

Implement of a 6-DOF Manipulator with Machine Vision and Machine Learning Algorithms

Pi-Yun Chen

Department of Electrical Engineering
National Chin-Yi University of Technology
Taichung, Taiwan
chenby@ncut.edu.tw

Neng-Sheng Pai

Department of Electrical Engineering
National Chin-Yi University of Technology
Taichung, Taiwan
pai@ncut.edu.tw

Guo-Shing Huang

Department of Electronic Engineering
National Chin-Yi University of Technology
Taichung, Taiwan
hgs@ncut.edu.tw

Wei-Lun Chen

Department of Electrical Engineering
National Chin-Yi University of Technology
Taichung, Taiwan
cwl_24@hotmail.com

Min-Hung Chou

Department of Marine Engineering
National Kaohsiung Marine University
Kaohsiung, Taiwan
chouhp88scott@webmail.nkmu.edu.tw

Che-Yuan Chang

Department of Electrical Engineering
National Chin-Yi University of Technology
Taichung, Taiwan
a2356689@yahoo.com.tw

Abstract –This paper explores the application of a Machine Vision and Machine Learning Algorithm to a Manipulator with six degrees of freedom (6-DOF). A Kinect sensor were used to extract images from a screen and obtain the relevant target information. Image processing was accomplished using a Scale-invariant feature transform (SIFT) Algorithm to capture image of the target object. The processed visual is rendered on the computer controller and Manipulator Learning is accomplished using a Reinforcement Learning Algorithm. Markov Decision Processes (MDP) were used to train the Manipulator to move to the location of the target object. Experimental results showed the Reinforcement Learning Algorithm proposed in this paper is effective and can be utilized on a 6-DOF Manipulator with reproducible results.

Keywords- Machine Vision, Scale-invariant feature transform (SIFT), Kinect, Machine Learning, Reinforcement Learning, Manipulator

I. INTRODUCTION

Robots can replace people for certain items in the production process, such as repetitive work, frequent, monotonous, and long lasting work, or work in dangerous and precarious environments. However, such worksites all have limitations and thresholds. If control is imprecise, or if the robot is affected by the environment, unexpected behavior can arise which might adversely affect the operation or even lead to failure of the task, including damage to the object and even the surroundings. The need to improve quality of a task so that the robot precise control and automation has become an important issue.

In this paper we introduced a Machine Vision and Machine Learning Algorithm to give the

manipulator visualization function. We chose to use the Kinect v2 sensor [1] for our experiments in view of the environmental configuration and the depth information that was needed [2]. A manipulator that has visualization function must also have the ability to detect and identify the target and its position in the image. In this paper we used SURF[3] and SIFT[4] simultaneously and compared the difference between the two.

After the system has obtained the necessary information about the environment, it is necessary to establish a route planning policy. Machine Learning is also needed and a Learning algorithm for the manipulator was used. Commonly used Machine Learning Algorithms include, Reinforcement Learning [5], and support vector Machine (SVM) [6]. We used Reinforcement Learning because it more closely matched the system used and allowed the manipulator to learn how to grab any target object.

II. SYSTEM ARCHITECTURE AND HARDWARE

The system architecture can be divided into Image processing, Machine Learning and the servo control system (see Figure 1).

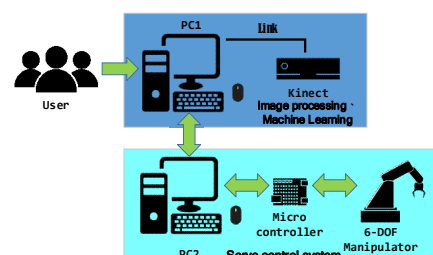


Figure 1. System architecture

The Image processing and Machine Learning end is the main system. The Kinect sensor is used to gather image data from the work area which then undergoes image processing and calculation. After the parameter data required by the system has been calculated, it is input to the Machine Learning Algorithm in the program for calculation. Finally, the results are input to the servo control system computer. The micro controller issues commands to the servo motors based on the calculation results and also stores the implementation results in the servo control system database as experience references. The system parameters are then adjusted to obtain better results.

III. VISUAL IMAGE PROCESSING AND IDENTIFICATION

The Machine Vision system includes the computer and other hardware needed to provide a visual function similar to that existing in humans and can realize the 3D location and identification of an object. The system can be used for object detection and recognition, image tracking, Machine navigation, and model building in 3D space. In this paper the operation included obtaining target object coordinate information which was then used to guide the manipulator to the object for pickup. The image recognition system flow is shown in Figure 2.

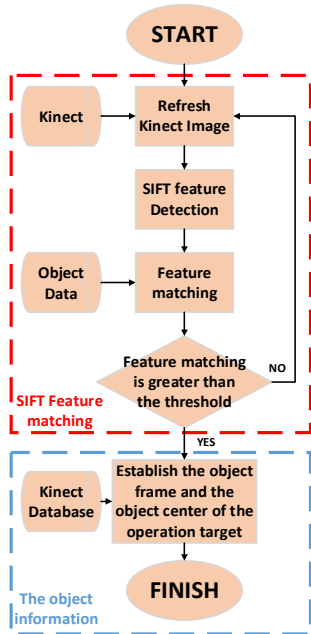


Figure 2. Image recognition system flowchart

A. Scale-Invariant Feature Transform

A.1 Keypoint Extraction

Image Keypoints include edges, corners, areas, ridges, and colors. The detection of Keypoints can be used to find the same scene or object in two different pictures. The most commonly used Keypoint detection algorithms used now are SURF[3] and SIFT[4]. The SIFT algorithm [4] calculation process involves four steps. Figure 3 illustrates the SIFT algorithm procedure.

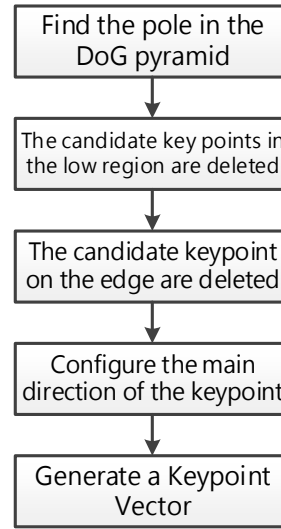


Figure 3. SIFT Flowchart

SIFT is an unchanging Keypoint detection and description transform algorithm used for translation, rotation, expansion, shrinking, and brightness change.

A.2 Difference of Gaussians

Difference of Gaussians (DoG) is a method used in Machine Vision. An original picture is enhanced by fuzzy grayscale from another grayscale picture. The Gaussian function equation is as follows (1):

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

Normally, the original picture is down sampled, and after two down samplings, the images are divided to give three images. Using m number of different standard deviation Gaussian functions to identify different scale, we obtained $m=5$. The Gaussian function standard deviation for these five scales are $\sigma, K\sigma, K^2\sigma, K^3\sigma, K^4\sigma$. We set σ as 2 and K as $\sqrt{2}$. Finally, the Gaussian function was used with different scales and three image scales for convolution, as shown below (2):

$$\begin{aligned} & Cov(x, y, K^t \sigma) \\ &= G(x, y, K^t \sigma) * I(x, y) \end{aligned} \quad (2)$$

$t \in 0, 1, 2, 3, 4$

Next, DoG was conducted on the two neighboring scales, shown below (3):

$$DoG(x, y, K^t \sigma) = (G(x, y, K^t \sigma) - G(x, y, K^{t+1} \sigma)) * I(x, y) \quad (3)$$

Using the DoG pyramid, we compared each pixel in the scale-space with the neighboring area. If the pixel point is a pole, then that point can be set as a Candidate Keypoint, as shown in Figure 4.

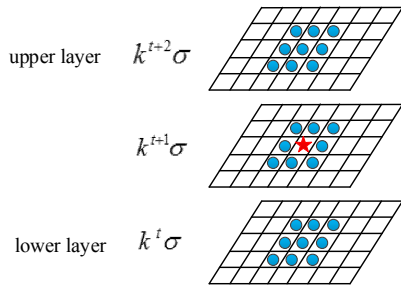


Figure 4. DoG pyramid search for the pole

A.3 Keypoint Main Orientation Distribution

A Keypoint is related to the orientation of its surrounding Description Vector. To achieve rotation resistance, each Keypoint must have a defined main orientation. By rounding up, SIFT calculates the standard deviation based on the center of the Keypoint (with five times the radius of the area) to obtain the Gradient value $m(x, y)$ and Gradient direction $\theta(x, y)$ of all the pixel points in the area. The equation (4) is defined as follows:

$$m(x, y) = \sqrt{[L(x+1, y) - L(x-1, y)]^2 + [L(x, y+1) - L(x, y-1)]^2} \quad (4 a)$$

$$\theta(x, y) = \tan^{-1} \left[\frac{L(x+1, y) - L(x-1, y)}{L(x, y+1) - L(x, y-1)} \right] \quad (4 b)$$

The Gradient direction obtained is used to divide 360 degrees into 36 aliquots of 10 degrees. The Gradient value is multiplied with the pre-set Gaussian function to make it a weight value. The orientation with the highest value is determined in the histogram so that the orientation becomes the main orientation of the Keypoint, as shown in Figures 5 and 6.

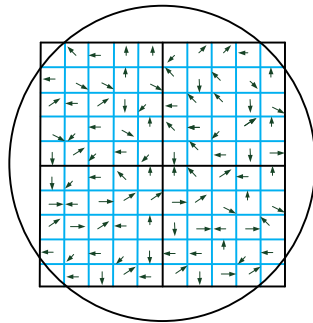


Figure 5. Gradient vector determination of main orientation

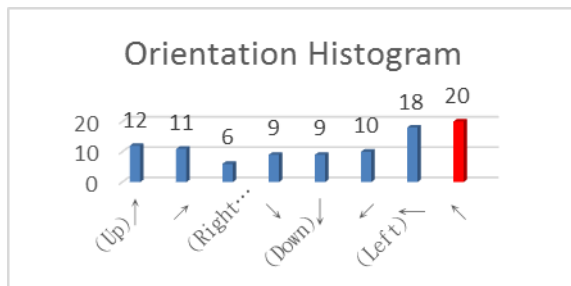


Figure 6. Orientation histogram

A.4 Constructing a Keypoint Vector

After the above steps, each Keypoint will possess three attributes, scale, position, and orientation. Next, each Keypoint must be described and a descriptor established. This will prevent any external changes from affecting the original description. Subsequently, we used the Keypoint as the center to obtain a 16×16 area, and then divided this area into 16 smaller 4×4 areas. After this the Gradient value in the areas was multiplied by the weighted value of the Gaussian function. Finally, the main orientation of each area was rotated clockwise. The total Gradient value of each orientation was recorded during the process, and produced a set of $4 \times 4 \times 8 = 128$ eigenvectors. This gave the rotation-invariant of each Keypoint, as shown in Figures 7 and 8.

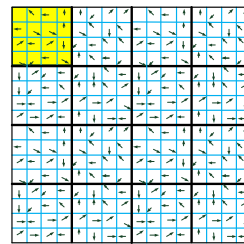


Figure 7. Gradient vector schematic diagram

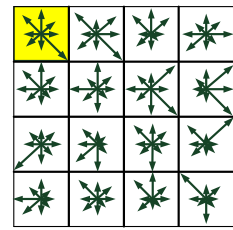


Figure 8. 128 Dimension eigenvector

B. Setting up the Kinect Sensor and the Work Area

The Vision system manipulator combination with the Kinect sensor will be significantly affected by the position of the Kinect sensor. We used the eye-to-hand setup to install the Kinect sensor behind the work area, as shown in Figure 9.

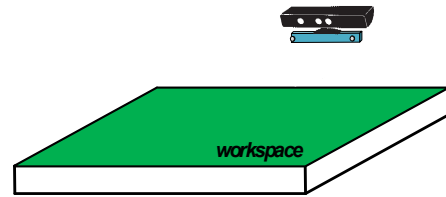


Figure 9. Kinect sensor installed behind the work area

After considering the preset condition, the mechanical limitations of the manipulator, and the Kinect sensor setup, we organized the entire manipulator work area, as shown in Figure 10.

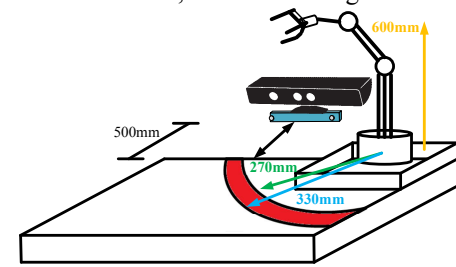


Figure 10. Manipulator workspace arrangement diagram

IV. MACHINE LEARNING ALGORITHM

Machine Learning includes three different types of Learning method, Reinforcement Learning, supervised Learning, and unsupervised Learning. In reinforcement the system learns from the environment and obtains rewards. In this paper a Machine Learning Algorithm in the manipulator was trained to grab the target object through Learning. For this paper, we added a database to the manipulator system to give it a memory function. Figure 11 is an interaction diagram showing the Learning agent and the environment.

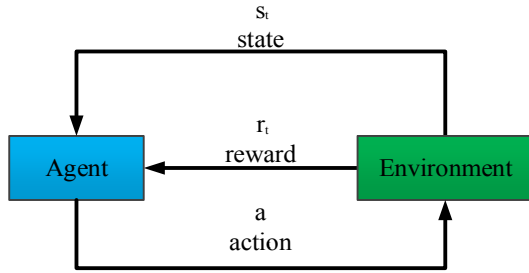


Figure 11. Interaction of the Learning agent with the environment

A. Reinforcement Learning

There are two choices of Reinforcement Learning Algorithm, the model-based and model-free [7, 8] types. Reinforcement Learning is viable, highly efficient and probably the best method for program training.

B. Model-Free Reinforcement Learning

Model-free reinforcement is more complex than model-based Learning. When calculating each value function of a policy in model-free Reinforcement Learning, the MDP reward function and the transition probability are unknown. Also, the policy selection method often employs a Greedy algorithm. However, if an action is selected by Greedy algorithm, another action will have no chance of selection. For this reason, an improved ϵ -greedy method is often used. Model-free Reinforcement Learning Algorithms primarily utilize one of these algorithms: Monte Carlo, state action reward state action (SARSA), and Q Learning.

V. EXPERIMENTAL RESULTS

Experiments were conducted to test and verify the methods used in this paper and the system was implemented on a 6-DOF manipulator. The Machine Vision system used a Kinect sensor which was connected to the main control PC through a USB 3.0 cable. The images collected were subject to Image processing calculation and the results were rendered on the PC, see Figure 12.

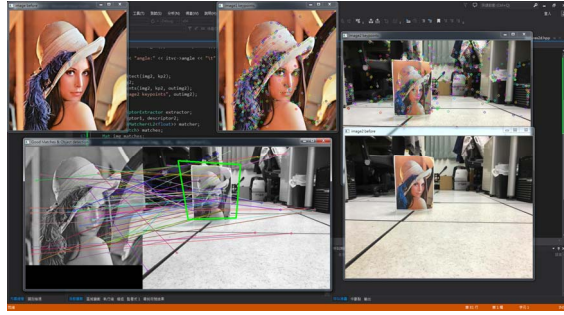


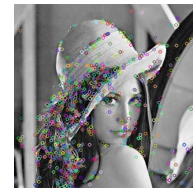
Figure 12. Visual image interface

A. SIFT Feature Detection and Matching Experiments

The image used as the target object in this paper is shown in Figure 13(a). Grayscale processing and SIFT feature detection identified all the SIFT descriptors of the picture, see Figure 13(b). Figures 14(a) and (b) show the Kinect sensor image before and after extraction. The SIFT features are identified by circles with marked radius in which the circles are the Keypoint positions. Feature matching was carried out after image sampling and SIFT feature detection and the SIFT feature matching experiment results are shown in Figure 15.

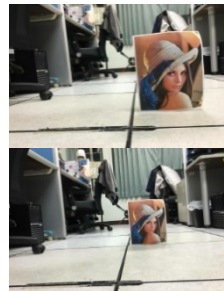


(a) Original picture



(b) SIFT feature detection

Figure 13. The sample picture used.



(a) Original picture



(b) SIFT feature detection

Figure 14. Kinect sensor extraction image

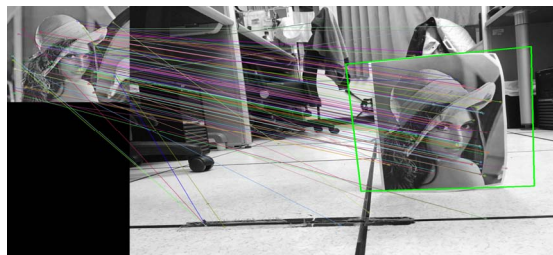


Figure 15. SIFT feature matching results

B. Machine Learning Experiments

The experimental results showed that the model-free testing algorithms could converge normally. Figure 16 shows the statistics after the

experiment parameter had been applied to the manipulator. Q-Learning was chosen as the best of these because it was faster than the others and the errors were smaller. The next best was SARSA followed by Monte Carlo.

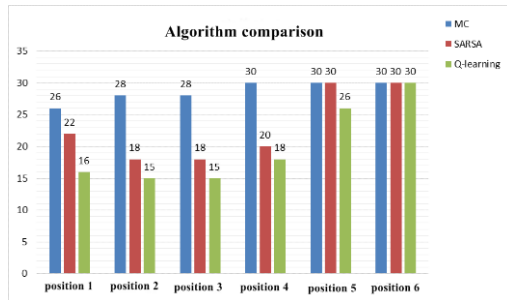


Figure 16. Comparison of Model-free Algorithms

An examination of Figure 16 shows the Q-Learning results to be the best. However, it should be noted that in model-free experiment position six, none of the three algorithms could complete the objective. The reason for this might have been that the error was too large for the algorithms to track to the best value or designated threshold.

C. Manipulator Control Experiments

The experiments showed that by means of image extraction and a Learning algorithm, learned policy parameters could be input to the servo control system. Repeated manipulator control experiments validated the policy and the system used in this paper. Successful grabbing of the target object clearly verified the premise that Reinforcement Learning can be applied to Manipulator, as shown in Figures 17.

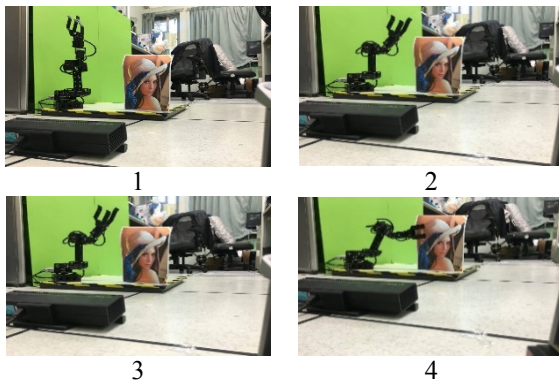


Figure 17. Manipulator target object pickup results

VI. CONCLUSION

We combined Machine Vision with a Machine Learning Algorithm to give Manipulators a sensory

system and Learning ability similar to that found in humans. This system was implemented in three stages. The first part was target object feature detection and matching. We used a Kinect sensor set up behind the manipulator work area. The target object Scale-invariant and Rotational-invariance characteristics, from the image in the work area, were extracted using the SIFT algorithm. Image depth information of the object was also extracted. In the second stage, the manipulator learned how to move to the location of the target object. This was achieved using a Reinforcement Learning Algorithm, a model was constructed and a transfer policy was set. Policy scoring was used to derive multiple sets of motor control parameters. The last stage was servo control. The parameters obtained from part two were input to the servo control system. Repeated attempts and experiments resulted in reproducible movement of the manipulator to the target object location.

ACKNOWLEDGMENT

This work is supported by the National Science Council, Taiwan, Republic of China, under Grand number MOST 105-2221-E-167-026.

REFERENCES

- [1] M. Camplani, T. Mantecon and L. Salgado, "Depth-Color Fusion Strategy for 3-D Scene Modeling With Kinect," *IEEE Transactions on Cybernetics*, Vol. 43, pp. 1560-1571, 2013.
- [2] N. Stefanoski, C. Bal and M. Lang, "Depth Estimation and Depth Enhancement by Diffusion of Depth Features," *2013 IEEE International Conference on Image Processing*, pp. 1247-1251, 2013.
- [3] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features," *European Conference on Computer Vision*, pp. 404-417, 2006.
- [4] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proceedings of IEEE International Conference on Computer Vision*, Vol. 2, pp.1150-1157, 1999.
- [5] M. Guo, Y. Liu and J. Malec, "A new Q-Learning algorithm based on the metropolis criterion," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 34, pp. 2140-2143, 2004.
- [6] V. N. Vapnik, "An Overview of Statistical Learning Theory," *IEEE Transactions on Neural Networks*, Vol. 10, pp. 988-999, 1999.
- [7] Reinforcement Learning, <http://www.algorithmdog.com/series/rl-series>
- [8] N. Tziortziotis and K. Blekas, "A Model Based Reinforcement Learning Approach Using On-Line Clustering," *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, Vol. 1, pp. 712-718, 2012.