining all objects in a frame base on the depth of its, the obtained result is shown in Fig. (5).

We tried for 30 distances and moving robot with 30 frames from the 4m far from the wall with 25mm-steps, all obstacles

(a) Color frame

(b) Depth frame



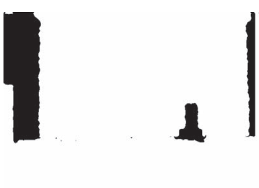
(c) Depth histogram

(d) Background removal frame

Figure 3: Original data and preprocessing data

(a) Cluster 2460

(b) Cluster 2845



(c) Cluster 3440

(d) Cluster 3907

Figure 4: Clusters in the frame

Figure 5: 9 objects in the frame

(a) Object 1

(b) Object 2

(c) Object 3

(d) Object 4

Figure 6: Obstacles concerned

| Objects | max dev (mm) | $\sigma$ (mm) | max dev % | $\sigma$ % |
|---|---|---|---|---|
| Object 1 | 124 | 61 | 6.5 | 2.9 |
| Object 2 | 73 | 35.1 | 2.2 | 1.1 |
| Object 3 | 74 | 32.5 | 2.4 | 1.2 |
| Object 4 | 90 | 40.8 | 3.7 | 1.9 |

Table I: Statistics of experience: *max dev* is the maximum deviation, $\sigma$ is the standard deviation,*max dev %* is the percentage of max deviation over the real depth, $\sigma$ % is the standard deviation percentage.

are detected. Now we consider to 4 objects in frames as Fig. (6).

The data statistic shows in Fig. (7) and table (I). The Fig. 7 plots the comparison of real distances and measured distances from 4 obstacles and robot-camera at difference positions.

The result over 4 objects shows that, the maximum error is 124 and the standard deviation is 61, they are 6.5% and 2.9% respectively. However the object 1 is a cylinder, thus it makes uncertain in depth. Considering 3 others objects, the maximum error is only 90 and the standard deviation is 40 the are 3.7% and 1.9% respectively. And the view points are different, that makes error in estimating real distance. The error percentage and error distance are acceptable for the indoor robot navigation problem, for example in corridors, room space.

## V. Conclusion

To solve the indoor robot navigation problem, the paper proposes an approach to identify the objects based on depth frames of Kinect camera. Steps to locate obstacles are presented by the *algorithm 2*. The result of the experience shows that, the approach is suitable for the indoor robot navigation problem. Besides that, the paper proposes an improved in the subtractive clustering algorithm to reduce the time processing.

Our next goals are to model 3-dimensional environment from detected obstacles for robot navigation systems; deploy the approach on FPGA hardware.

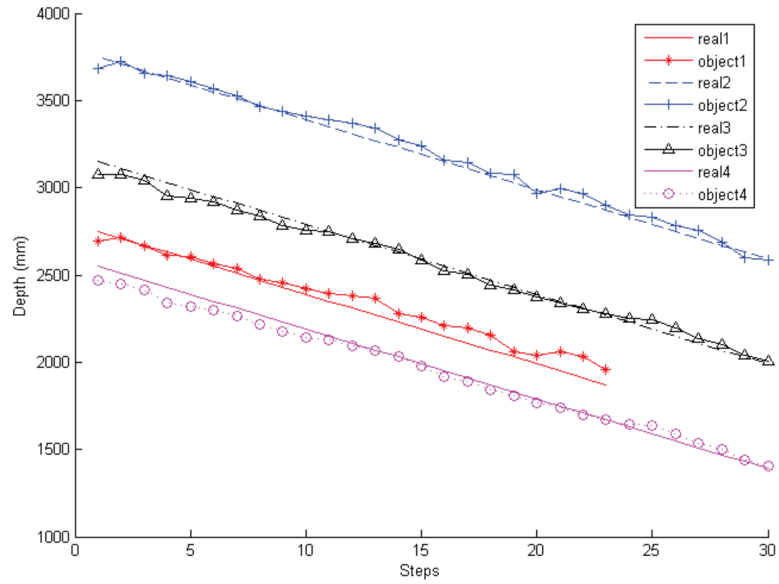



Figure 7: Comparison between real distances and measured distances.

## Acknowledgment

This work was funded by the Research Fund RFIT@LQDTU, Faculty of Information Technology, Le Quy Don Technical University.

## References

[1] S. L. Chiu, *Fuzzy Model Identification Based on Cluster Estimation*,Journal on Intelligent Fuzzy Systems, vol. 2, pp. 267-278, 1994.

[2] S. L. Chiu, *Extracting Fuzzy Rules from Data for Function Approximation and Pattern Classification*, Fuzzy Information Engineering: a Guide Tour of Applications, pp. 149-162, 1997.

[3] Demirli K., S. X. Cheng, P. Muthukumaran, *Subtractive clustering based modeling of job sequencing with parametric search*, Fuzzy Sets and Systems, Vol. 137, 235-270,2003.

[4] JunYing Chen, Zheng Qin and Ji Jia, *A Weighted Mean Subtractive Clustering Algorithm*, Information Technology Journal, Vol. 7 (2): 356-360, 2008.

[5] Kim, DW., K.Y. Lee and K.H. Lee, *A Kernel Based Subtractive Clustering Method*,Pattern Recog. Lett., Vol. 26(7):879-891, 2005.

[6] Long Thanh Ngo, Binh Huy Pham, *Approach to Image Segmentation Based on Interval Type-2 Fuzzy Subtractive Clustering*, Springer LNAI, Vol. 7197, 1-10, 2012.

[7] J. Mendel and R. John, *Type-2 fuzzy set made simple*, IEEE Trans. On Fuzzy Systems, vol. 10(2), pp. 117127, 2002.

[8] N. Karnik and J.M. Mendel, *Operations on Type-2 Fuzzy Sets, Fuzzy Sets and Systems*, Vol 122, pp.327-348, 2001.

[9] C. Hwang and F. C. Rhee, *Uncertain Fuzzy Clustering: Interval Type-2 Fuzzy Approach to C-Means*, IEEE Trans. on Fuzzy Systems, Vol. 15, pp 107-120, 2007.

[10] H.Y. Shen, X.Q. Peng, J.N. Wang, Z.K. Hu, *A Mountain Clustering Based on Improved PSO Algorithm[A]*. First Int' Conf' of Advances in Natural Computation, 477-481, 2005.

[11] Myron Z. Brown, Darius Burschka, Gregory D. Hager, *Advances in Computational Stereo*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 25(8):993-1008, 2003.

[12] Xiaofeng Ding, Lizhong Xu, Huibin Wang, Xin Wang, Guofang Lv, *Stereo depth estimation under different camera calibration and alignment errors*, Applied Optics, vol. 50(10):1289–1301, 2011.

[13] Khoshelham, K. (2011), *Accuracy analysis of kinect depth data*, In: ISPRS workshop Laser Scanning 2011, pp.29-31,2011.

[14] Lu Xia, Chia-Chih Chen, J. K. Aggarwal, *Human Detection Using Depth Information by Kinect*, Computer Vision and Pattern Recognition Workshops (CVPRW), pp.15-22, 2011.

[15] Chih-Wen Wu, Jin-Lian Chen, Jung-Hua Wang, *Self-organizing mountain method for clustering*, IEEE SMC 2001.

[16] Melin, P., Mendoza, O., Castillo, O., *Face Recognition With an Improved Interval Type-2 Fuzzy Logic Sugeno Integral and Modular Neural Networks*, IEEE Trans. on Systems, Man, and Cybernetics, Vol.1(5), 1001-1012, 2011.

[17] Melin, P., Mendoza, O., Castillo O., *An improved method for edge detection based on interval type-2 fuzzy logic*, Expert Systems with Applications, Vol.37(12), pp.85278535, 2010.