Proceeding of the 11th World Congress on Intelligent Control and Automation
Shenyang, China, June 29 - July 4 2014

# Depth Image Denoising and Key Points Extraction for Manipulation Plane Detection

Shuang Ma[1, 2], Changjiu Zhou[2], Liandong Zhang[2], Wei Hong[1], Yantao Tian[1]

1. College of Communication Engineering, Jilin University, Changchun 130025, China
E-mail: mashuang11@mails.jlu.edu.cn; hongweijilin@163.com; tianyt@jlu.edu.cn

2. Advanced Robotics and Intelligent Control Centre, Singapore Polytechnic, 500 Dover Road, 139651, Singapore
E-mail: {ZhouCJ, zhangld}@sp.edu.sg

***Abstract*— The handling of twist-locks has been a heavy burden for the container industry. To address this challenge, we are developing a customized mobile manipulator for handling the twist-locks. In this paper, we propose a fast normal computation algorithm for depth image, which is able to use normal deviation along eight directions to extract key points for segmenting points into objects on manipulation support plane in an unstructured table top scene. Before further processing point clouds, a bilateral filter is used to denoise depth images. To evaluate the effectiveness of the bilateral filter, eight direction angles are also used to observe the effectiveness of filter. To further evaluate the proposed approach, a median filter is also used for comparison with the bilateral filter. Experimental results show that the fast surface normal computation based on depth image and eight directions to determine a point are feasible for plane detection.**

## I. INTRODUCTION

Today leading terminal operators aim to achieve a fully automated container handling process. With the advances of the global economy and the development of container transportation has grown rapidly, an ever increasing number of goods over the world are being put into containers and loaded onto ships to be carried to their respective destinations. Containerization is regarded as the inevitable trend of the development of international transportation. During transferring, the twist-locks are used for locking a container into place. A twist-lock and corner casting together form a standardized rotating connector for securing shipping containers. So far the quite heavy twist-locks are still manually operated, and the handling sequence is shown in Fig. 1. The twist-lock handling system is the weakest link for port automation and is the most difficult automation part in the whole port automation field [1, 2, 3, 15].

Automation is the next step towards better planning security and safety at work by replacing stevedores working under or adjacent suspended loads in a high traffic area. Fully automated process also can provide an optimum efficiency in container handling. So there are the robotic technology needs to replace the stevedores' heavy work. It's the technical challenge to do twist-locks handling automatically in unstructured port environment. The robotic manipulation system requires a vast set of perception and manipulation capabilities. Real-time 3D perception of the surrounding environment is a crucial precondition for the reliable and safe application of mobile manipulation in dynamic environments. Plane detection is a prerequisite to a wide variety of vision tasks for finding task-specific locations. Both the detection of graspable objects and support manipulation plane highly depend on the quality of the estimation of surface normal. The normal estimation is important for depth image segmentation.


Fig. 1 Twist-lock coning and deconing sequenc

In this paper, we present algorithms to extract key points for segmenting points into objects on manipulation support plane. The points of are detected by surface normal and eight directions in an unstructured table top scene. For all points collected from Kinect, are inevitable polluted by noise, we involve bilateral filter to reduce the noise for getting more accurate results, and then evaluate the filter effect by eight direction angle.

The remainder of this paper is organized as follows: After giving an overview on related work in Section II, bilateral filter is briefly introduced in Section III, and then the methods for computing local surface normal and point identification located on support plane are described in detail in Section IV. The experiment results and analysis are given in Section V. Finally, we give a short brief conclusion and the future work in Section VI.

This work is supported by Translational and Innovation Fund from Ministry of Education, Singapore (MOE2013-TIF-1-G-057 and MOE2011-TIF-1-G-050), and Singapore Polytechnic TIEFA Research Grant (11-27801-36-R140). The work is also supported in part by the National Natural Science Foundation of China (NSFC) through grants No. 50505004 and No. 51275065.



978-1-4799-5825-2/14/$31.00 ©2014 IEEE

## II. RELATED WORK

There have been many efforts in developing automated twist-lock handling solutions [1,2]. To address this challenge, we are developing a customized mobile manipulator for twist-lock grasping, which is shown in Fig. 2. Hardware system is mainly composed of Microsoft Kinect [4], ABB IRB120 robot and one ROBOTIQ three finger adaptive gripper [5]. In consideration of rust and painting peeled off after long time use, in addition to light condition problem, color information obtained from ordinary camera is difficult to describe twist-locks characteristics in port environment. So the proposed approach only based on depth information can address the issues of twist-lock handling. With the advent of low-cost Microsoft Kinect, it can obtain RGB and corresponding depth information simultaneously [2,15]. Since depth images are insensitive to changes in lighting conditions, so we propose to use Microsoft Kinect to perceive environment instead of other depth perception sensors to simulate to obtain depth information. In the same time, all the algorithms are applied to the low-cost Microsoft Kinect, and can be used for other high-precision range camera sensors used in the real port environment.

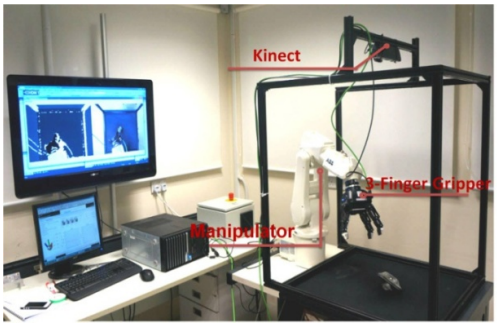



Fig. 2 Platform overview

Operating in complex real-world environments requires powerful perception and action capabilities. Perceiving the geometry of environmental structures surrounding the robot is a crucial prerequisite for applying mobile manipulation in the real-time environments. The objects to be handled are often cluttered in highly dynamic environment. The typical task of our handling system is to grasp and install objects on platform. This involves detection of objects, support plane and recognition of objects in robot's workspace. We attempt to use depth value to segment interesting area, and extract key points depending on normal vector at each pixel. We can apply RANSAC to fit support plane based on points belonging to flat. Then we employ the distance from point to plane to fined object clutter. All the proposed process is based on normal vector at each pixel, in order to get better detection results, noise filter is inevitable. Here we focus on less complex but fast methods to obtain an initial segmentation of the plane in real-time that can also be used in a variety of applications such as wall, obstacles and so on. The plane consists of smooth surface patches. And the point belonging to smooth surface is detected by eight directions of adjacent surface normal [11]. This preprocessing work is introduced in this paper, and our paper work is illustrated in Fig. 3.

With the advent of low-cost depth imaging camera, recently more and more image processing work focused on raw 3D point clouds for detection, recognition and pose estimation. The Kinect sensor we use is of quick reaction, very low cost but low accuracy of depth information including a lot of noise. Noise can be systematically introduced into depth images during acquisition and transmission. One fundamental problem of depth image processing is to effectively remove noise from an image while keeping its features intact. The simplest and the most common filter is median filter [6]. It is used extensively in smoothing and de-noising applications. For depth image denoising, we want to reduce noise depth image while preserving edge information. Bilateral filtering [7] for gray or color image can smooth images and preserve edges. Motivated by image filtering, we apply bilateral filter to depth image. The preprocessing operation is very important to next detection step. Both the detection of graspable objects and support manipulation plane as well as obstacles segmentation highly depend on the quality of the estimation of surface normal.

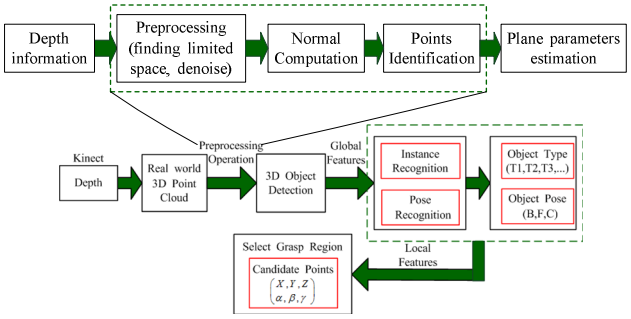



Fig. 3 Overview of the proposed perception process

Point normal estimation is a basic component of many segmentation approaches as well as in our paper. Most methods use a form of least squares, cross product, RANSAC, or PCA to fit a plane into a set of neighboring points [8, 9, 10, 11]. As most methods focus on arbitrary 3D point clouds, emphasis is on efficient selection of those points. Exploiting the intrinsic grid structure of range images, rough normal estimations can be computed much faster from the cross product of tangential vectors. For example, [10, 11] a method employing integral images to yield real-time performance is proposed. In this work, we follow similar cross product approach to compute normal vectors.

Rusu et al. [8] propose to segment point clouds into objects on planar surfaces. They suggest using RANSAC to detect planes and to extract shape primitives on the objects. Point clusters supported by these planes, lying above the plane and within its 2D bounding box after projection, are considered as objects. So plane detection is a prerequisite to a wide variety of vision tasks. Many existing segmentation algorithms aim for a simultaneous recognition of objects and their pose. To this end, found image features are matched to a database of known objects.

Various feature extraction methods have been proposed, including 3D-augmented SIFT features, and features directly obtained from range images such as a viewpoint feature histogram [12], depth-encoded Hough voting [13], point pair features [8]. These approaches robustly recognize partially occluded objects and correctly estimate their pose from stored 3D models, but are always restricted to the known set of objects. Point Cloud Library [14] aims to fit specific object models, the proposed approach is model-free and can successfully handle unknown, stacked, and nearby objects. Compute range images from point clouds, extract borders and key-points and use 3D feature descriptors to find and match repetitive structures. The above approaches show good results when processing accurate 3D range data, but tend to have high runtime requirements.

## III. BILATERAL FILTER FOR DEPTH IMAGE

Bilateral filtering for gray and color images [7] is non-linear, edge-preserving and noise-reducing smoothing filter, by means of a nonlinear combination of nearby image values. The method is non-iterative, local, and simple. It combines gray levels or colors based on both their geometric closeness and their photometric similarity, and prefers near values to distant values in both domain and range. It extends the concept of Gaussian smoothing by weighting the filter coefficients with their corresponding relative pixel intensities. Pixels that are very different in intensity from the central pixel are weighted less even though they may be in close proximity to the central pixel. This is effectively a convolution with a non-linear Gaussian filter, with weights based on pixel intensities. This is applied as two Gaussian filters at a localized pixel neighborhood, one in the spatial domain, named the domain filter, and one in the intensity domain, named the range filter. A very intuitive mathematical approach is given as follows,

$$I^{filtered}(x,y)=\frac{1}{w_p}\sum_{q\in\Omega}G_{\sigma_s}(\|p-q\|)G_{\sigma_r}(\|I_p-I_q\|)I_q \quad (3.1)$$

$$w_p=\sum_{q\in\Omega}G_{\sigma_s}(\|p-q\|)G_{\sigma_r}(\|I_p-I_q\|) \quad (3.2)$$

$$G_{\sigma_s}(\|p-q\|)=\exp(-\frac{\|p-q\|}{2\sigma_s^2}) \quad (3.2)$$

$$G_{\sigma_r}(\|I_p-I_q\|)=\exp(-\frac{\|I_p-I_q\|}{2\sigma_r^2}) \quad (3.2)$$

Where

$G_{\sigma_s}$ is the spatial weight; $\|p-q\|$ is the Euclidean distance between p and q;

$G_{\sigma_r}$ is the range weight; $\|I_p-I_q\|$ is the suitable measure of distance between the two intensity value $I_p$ and $I_q$ ;

I is the original image to be filtered;

Ω is the window centered in q;

$I^{filtered}$ is the filtered image.

Depth images obtained from low-cost provide, for every pixel, in addition to the usual color values, their distance from the camera. The noise exists in depth image, and some depth information will be missed. It is necessary to filter out noisy point in the depth image. Motivated by bilateral filter used in gray or color image, we deploy bilateral filter to depth image to reduce noise while preserving edge. Removing noise is an important preprocessing operation for our detection algorithm.

## IV. KEY POINTS EXTRACTION FOR PLANE DETECTION

We outline the overview of key points extraction for plane detection method. It can be divided into two main parts: the fast computation of normal vector at every pixel, and the point identification of flat surface for detecting manipulation support plane. After the two step analyses, we can combine these low level segmentations into object level segmentation further.

The crux of a segmentation technique is to detect surface discontinuities and smooth surface regions in a range (or depth) image. Looking more closely at the depth image, we can see that it has obvious depth contrast between foreground and background, we can also see that around object edges: discontinuous jumps of depth image, suddenly changes of the surface normal direction. Consequently, local geometric features form the fundamental basis for extracting semantic information from 3D point clouds, such as surface normal or curvature at a point. The common way for determining the normal to a point on a surface is to approximate the problem according to the point's local neighbourhood points.

### A Fast computation of local surface normal

The basic idea of the normal estimation method is to determine local surface normals from the cross product of two tangents to the surface. In mathematics, the cross product is an operation on two tangential vectors in three-dimensional space. The cross product of two vectors produces a third vector which is perpendicular to the plane in which the first two lie. The computation of normals is conducted in the local image coordinate frame. Less accurate but considerably faster method is to consider pixel neighborhoods instead of spatial neighborhoods.

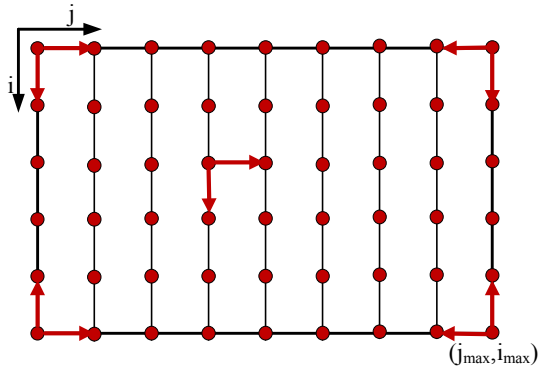



Fig. 4 Principle of fast normal computation using integral images

The determination principal of local surface normal is shown in Fig. 4. For each pixel in the depth image, the tangents are estimated from local pixel neighbours. In the

simplest case, both tangents could be calculated from just the horizontal and vertical neighbours, respectively. However, this approach would be highly prone to measurement noise. As shown in the Fig. 4, we first create two tangential vectors at point P, P is the pixel point in depth image, the normal vector corresponding to (*i,j*)th point *N*(*i,j*) can be calculated as follows:

$$\begin{aligned}
&\text{If } i \neq i_{\max}, j \neq j_{\max},\\
&\quad N(i,j) = cross(\overrightarrow{P_{(i,j)}P_{(i,j+1)}}, \overrightarrow{P_{(i,j)}P_{(i+1,j)}}),\\
&\text{else}\\
&\quad \text{if } i = i_{\max}, j \neq j_{\max},\\
&\qquad N(i,j) = cross(\overrightarrow{P_{(i,j)}P_{(i,j+1)}}, \overrightarrow{P_{(i,j)}P_{(i-1,j)}}),\\
&\quad \text{else}\\
&\qquad \text{if } i \neq i_{\max}, j = j_{\max},\\
&\qquad\quad N(i,j) = cross(\overrightarrow{P_{(i,j)}P_{(i,j-1)}}, \overrightarrow{P_{(i,j)}P_{(i+1,j)}}),\\
&\qquad \text{else } i = i_{\max}, j = j_{\max},\\
&\qquad\quad N(i,j) = cross(\overrightarrow{P_{(i,j)}P_{(i,j-1)}}, \overrightarrow{P_{(i,j)}P_{(i-1,j)}}).\\
&\text{End}
\end{aligned}$$

**B** Identification of flat surface plane or edges

According to above, we can use cross product to get all point normals easily. For the detection of the surface plane or edges, it is based on the computation of the angle among surface normals of adjacent image points, which can be efficiently done employing the dot product of two vectors:

$$\cos(N_1, N_2) = N_1 \bullet N_2 \qquad (4.2)$$

To judge the point belonging to plane or edge, we involve eight directions based on the neighboring pixels of a point, which is show in Fig. 5, defined as north (N), east (E), west (W), south (S), as well as northeast (NE), southeast (SE), southwest (SW), northwest (NW).

The eight directions are determined in a neighbor window, for example, the angle of north orientation of the edge (relative to the pixel grid) is given by:

$$\cos(\theta_N) = \frac{1}{w}\sum_{k=1}^{w} N_{(i,j)} \bullet N_{(i,j+k)} \qquad (4.3)$$

Here, w is the number of pixels along north direction in the neighborhood. Note that, we don't need to calculate the real angle value to judge point belong to plane or edge. According to the characteristic of cosine trigonometric function, the minimal cosine value of eight directions is corresponding to the strongest angular deviation.

$$\min\{\cos(\theta_E), \cos(\theta_{NE}), \cdots, \cos(\theta_{NE})\}$$

While large cosine value is close to 1, corresponding to flat surface, smaller cosine value indicates sharp curvature or object edges.

Our ultimately aim is to segment manipulation plane and object edges on an object level. So after all point identification, it needs to integrate homogeneous pixels into patches. Then we apply a simple region growing algorithm to integrate surface patches.

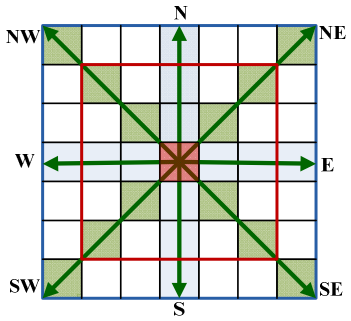



Fig. 5 Curvature determination based on normal direction

## V. EXPERIMENT RESULTS AND ANALYSIS

In this section, we will evaluate the efficiency of proposed approach on our twist-lock manipulation system. Both the detection of graspable objects and support manipulation plane as well as obstacles segmentation highly depend on the quality of the estimation of surface normal. So the accuracy of surface normal is vitally important for identification, segmentation in the next step.

Note, that the computation of surface normals is directly performed on the raw depth image, instead of the 3D point cloud. That is, the 2D image coordinates are augmented by the depth value to yield valid three-dimensional vectors. This procedure yields much more distinct changes of the normal direction at the boundary of objects, because the smoothing effect due to 3D projection is avoided. Before noise filtering, we need to define the robot arm reachable region, since the region that the arm of our robot can reach is limited, we can ignore the area that is too high or too low above our robot. Every point of the input point cloud is denoted with x, y and z coordinates, we filter out all points with too high or too low based on z-values, which is shown in Fig. 6. For defined region depth image, we use the fast normal computation proposed in this paper, the result is shown in Fig. 7. From the picture, we can see that noisy points caused normal direction deviation obviously.

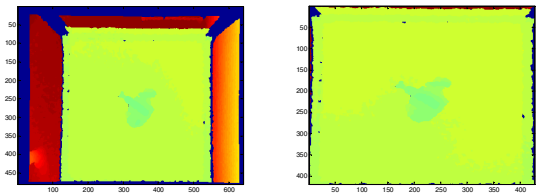


Fig. 6 Original depth image and defined region image

In order to reduce sensor noise and to obtain smooth and stable surface normal estimation, we apply bilateral filter to raw depth image. The parameters for depth image bilateral filter are: $\Omega$:5×5 , $\sigma_s$=3 , $\sigma_r$=32 . To evaluate the filter affectivity, we compare bilateral filter with median filter. To evaluate the effect of these two filter method for estimating normal in our paper, because the noise point is very difficult to find from depth image, we conduct a sequence points of eight orientations based on estimated surface normal in its neighborhoods. That can observe the normal direction changes more directly. In our paper, we

employ a threshold T =0.85, the identification of plane or curvature is according to the following:

$$\begin{cases} \mathrm{T}=\cos\theta \geq 0.85, \textit{flat} \\ \mathrm{T}=\cos\theta < 0.85, \textit{curvature}, \textit{edge} \end{cases}$$

We select some sensitive points as being typically analysis on depth images, the results are shown in (c)

**Fig. 8**.

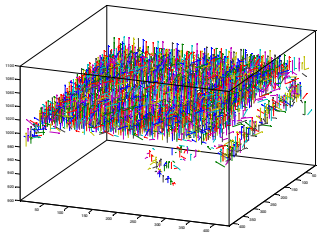


Fig. 7 Fast normal computation results without noise filter

From the results, we can see that the performance of bilateral filter is better than traditional median filter to reduce noise. From the last result, it shows that bilateral filter can reduce sensor noise and obtain smoother and more stable for our point identification process.

An important part of our work is the window size S of eight orientations for each point. It will bring a lot of computation when S=3*3. We compare the window size S=5*5 with S=7*7. The bigger window size for direction estimation is notoriously inaccurate. We manually collect flat surface points and edge points from different depth images. Eight directions analysis is used for sample points, the identification result of statistical analysis is shown Fig.9. Finally, we set the window size S=5*5.

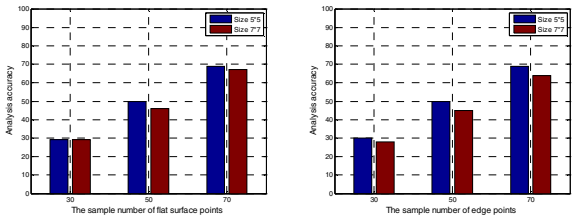
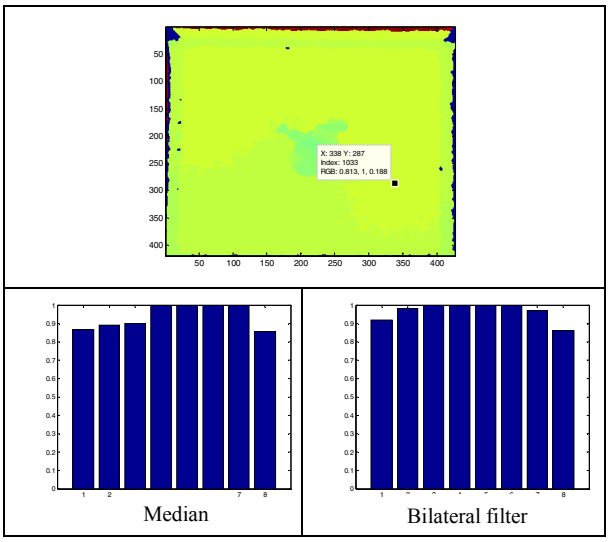


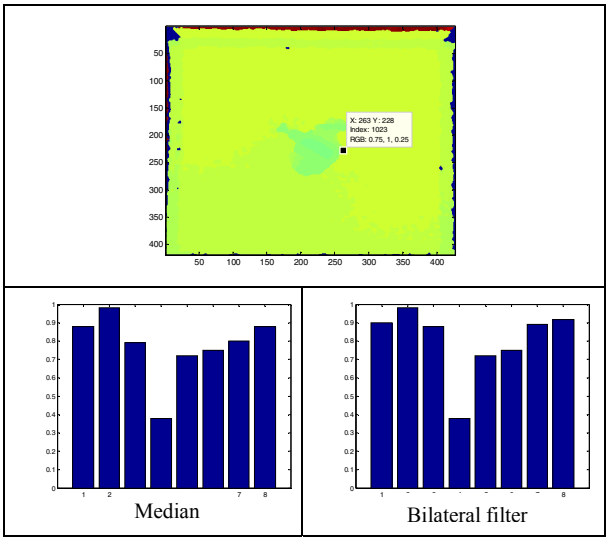

Fig 9 window size impact analysis

## VI. CONCLUSION

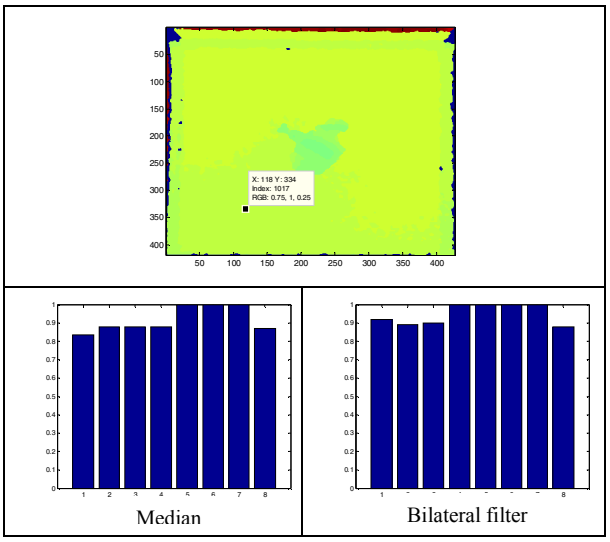
In this paper, we propose a fast normal computation algorithm for depth image, and it's able to use normal deviation along eight directions to extract key points for segmenting points into objects on manipulation support plane in an unstructured table top scene. All the points collected from Kinect, are inevitable polluted by noise, we involve a bilateral filter to denoise depth image. The eight direction angles can also be used to observe filter effectiveness. To further evaluate the proposed approach, a median filter is also used for comparison with the bilateral filter. Experimental results show that the fast surface normal computation based on depth image and eight directions to determine a point are feasible for plane detection. In the future work, we are developing a new framework based on cognition theory of working memory aiming to recognize object adaptively, the preprocessed data is regarded as cognitive system input.




(a)




(b)




(c)

Fig. 8 (a),(b),(c) Noise Filter analysis results

## REFERENCES


[1] M. Chellappa, "The weakest link," *Proceedings of International Maritime-Port Technology and Development Conference* (MTEC), pp. 278-282, 2011.

[2] S. Ma, C. Zhou, L. Zhang, W. Hong and Y. Tian, "3D Object Recognition Using Kernel PCA Based on Depth Information for Twist-lock Grasping," in *Proceeding of IEEE International Conference on Robotics and Biomimetics*, pp. 2667-2672, 2013.

[3] R.B. Rusu, N. Blodow, Z.C. Marton, and M. Beetz, "Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in human environments," in *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (IROS2009), pp. 1–6,.2009.

[4] ROBOTIQ [Online]. Available: http://robotiq.com/en/.

[5] K.I Kim, K. Jung, and H.J. Kim, "Face recognition using kernel principal component analysis," *IEEE Signal Processing*, Letter, Vol.2, No.9, pp. 40–42, 2002.

[6] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Trans. Acoustic., Speech, & Signal Processing*, vol. 27, no. 1, pp. 13–18, 1979.

[7] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proceedings of the 1998 IEEE International Conference on Computer Vision*, pp. 839-846, 1998.

[8] R.B. Rusu, N. Blodow, Z.C. Marton, and M. Beetz, "Close-range Scene Segmentation and Reconstruction of 3D Point Cloud Maps for Mobile Manipulation in Domestic Environments ," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1–6, 2009, 2009

[9] E. Castillo, H. Zhao, "Point Cloud Segmentation via Constrained Nonlinear Least Squares Surface Normal Estimates", *Recent UCLA Computational and Applied Mathematics Reports*, 2009

[10] D. Holz, S. Holzer, R.B. Rusu, and S. Behnke, "Real-Time Plane Segmentation using RGB-D Cameras," *RoboCup Symposium, Springer*, pp. 307-317, 2012.

[11] A. Uckermann, C. Elbrechter, R. Haschke, and H. Ritter, "3D Scene Segmentation for Autonomous Robot Grasping," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), pp. 20155-2162, 2012.

[12] R.B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), pp. 1734-1740, 2010.

[13] M. Sun, B. Gary, B. Xu, and S. Savarese, "Depth-Encoded Hough Voting for Joint Object Detection and Shape Recovery," *the 11th European Conference on Computer Vision*, pp. 658-671, 2010.

[14] R.B. Rusu, and S. Cousins, "3D is here: Point Cloud Library (PCL)," 2011 *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1-4, 2011.

[15] L. Zhang, C. Zhou, X. Han, S. Ma, and R. Li, "Twist-lock pose estimation and grasping based on CAD Model," *Proceedings of the 3rd IFTOMM International Symposium on Robotics and Mechatronics*, pp. 739-747, 2013.