



OBJECT RECOGNITION SYSTEM IN REMOTE CONTROLLED WEAPON STATION USING SIFT AND SURF METHODS

Midriem Mirdanies ^a, Ary Setijadi Prihatmanto ^b, Estiko Rijanto ^a

^a Research Center for Electrical Power and Mechatronics, Indonesian Institute of Sciences (LIPI)
Komp LIPI Bandung, Jl. Sangkuriang, Gd. 20. Lt. 2, Bandung 40135, Indonesia

^b School of Electrical Engineering And Informatics, Bandung Institute of Technology (ITB)
Jl. Ganesha No 10, Bandung 40132, Indonesia

Received 8 April 2013; received in revised form 12 November 2013; accepted 13 November 2013

Published online 24 December 2013

Abstract

Object recognition system using computer vision that is implemented on Remote Controlled Weapon Station (RCWS) is discussed. This system will make it easier to identify and shoot targeted object automatically. Algorithm was created to recognize real time multiple objects using two methods i.e. Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) combined with K-Nearest Neighbors (KNN) and Random Sample Consensus (RANSAC) for verification. The algorithm is designed to improve object detection to be more robust and to minimize the processing time required. Objects are registered on the system consisting of the armored personnel carrier, tanks, bus, sedan, big foot, and police jeep. In addition, object selection can use mouse to shoot another object that has not been registered on the system. Kinect™ is used to capture RGB images and to find the coordinates x, y, and z of the object. The programming language used is C with visual studio IDE 2010 and opencv libraries. Object recognition program is divided into three parts: 1) reading image from kinect™ and simulation results, 2) object recognition process, and 3) transfer of the object data to the ballistic computer. Communication between programs is performed using shared memory. The detected object data is sent to the ballistic computer via Local Area Network (LAN) using winsock for ballistic calculation, and then the motor control system moves the direction of the weapon model to the desired object. The experimental results show that the SIFT method is more suitable because more accurate and faster than SURF with the average processing time to detect one object is 430.2 ms, two object is 618.4 ms, three objects is 682.4 ms, and four objects is 756.2 ms. Object recognition program is able to recognize multi-objects and the data of the identified object can be processed by the ballistic computer in realtime.

Keywords: RCWS, object recognition, shared memory, SIFT, SURF, opencv, C language, kinect™.

I. INTRODUCTION

Defense system of a country requires reliable weapon systems [1], one of them is Remote Controlled Weapon Station (RCWS). RCWS is commonly used in modern combat equipment such as Tank or Armoured Personnel Carrier (APC). RCWS is a weapons system that can be operated remotely from vehicle cabin so the gunner is protected safely.

This paper describes the results of the computer vision research which is implemented on RCWS. This system will make it easier to recognize and to shoot targeted objects

automatically, including to shoot multiple objects. Kinect™ is used to capture RGB images and the object coordinates x, y, and z in meters [2] which can be seen in Figure 1. Kinect™ consists of infrared cameras, RGB camera, infrared laser projector, microphone, and tilt motors.

The research was performed at the Mechatronics Lab. - Research Center for Electrical Power and Mechatronics (Indonesian Institute of Sciences), and the School of Electrical Engineering And Informatics Lab. - Bandung Institute of Technology (ITB). RCWS model tested to move the weapon model in the Laboratory. The weapon model can be seen in

* Corresponding Author. Phone: +62-22-2503055
E-mail: midriem.mirdanies@lipi.go.id

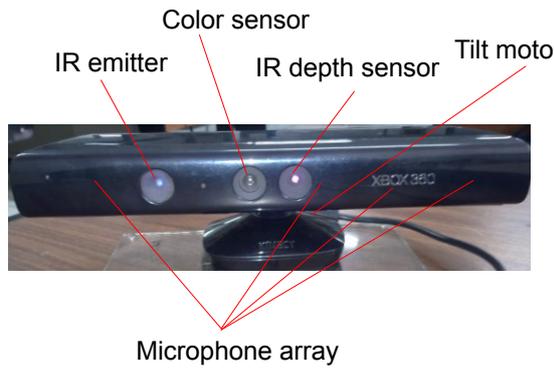


Figure 1. Kinect™ Sensor used in this research



Figure 2. Weapon model

Figure 2. Kinect™ is placed in the front side and separated from the weapon model.

Methods that can be used in tracking and object recognition have been reported by Yilmaz [3]. Methods to recognize objects can be based on color [4], shape, or other. Methods used in this research is Scale Invariant Feature Transform (SIFT) [5], [6] and Speeded Up Robust Features (SURF) [7] combined with the K-Nearest Neighbors (KNN) and Random Sample Consensus (RANSAC) for verification. The programming language used is C with visual studio IDE 2010 and Open CV libraries [8], [9], [10]. Experiment was done to compare the accuracy and process speed of the both methods. An algorithm is developed to detect multiple objects simultaneously, while improving object detection results to be more robust and to minimize the time required for the object recognition. The recognized objects are sent to the ballistic computer through Local Area Network (LAN) using winsock [11] for further processing. Object recognition program is divided into three section: capturing images from kinect™ and simulation results, object recognition process, and object data transfer to the ballistic computer. Communication between programs was performed using shared memory. With shared memory, data on the memory can be accessed by multiple programs [12].

Concept used in SIFT and SURF method is to find interest points, also called keypoint or

feature points. The idea is to find the unique features in the images and perform analysis on that feature.

The advantage of using SIFT and SURF methods than some other methods is scale invariant, i.e. the object that had been detected may have different scale than the comparison object. It also supports object rotation and can detect objects even if partially visible.

There are four major stages in the SIFT method to generate the set of image features: scale-space extrema detection, keypoint localization, orientation assignment, and calculating keypoint descriptor [13]. After doing these stages, features will be obtained which is used as a descriptor of an object to be processed further. In the process, SIFT uses Laplacian filter. Laplacian filter in difference scale is calculated based on Gaussian filter. The stages of SURF almost the same as SIFT, but it uses Hessian determinant filter, not the Laplacian filter. The main difference between the SURF and the SIFT is the speed and accuracy, typically SURF gives faster process, whereas SIFT is more accurate in finding its matching feature [14].

II. REMOTE CONTROLLED WEAPON STATION DESIGN

Figure 3 shows the RCWS design conducted in this research. There are three main parts on the RCWS system, i.e. computer vision, ballistic

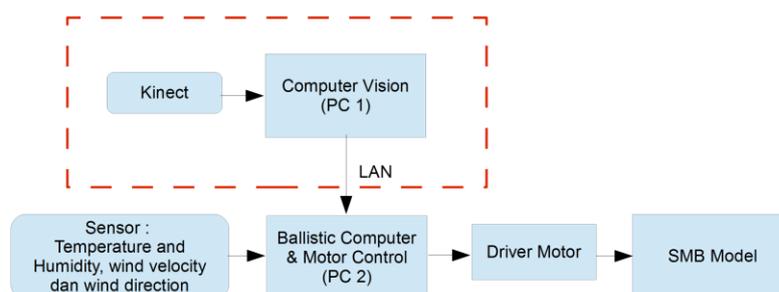


Figure 3. RCWS Design

computer, and motor control. The dotted lines in Figure 3 are area of this research. There are two computers used, the first computer is for computer vision, and the other is for ballistic computer and motor control. Communication between both computers is done via LAN.

Image from kinect is processed by the first computer to recognize object. Objects coordinate that are detected will be sent to the second computer to do ballistic calculation. Ballistic calculation is performed by several parameters, i.e. the object coordinates, projectile, and other data from the temperature sensor, wind velocity and wind direction in order to obtain the correction of shooting angle (azimuth and elevation), then the motor control system will give the command to the motor driver to move the weapon model in order to direct towards a selected object.

III. COMPUTER VISION DESIGN

Two computer programs have been created. The first is used to store data in a database (training data) that can be seen in Figure 4. The second is realtime object recognition process that can be seen in Figure 6 and Figure 7.

The database file was created with ".yml" extension and it is saved separately for each object and method. It is intended to be efficient, so only the required files are used. This would avoid the buildup of data in a single file. The process storage of objects data to the database is done by selecting the desired object using mouse

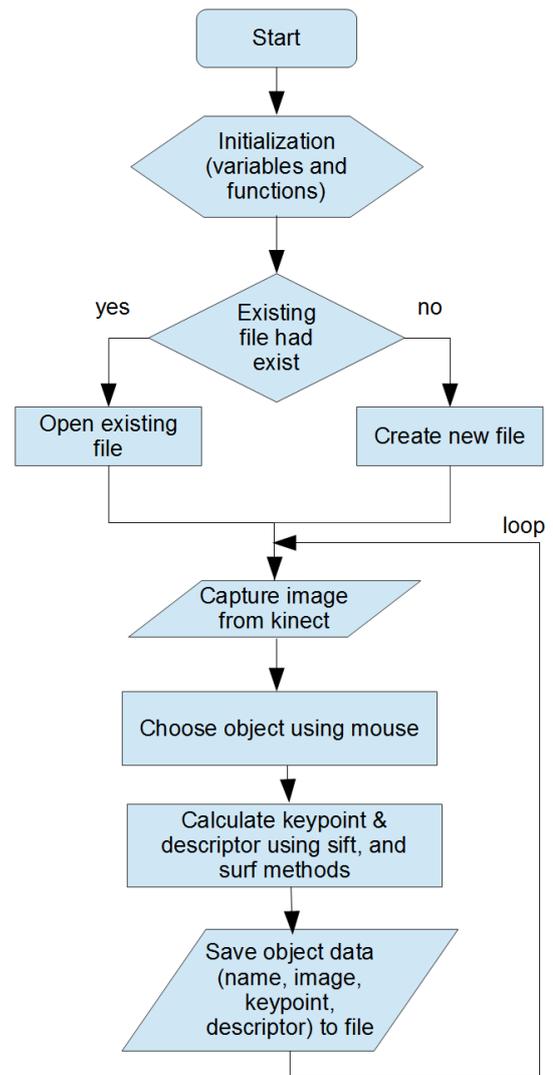


Figure 4. Object data storage process in the database



Figure 5. Objects stored in the database

Table 1.
Object database file

No	File name	Object	Method	The amount of data			
				Object Name	Images	Keypoints	Descriptors
1	SIFT_bus.yml	Bus	SIFT	1	36	36	36
2	SURF_bus.yml	Bus	SURF	1	36	36	36
3	SIFT_mobil_sedan.yml	Sedan	SIFT	1	36	36	36
4	SURF_mobil_sedan.yml	Sedan	SURF	1	36	36	36
5	SIFT_tank.yml	Tank	SIFT	1	38	38	38
6	SURF_tank.yml	Tank	SURF	1	38	38	38
7	SIFT_panser.yml	APC	SIFT	1	38	38	38
8	SURF_panser.yml	APC	SURF	1	38	38	38
9	SIFT_big_foot.yml	Big foot	SIFT	1	48	48	48
10	SURF_big_foot.yml	Big foot	SURF	1	48	48	48
11	SIFT_jeep_polisi.yml	Police jeep	SIFT	1	49	49	49
12	SURF_jeep_polisi.yml	Police jeep	SURF	1	49	49	49

so it is user friendly. Objects used consist of toys APC, Tank, Bus, Sedan, Big foot, and Police jeep. Figure 5 shows the objects, while the database file object is listed in Table 1.

Object was trained from several sides. Each file contains object name, images, key point, and the object descriptor. All data is stored in the database so there is no need to look for key point and descriptor again during the process of object recognition. This method will speed up the program execution.

Figure 6 shows real time objects identification flowchart. In the initial phase, all databases are loaded into memory, to speed up the execution time. The experiment result shows that the speed of the object identification process is faster than if it is made directly to the database. The next step is to compare all the data in the database that has been loaded into memory with the image data from kinect™ to find a match object. This process is based on an algorithm created by Robert Lagainere [14].

The experimental results show that the objects identification becomes more robust. However, this study has done some modifications, such as addition of data checking i.e. keypoint checking before matching process, symMatches before RANSACTest, outMatches before the conversion of keypoint to point process, and point before the findFundamentalMat. The experimental result shows that realtime process will generate error if the data input does not exist or insufficient extent. The algorithm was also modified to be able to choose the method used (SIFT or SURF method) by changing the parameter setJenisFeature.

In addition, the searching process of the keypoint and descriptor for each compared image

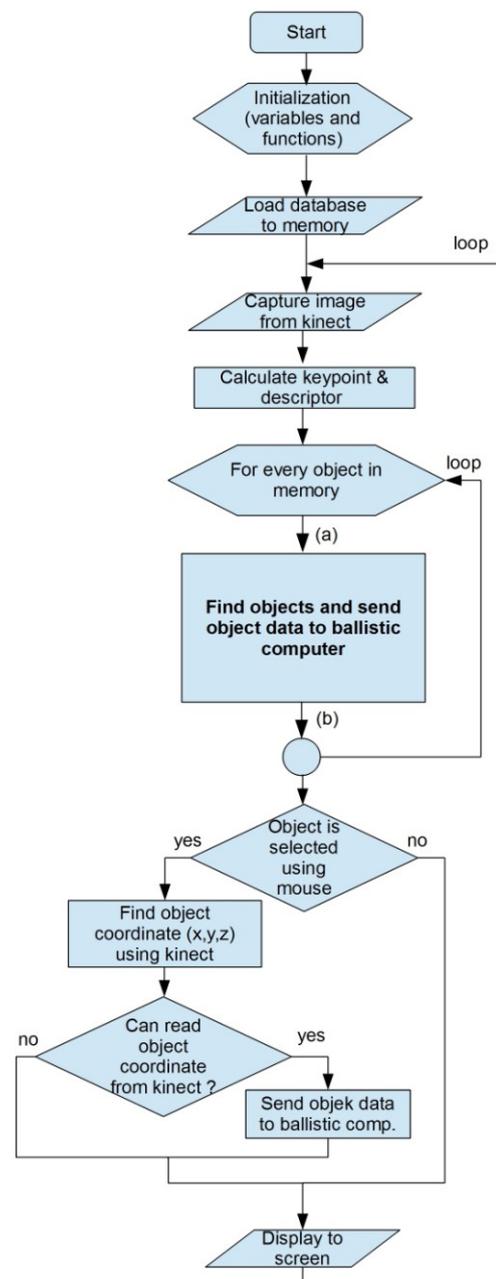


Figure 6. Real time object recognition flowchart

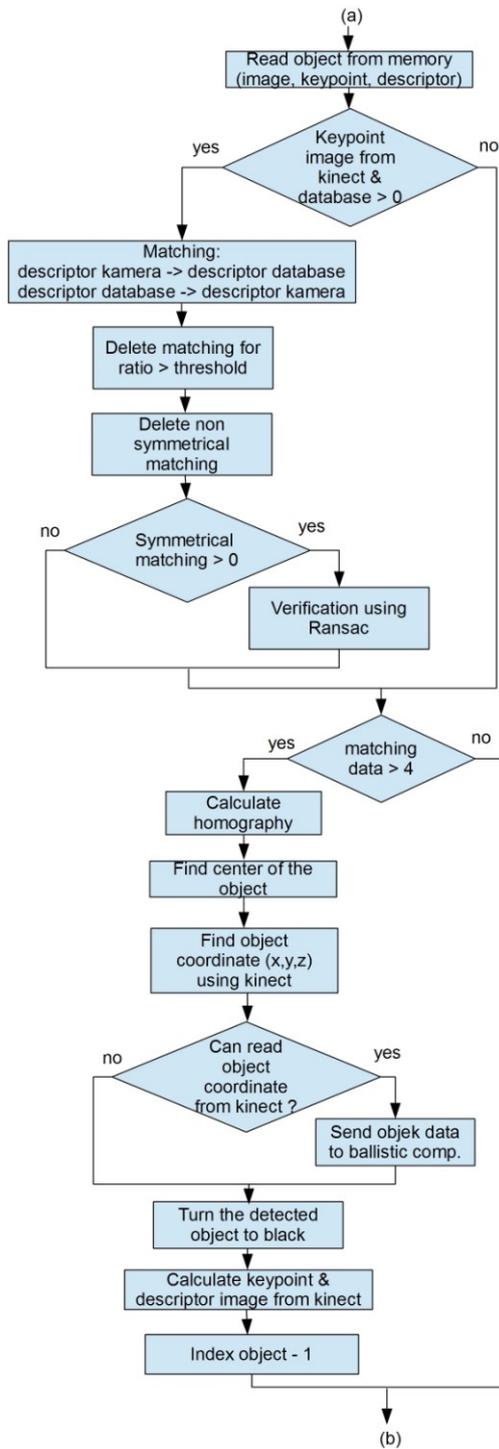


Figure 7. Find objects and send object data to ballistic computer

is not performed continuously, but only as needed, i.e. the image from the kinect™ is done only at the beginning when image from kinect™ will be compared with the database and if there is a marking of detected objects for every image. As a result, it is not necessary for the image to search a key point and descriptor again, because it's already stored in the database. This way will speed up the program execution process. Section of finding objects and sending data objects to ballistic computer in Figure 6 consists of matching process, calculating homography,

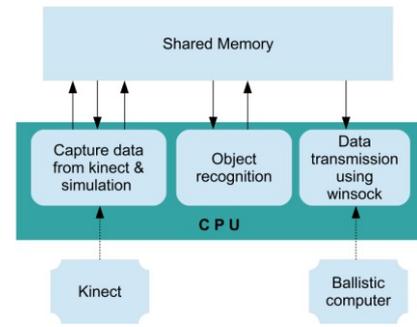


Figure 8. Distribution of object identification programs with shared memory mechanism

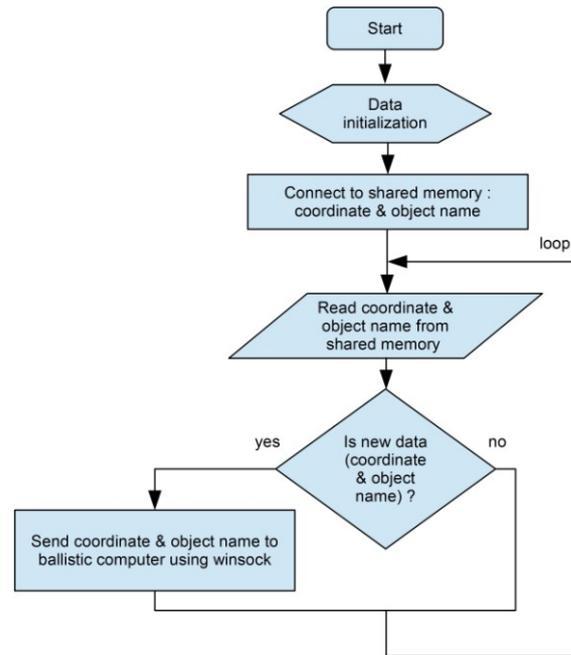


Figure 9. Distribution of object identification programs with shared memory mechanism

finding object coordinate and sending to ballistic computer. Detail process of finding objects and sending data objects to ballistic computer can be seen in Figure 7.

Realtime object recognition program is divided into three parts and interact each other using shared memory as shown in Figure 8. Three parts of data are shared between programs: captured image from kinect™, object identified object data (name and center of the object), and a complete objects data (name and coordinate x, y, z in meter). Handle file mappings used for each data are “Global\\MatMappingObject”, “Global\\ObjekMappingObject”, and “Global\\DtLngkpMappingObject”.

Figure 9 shows the flowchart of the delivery of the detected object data to the ballistic computer flowchart, Figure 10 shows the object recognition flowchart, and Figure 11 shows flowchart of the real time images captures from kinect™ and the detected object display to the screen for simulation.

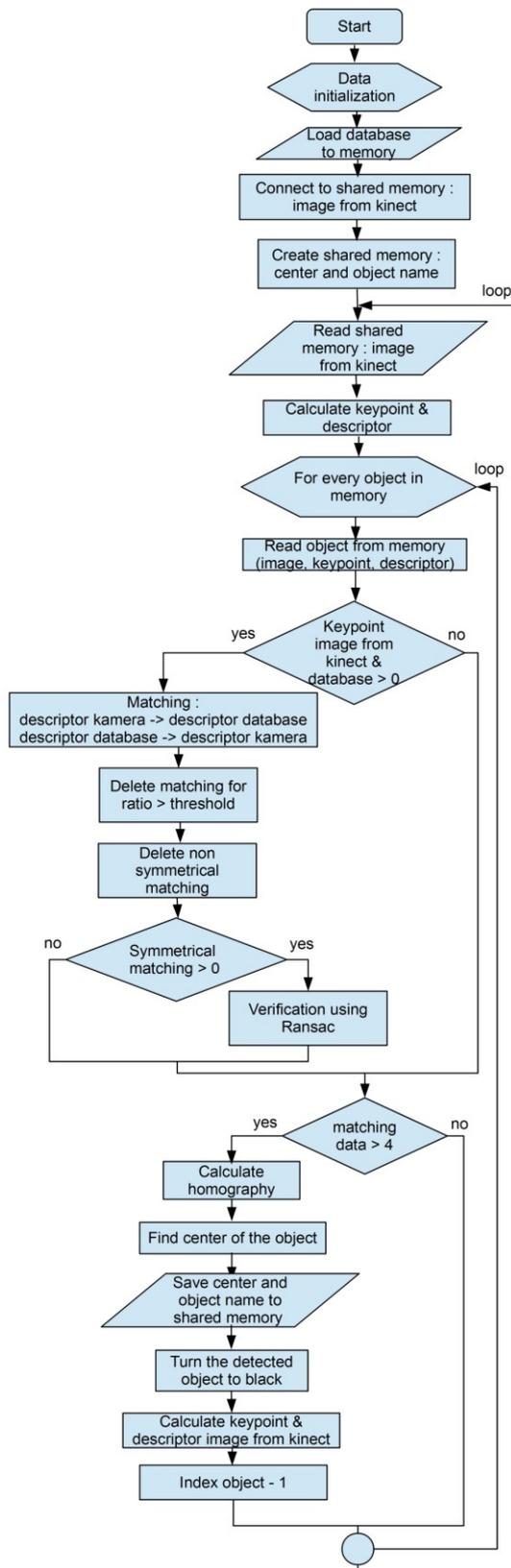


Figure 10. Object recognition flowchart

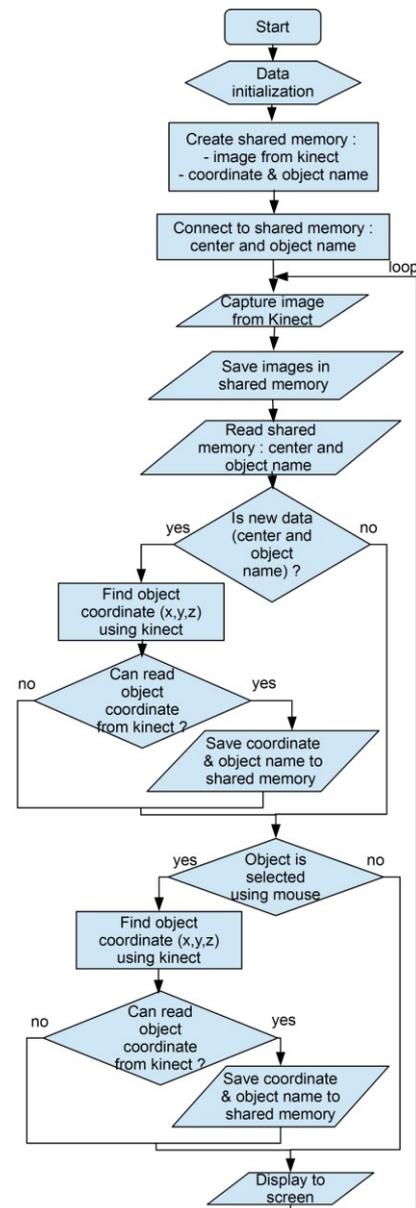


Figure 11. Flowchart of the image capturing process from kinect™ and simulation results



Figure 12. Laptops used in experiments

IV. RESULT AND DISCUSSION

The computer used in this experiment is a laptop with the following specifications: processors Intel® Core™ i5-2450M CPU @ 2.50 GHz, 8 GB RAM, Microsoft Windows 7 64 bit, and Nvidia Geforce 410M, which can be seen in Figure 12.

A. Object Recognition Algorithm

Figure 13 is an experiment result displaying the detection of an APC. Name and coordinates x , y , and z of the detected object is displayed on the screen and command prompt. The program execution time is displayed in the command prompt.

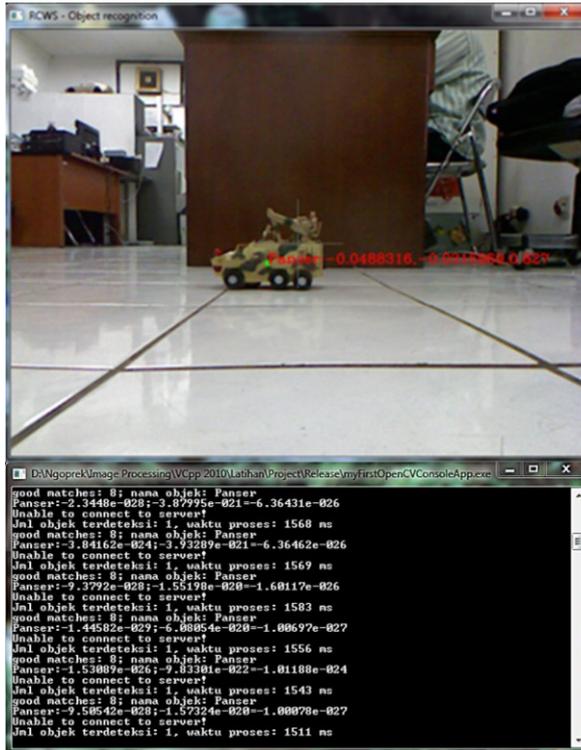


Figure 13. Experiment to detect APC

Experiments were performed to detect a tank and APC objects in the Lab, ten times for each object from several positions. Experiments to test speed of the three methods were done by calculating the process time for every iteration from the key point calculation from kinect™ to the completion of comparisons between the data and the database. Table 2 lists experiment results concerning accuracy and processing time of the three methods.

From Table 2 can be seen that the SIFT method provides better accuracy and faster processing time. The SIFT method can detect 13

Table 2.

Experiment results concerning accuracy and processing time using SIFT and SURF methods

No	Object name	Detection		Process time (ms)	
		SIFT	SURF	SIFT	SURF
1	Panser	√	√	1,511	2,329
2	Panser	√	√	1,489	2,505
3	Panser	√	-	1,526	-
4	Panser	√	√	1,519	2,634
5	Panser	√	√	1,508	2,554
6	Panser	-	-	-	-
7	Panser	-	-	-	-
8	Panser	√	-	1,482	-
9	Panser	√	-	1,521	-
10	Panser	√	-	1,583	-
11	Tank	√	√	1,511	2,876
12	Tank	-	-	-	-
13	Tank	-	-	-	-
14	Tank	√	√	1,531	2,825
15	Tank	√	√	1,541	2,829
16	Tank	√	√	1,525	2,787
17	Tank	-	-	-	-
18	Tank	-	-	-	-
19	Tank	-	-	-	-
20	Tank	√	-	1,502	-

times in 20 experiments (65%), with an average processing time of 1519.2 ms. The SURF method can detect 8 times (50%), with an average processing time of 2667.4 ms. Based on experiments it can be seen that the less feature point (uniqueness) in an image and the ruination of lighting, the smaller the possibility of the object can be detected.

It was also found that the process time of the SIFT method being faster than the SURF method

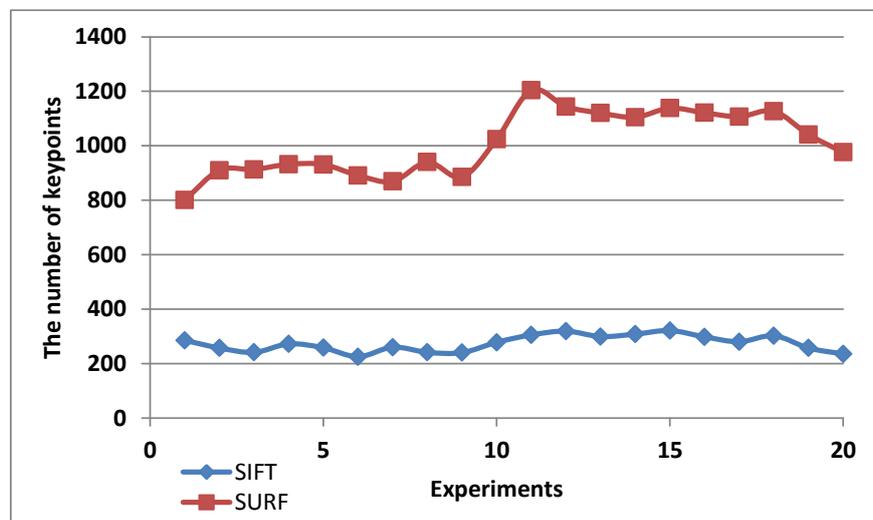


Figure 14. The number of keypoints using SIFT and SURF methods

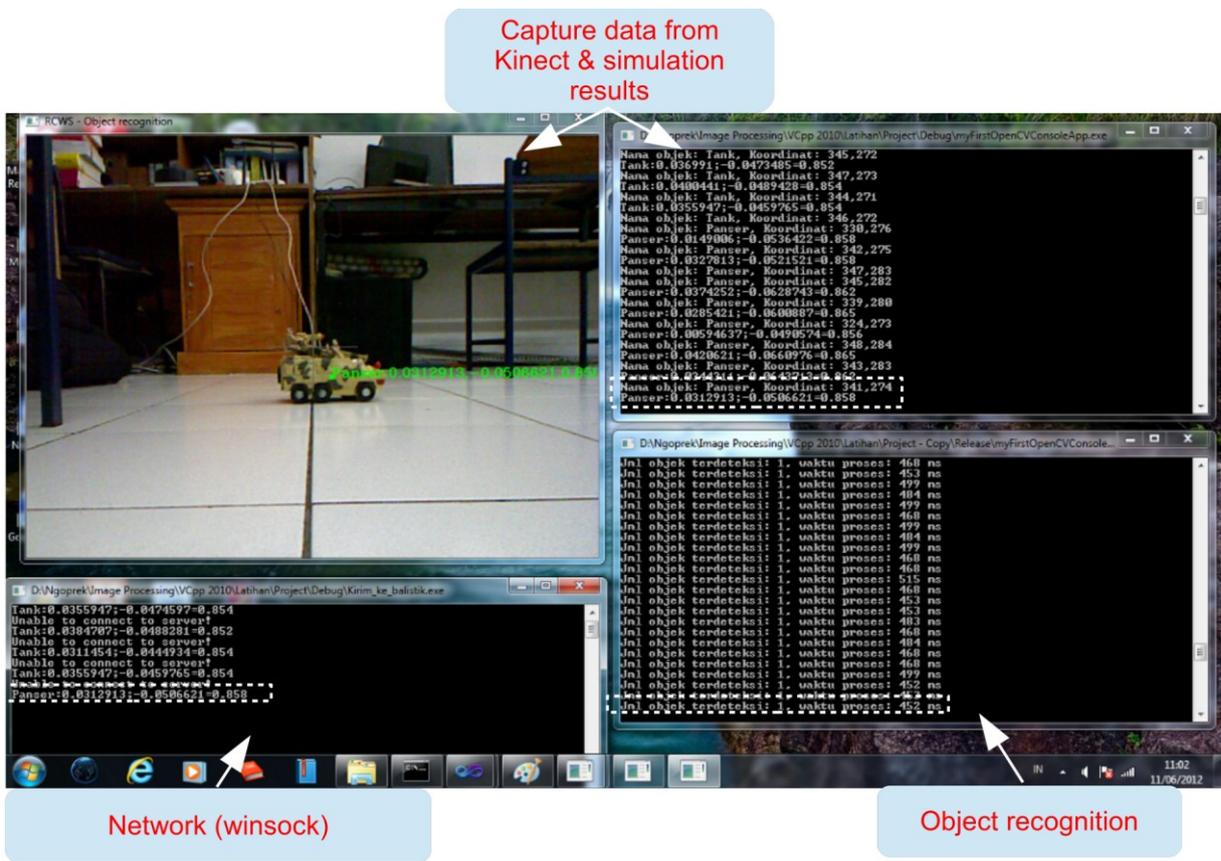


Figure 15. Communication experiment using shared memory mechanism

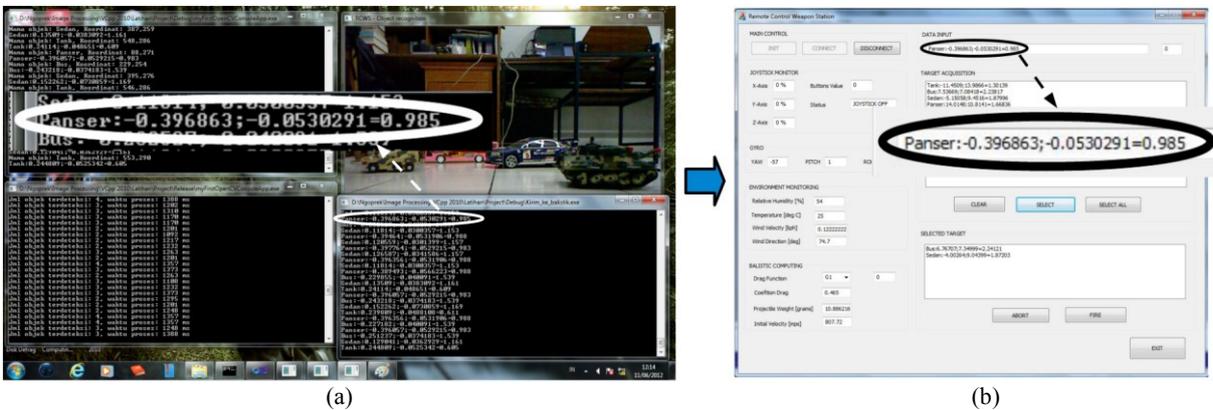


Figure 16. Data transmission experiments via LAN; (a) computer vision; (b) ballistic computer and motor control

because the number of key point in the SIFT method is less than the SURF method. Figure 14 shows the comparison of number of key point using SIFT and SURF method. Twenty times of the experiments results show that the average number of SIFT key point is 274.1, while SURF is 1008.9.

B. Shared Memory

Figure 15 shows experiment result of the communication between the three programs using shared memory. In Figure 15 it can be seen that the three programs can communicate each other

using shared memory. It can be seen from the data congruence displayed on the command prompt of each program with the data display on the screen.

Experiments to see a comparison of the process time before and after the program being divided into three parts using the SIFT method can be seen in Table 3. The object recognition process was calculated since the image from kinect™ was loaded from the shared memory until the whole process of comparisons between the data from kinect™ and the database is complete. The experiments results proved that

Table 3.

Experiments to see object recognition process time after program was divided into three parts

No	Object	Detected (using SIFT method)	Processing time (ms)
1	Panser	√	418
2	Panser	√	407
3	Panser	√	430
4	Panser	√	405
5	Panser	√	409
6	Panser	-	-
7	Panser	-	-
8	Panser	√	387
9	Panser	√	409
10	Panser	√	468
11	Tank	√	453
12	Tank	-	-
13	Tank	-	-
14	Tank	√	461
15	Tank	√	481
16	Tank	√	461
17	Tank	-	-
18	Tank	-	-
19	Tank	-	-
20	Tank	√	404

object recognition process become faster than before, with an average processing time is 430.2 ms. Experiment to see object recognition time in the case of two to four objects at a time can be seen in Table 4.

Table 4 indicates that the average time required to detect two objects is 618.4 ms, three objects is 682.4 ms and four objects is 756.2 ms. Figure 16 shows flow of data sent from computer vision to ballistic computer or servo motor control system. In Figure 16 it can be seen that data from the computer vision was received

Table 4.

Experiments to see object recognition process time to detect two to four objects at once

No	Object Number	Processing time (ms)
1	2	637
2	2	619
3	2	634
4	2	605
5	2	597
6	3	677
7	3	688
8	3	683
9	3	701
10	3	663
11	4	748
12	4	755
13	4	759
14	4	765
15	4	754

correctly by the ballistic computer that is Panser: -0.396863 ; $-0.0530291 = 0.985$. Experiments to examine the performance of RCWS in the case of moving weapon model towards a selected object can be seen in Figure 17. Experiments results show that the weapon model can move towards a selected object, which is proven by the laser pointer direction that leads to the object.

V. CONCLUSIONS

Object recognition using SIFT and SURF methods have been successfully implemented on RCWS. The object recognition algorithms can detect multiple objects simultaneously. Experiments results show that the SIFT method gives better performance than the SURF method. Process time can be minimized by optimizing the algorithm, and dividing the program into three



Figure 17. RCWS experiments in the case of moving weapon model

parts, each part runs independently and communicates with each other using shared memory. The average processing time to detect an object is 430.2 ms, two object is 618.4 ms, three objects is 682.4 ms, and four objects is 756.2 ms. The data of the identified object can be processed by the ballistic computer in realtime manner and the weapon model can move towards the desired object. Further research may be done by increasing training data number for each object to obtain more robust object recognition.

ACKNOWLEDGEMENT

Authors would like to thank to the Research Center for Electrical Power and Mechatronics - Indonesian Institute of Sciences, School of Electrical Engineering And Informatics - Bandung Institute of Technology, and the Ministry of Research and Technology for the opportunity and financial support, also to Iwan Muhammad Erwin, Adi Yasri Bahri, Riyo Wardoyo, and all those who have helped conducting this research.

REFERENCES

- [1] M. Mirdanies, *et al.*, *Kajian Kebijakan Alutsista Pertahanan dan Keamanan Republik Indonesia*. Jakarta: LIPI Press, 2013.
- [2] N. V. Bagwe, "A study of the Microsoft's Kinect camera," Indian Institute of Technology, Bombay 2012.
- [3] A. Yilmaz, *et al.*, "Object Tracking: A Survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.
- [4] A. Djajadi, *et al.*, "A Model Vision of Sorting System Application using Robotic Manipulator," *Telkomnika*, vol. 8, no. 2, pp. 137-148, 2010.
- [5] D. L. G. "Object Recognition from Local Scale-Invariant Features," in *International Conference on Computer Vision*, Corfu, 1999.
- [6] G. Bradski and A. Kaehler, "Learning OpenCV," M. Loukides, Ed., ed. Sebastopol: O'Reilly Media, Inc., 2008.
- [7] H. Bay, *et al.*, "SURF: Speeded Up Robust Features," *Computer Vision - ECCV*, vol. 3951, pp. 404-417, 2006.
- [8] I. Corp., "The OpenCV Reference Manual (Release 2.4.2)," ed, 2012.
- [9] I. Corp., "The OpenCV Tutorials (Release 2.4.2)," ed, 2012.
- [10] I. Corp., "The Open CV User Guide," ed, 2012.
- [11] Microsoft. Using Winsock (Windows) [Online]. Available: <http://msdn.microsoft.com/en-us/library/windows/desktop/ms740632%28v=vs.85%29.aspx>
- [12] Microsoft. Creating Named Shared Memory [Online]. Available: <http://msdn.microsoft.com/en-us/library/windows/desktop/aa366551%28v=vs.85%29.aspx>
- [13] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
- [14] R. Laganiere, *OpenCV 2 Computer Vision Application Programming Cookbook*. Birmingham: Packt Publishing, 2011.