3D Hand Recognition for Telerobotics

Saad Hafiane, Yasir Salih, Aamir S. Malik
Centre for Intelligent Signal and Imaging Research
Universiti Teknologi PETRONAS
31750 Tronoh, Perak, Malaysia

***Abstract*—This paper presents a system for recognition of 3D hand gestures for the purpose controlling and manipulating robots. This objective of the work is to allow the robot to mimic or imitate the recognized gesture which can be used for remote manipulation of a robotic arm to perform complex task (teleoperatoin). Telerobotics systems rely on computer vision to create the human-machine interface. In this project, hand tracking was used as an intuitive control interface because it represents a natural interaction medium. The system tracks the hand of the operator and the gesture it represents, and relays the appropriate signal to the robot to perform the respective action in real time. The study focuses on two gestures, open hand, and closed hand, as the NAO robot is not equipped with a dexterous hand. SURF features points have been used to represent the hand gesture and face to hand distance was used to gauge the depth of the hand. This system has been test with Aldebaran NAO robot for performing different gesture imitation task for picking and placing objects.**

Keywords-2D hand gestures; telerobotics; NAO robot; human-machine interface; SURF

I. INTRODUCTION

In robotics, the subdomain that covers using a mainly wireless connection to remote control a robot is called telerobotics. [1]. It is the marriage of two disciplines, teleoperation and telepresence [2].

Teleoperation aims to give the control of a robot to an operator from a distance in cases where the physical presence of the person is cumbersome or hazardous and creating an autonomous robotic system intricate. In order for teleoperation to be a viable option, it has to offer a level of manipulation close to the intuitive human hand motion which hand gesture recognition offers.

A. Gesture Recognition

In order to recognize a gesture the shape of the hand and the hand itself must be detected in the frame, to do so multiple object recognition algorithms exist. These algorithms generally fall in the following categories:

- Appearance-based methods: Use example images (called templates or exemplars) of the objects to perform recognition
- Feature-based methods: Interesting features in the object are extracted and described, and then a search is used to find feasible matches between object features and image features.

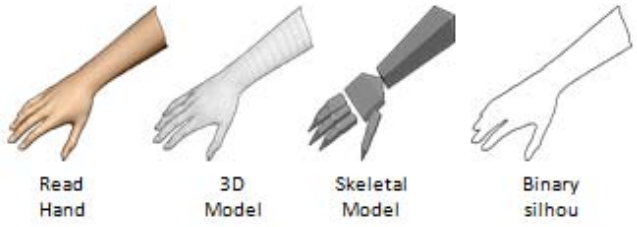

Figure 1. Models of hand

In this study the SURF (Speeded-Up Robust Features) feature extractor was used, combined with 2 nearest neighbor (2NN) search was used to match the hand gestures in the camera frames [SURF]. The gestures were stored in templates and loaded during the initialization of the system.

B. Robot Control

The control of the NAO robot is facilitated by the Naoqi library, in this project we are mainly focused on controlling the arm and grip of NAO. This can be done in manual mode by sending the updated joints angles to NAO. This method is very low level and does not take into account the balancing and the physics involved.

Another way to implement the motion control of NAO, is by enabling the whole body effector control, which controls the whole body to balance it and be able to perform the action needed. During experimentation, an initial single threaded program was developed, where the match was found and the motion was sent to the robot to be executed. Due to the Blocking nature of the whole body effector control of NAO, the performance dropped considerably and the frame rate dropped to 0.4fps. This was caused by the fact that the program waits for the motion to finish before processing the new frame.

To solve this problem a new multithreaded version was implemented where the gesture recognition and the robot telecontrol were implemented in separate threads that communicate through queues. The results were improved as the frame rate returned to normal; however the update speed of the robot motion was still slow, as the robot has to finish the motion before being able to execute a new one, even if the motion that is being performed is obsolete. To counter this problem, the whole body effector control of NAO was be replaced by a non-blocking and will therefore remove the lag created by the whole body effector control of NAO [4].

978-1-4799-0210-1/13/$31.00 ©2013 IEEE

II. LITERATURE REVIEW

Telemanipulation, is the combination of two words tele which means remote, and manipulation. Therefore telemanipulation can be defined as teleoperation where a human operator remotely controls a mechanism manually, in order to manipulated the environment where the robot is present [5]. In recent years, many researchers have focused on creating telerobotics system using robots equipped with dexterous robotic hands in order to allow the operator to manipulate the remote environment intuitively. Different strategies were followed in order to control the robotic arm, as some systems used joysticks or space ball to operate them [5], when others used trackers fixed to wrist of the operator, in order to map the motion of the arm of the operator to the robotic arm [6]. Computer vision was used in other telerobotics systems to allow the operator to manipulate the robot without requiring a physical tool [3]. Often the hand gesture is mapped from the operators hand to the robot using a dataglove, as it is easy and robust [7].

III. METHODOLOGY

The flowchart below (Figure2.) shows detailed description of the proposed algorithm.

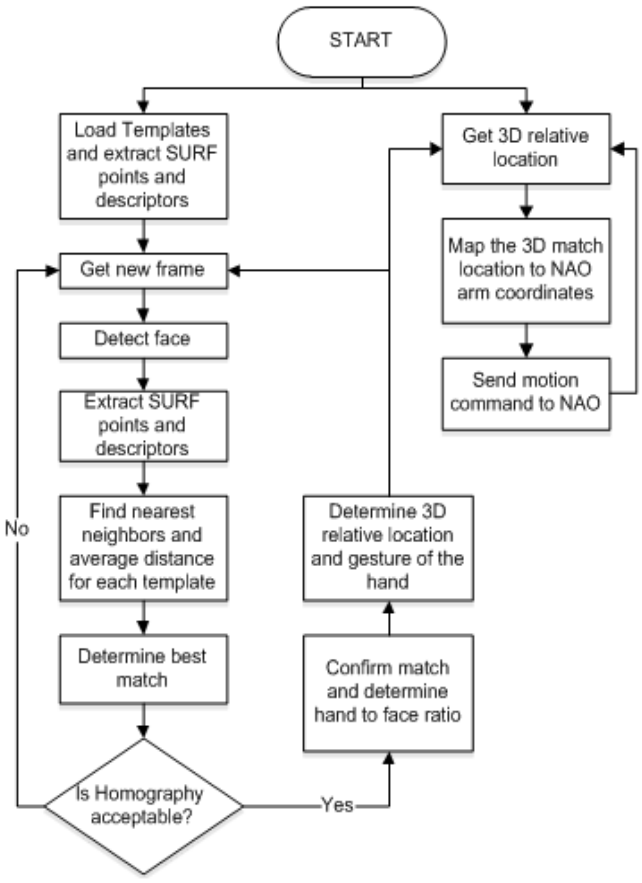

Figure 2. Flowchart for the geasutre recognition and robot control system

The system consists of two threads running in parallel; the first thread runs the computer vision program while the second controls the NAO robot. The computer vision thread starts by loading the hand gestures' templates from the respective folders. The templates are easily created a program written specifically for that purpose, which automates capturing, creating, configuring, and managing the gestures' templates database. Once the templates are loaded, the extraction of the SURF features of each template begins, and all SURF features are categorized by the gesture they indicate.

This concludes the initialization of the computer vision thread and initiates the recognition routine. The system grabs a frame from the video camera, and then searches for a face inside it by using a frontal face Haar classifier. If a match is found, its size and location on the frame is stored. Next the SURF features of the frame are extracted and stored. The hessian threshold for the SURF extraction is selected after testing a range of values, as will be shown in tests results.

A. Speeded-UP Robust Features (SURF)

SURF (Speeded-Up Robust Features) [8] is a scale- and rotation-invariant detector and descriptor. To extract the points of interest, the method relies on integral images in the summation for image convolution in order to approximate the determinant of hessian, in addition to 2D Haar wavelets. For describing the interest points, SURF uses the distribution of the intensity neighboring the point [9]. The OpenCV implementation of the SURF algorithm was used in this system.

The hessian value represents the local curvature of the image intensity, which describes the strength of the feature extracted and therefore is a good indicator on its robustness. The surf extraction was tested at different hessian thresholds, 50, 100, 200, 300, 500, and 1000. In other words the higher the hessian threshold the fewer and the stronger the features are. The tests showed a direct relation between the hessian threshold and the frame rate achieved by the system.

Due to the fact that extraction of the SURF feature descriptors from the templates is only done during the initialization of the system, a hessian threshold of 100, with an octave value of 2 (with each next octave the feature size is doubled), and using 2 layers per octave was chosen, in order to extract a larger number of points. On the other hand for the extraction of the surf points from the camera frames, a hessian threshold of 300 was selected as a compromise between the number of points extracted and the time needed for extraction.

B. Homography Extraction

1) Nearest Neghbor Matching

In order to determine the location of the match between the gesture template and the frame, a 2NN (2 nearest neighbor) search was used determine the two closest points to each point of the gestures templates. The FLANN library was used to perform the nearest neighbor search, and compute the respective distances [10]. During the 2NN search, all points with different Laplacians were ignored in order to speed up the search. Using the 2NN points, the nearest neighbor was confirmed as a plausible match if the distance from it is smaller than 60% of the distance from the second nearest neighbor. The value of 60% was determined by trial and error. The matched gesture is determined by the least average distance of the confirmed points.

2) Computing Homography

Relying on the matched points using the nearest neighbor search, the RANSAC (RANdom SAmple Consensus) algorithm is used to remove the erroneous matches while computing the 3 by 3 homography matrix H that describes the affine transformation between the gesture's template and the match in the frame [18].

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H \times \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} \qquad (1)$$

3) Homography Testing

Once the homography between the template points and the frame points is computed, four points that represent a square of length 100 are projected using the homography matrix. Due to the fact that the appearance of the hand gesture changes significantly with its orientation, due to shadows and luminance variance, a practical match orientation range of ±30º was found to be acceptable in this situation. Based on that, the homography is tested by projecting four points that represent a square of side length 100, and testing the projected square. An acceptable transformation is expected to keep the general form of the square; hence, the acceptable maximum ratio between the lengths of opposite sides of the projected square was set to 3.Furthermore the acceptable angle of rotation of the square was limited to the range ±30º. Finally if any criteria is not satisfied the homography is not accepted, and no match is declared to be found.

4) Relative 2D Location

Using the confirmed homography, the center point of the matched gesture template is projected over the frame, giving therefore the absolute location of the match. In order to send a location that will be easily translated to a motion that uses the full range of motion of NAO, a normalized location is computed and then sent.

$$x_{norm} = \frac{x}{x_{max}} \qquad (2)$$

$$y_{norm} = \frac{y}{y_{max}} \qquad (3)$$

C. Depth Extraction

Using a 2D camera, the distance between an object and the camera can be easily determined if the size of the object and characteristics of the camera are available. However in our system the location of the hand relatively to the camera is invaluable. In order to extract the 3rd spatial coordinate of the hand, the location of the hand must be determined relatively to the body of the operator.

1) Face Detection

To determine the location of the hand relatively to the body, the location of the later must be determined. In this system the face was chosen as an approximated reference to the body. In order to detect the face in the frame, Haar classifiers were used. We relied on the "haarcascade_frontalface_alt.xml" Haar cascade provided with the OpenCV library, to detect the frontal side of the face of the operator. Once the face is detected, its size is saved for later use.

2) Hand to Face Distance

In order to determine the hand to face distance, some approximations and assumptions have to be made. Numerous studies were conducted in the field of human body proportions, for centuries artists relied on approximation and calculated proportions, to describe the human body. One of them is the fact that the actual length of the hand can be approximated with the length of the face. Figure 3 shows the size difference of the hand and face in the frame due to the distance between them relatively to the camera lens. The distance between the hand and the face can be calculated as follows:

$$\tan(\theta) = \frac{h}{D-d} = \frac{\acute{h}}{D} \quad \rightarrow \quad d = D\left(1 - \frac{h}{\acute{h}}\right) \qquad (4)$$

In the above equation, (θ) is the angle of view of the camera occupied by the hand. (h) is the actual height of the hand. $(\acute{h})$ is the apparent eight of the hand, (D) is the distance between the face and the camera while (d) is the distance between hand and the face.

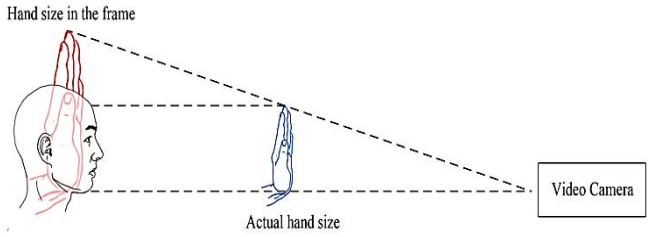

Figure 3. Optical projection of the hand

Eq. (4) describes the absolute distance from hand to face, however in our system, the ratio of the hand to face distance to the maximum hand to face distance, in order to use the full range of motion of NAO even if the operator's size changes. By setting the operator at twice his arms' length from the camera, the normalized hand to face distance (depth) becomes as shown in Eq. (5).

$$D = 2d_{max} \quad \rightarrow \quad depth = \frac{d}{d_{max}} = 2\left(1 - \frac{h}{\acute{h}}\right) \qquad (5)$$

IV. Results and Analysis

In order to test the performance of the system created, two aspects were taken into consideration. First the objective performance of the system, which consist of the gathering the data about the system's frame rate performance, positive matches rate, the robotic system's lag compared to the motion of the operator and others. Finally the subjective performance of the system was tested which consists of the perceived responsiveness of the system, the ease of operation, the learning curve faced while using the system for the first time, and how intuitive is it to perform a practical task. Tests were conducted on two fronts:

a. Synthetic tests to get the processing performance of the system.
b. Task oriented tests where the robot had to successfully perform a practical task.

A. Synthetic Tests

The results were obtained using a Core2 Quad 2.67GHz Intel processor, running OpenCV2.4.2 on Windows 7 64bits. TABLE I shows the frame rates achievable by the system for different hessian thresholds. It can be clearly seen that the frame rate increase with the increase of the hessian threshold.

TABLE I. FRAME RATE PERFOMANCE

Algorithm	Resolution	Hessian threshold	Frame rate (fps)		
			Open Hand	*Closed Hand*	*Average*
SURF	VGA @30fps	300	9.17	9.08	9.125
		500	13.35	13.20	13.275

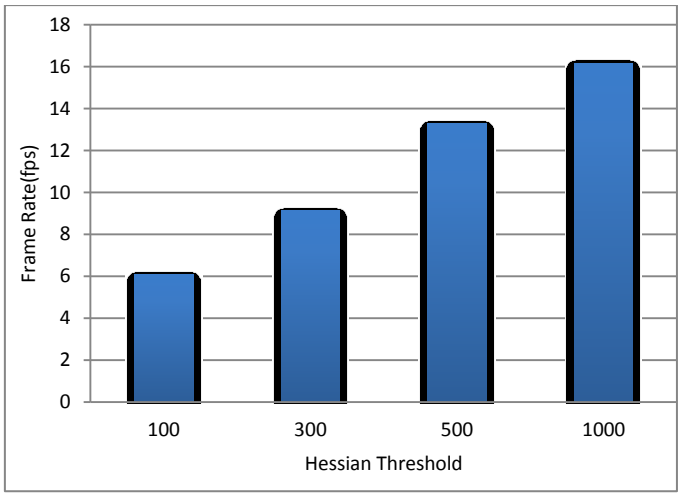

Figure 4. Effect of the Hessian threshod on the frame rate perfomance.

TABLE II shows the repeatability rates achievable by the system for different hessian thresholds. It can be clearly seen that the repeatability rate decrease with the increase of the hessian threshold. Is this experiment, repeatability refers to the ability to repeat the procedure of extraction and matching on different frames and still get the same results, in other words it represents the robustness of the system.

TABLE II. MATCHING REPEATABLILITY

Gesture	Hessian threshold	Number of frames with positive match	Repeatability (%)	Average frame rate (fps)
Open hand	100	725	72.5	6.15
	300	678	67.8	9.17
	500	586	58.6	13.35
	1000	385	38.5	16.10
Closed hand	100	705	70.5	6.05
	300	667	66.7	9.08
	500	529	52.9	13.20
	1000	332	33.2	16.24

*Tests conducted at VGA@30fps over 1000 frames

TABLE III. REPEATABILITY RESULTS SUMMARY

Hessian threshold	Average Repeatability (%)
100	71.5
300	67.25
500	55.75
1000	35.85

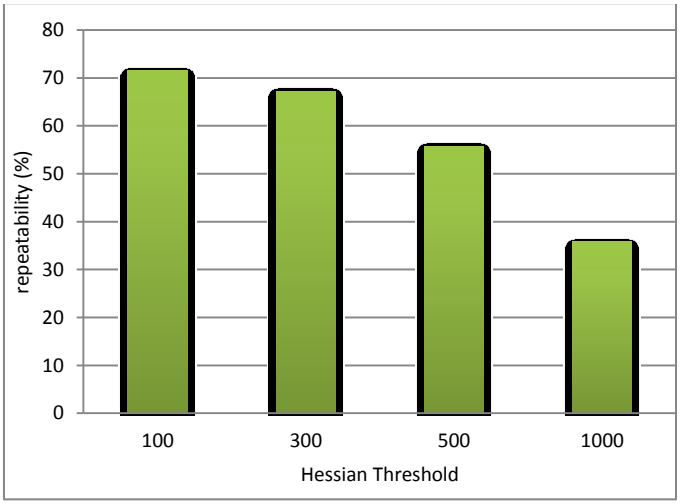

Figure 5. Effect of Hessian threshold on the matching repeatability

B. Task Oriented tests

Multiple tasks were conducted in order to test the telerobotics system created. Tasks included picking up an object that was handed to the robot, grasping it, and placing it in a boxed. The test was conducted by an untrained operator over multiple trial runs in order to investigate: the performance of the system in a real world application, the intuitiveness of the system to an untrained operator, and the ease of learning and familiarizing with the system.

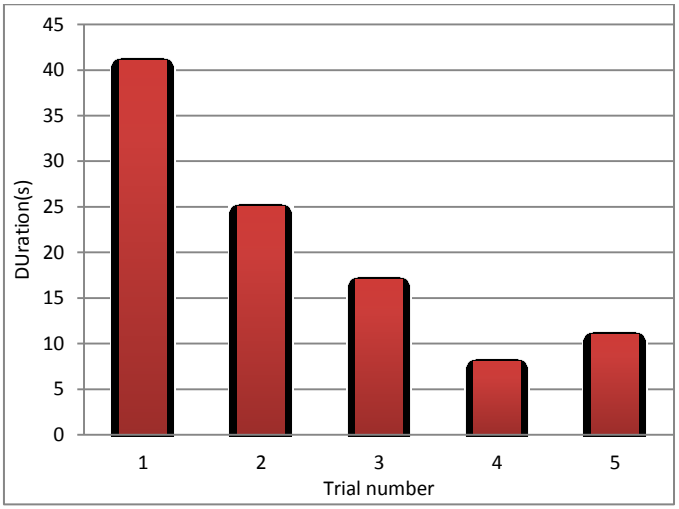

Figure 6. Successive trail durations.

Five trial runs were conducted and were timed until the object was successfully transported to the box. The chart in Figure 6 clearly shows a decrease in the time required to accomplish the task by an untrained operator. The time required to grab the object and put it in the box took on the fourth trial a fifth of the time it took for the first trial. This clearly shows a very quick familiarization with the system. These results demonstrate the ease and intuitiveness of the system developed, in addition to a good responsiveness. The system showed response delay of 0.4 seconds on average which translates in easier control.

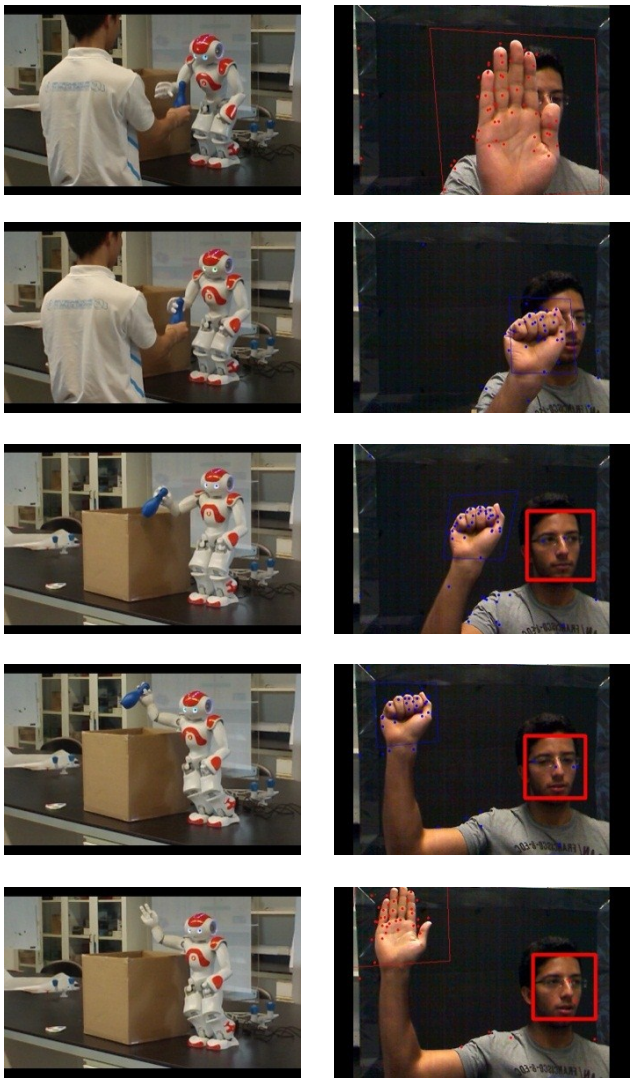

Figure 7. Sequence of images taken during the task oriented tasks, From top to bottom: reaching, grabbing, transporting, and dropping.

C. *Resutls Summary*

The system was implemented using SURF (Speeded-Up Robust Features), the feature points of the template and the frames were matched using squared difference nearest neighbor, and unwanted points were removed using RANSAC during the homography computation. This resulted in a 9.125fps and a repeatability of 67.25%, using a hessian threshold of 300 on a VGA sized frame.

The SURF points were extracted using the OpenCV implementation of the surf algorithm. The FLANN library was used to perform the nearest neighbor search, and compute the respective distances [8]. Once the points were matched, points were selected based on their distances.

Next a 3x3 homography matrix was computed while ignoring outliers using the RANSAC algorithm. Further testing of the homography matrix was set up to check that the homography describes a direct affine transformation. Once the homography is confirmed to be correct, the matches are projected using the homography and the location of the match is identified within the 2D plane.

Using Haar classfiers for the frontal face, the face is detected in the frame and, the approximate actual hand size is found. By using the relation between the actual hand size and the apparent hand size, the distance between the face and the hand is estimated. And therefore the depth data is extracted.

By combining the depth data with the 2D location, the hand gesture is recognized and located in the 3D space

The NAO robot was teleoperated using a TCP connection over Wi-Fi, and its body motion was set to self-balance during the arm motion in order to maximize the arm motion range. Due to the blocking nature of the whole body effector control of NAO, it was be replaced by a non-blocking method to remove the lag and improve the performance. The program was implemented in a single thread fashion initially, but was later made into a multithreaded version in order to improve the performance.

V. CONCLUSION

The paper presented a telerobotic system using the NAO robot. The system is operated by hands gestures which are represented using SURF features. Nearest neighbor approach was used to match features points of every new frame with template and then RANSAC approach is used to filter the matching results. The system showed a good frame rate performance of 9.125fps and with 67% repeatability at VGA frame size. The depth of the hand was inferred using face to hand size ratios base on Haar classifers of the frontal face. This NAO robot was operated using TCP connection over wifi and good gesture recognition and robot motion control was achieved as was shown in the previous results.

Further improvement of this system can be achieved by using a GPU implementation of SURF as the SURF algorithm is highly parallelizable, as the frame rate can reach 105fps [15]. In addition a new feature extractor/descriptor called FREAK (Fast Retina Keypoint), can provide better results in frame rate and rotation invariance [18] compared to SURF.

REFERENCES

[1] Haiying Hu, Jiawei Li, Zongwu Xie, Bin Wang , Hong Liu and Gerd Hirzinger , “A Robot Arm/Hand Teleoperation System with Telepresence and Shared Control”, Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Monterey, California, USA, 24-28 July, 2005

[2] Shuai Jin, Yi Li, Guang-ming Lu1, Jian-xun Luo, Wei-dong Chen, Xiao-xiang Zheng1, “SOM-based Hand Gesture Recognition for Virtual Interactions”, IEEE International Symposium on Virtual Reality Innovation 2011, 19-20 March, Singapore

[3] Juan Wachs, Uri Kartoun, Helman Stern, Yael Edan, "Real-time hand gesture telerobotics system using fuzzy c-means clustering", WAC 2002, Fifth Biannual World Automation Congress, June 9-13, 2002, Orlando, Florida

[4] D. Gouaillier, V. Hugel, P. Blzevic et. al, "Mechatronic design of NAO humanoid", in Proceedings of IEEE International Conference on Robotics and Automation, 2009, p.769-774, Kobe, Japan

[5] Paulo Menezes, Frédéric Lerasle and Jorge Dias`, "Data Fusion for 3D Gestures Tracking using a Camera mounted on a Robot", Proceedings of the IEEE 18th International Conference on Pattern Recognition (ICPR'06), 2006

[6] Doe-Hyung Lee, Kwang-Seok Hon , "Game Interface using Hand Gesture Recognition", School of Information and Communication Engineering, Sungkyunkwan University, 300 Chunchun-dong, Jangan-gu, Suwon, Kyungki-do, 440-746, Korea

[7] M. A. Diftler, R. Platt, Jr, C. J. Culbert, R.O. Ambrose, W. J. Bluethmann. "Evolution of the NASA/DARPA Robonaut Control System". Proceedings of the 2003 IEEE International Conference on Robotics & Automation. Taipei,Taiwan, September 14-19, 2003. 2543-2548.

[8] H. Bay, T. Tuytelaars, L. Gool, "SURF: Speeded up robust features", in Proceedings of the European Conference in Computer Vision, 2006, p.404-417.

[9] H Bay, A Ess, T Tuytelaars, L Vangool. "Speeded-Up Robust Features (SURF)". International Journal on Computer Vision and Image Understanding (2008).

[10] Marius Muja and David G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration", in International Conference on Computer Vision Theory and Applications (VISAPP'09), 2009.

[11] Y.-D. Jian and C.-S. Chen, "Two-View Motion Segmentation with Model Selection and Outlier Removal by RANSAC-Enhanced Dirichlet Process Mixture Models," *International Journal of Computer Vision*, vol. 88, no. 3, pp. 489–501, Jan. 2010.

[12] R. Marín, J. S. Sánchez, P. J. Sanz, "Object Recognition and Incremental Learning Algorithms for a Web-based Telerobotic System", Proceedings of the 2002 IEE International Conference on Robotics & Automation, Washington DC, May 2002.

[13] Paolo Fiorini , Roberto Oboe . "Internet-based telerobotics: problems and approaches". Proceedings of the 8th International Conference on Advanced Robotics, 1997. ICAR '97.

[14] Liu Jianbang, Lai Xuzhi, Wu Min, Chen Xin. "Design of embedded Telerobotics system". Proceedings of the 27th Chinese Control Conference July 16-18, 2008, Kunming,Yunnan, China.

[15] Costas S. Tzafestas, Nektaria Palaiologou, Manthos Alifragis. "Virtual and remote robotic laboratory: comparative experimental evaluation". IEEE transactions on education, vol. 49, no. 3, august 2006, p360 – 369.

[16] Bartlomiej Stanczyk , Martin Buss . "Development of a telerobotic system for exploration of hazardous environments". Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems September 28. October 2,2004, Sandal, Japan.

[17] M.L. Turner, R.P. Findley, W.B Griffin, M.R. Cutkosky, and D.H. Gomez, "Development and Testing of a Telemanipulation System with Arm and Hand Motion", Proc. ASME Dynamic Systems and Control Division (Symposium on Haptic Interfaces for Virtual Environments and Teleoperators), DSC-Vol. 69-2, 2000, pp. 1057-1063.

[18] David Gouaillier, Vincent Hugel, Pierre Blazevic Chris Kilner, Jérôme Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre and Bruno Maisonnier "Mechatronic design of NAO humanoid", 2009 IEEE International Conference on Robotics and Automation Kobe International Conference Center Kobe, Japan, May 12-17, 2009.