Gesture-based Human-Robot Interaction and Control

Saso Koceski[1], Natasa Koceska[1]and Predrag Koceski[1]

Abstract – **Aiming at the use of hand gestures for human-computer interaction, this paper presents a novel approach for hand gesture-based control of mobile robot's freight ramp. The research was mainly focused on solving some of the most important problems that current HRI (Human-Robot Interaction) systems fight with. Presenting a simple approach to recognizing gestures through image processing techniques and a single web camera, we address the problem of hand gestures recognition using motion detection and algorithm based on histograms, which makes it efficient in unconstrained environments, easy to implement and fast enough.**

Keywords – **Human-Robot Interaction (HRI), gesture-based control.**

I. INTRODUCTION

Advances in computer technology, artificial intelligence, speech simulation and understanding, and remote controls have led to breakthroughs in robotic technology that offer significant implications for the human computer interaction community.

Human-Robot Interaction (HRI) can be considered as one of the most important Computer Vision domains. In HRI based systems, the communication between human operators and robotic systems should be done in the most natural way. Typically, communication is done through hands/head postures and gestures. This type of communication provides an expressive, natural and intuitive way for humans to control robotic systems. One benefit of such a system is that it is a natural way to send geometrical information to the robot, such as: up, down, etc. Gestures may represent a single command, a sequence of commands, a single word, or a phrase and may be static or dynamic. Such a system should be accurate enough to provide the correct classification of hand gestures in a reasonable time. Human-robot interaction using hand gestures provides a formidable challenge. This is because the environment contains a complex background, dynamic lighting conditions, a deformable hand shape, and real-time execution requirement. There has recently been a growing interest in gesture recognition systems. For example, Stergiopoulou and Papamarkos [1] proposed YCbCr segmentation. In [2] a real-time hand posture recognition using 3D range data analysis is proposed. Ribeiro and Gonzaga [3] proposed different approaches of real time GMM (Gaussian Mixture Method) background subtraction algorithm using video sequences for image segmentation. Mariappan [4] uses motion detection algorithm for gesture recognition. Popa et al. [5], proposed trajectory based hand gesture recognition using kernel density estimation and the related mean shift algorithm. Bugeau and Pérez [6] proposed a method for detecting and segmenting foreground moving objects in complex scenes using clusters.

The main objective of this work was the developing of a control system for a robot freight ramp, based on gesture recognition. We aim to develop a cheap and robust control system without using data gloves or colored gloves, or other devices. With that purpose, we decided to use a generic webcam for the image acquisition process, and we have defined a gesture vocabulary for the telerobotic control, using motion detection and gesture recognition algorithm based on histograms, which makes it suitable for real time control, easy to implement and efficient in unconstrained environments.

II. FREIGHT RAMP WORKING PRINCIPLE

Since, this work is focused on the HRI and the controller development, the details about the conducted research on the ramp mechanics, will not be discussed here. Instead of that, the main working principle of the mobile robot's ramp will be described. As shown on Fig. 1. the robot ramp has two degrees of freedom:

-rotation around the Y axis which can be obtained by one stepper motor

-translation along the Z axis which can be obtained by stepper-based linear actuators.

In classic mechanical engineering, linear systems are typically designed using conventional mechanical components to convert rotary into linear motion. Converting rotary to linear motion can be accomplished by several mechanical means using a rotary motor, rack and pinion, belt and pulley, and other mechanical linkages, which require many components to couple and align. Although these methods can be effective, they each carry certain limitations. Conversely, stepper motor-based linear actuators address all these factors and have fewer issues associated with their use. The reason is that rotary-to-linear motion is accomplished in the motor itself, which translates to fewer components, high force output, and increased accuracy. The ramp has a warning light

[1]Saso Koceski, Natasa Koceska and Predrag Koceski are with the Faculty of Computer Sciences, University "Goce Delcev" – Stip, Krste Misirkov bb, 2000 Stip, Macedonia, E-mail: saso.koceski@ugd.edu.mk, natasa.koceska@ugd.edu.mk, pkoceski@gmail.com

which is activated whenever the ramp performs a movement and is deactivated when the ramp has finished the last movement. This light can be used as a visual feedback for the operator. The ramp is equipped with a sensor, which measures the angle between 90° ... -10°. In open position the angle is 5°. In closed position the angle is 90°. Another position sensor is used to calculate the vertical position (pos) as a value between 0 ... 100%. If the lift is up the position is 100%. If the lift is down the position is 0%. It is only possible to operate the lift upwards / downwards if the door is fully opened. If the door is open the tilt can be adjusted between +5° and -10° with the resolution of 3°. Adjusting the tilt is only possible if the door is fully open.

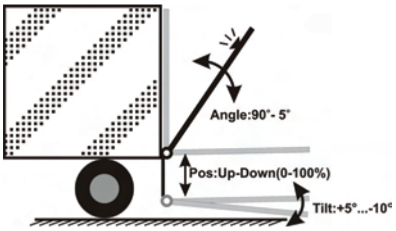

Fig. 1. Freight ramp working principle

III. Hand gesture language

Application specific vocabulary of 12 gesture poses (Fig. 2) was designed for robot freight ramp control tasks. The first two gestures control the ramp opening and closing. Next two gestures move the ramp up and down. Stop hand gesture stops any action the ramp performs moving the system into the Stopped position. StopTilt gesture stops any tilt movement the ramp performs and takes the ramp into the Stopped position. All other hand gestures control the tilt angle of the ramp.

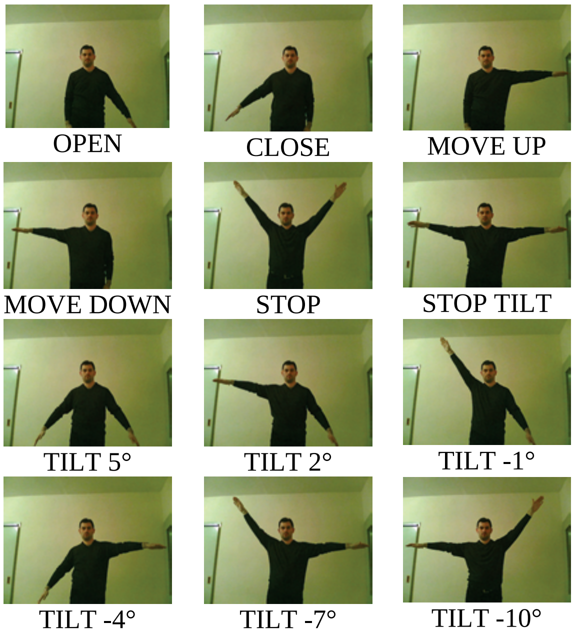

Fig. 2. Hand gesture vocabulary

IV. Control system architecture

The system architecture is based on client-server model Fig. 3. To control a robot's ramp movement, the operator evokes a gesture from gesture vocabulary. The user performs his hand gestures in front of a web cam, which captures a video stream with a frame rate of 30fps. The motion detection is performed and hand gesture is classified by the client control software application, which is running on a client console.

Each recognized gesture is converted into a command, and then it's sent through TCP/IP protocol to a distant robot PC server for execution by the robot Fig. 4.

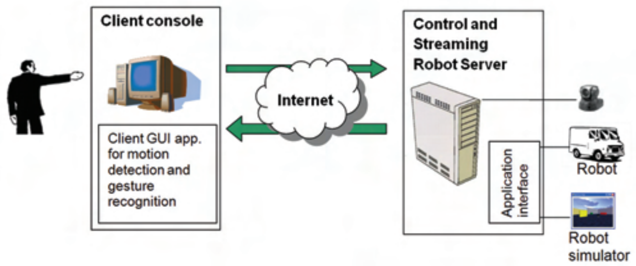
Client console
Client GUI app. for motion detection and gesture recognition
Internet
Control and Streaming Robot Server
Application interface
Robot
Robot simulator

Fig. 3. System architecture

In the case of no match, the system outputs "No gesture detected" message, with suggestions for the possible operator actions, which correspond to commands applicable in the current robot state. The application context restricts the possible visual interpretations and the user exploits the immediate feedback to adjust their gestures.

Robot server can be connected through the unified application interface both to the real robot and the robot simulator. A web cam with a streaming server provides a real time view (feedback) of the robot movements. Captured video is sent to the client using the FTP protocol. Besides the video feedback, a graphic and audio message for the current robot action/state, received by the robot sensors, is communicated via TCP/IP to the client application, and is presented to the user. When the robot completes an action, it is also reported to the operator. General control algorithm consists of three phases: a) Image acquisition; b) Image segmentation; c) Feature extraction and gesture recognition.

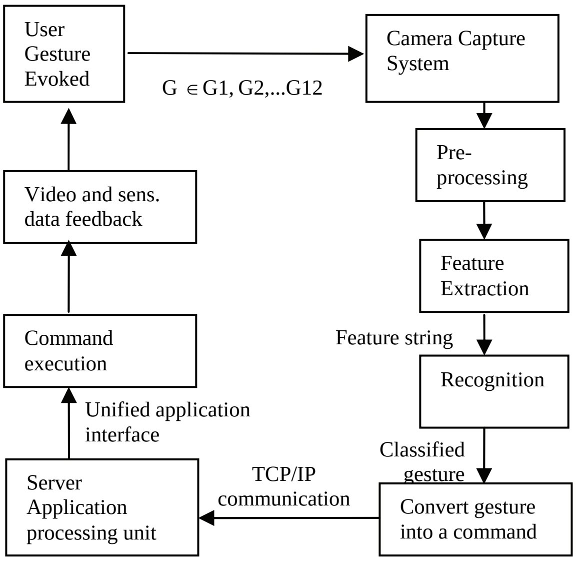

Fig. 4. Gesture recognition control flow diagram

V. Motion detection and object extraction

In this research motion detection was used for locating a potential region of interest (ROI). Motion detection approach applied in our algorithm is not bound to any specific video stream format/protocol. Instead of this, it just analyze consequent video frames, which makes them free from any video processing routines and makes them applicable to any video stream format.

This approach, based on finding inter-frame differences, tries to use simple techniques of modeling scene's background and updating it through time to get into account scene's changes. To take these changes into account, we are going to apply adaptive background modeling approach which tends to decrease the differences between two given images. The reference image is constantly updated with newly arriving image using the Eq.1:

$$\forall x, \forall y\; R_t(x,y) = \frac{N-1}{N}\cdot R_{t-1}(x,y) + \frac{1}{N} I(x,y) \qquad (1)$$

With R standing for the reference image and I for the newly arrived frame. The formula calculates a running average over all frames, with a weighting that decreases exponentially over time. For setups in front of a wall or white board, we found that the user practically never rests with his hand in the same position for more than 10 seconds, which implies a value of about 500 for N.

The main problem with this type of reference image updating is that dark, non-moving objects, such as the body are added to the reference image. If the user moves his/her hand into regions with dark objects in the reference image, there is not sufficient contrast between foreground and background to detect a difference. For this reason, we can update dark regions slowly, but light regions instantly. The value N in Eq. 1 is calculated as in Eq. 2.

$$\forall x, \forall y\; N(x,y) = \begin{cases} 1\; for\, I_{t-1}(x,y) - I_t(x,y) \le 0 \\ \approx 500\; for\, I_{t-1}(x,y) - I_t(x,y) > 0 \end{cases} \qquad (2)$$

With this modification, the algorithm provides the maximum contrast between foreground and background at any time.

This kind of modeling approach gives the ability of more precise highlighting of motion regions and makes the algorithm adaptive to unconstrained dynamic scenes.

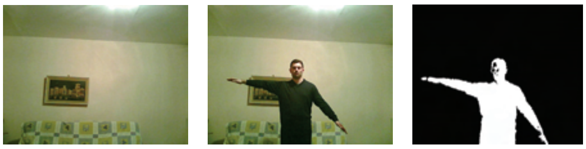

Fig. 5. Image differencing with reference image. From left to right: reference image, input image, output image

By applying a differentiation operation on the current and reference image, it is possible to see absolute difference between the two images (Fig.5).

VI. Hands gesture recognition

The algorithm for hand gesture recognition takes into consideration human's body proportions. It is based on histogram analysis, and presents two advantages: implementation simplicity and execution quickness. The core idea is based on analyzing two kinds of object's histograms: horizontal and vertical histograms (Fig. 6).

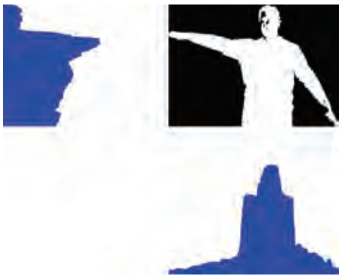

Fig. 6. Horizontal and vertical histogram of the detected human's body

Analyzing the horizontal histogram, one can observe that the hands' areas have relatively small values on the histogram, and the torso area is represented by a peak of high values. Considering relative proportions of humans' body and the horizontal histogram, we may say that human hand's thickness can never exceed 30% percent of human's body height. So, hands' areas and torso area can be easily classified, their length can be determined. Both experimentally and analytically, it can be proven that in the case of not raised hand, in most cases, the width on horizontal histogram will not exceed 30%. Otherwise it can be considered that the hand is raised somehow. When the hand is raised it can be raised in three different positions: raised diagonally down, raised straight or raised diagonally up. In order to determine the exact hand's position when it is raised, the vertical histogram of the hands objects only (Fig. 7), will be analyzed.

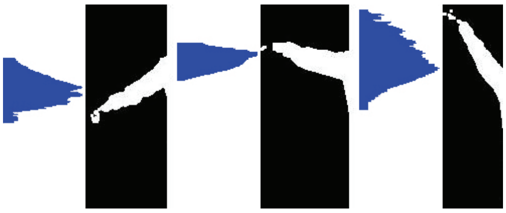

Fig. 7. Vertical filtered histograms of the raised hand

In some cases the histograms may be cluttered with noise, which may be caused by light conditions and shadows. Therefore two additional preprocessing steps of filtering will be performed, on the vertical histogram:

- removing the low values from the histogram, which are lower than some threshold percentage value of maximum histogram's value. This will remove some artifacts caused by the shadows.

- removing all peaks, which are not the highest peak. This will remove the strong shadows and other artifacts caused by the environment changing conditions. After these preprocessing steps exact hand position will be determined, taking into consideration one more assumption about body proportions, length of the hand is much bigger than its width. Therefore, one may observe that in the case of straight raised hand its histogram should have quite high, but thin peak, for the diagonally up hand its shifted to the top of vertical histogram, but the peak of the diagonally down hand is shifted more to the center (Fig. 7).

VII. EXPERIMENTAL RESULTS

In order to evaluate the performances of the developed control algorithm, physics-based robot simulator and server applications are deployed on PC with AMD Athlon 64 Processor on 2.4GHz with 1GB of RAM, NVIDIA GeForce FX5500 with 256MB memory and Windows XP operating system. Gesture recognition client application is implemented in C# language and is deployed on HP Pavilion dv62000 with Intel Core 2 Duo Processor on 2.2GHz with 2GB RAM, it has built in WebCam with 1.3 Megapixels resolution. Both computers are connected on (8 Mbps) Internet connection.

The performances of the developed control method were tested in two different conditions:

1. with sample images grabbed from static scenes in which the lighting and the distances from the camera were approximately constant.
2. in unconstrained scenes with changes of lighting, moving objects and different distances from the camera.

The accuracy was measured for 18 untrained users (9 males, 9 females with different body sizes) in both cases. Each user tried all command gestures 95 times in front of the camera watching the monitor. Table I shows the rates of successfully performed actions by the robot simulator. The results are average of all users.

TABLE I
HAND GESTURE DETECTION RATES

Gesture	Static scene detection rate	Unconstrained scene detection rate
Open	93.2%	89.7%
Close	91.5%	90.4%
Move up	94.9%	91.8%
Move down	95.1%	92.2%
Tilt -4°	94.6%	91.3%

VIII. CONCLUSION

This paper presents a fast, robust and accurate method for hand gestures recognition under unconstrained scenes. Experimental results show satisfactory recognition percentage of the gestures. The failure of the system to recognize the gesture is mainly due to the very changeable lighting conditions and moving objects (persons) entering the scene, operator’s failure to move the hand to the proper posture. It must be emphasize that after a short experience operators get used to the system.

REFERENCES

[1] E. Stergiopoulou and N. Papamarkos, "A new technique for hand gesture recognition", Proceedings of IEEE International conference on Image Processing, Atlanta, pp. 2657-2660, Oct. 8-11, 2006

[2] Malassiotis, S. and Strintzis, M. G., "Real-time hand posture recognition using range data," Image Vision Comput. 26(7), 1027-1037 (2008).

[3] Hebert Luchetti Ribeiro, Adilson Gonzaga, "Hand Image Segmentation in Video Sequence by GMM: a comparative analysis", Symposium on Computer Graphics and Image Processing (SIBGRAPI'06), 2006

[4] R. Mariappan, "Video Gesture Recognition System," iccima, vol. 3, pp.519-521, International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), 2007.

[5] Popa, D., Simion, G., Gui, V., and Otesteanu, M. 2008. Real time trajectory based hand gesture recognition. WSEAS Trans. Info. Sci. and App. 5, 4 (Apr. 2008), 532-546..

[6] Bugeau, A. and Pérez, P. 2009. Detection and segmentation of moving objects in complex scenes. Comput. Vis. Image Underst. 113, 4 (Apr. 2009), 459-476.

[7] Nielsen, J. “Enhancing the explanatory power of usability heuristics.” In Proc. of the CHI 94 Conference on Human Factors in Computing Systems, Boston, MA, ACM.