Octree Segmentation Based Calling Gesture Recognition for Elderly Care Robot

Xinshuang Zhao
Intelligent Systems Research Institute
Sungkyunkwan University
Suwon, Rep. of Korea
+8231-299-6465
ztchit@hotmail.com

Ahmed M. Naguib
Intelligent Systems Research Institute
Sungkyunkwan University
Suwon, Rep. of Korea
+8231-299-6471
ahmed.m.naguib@gmail.com

Sukhan Lee[1,2,*]
[1]Intelligent Systems Research Institute
[2]Department of Interaction Science
Sungkyunkwan University
Suwon, Rep. of Korea
+8231-299-6470
lsh@ece.skku.ac.kr

ABSTRACT

This paper presents a method of calling gesture recognition by isolating the head and hand of a caller based on octree segmentation. The recognition of calling gestures is designed here mainly for elderly to call a service robot for their service request. A big challenge to solve is how to make the calling gesture recognition work in a complex environment with crowded people, cluttered and randomly moving objects, as well as illumination variations. The approach taken here is to segment out individual people from the 3D point cloud acquired by Microsoft Kinect or ASUS Xtion Pro and detect their heads and hands in certain geometric configurations. The segmentation is done fast by representing the 3D point cloud in octree cells and clustering those octree cells connected by the neighborhood relationship. The head and hand in a certain geometric configuration are identified from the candidate regions defined with a segmented object and by detecting the shape and color evidences. Color model in HSV color space also discussed to well define the skin color model. The proposed method has been implemented and tested on "HomeMate," a service robot developed for elderly care. The result of performance evaluation is given.

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering-Algorithms; I.4.8 **[Image Processing and Computer Vision]**: Scene Analysis-Color, Object recognition.

General Terms

Algorithms, Performance, Reliability, Experimentation

Keywords

Octree Segmentation, Human Robot Interaction, Elderly Care Robot, Gesture Recognition.

IMCOM (ICUIMC)'14, January 9–11, 2014, Siem Reap, Cambodia.
Copyright 2014 ACM 978-1-4503-2644-5 …$15.00.

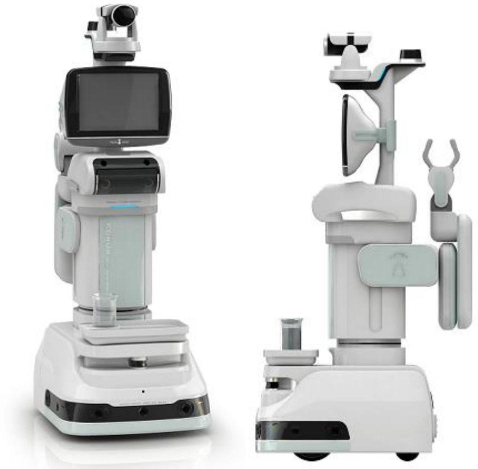


Figure 1. "HomeMate" a Service Robot developed for Elderly Care.

1. INTRODUCTION

With the continuous development of society, especially in some developed countries and part of developing countries, growing aging population increased healthcare costs. More and more consumer robots are designed for elderly with services as errand, medicine delivery, video chatting, etc. These robots usually work in home, hospital, or health care center environments that have highly cluttered background; such as moving people, noisy sound, etc. Our research mainly focuses on locating robot caller out of a cluttered environment.

Most current user interfaces for HUMAN ROBOT INTERACTION (HRI), revolve around hand-controllers such as IR remote control, gamepad, computer keyboard and mouse, or even a sensor gloves [1] [2]. With the popularity of smart phones, using smart phone to control robot is becoming more and more popular. Furthermore, voice commands and vision based human robot interaction interface, such as face detection, hand gesture, or body motion recognition, also are being developed and applied. These HRI interfaces can be experience as a more natural way compared to hand held devices [3]. Recently, sensors such as Microsoft Kinect and ASUS Xtion pro are being used in Somatic Games. These sensors provide reasonable capabilities for facial expression, gesture recognition, and voice recognition. They provide low cost/high quality way for HRI [4].

A number of previous research works started focusing on hand gesture interface in Human Robot Interaction [5] [6] [7] [8] and [9]. With the developing of structure light sensor such as Kinect and Xtion Pro, hand gesture with depth information can now produce satisfying results [10] [11], and [12]. In hand gesture recognition, a key point is hand region segmentation. Ajallooeion [13] presented a method based on saliency map to rapidly find hand region. Cheng Bor-Jeng, et al [14] presented a face related method for hand finding. Viola and Jones' Haar feature detector detects face directly, and then compute the skin color similarity between face and hand.

Depth information cannot be ignored for hand segmentation. When a human is doing a gesture, normally the hand, doing the gesture action, is located in front of human body; simple use of a depth threshold can isolate the hand. Many researchers have presented this method for hand segmentation [15] [16], and [17]. Skin color based hand segmentation is another common method. Matthew Tang [18] proposed another way which combined skin color and depth information for hand gesture recognition, and obtained good results.

The above methods were focusing on one agent performing the gesture in an empty workspace, however, in our indoor demand, a cluttered and noisy environment is expected and we are proposing an approach to resolve it. By taking advantage of our fast octree representation and segmentation capabilities, we can rapidly segment spatially displaced clouds in workspace out of acquired 3D point clouds generated by RGBD sensor. These segments might include desks, chairs, people or other possible things. Geometric constrains used to define the candidate region of heads and hands in 3D space, along with skin color detector, can locate heads and hands of candidate agents. A hand motion tracking algorithm investigates the gesture of each agent looking for a predefined specific motion gesture. Once it's identified, this agent becomes a caller. Caller's position is, then, given to robot directly for robot motion action.

The rest of this paper is organized as following. In section 2, algorithm over view and the specific gesture design for elderly people is introduced. Section 3 describes octree segmentation for object extraction from raw 3D cloud points. The geometrical constrains is explained in section 4. The head and hand detection in each human like segment is presented in section 5. In section 6, experiments and conclusions show in this section.

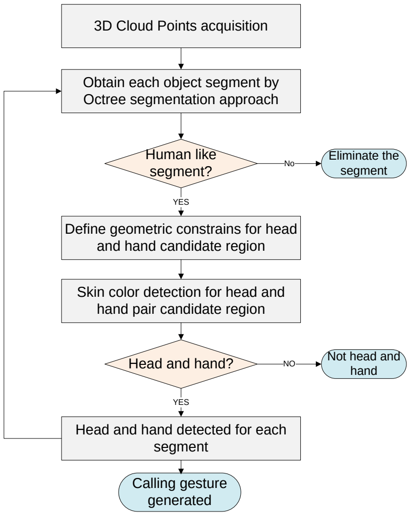

3D Cloud Points acquisition
Obtain each object segment by Octree segmentation approach
Human like segment?
No
Eliminate the segment
YES
Define geometric constrains for head and hand candidate region
Skin color detection for head and hand pair candidate region
Head and hand?
NO
Not head and hand
YES
Head and hand detected for each segment
Calling gesture generated

Figure 2. Overall process flowchart.

2. ALGORITHM OVER VIEW

Our approach can be simply divided into data acquisition, octree segmentation, skin color detection for finding caller's head and hand, calling gesture recognition parts (Fig. 2). Gestures can be defined as dynamic gestures, static gestures or combined dynamic or static elements gestures. In this paper, the calling gesture is mainly designed for elder people when calling for elderly care robot. Thus, this gesture should be simple, easy, and natural; hence, static gesture is more suitable. Gesture of lifting hand with palm facing forward is defined as calling gesture (Fig. 3). Further dynamic gestures, such as hand shaking, are introduced to distinguish callers from other agents lifting their hands unintentionally. According to our method, some necessary explanations for elder people to use this kind of calling gesture are as following: 1) Make sure the hand which one is calling the robot located in a nearest position refer to camera plane (less than 3.5 meters). 2) Only use one hand for calling gesture, just simply raise the hand. 3) The gesture hand has to be lower than head height.

3. OCTREE SEGMENTATION

From the sensor such as Microsoft Kinect and ASUS Xtion Pro, 3D cloud points have been generated. These 3D cloud points' clusters represent the objects and background in FOV of the sensor. With these information supplied by 3D cloud points, objects are difficult to extract from cluttered background. Many previous works have presented many algorithms for 3D object segmentation such as K-means clustering [19], Expectation-Maximization (EM) [20] algorithm, mean shift clustering and etc. most of these approaches either provides good results with high computational costs, or introduced a performance trade-off for real-time performance.

Michael Potmesil presented a way to generate 3D solid objects from octree models [21] by constructing octree model from range image using depth information. As a nonlinear data structure, Octree produce a tree like hierarchy structure with each internal node containing 8 branches [22], according to 3D object segmentation with 3D cloud points, octree is improved to represent multi-scale and 3 dimensions models, it partitions a 3 dimension space by recursively subdividing it into eight octants which shown on Figure 4. Jaewoong Kim presented a fast way to find neighbor cells for multiple octree representation [23], through this approach we can give key number to each segment, it was sufficient for identifying different objects.

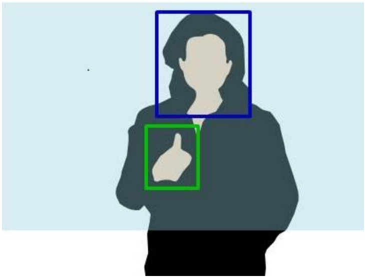


Figure 3. Calling gesture design.

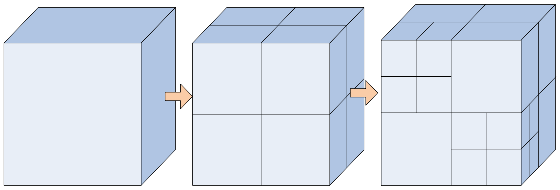


Figure 4. Octree construction for 3D model.

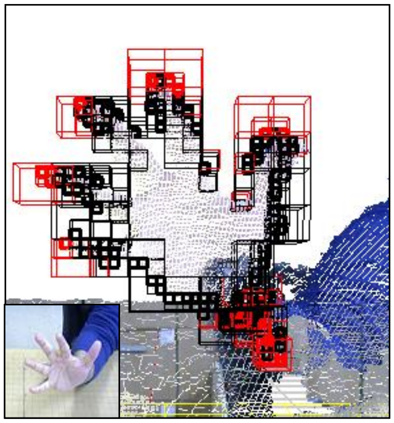


Figure 5. Octree-based extreme cell for finger tips detection.

As we are expecting a highly cluttered environment, with a wide workspace volume, we have chosen to separate 3D objects using our rapid octree based objects segmentation. This will not only robustly segment the workspace into potential segments, but also will provide a structured addressing system that will allow us to instantaneously access neighbor cells to inspect their geometrical formation. This inspection can help us analyze and extract valuable geometrical shape descriptors for very low computational costs, such as corners shown in figure 5.

3.1 Object Segmentation through Octree Representation

In this method only distance between cells are used for object segmentation. After fast octree cells segmentation, in each cell, there are amount of 3D cloud points, with their associated color and location information. Cells that include few cloud points, for example, 3 points or less, can be considered as gap or blank cells (Fig. 6). By contrast, if an octree cell has enough point cloud density, this cell is considered as effective cell.

For optimum selection of octree cell size, as well as the required density of a point cloud to allow a cell to exist, we have developed an adaptive algorithm that determines cell size according to certain constraints and choose density limit based on computed cell size and distance of each cell.

Octree cell size is bounded by 4 constraints:

C1.Memory capacity that limits the amount of cells we can generate at an instance

C2. Cell size should be less than expected person-person distance in workspace environment (people density)

C3. Cell size should be larger than point-point distance of 3D sensor at maximum desired depth

C4. Cell size should be larger than expected 3D sensor noise/uncertainty at maximum desired depth

Since windows operating system allows a process to write no more than 4GB of page file, we can only store 221 cells at a time. This means that cell size cannot be less than workspace volume / 7^3.

$$C1 = \frac{V}{343}$$

According to environment setting, we should guarantee that each 2 agents are segmented successfully. This can be guarantee only by enforcing a minimum length expected to exist in target environment. Using this pre-knowledge, we can compute the second constraint as follows:

$$C2 = \frac{minimum\_lengthResolution}{2.nDistance.Tan\left(\frac{FOV}{2}\right)}$$

Where “Distance” is the maximum distance between any point in workspace volume and the sensor depth CCD, “Resolution” is sensor depth CCD resolution, and “n” is the minimum 3D point cloud density in a cell to exist (this should be computed both horizontally and vertically)

To guarantee that a cell would at least have a point inside of it, 3rd constraint requires us to guarantee that cell size is always larger than point-point distance anywhere in workspace.

$$C3 = \frac{2.nDistance.Tan\left(\frac{FOV}{2}\right)}{Resolution}$$

And finally, constraint 4 demands for cell size to be larger than 2 times the uncertainty of depth measurement at weakest measurement location in workspace. In order to formulate this constraint, we should model the error bound of geometric measurement of 3D sensor at a specific depth (as shown in figure 7):

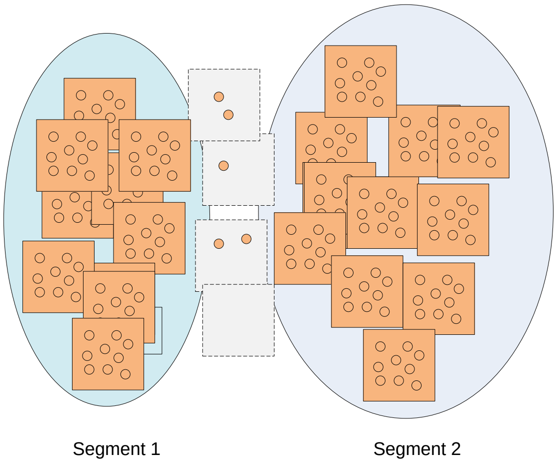


Figure 6. Cell cluster approach. Note that these cells are separated into two groups by compute the grayscale value of 3D cloud points inside each cell and check the distance of effective cells.

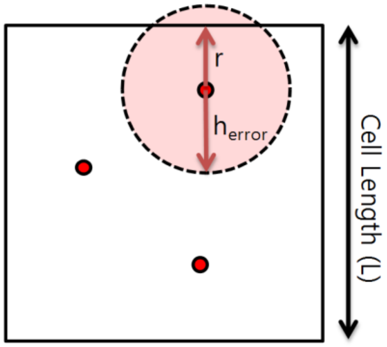


Figure 7. Probability of point cloud depth error to validate a false octree cell.

$$P_{noise} < threshold$$

$$P_{noise=} \int_{r=0}^{h_{error}} \left[\frac{1}{L}\left(\frac{h_{error} - r}{2h_{error}}\right)\right]^{n} dr$$

$$= \left(\frac{1}{2h_{error}L}\right)^{n} \int_{r=0}^{h_{error}} [(h_{error} - r)]^{n} dr$$

$$= \left(\frac{1}{2h_{error}L}\right)^{n} \frac{(h_{error} - r)^{n+1}}{n+1}\Bigg|_{h_{error}}^{0}$$

$$= \left(\frac{1}{2h_{error}L}\right)^{n} \left[\frac{(h_{error})^{n+1} + (h_{error} - h_{error})^{n+1}}{n+1}\right]$$

$$= \frac{h_{error}}{(n+1)(2L)^{n}}$$

at $n = 3$:

$$= \frac{h_{error}}{32L^{3}} < \text{threshold}$$

$$C4 = \sqrt[3]{\frac{dpth_{error} sin(tilt)}{32 \cdot threshold}}$$

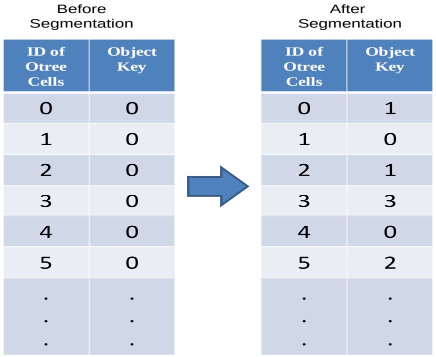
Given a specific volume of interest and expectation of people density, Our adaptive algorithm iterates over the above 4 constraints to try to find the optimum cell size that satisfies all of them with maximum number of required point cloud density per cell (n).

In order to guarantee segmentation, we computed cloud points inside cells and for object segmentation, only distance between two effective cells was considered as the separate condition.

The distance between two effective cells should be large than 1 cell size. After segmentation, each cell is assigned a segment key (Fig. 8). These keys are very important for head and hand detection, with them head and hand can be paired in each segment, which can avoid the situation of pairing hand and head of different agents in a cluttered environment.

3.2 Human like Objects Extraction

We set a design assumption for two agents to be at least one cell size apart. In real environment, however, people might have some connection with desks, chairs or other people. In this situation, the connected object will be detected as one object (Fig. 9). Hence, a simple shape constrain will be applied for eliminate the segments which are not likely to be human. We check the size of each segment, if the size is too large or too small; it will be eliminated, if the size is similar as normal human beings, the skin color


Before Segmentation

ID of Otree Cells	Object Key
0	0
1	0
2	0
3	0
4	0
5	0
.	.
.	.
.	.

After Segmentation

ID of Otree Cells	Object Key
0	1
1	0
2	1
3	3
4	0
5	2
.	.
.	.
.	.

Figure 8. Object key for each segment.

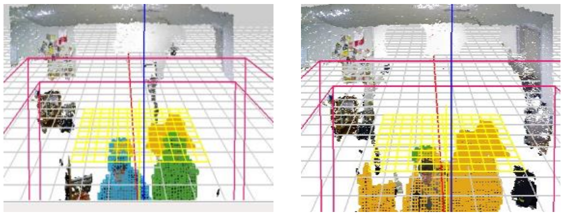


Figure 9. Object segmentation results by octree. The left figure shows each segment is successfully separated, the right figure shows when there is connection among the objects, these objects will be separated as one object. This kind of segment will eliminated for head and hand finding.

`

detection approach will check whether the segment has skin color information.

4. DEFINE HEAD AND HAND REGION

After obtaining human like segments from octree approach, we find head and hand for each segment. Head and hand candidate region are defined as follows:

4.1 Head Candidate Region Define

In previous work, human's face can be detected by Haar like feature face detector. Different from it, in our method, after we obtain human like segments, some simple geometric constrains were defined to determine head candidate regions. For each human like segments, normally head take the highest part of that segment. The highest cell can be easily computed from octree segmentation, make highest cell as reference point, a square region with 30cm of each side was drew for each segment as the head candidate region (Fig 10).

4.2 Hand Candidate Region Define

Purpose of our method is to design a simple calling gesture for elder people when call the elderly care robot. As simple calling gesture design before, only one hand to be detected for one people. Some elder people might be sitting in a wheel chair. Firstly we define a hand candidate region in 55cm lower than highest cell. Normally, when people are doing a calling gesture, hand supposed to be farthest part in front of human body. Depth information can be obtained by Kinect or Xtion Pro and used to define a hand candidate region.

Similar to compute highest cell for each segment, the nearest cell, with respect to camera plane, can be computed. With this nearest

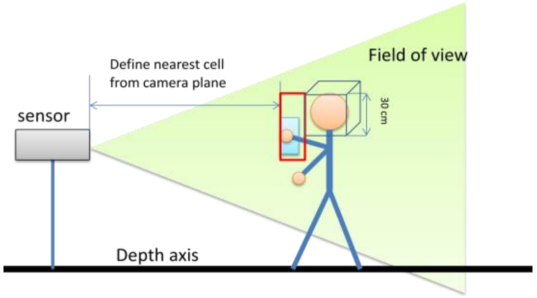


Figure 10. Overview of head and hand candidate region define. The blue cube assume to be head candidate region with 30 cm long of each side, blue rectangle is the candidate region for hand and red rectangle is the hand posture region.

cell as reference point, a 20cm x 20cm x 20cm region drew for hand candidate region (Fig. 10).

5. HEAD AND HAND FINDING BASED ON COLOR DETECTION

Head and hand candidate regions are defined through the approach above. Through the candidate region head and hand cannot be determined directly, more features are needed for head and hand detection. M. Ajalloooeian, A. Borji, and etc presented a way using saliency map to fast find potential hand regions [13], but most of other methods are using skin color detection for hand segmentation as the first step for gesture recognition due to the fact that hand and head skin color is quite different from other objects' color in background in most situations. By contrast, our approach defined potential region first and uses skin color information to check whether it is head and hand or not.

5.1 Color Space for Skin Color Detection

R, G, B color space is most commonly used color space in digital image processing , the luminance of R, G, B pixels are with linear combination relation with R, G, B values, in other words, when using R, G, B color space for skin color detection, it is sensitive to illumination changes. But some extended R, G, B color space algorithms are used for skin color detection and obtained satisfying results. TV color spaces which include YUV, YIO, and YCbCr are also used for skin color detection because they are not affected by changing illumination intensity and invariance by human race. In our method, perceptual illumination-invariant color space such as HSV is used for skin color detection.

5.2 Head and Hand Segmentation using HSV Color Space

H, S, V stand for Hue, Saturation and Value respectively. HSV color space can be obtained by a nonlinear transformation form RGB color space. Even though Value of the color is useful for a preceding step in our approach, we are not interested in analyzing it, since it's directly proportional to illumination intensity.

Instead, we are interested in classifying the color distribution in HS subspace to identify whether it belongs to human skin or not. First, we collect the RGB vector of candidate region of interest, then, we compute HSV vector using the following formulas:

Previous works relied heavily on threshold schemes for classifying skin color [24] [25] and [26]. This, however, lacks the robustness we seek for out open environment with uncontrollable illumination. Instead, we have designed a 2-layerclassification method that cluster HS distribution and match each cluster to training group of skin classifiers.

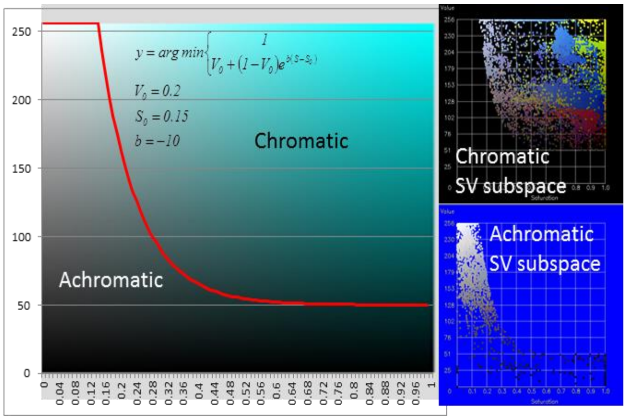


Figure 11. Chromatic/Achromatic pixels segmentation.

$$h = \begin{cases} undefined, & if\ max = \min \\ 60° \times \dfrac{g-b}{max-min} + 0°, & if\ max = r\ and\ g \geq b \\ 60° \times \dfrac{g-b}{max-min} + 360°, & if\ max = r\ and\ g < b \\ 60° \times \dfrac{b-r}{max-min} + 120°, & if\ max = g \\ 60° \times \dfrac{r-g}{max-min} + 240° & if\ max = b \end{cases}$$

$$s = \begin{cases} 0, & if\ max = 0 \\ \dfrac{max-min}{max} = 1 - \dfrac{min}{max}, & otherwise \end{cases}$$

$$v = max(r,g,b)$$

First, we try to exclude achromatic pixels since their HS distribution will be close to singularity and will thus considered outliers. Clustering of chromatic/achromatic colors is usually conducted in SV space. While in many researches, clustering of chromatic/achromatic colors is done by a simple linear threshold in SV subspace, we preferred to model an experimental threshold and represent it accurately in an exponential form as shown in figure 11.

We then collect chromatic pixels and represent their distribution in HS subspace. In contrast to common approaches of representing HS distribution in terms of peak histograms, we rely on a clustering based approach. Peak histogram tends to be biased towards high peaks and usually ignore low density colors. We would like to have a representation model that represents every color patch in input vector regardless of its density. For that purpose, we first represent HS space into weighted quad tree cells, which in a way, forms local histograms. We then apply our rapid cell-based clustering to separate and locate color patches. Clustering is performed by analyzing ratio of cell density to density of neighbor cells and comparing it to an adaptive threshold. This will suppress local minima near to high peaks regardless of their absolute values. Second step is to extend scope

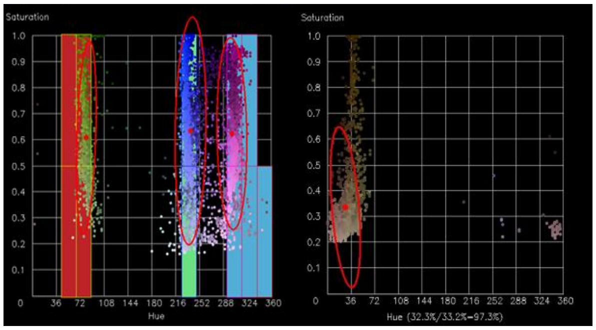


Figure 12. Chromatic pixels clustering and distribution modeling, left figure shows multi-color results and right one shows the skin color results.

of high peaks to cover locally suppressed neighbors. Figure 12 shows the clustering results a multicolored test objects and skin color clustering results.

For selection of quad tree cell size, we have conducted experimental analysis of one color distribution and determine cell size that is larger than this variance.

Finally, for color representation, we represent each cluster as a 7D feature-space point by extracting the following 7 features from HS subspace distribution: 1. mean H, 2. mean S, 3. Eigen vector 1 angle, 4. Eigen value 1, 5. Eigen values ratio, 6. Weight ratio, 7. Scattering index. Weight ratio is the ratio of number of pixels in that cluster to the total number of input pixels (including achromatic pixels). Scattering index is a volumetric distribution index that is computed from the ratio of cluster pixels volume to total input pixels volume.

For matching, we first have built a database of skin color classifier using 6 different skin-colored people, each generating 100 sample at different illumination conditions. We then represented their distribution statistically as a 7D hyper ellipsoid. We computed likelihood probability for each cluster by measuring the Mahalanobis distance from the cluster 7D point to classifier hyper ellipsoid. We, finally, assigned a probability to each candidate region according to the sum of all cluster's likelihood. Regions with probabilities below certain threshold were then discarded. We detected head first, and then highest candidate probability in a specific geometric space is assigned as a hand.

6. EXPERIMENT AND CONCLUSION

6.1 Experiments and Results

In order to test robust of our calling gesture recognition algorithm, we set hundreds of tests in several specific situations and evaluated the test results independent of the robot system, furthermore, this calling gesture component also was installed to our elderly care robot to test how it works in a robot system.

For evaluating the our method, a laptop with Intel(R) Core(TM) i7-2640M CPU @ 2.8G Hz, RAM of 4.00GB was used, both of the Microsoft Kinect and ASUS Xtion Pro were used for testing. We used ASUS Xtion Pro for our algorithm tests and used Microsoft Kinect ran at the robot system for practical tests.

Table 1, 2, 3 and figure 13, 14, 15 shows the performance of our gesture recognition system.

Initially, original octree size was set as cube with 3 m of each side, the octree center is 2.25 m from the camera plane, and maximum levels of octree segmentation were 6. As these parameters, octree segmentation cover range was 0.75 m to 3.75 m with respect to the camera plane. For the algorithms tests, the first group was set to test whether our method can detect multiple calling gestures at the same time. Several people were asked to do calling gestures in front of the sensor, and made environment background as natural as possible. Test results presented that our method detected most of the calling gestures both in one people and multiple people situation (Fig. 16).

Second group tested calling gesture detection rate in different illumination conditions and different distance (Fig. 17). The results show that extremely dark illumination condition caused the calling gesture recognition rate dropped down. The third group of algorithm tests presented calling gesture detection rate according to different distance from the camera plane. The results show that within the range from 0.75m to 3m, the calling gesture detection results were satisfying but if the distance was larger than 3m, detection rate also drained in the case of the range of the sensor.

We also installed our calling gesture recognition system into our Home-Mate robot for practical test (Fig. 18). In robot system, our method successfully worked in the robot, after calling gesture generated, the robot detected the user and received the position information of the caller then moved to the caller, full calling gesture demo video is given link at reference [27].

This simply calling gesture is easily operated by elder people, they just lift their hand with 3 times shaking more than 5 cm of the center of hand in left and right position, and they can call the robot to come for service. The computation time for detecting hand gesture was around 200 ms.

6.2 Conclusions and Future Works

This paper presents an octree segmentation based calling gesture recognition algorithms using sensors as Microsoft Kinect and ASUS Xtion Pro. And our algorithm is feasible to be run at elderly care robot, and our calling gesture is a kind of simply design for elder people. Octree segmentation, geometric constrains and skin color detection was integrated in this method. This paper also presented some robust, high ratio detection results for calling gestures.

For future work, we will extend our work to more complicated situations. There are also some weaknesses of our current method now, octree segmentation now take some computation time, in that case for the calling gesture is not a really real-time detection, a faster octree segmentation algorithms will be used. Second, if two people have some physical connections, they are separated as one object, using our height and depth information, just one of them can be detected. Skin color detection sometimes affected by illumination variance, more support features will be considered in future work.

Table 1. Performance in bright illumination

Distance (m)	TPR	**FPR**
0.75~1.50	92.81%	5.43%
1.50~2.25	93.86%	6.35%
2.25~3.00	94.12%	8.99%
3.00~3.75	73.21%	13.27%

Table 2. Performance in normal illumination

Distance (m)	TPR	FPR
0.75~1.50	93.02%	6.79%
1.50~2.25	92.33%	7.98%
2.25~3.00	90.01%	9.32%
3.00~3.75	74.87%	14.23%

Table 3. Performance in dark illumination

Distance (M)	TPR	FPR
0.75~1.50	80.79%	11.42%
1.50~2.25	79.44%	12.53%
2.25~3.00	77.42%	14.37%
3.00~3.75	65.78%	19.66%

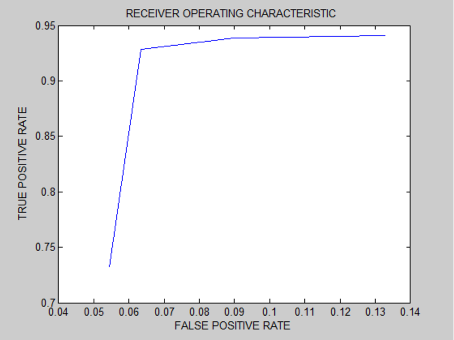


Figure 13. Receiver Operating Characteristic (ROC) of bright illumination results.

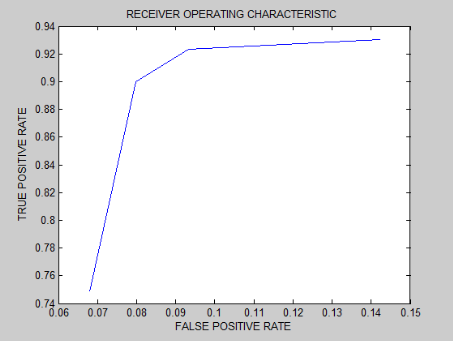


Figure 14. Receiver Operating Characteristic (ROC) of normal illumination results.

7. ACKNOWLEDGMENTS

This research was performed in part for the KORUS-Tech Program (KT-2010-SW-AP-FSO-0004), in part for NRF-2013M1A3A02042335, and in part for ITRC NIPA-2013-(H0301-13-3001), funded by the Korean Ministry of Science, ICT and Planning (MSIP).

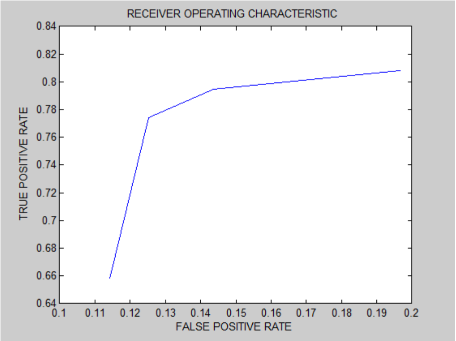


Figure 15. Receiver Operating Characteristic (ROC) of dark illumination results.

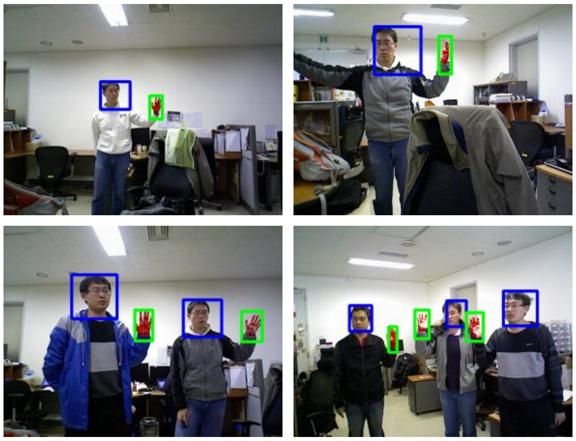


Figure 16. Test results of calling gesture recognition in single and multi-people situation. It shows that both in single people and multi people situation, our method can successfully detected most of the calling gestures.

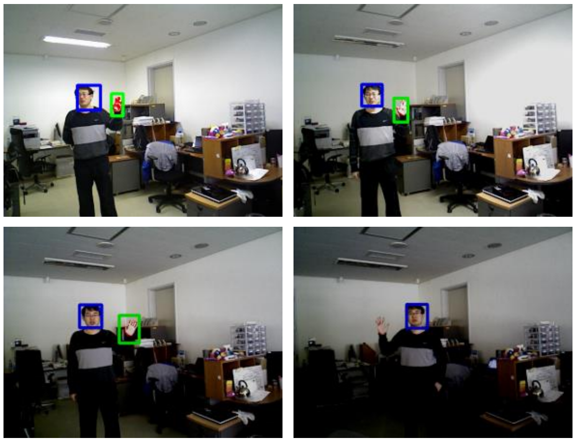


Figure 17. Test results of calling gesture recognition in different illuminations. We checked from left to right and top to bottom, the illumination decreasing. Test results show calling gesture detection rate decreasing with the illumination decreased.

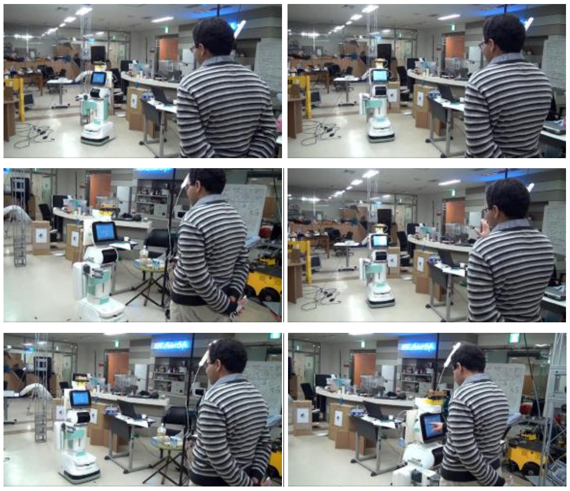


Figure 18. Calling gesture recognition in robot system test. The picture above are the screen shots of calling gesture demo, [27] show the full demo

8. REFERENCES

[1] Vladimir I. Pavlovic, Rajeev Sharma, T. 1997. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, 677-695.

[2] J. Triesch and C. 1997. Robotic Gesture recognition, In *Gesture workshop*, 233-244.

[3] Kakade, Deepali N., and J. 2012. Dynamic Hand Gesture Recognition: A Literature Review, *International Journal of Engineering* 1.9.

[4] Suarez, Jesus, and R. 2012. Hand gesture recognition with depth images: A review, In *RO-MAN*, IEEE, 411-417.

[5] Murthy, and R. 2009. A review of vision based hand gestures recognition, *International Journal of Information Technology and Knowledge Management* 2, no. 2, 405-410.

[6] Weinland, Daniel, Remi Ronfard, and E. 2011A survey of vision-based methods for action representation, segmentation and recognition, *Computer Vision and Image Understanding* 115, no. 2, 224-241.

[7] Shastry, Karthik R., Manoj Ravindran, M. V. V. N. S. Srikanth, and N.2010. Survey on Various Gesture Recognition Techniques for Interfacing Machines Based on Ambient Intelligence," *arXiv preprint arXiv*: 1012.0084.

[8] Zabulis, X., H. Baltzakis, and A.2009. Vision-based hand gesture recognition for human-computer interaction, *The Universal Access Handbook, Human Factors and Ergonomics*, 34-1.

[9] Ben Jmaa, Ahmed, Walid Mahdi, Yousra Ben Jemaa, and A. 2009.A new approach for digit recognition based on hand gesture analysis.

[10] Xia, Lu, Chia-Chih Chen, and J. 2011. Human detection using depth information by Kinect, *Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE Computer Society Conference on, 15-22.

[11] Doliotis, Paul, Alexandra Stefan, Christopher McMurrough, David Eckhard, and V. 2011 "Comparing gesture recognition accuracy using color and depth information," *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, p. 20. ACM.

[12] Yang, Cheoljong, Yujeong Jang, Jounghoon Beh, David Han, and H. 2012. Gesture recognition using depth-based hand tracking for contactless controller application, *Consumer Electronics (ICCE), IEEE International Conference* on, 297-298.

[13] Ajallooeian, M., A. Borji, B. N. Araabi, M. Nili Ahmadabadi, and H. 2009. Fast hand gesture recognition based on saliency maps: An application to interactive robotic marionette playing, *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, 841-847.

[14] Chen, Bor-Jeng, Cheng-Ming Huang, Ting-En Tseng, and L. 2012. Robust head and hands tracking with occlusion handling for human machine interaction, *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference* on, pp. 2141-2146.

[15] Liu, Xia, and K. 2004. Hand gesture recognition using depth data, *Automatic Face and Gesture Recognition*, 529-534. IEEE, 2004.

[16] Droeschel, David, Jorg Stuckler, and S. 2011. Learning to interpret pointing gestures with a time-of-flight camera, *Human-Robot Interaction (HRI), 2011 6th ACM/IEEE International Conference,* 481-488. IEEE, 2011.

[17] Mo, Zhenyao, and U. 2006. Real-time hand pose recognition using low-resolution depth images, *Computer Vision and Pattern Recognition*, vol. 2, 1499-1505. IEEE, 2006.

[18] Tang, M. 2011 Recognizing hand gestures with Microsoft's kinect, Website: http://www.stanford.Edu/class/ee368/Project_11/Reports/Tang_Hand_Gesture_Recognition. pdf (2011).

[19] MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 281-297, p. 14.

[20] Dempster, Arthur P., Nan M. Laird, and D. 1977. Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society. Series B* (Methodological) (1977): 1-38.

[21] Meagher, D. 1982. "*Geometric modeling using octree encoding*," *Computer Graphics and Image Processing* 19, no. 2 (1982): 129-147.

[22] http://en.wikipedia.org/wiki/Octree.

[23] Kim, Jaewoong, and S. 2009. Fast neighbor cells finding method for multiple octree representation. In *Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium on*, 540-545. IEEE, 2009.

[24] Gasparini, Francesca, and R. 2006. Skin segmentation using multiple thresholding. *Proceedings of SPIE*, vol. 6061, 128-135.

[25] Tsekeridou, Sofia, and I. 1998. Facial feature extraction in frontal views using biometric analogies. *Proceedings of the IX European Signal Processing Conference*, vol. 1, 315-318. 1998.

[26] Sobottka, Karin, and I. 1998. A novel method for automatic face segmentation, facial feature extraction and tracking, Signal *processing: Image communication* 12, no. 3 (1998): 263-281.

[27] https://www.youtube.com/watch?v=fQuysVqs2F0.