

# An Object Finding Robot Vision System for the Indoor Living Environment Based on Saliency Map and SIFT

Jyun-Han Wei<sup>1</sup>, Shih-Hung Wu<sup>\*2</sup>

<sup>12</sup>Dept. of CSIE, Chaoyang University of Technology  
No.168, Jifeng E. Rd., Wufeng Dist, Taichung city, Taiwan(R.O.C.)

<sup>1</sup>s9827635@cyut.edu.tw

<sup>2\*</sup>shwu@cyut.edu.tw

*\*Correspondence author*

**Abstract— This paper presents an object finding robot vision system that can work in indoor living environment. With the robot vision system, a robot can find a specific object that a user appointed in an office or a house. This system is based on an improved saliency map and the scale-invariant feature transform (SIFT) image feature. The experiment shows that our robot vision system works well in everyday life environment.**

**Keywords— Robot vision, Saliency map, Object finding**

## 1. INTRODUCTION

In recent years, the home-service robots are working with people in offices or houses. They often equipped with cameras to perceive the environment. A robot vision system process the image and return the information in the image, the information is used to answer “Which way should I go?”, “Where is the object?”, or “There is a man ahead”...etc. Since the image’s information is plentiful, it makes the image process difficult.

In this paper, our goal is to build a robot vision system that can guide a robot to find a specific object in the indoor living environment. The specific object is known and must be in the environment somewhere. The robot will scan the place, and use a probability map to show the probability that the object position might be. The robot then according to the probability map, move to the most likely position to find the object.

Object finding is an important component of object fetching; a service robot is expected to bring back various objects for human in office or house. However, it is hard to find the object not in the known place but in all possible places.

Thus, a robot vision system is crucial to find the object first.

The rest in the paper is organized as follows, section 2 is the background of the technologies we used, section 3 is out improvement of the methods, section 4 is the experiments and results, final section is the conclusions.

## 2. BACKGROUND TECHNOLOGIES

In this section, we will investigate the literature of the technologies we used.

### 2.1. Saliency map

Saliency map is the technologies that transform an image to highlight the object in interest. The technology will emphasize the intensity of the region of interest, and reduce the intensity of other pixels. The idea is to simulate human’s attentions; human’s eyes pay attention to the selected object at one time, not to the different things of background. So in the saliency map, the important things will be emphasized, and the intensity will be stronger than the background.

The saliency map technology has many applications. Feng et al. [1] used it on the Content-based image retrieval (CBIR), a saliency map is constructed from the input image. According to the saliency map, salient region is identified and used to retrieve relative images from database. Saliency map is also used to the scenes recognition [2][3].

In our case, we expect that the robot can find the specific object, than the robot can manipulate the object later. For example, in Fig. 2, there are two cans in front of the robot, supposed that the user want the red one. In a traditional saliency map, like Fig. 2 (b) is not good for the robot to recognize the specific object. We wish the

saliency map is more like Fig. 2(c), where the object in want is highlighted, and then the robot will not get the wrong objects in the scene.



Fig. 1 Saliency map and input image.

The saliency map construct process is defined as below:

$$SM(I) = \sum_{l=1}^L \sum_{t \in \theta} g(I_l, t) + C(I_l) \quad (1)$$

SM is the saliency map, I is the input image. The saliency map is the summation of Gabor filter result and the color information of the input image. Where L is the level of image pyramid, usually is set to three,  $\theta$  is the gabor filter angles, and the  $g()$  is the gabor filter. The  $C()$  is the color information, the formula is defined as:

$$\begin{aligned} C(I_l) &= \frac{1}{4} \sum_{d \in R, G, B, Y} c_{d_{I_l(i,j)}} \quad (2) \\ &= \frac{1}{4} R_{(i,j)} + G_{(i,j)} + B_{(i,j)} + Y_{(i,j)} \\ R_{(i,j)} &= r_{I_l(i,j)} - (g_{I_l(i,j)} + b_{I_l(i,j)})/2 \quad (3) \\ G_{(i,j)} &= g_{I_l(i,j)} - (r_{I_l(i,j)} + b_{I_l(i,j)})/2 \\ B_{(i,j)} &= b_{I_l(i,j)} - (r_{I_l(i,j)} + g_{I_l(i,j)})/2 \\ Y_{(i,j)} &= \frac{(r_{I_l(i,j)} + g_{I_l(i,j)})}{2} - \frac{(r_{I_l(i,j)} - g_{I_l(i,j)})}{2} - b_{I_l(i,j)} \end{aligned}$$

Where the  $(i, j)$  is the coordinate of image I.

## 2.2. SIFT

Keypoint identification methods, such as SIFT [4] or SURF [5], have been used to recognition objects of robot vision [6], or used to representation of pseudo-objects [7] in CBIR.

Scale-invariant feature transform (SIFT) is a well-known image feature and widely used in the image recognition. SIFT is propose [4], here we give a brief description.

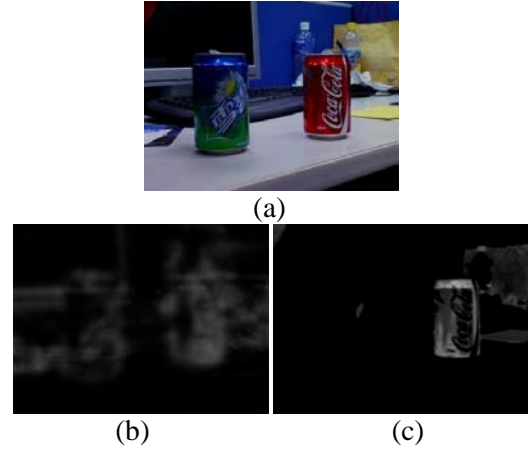


Fig. 2 A Saliency map example, (a) two cans in an image, (b) saliency map of (a), (c) saliency map of emphasizes the Coke can.

At the first step, the system detects the extreme in scale space, use the Difference of Gaussian (DoG) :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (4)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5)$$

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y, \sigma) \quad (6)$$

$$= L(x, y, k\sigma) - L(x, y, \sigma)$$

$D(x, y, \sigma)$  is the DoG function, compute the difference two nearby scale.

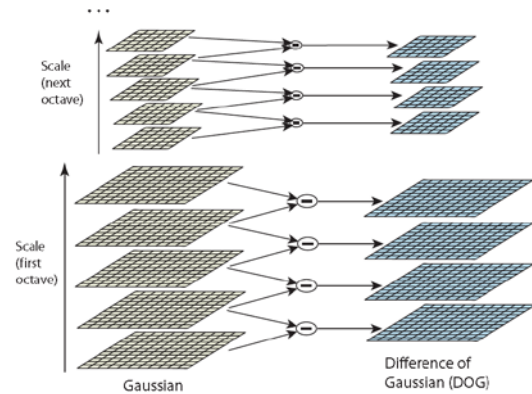


Fig. 3 Difference of Gaussian [4]

Then find the local maximum or minimum, compare with the neighbors, and the near scale.

When the key point candidate is find out, then reject the low contrast and the edge.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (7)$$

$$\hat{x} = -\frac{\partial^2 D^{-1} \partial D}{\partial x^2 \partial x} \quad (8)$$

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (9)$$

If the  $|D(\hat{x})|$  less than 0.03, the keypoint will be remove. Then the edge eliminate used  $2 \times 2$  Hessian matrix.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \quad (10)$$

$$T_r(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (11)$$

$$\text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (12)$$

$$\frac{T_r(H)^2}{\text{Det}(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(\gamma\beta + \beta)}{\gamma\beta^2} \quad (13)$$

$$= \frac{(\gamma + 1)^2}{\gamma}$$

Check the threshold:

$$\frac{T_r(H)^2}{\text{Det}(H)} < \frac{(\gamma + 1)^2}{\gamma} \quad (14)$$

If the value is bigger than the threshold, then remove the keypoint candidate.

Therefore, the key point is find out, then assignment the orientation:

$$m(x,y) = \sqrt{\frac{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}{2}} \quad (15)$$

$$\theta(x,y) = \tan^{-1} \left( \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \right) \quad (16)$$

where  $m(x,y)$  is the gradient, and  $\theta(x,y)$  is the orientation. The orientation histogram has 36 bins. Peaks in the orientation define the domain direction.

Fig. 5 is the keypoint descriptor, computing the gradient and orientation, then weighted by Gaussian window, and summarizing into  $4 \times 4$  windows, the arrow is mean the direction and sum.

In [4], the author is use  $4 \times 4$  window and 8 orientations get best result. So the key point descriptor is a  $4 \times 4 \times 8 = 128$  dimensions feature vector.

The key point matching is based on the Euclidean distance:

$$d = \sqrt{\sum_{i=1}^{128} (S_i - T_i)^2} \quad (17)$$

where  $S_i$  and  $T_i$  are two SIFT keypoints, if the distance is smaller than threshold, then can be see a pair.

Although the SIFT can effective match the feature points, but in our case it's will be affected by the distance. When the object is closed, the effect will be good; else the effect is lower when the object is far.

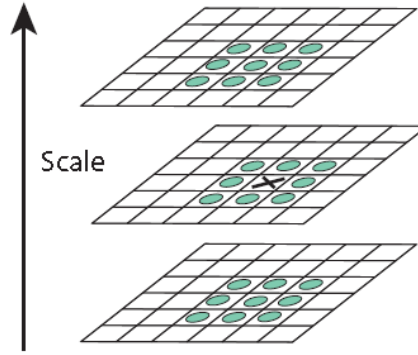


Fig. 4 Compare with neighbors, to find the maximum or minimum [4]

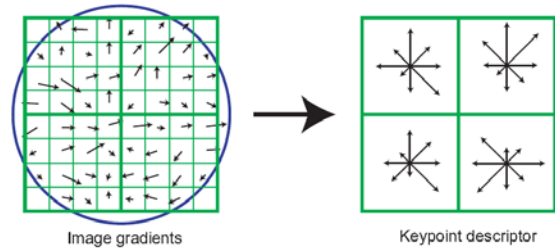


Fig. 5 Descriptor orientation [4]



Fig. 6 Difference distance of the object with camera, the effect of SIFT will be affected.

### 2.2.1 SIFT clustering

Generally the objects are not plat, there are many possible angles that facing the robot. So the template of an object should consider the SIFT feature points in various angles. In Fig. 8, there are 1052 SIFT feature points. To reduce the number of feature points that need to match, our system uses the K-means clustering algorithm to find representative points.

### 2.3. Probability map

Previous work [8] proposed way that a visual search system can be used to guide robot move based on the probability map. The space is separate into grids; in the initial, the probability map can be zero or give the initial probability due to some reason, like hint the object possible where is.



Fig. 7 Difference sides of the object Coke can.



Fig. 8 The SIFT feature points in difference sides of the object Coke can.



Fig. 9 Probability map, the “R” is mean robot, “X” is mean the obstacle.

The probability is updated according to the result of a vision system. When the probability map is updated, the robot will move to the next position. The moving strategy can be use many difference ways, such as move to the highest probability position, or the highest probability and near with robot.

When the robot arrive the new position, scan the surrounding of the robot, compute the new

probability to update the probability map to move robot to next position, until the robot find the object.

### 3. OUR SYSTEM

In this section we will show our method.

#### 3.1 System flowchart

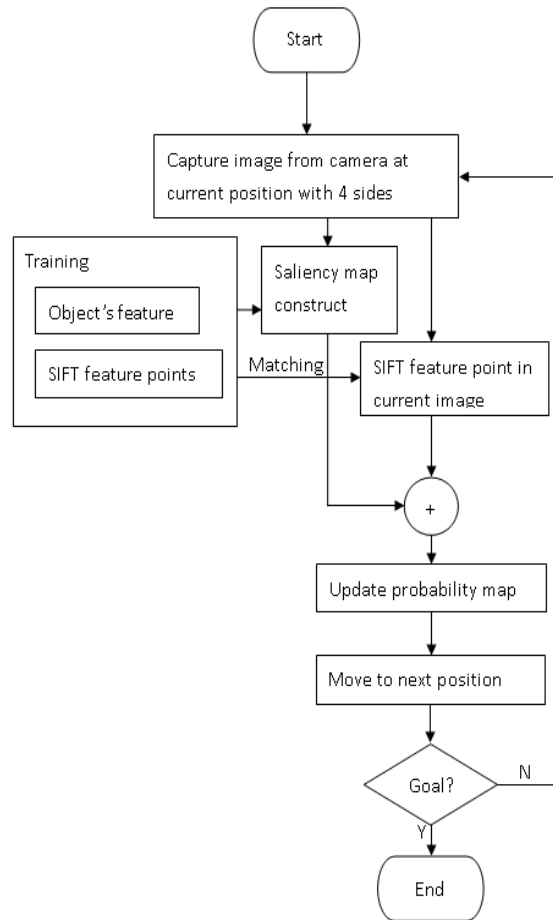


Fig. 10 The system flow chart.

At first, we get the specific object’s feature to join saliency map construct, and the SIFT feature point used in the match. Therefore, the score of input image will be used to update the probability map.

When the probability map is updated, the robot will move to the high probability position.

The final goal is the robot move to the position of object.

#### 3.2 Saliency map module

The saliency map is designed to provide the information of a specific object in a given image. We modify the traditional saliency map by adding the features of the specific object. The first one is the color of the specific object. In our previously work [9], we add the color information to the saliency map, and we test it in data set obtain good result. In this paper we do it in the real live environment.

As we proposed [9], we used the object's feature to construct the saliency map:

$$C(I_1) = \frac{1}{4} \sum_{d \in R, G, B, Y} F_d c_{d_{I_1(i,j)}} \quad (18)$$

We assigned a weight to the color component.

$$F_r = f_r / M \quad (19)$$

$$F_g = f_g / M$$

$$F_b = f_b / M$$

$$F_y = f_y / M$$

$F_{r,g,b,y}$  are the weight of each color R, G, B and Y, the  $f_{r,g,b,y}$  are the count histogram bin from bin  $x$  to the bin 255; and the  $M$  is normalization factor.

$$f_r = \sum_x^{255} N_x^r \quad (20)$$

$$f_g = \sum_x^{255} N_x^g$$

$$f_b = \sum_x^{255} N_x^b$$

$$f_y = \sum_x^{255} N_x^y$$

$$M = \sum_x^{255} f_{r,g,b,y}$$

The Fig. 11 is use the weight color component construct saliency maps.

### 3.3 Probability map module

Our goal is find specific object in live environment, such as laboratory, kitchen, living room...etc. The object is possible at anywhere in a live space. So the robot need the reason to move, the probability of object position is a reason. [8] was show that, the probability map can help the robot to find the object.

In our paper we will use the probability update function to update the probability map. In the first, the space will be parting to several equal squares, it's can be seen as a set  $C = \{c_1, c_2, \dots, c_i, \dots, c_k\}$ , each grid have a probability  $P$  with the time  $\tau$ , the  $P_\tau$  is calculate by the following:



Fig. 11 The image takes in real live environment and its saliency map.

$$P_\tau(c_i) = \frac{\alpha v_\theta + \beta s_\theta}{V + S} \quad (21)$$

The  $\theta$  is the direction of robot face to, and the  $v_\theta$  and  $s_\theta$  are the saliency map method and SIFT method respectively.

$$v_\theta = \left( \sum_{x=0}^m \sum_{y=0}^n SV(x,y) \right) / B \quad (22)$$

$v_\theta$  is the saliency value of current image,  $SV(x,y)$  is the saliency value of pixel  $(x,y)$ ,  $m$  and  $n$  is the width and height of image, and  $B$  is the sum of non-zero pixels.

$$V = \sum_{t \in \theta} v_t \quad (23)$$

$$S = \sum_{t \in \theta} s_t \quad (24)$$

And the probability will be normalized.

The probability map update the same direction  $\theta$  grids, and will be reduced by the distance with the robot's position.

The next time  $\tau + 1$ 's probability is shown as:

$$P_{\tau+1}(c_i) = \left( \frac{\alpha v_\theta + \beta s_\theta}{V + S} + P_\tau(c_i) \right) / \sum P(C) \quad (25)$$

## 4. EXPERIMENTAL



In this section we will show the experiment environment and the process, and final we will show the result.

#### 4.1 Experimental environment

In our case, we use the room size is 3.5m × 5m, the room was be divide into 50cm × 50cm grids.



Fig. 12 The environment.

The probability update function is as [8], consider the probability intensity and reduce, if the region's probability is increase, but there isn't having object, the probability will be weak next scan. And final, our goal is led the robot to the front of the object.

#### 4.2 Satisfy k value

We want to find the best k value in our experiment, so we take the k value from 1 to n, n is the total SIFT key points of the template object. We put the object in front of camera, the distance from 50cm to 200cm.

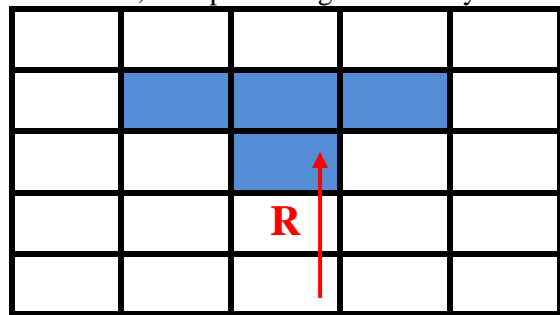
Fig. 13 is the SIFT key point matching result with the cluster centers, X-axis is the cluster quantity, the Y-axis is the matching pairs. As previously, when the object is near camera, the threshold is loose (0.6), the matching result is better than others. But there are still many noises

like key point of background, so we only take the object from the image, to do the match with cluster centers, Fig. 14 is the result. In the near distance and loose threshold, the match result is best.

The Fig. 15 is the match result. We want to find the best k value. The match pair of object less than originally image (with background) is that, because the noise (SIFT key point of background) is be reduced.

#### 4.3 Experiment result

In our experiment, we set the scan strategy is scan around the robot with 4 orientations, each scan will get the saliency map and the SIFT match result, the update range is a "T" symbol.



The "R" is robot position, the update range is the robot face to.

In this experiment, according the Fig. 13, Fig. 14 and Fig. 15, we decide the k value is 673. When the k is 673, the match pairs in noise background is 645 pairs, and the cut of the object without background, the match pairs is 86 pairs when the distance of object and camera is 50 cm.

The following is the process of the robot scan, move and final move to the object's position.

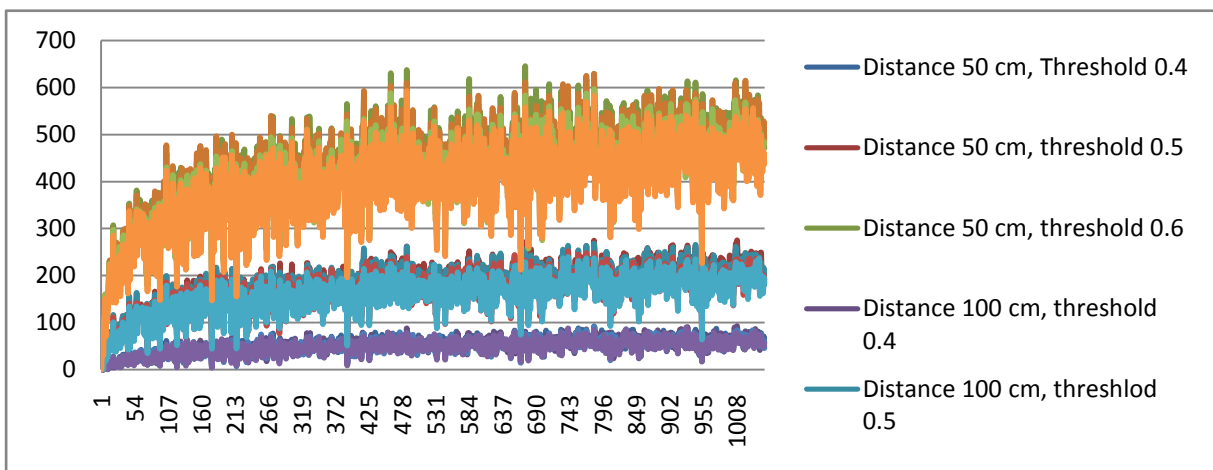


Fig. 13 SIFT key point matching pair with the difference distance and threshold.

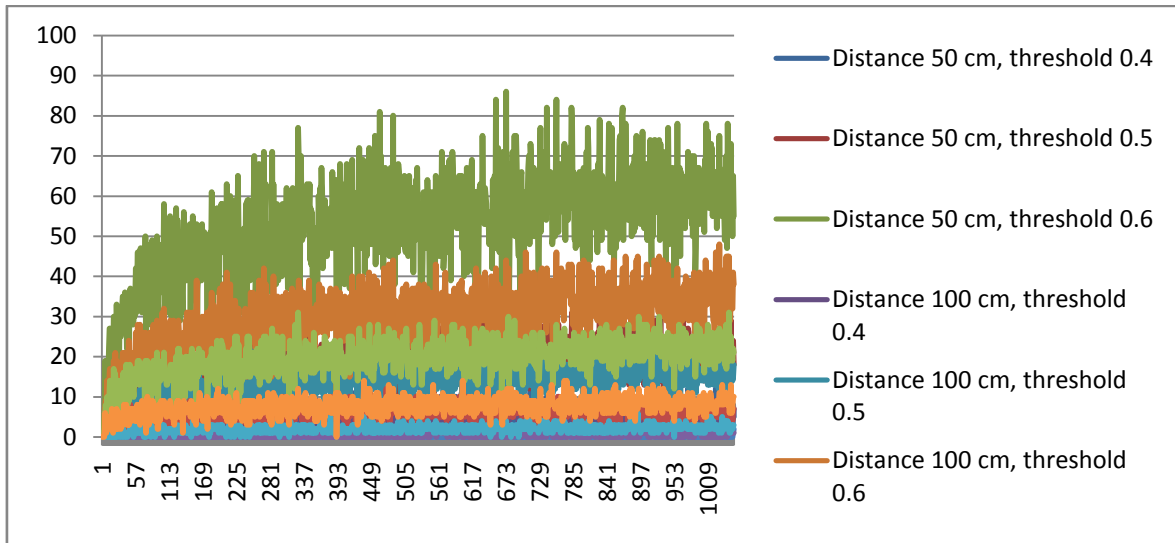


Fig. 14 SIFT key point matching with cut the object without background.

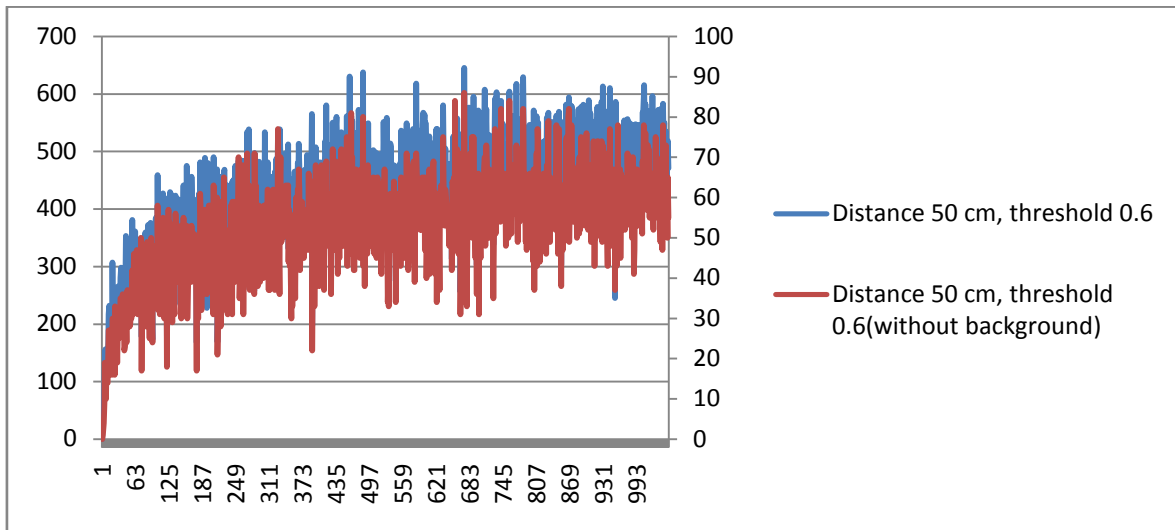
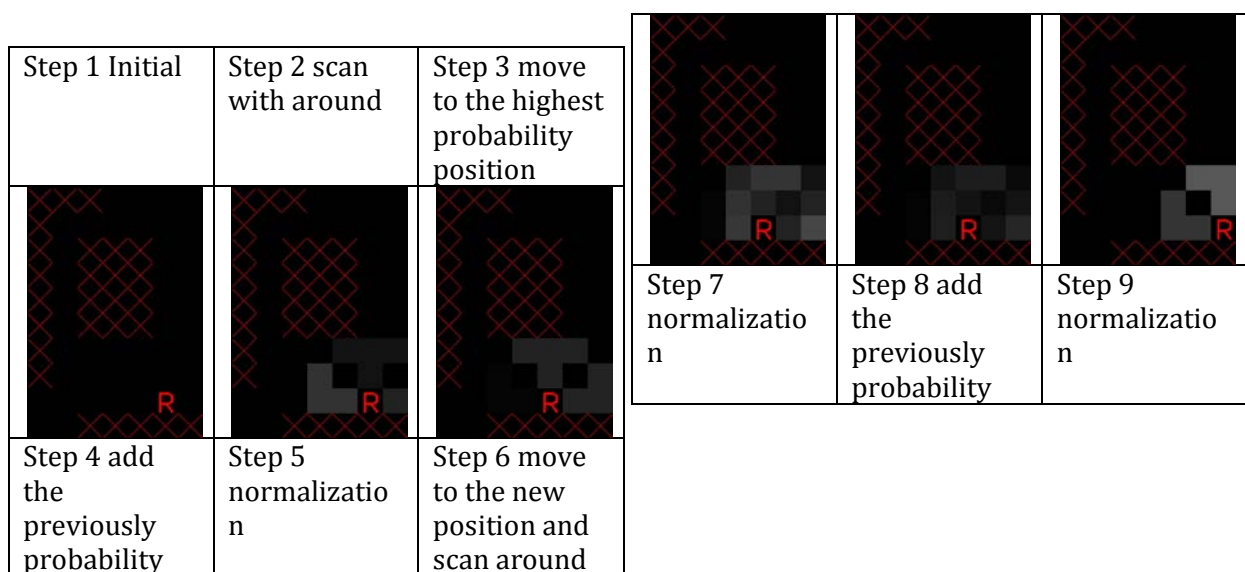










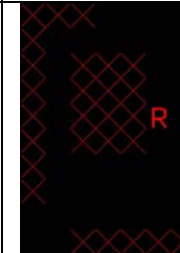





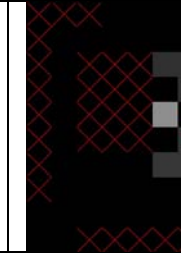





Fig. 15 SIFT key point matching result of same distance 50 cm, same threshold 0.6, with background and without background.



			to the new position and scan around	normalization	the previously probability
Step 10 move to the new position and scan around	Step 11 normalization	Step 12 add the previously probability			
			Step 25 normalization	Step 26 move to the goal	
Step 13 normalization	Step 14 move to the new position and scan around	Step 15 normalization			
			In this experiment, the robot moves 7 times, finally the robot find the goal, the object's position.		
Step 16 add the previously probability	Step 17 normalization	Step 18 move to the new position and scan around	<b>5. CONCLUSIONS</b>		
			In this paper, we report a robot vision system that can find specific object in the indoor living environment. The system incorporates with computer vision and the probability map technologies such that a robot can move to the object's position. When the robot arrive the object's position, then the robot can manipulate the objects for various purposes. The robot vision system is based on a modified saliency map technology and SIFT.		
Step 19 normalization	Step 20 add the previously probability	Step 21 normalization	In the feature we will take the depth information in the image. The depth information may be helpful when the robot is near the object on both object recognition and object manipulation.		
			<b>REFERENCES</b>		
Step 22 move	Step 23	Step 24 add	[1] S. Feng, D. Xu, X. Yang, <i>Attention-driven salient edge(s) and region(s) extraction with application to CBIR</i> , The journal of Signal Processing, Volume 90, Issue 1, pages 1–15, January 2010.		
			[2] L. Itti ,C. Koch, E. Niebur, <i>A model of saliency-based visual-attention for rapid scene analysis</i> , IEEE Transactions on		



- Pattern Analysis and Machine Intelligence, 20, No. 11, 1254–1259, 1998.
- [3] J. J. Bonaiuto, L. Itti, *Combining attention and recognition for rapid scene analysis*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'05), San Diego, CA, USA, 20–26 June 2005.
- [4] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision, 60(2), 91–110, 2004.
- [5] H. Bay, T. Tuytelaars, L.V. Gool, *SURF: speeded up robust features*, In: Proceedings of the European Conference on Computer Vision, vol. 3951, pp. 404–417, 2006.
- [6] D. Lee, G. Kim, D. Kim, H. Myung and H.T. Choi, *Vision-based object detection and tracking for autonomous navigation of underwater robots*, The journal of Ocean Engineering, Volume 48, Pages 59–68, July 2012.
- [7] K.T. Chen, K.H. Lin, Y.H. Kuo, Y.L. Wu and W.H. Hsu, *Boosting image object retrieval and indexing by automatically discovered pseudo-objects*, the Journal of Visual Communication and Image Representation, Volume 21, pp. 815-825, Issue 8, November 2010.
- [8] K. Shubina and J. K. Tsotsos, *Visual search for an object in a 3D environment using a mobile robot*, the journal of Computer Vision and Image Understanding, volume 114, issue 5, pp. 535-547, 2010.
- [9] J.H. Wei, S.H. Wu, L.P. Chen, W.T. Hsieh and S.C. Chou, *Robust Object Finding Vision System based on Saliency Map Analysis*, The 4th International Conference on Awareness Science and Technology(iCAST 2012), Seoul, Korea. Aug 21-24, 2012 2005.