EURASIAN JOURNAL OF MATHEMATICAL
AND COMPUTER APPLICATIONS
ISSN 2306–6172
Volume 2, Issue 4 (2014) 81–89

EMBEDDED GESTURE RECOGNITION SYSTEM FOR ROBOTIC APPLICATIONS [1]

A.Begalinova, A. Shintemirov

Abstract

Recent developments of system-on-chip (SOC) technologies have found a lot of applications in robotic embedded system design. Powerful miniature SOC boards allow development of mobile robots with advanced onboard computation performance executing complex data processing and control algorithms. This paper presents a preliminary results of an embedded system design of a mobile robot platform implementing gesture recognition and control. The system is designed using a BeagleBoard-xM SOC board and an Asus Xtion Live 3D camera. System implementation and analysis of a human hand detection and tracking from depth images are discussed.

Key words: Embedded system, hand gesture recognition, hand tracking, BeagleBoard-xM, Asus Xtion Live depth camera

AMS Mathematics Subject Classification: 68T40, 68T45.

1 Introduction

Human-robot interaction is expected to meet modern demands on intuitive control with human hand gestures. In recent years robot tasks encountered some modifications due to development of novel 3D vision technologies, that led to increased importance of robot vision research in robotics community. In order to reduce human-robot programming data flow, gesture-based control was introduced as the most flexible and natural way for interaction with the machine [1].

In particular cases when robot is manipulated by hand gestures, the mobility of the machine platform can be enhanced [2]. A mounted camera on a robot platform can represents the source of vision for the robot automation and provide surrounding environment information to controller that will allow natural motion of the robot. This paper presents preliminary results of designing a real-time hand gesture interaction system equipped with the on-board 3D depth camera, which will be set up on a hexapod robot platform. The system obtains human 3D shape and extracts the hand position coordinates, subsequently reacting to the hand gestures. In overall, the designed embedded system can detects a human body in front of the installed 3D (depth) camera and perform hand gesture recognition. Thus, an interactive robot platform control can be achieved by gesture commands interpretation using novel algorithms proposed in this paper. The hand gesture recognition and robot navigation systems was conducted and tested in order to show effectiveness of such human-robot interaction system.

[1] A conference version of this paper was presented on the 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT 2014), Astana, Kazakhstan, October, 2014

The main focus of this paper is following:

1. to propose a real time human-robot control system based on hand gestures, which makes human-robot communication accessible. Hand gesture recognition and hand tracking system are implemented on the Asus Xtion Live depth camera sensor, which provides high vision performance.

2. The designed system can be controlled remotely through the BeagleBoard-xM SOC board placed on a robot platform. The data from the camera is received and processed on the BeagleBoard-xM board, then required commands are sent to the robot controller to navigate the robot. All the manipulation is done in real time.

This paper is organized as follows. At first, a short overview of the hardware and software applied in the designed system is presented. Then, the proposed hand gesture recognition algorithm is described.

2 System Design

The whole system design can be broken into two parts: vision system design and robot navigation system design.

2.1 Vision System Design

2.1.1 Depth camera

The Asus Xtion Live depth camera (http://www.asus.com/) is a one of the popular depth sensor cameras that is able to structure a 3D environment from video by infrared light pattern. The camera is composed from a depth sensor and a RGB camera. Applications developed for this camera are able of extracting full body skeletons from images and tracking gestures so efficiently that these gestures can be used in real time as inputs for robotic systems. The camera is very compact and does not require additional power supply which makes it easy to mount the camera on top of a robot platform.

In the proposed application the Asus Xtion Live camera is used as an input device, which captures depth images at 640x480 resolution. The camera will be placed on the robot platform and a user can control the robot standing in front of it. The video is processed using external libraries and necessary command is sent back to the vehicle.

2.1.2 Embedded System

In order to achieve the mobility and robustness of the embedded system it was decided to implement the vision system using a microcontroller board that will be mounted on the robot platform. After testing several system-on-chip (SOC) options the BeagleBoard-xM was selected. Beagleboard-xM is the singleboard computer, with all the power of a modern basic computer. The BeagleBoard-xM has a DM3730CBP 1GHz ARM Cortex A8 processor running at 1 GHz with extra memory x32 512MB @ 166MHz of low-power DDR RAM. The board is equipped with high speed 4 USB, USB OTG ports, DVI-D (HDMI connector), S-Video and 10/100 Ethernet and other ports (RS-232, JTAG connector, Camera port, unsoldered: I2C, I2S, SPI, MMC/SD). After plugging the necessary peripherals the board can be used as a fully assembled working machine [3, 4].

Figure 1: Asus Xtion Live depth camera and Beagleboard-xM platform.

The authors started its first system implementation using native compilation on the board instead of cross-compilation. The 3D camera was connected to the BeagleBoard-xM through an USB interface as shown in Fig 1. The output of the video was displayed on a desktop monitor connected to the BeagleBoard through a HDMI port. Interfacing with the robot platform will be done using serial connection established with the ROBOTIS CM700 servomotor controller, which will transmit/receive signals from/to the Beagleboard-xM.

2.1.3 System Software

The last stable embedded Linux Ubuntu distribution for ARM processors was loaded to the Beagleboard-xM board. In addition, external libraries and drivers were installed in order to execute hand detection and recognition features. The OpenNI cross-platform framework providing a set of open source APIs for writing applications was utilized and installed on the board. OpenNI aims to form a standard API that enables: vision and audio sensors, vision and audio perception middleware. The OpenNI standard API enables interaction application developers to track real-time 3D scenes received from the input sensor (for example, Asus Xtion Live camera).

As a recognition tool that can interpret the user interaction with the sensor the NITE middleware [5] is used. It includes both computer vision and movement tracking algorithms. The NITE provides sample applications which is comprehensive on natural user interaction, which includes gesture as push, steady, swipes, waves. The authors use a gesture wave as a sign to start tracking the hand and execute new gestures recognition method.

2.2 Library installation process

To achieve the desired camera operation necessary software libraries must be installed on the clean Linux Ubuntu system. This is done by running the following commands in the exact order as shown below.

2.2.1 Update the Linux and install additional libraries

- sudo apt-get update

- sudo apt-get install g++ python libusb-1.0-0-dev freeglut3-dev openjdk-6-jre openjdk-6-jdk doxygen graphviz git make man g++-4.7multilib gcc-4.7multilib libc6devarmhf libc6devarmel

2.2.2 Install the OpenNI library

- git clone https://github.com/OpenNI/OpenNI.git
- cd OpenNI/Platform/Linux/CreateRedist
- chmod a+x RedistMaker

2.2.3 Remove the "mfloatabi=softfp" flag from Platform/Linux/Build/Common/Platform.ARM file

- ./RedistMaker
- cd ../Redist/OpenNI-Bin-Dev-Linux-Arm-v1.5.2.23/
- chmod a+x install.sh
- sudo ./install.sh

2.2.4 Install additional Xtion driver

- git clone https://github.com/PrimeSense/Sensor.git
- cd Sensor/Platform/Linux/CreateRedist

2.2.5 Remove the "mfloatabi=softfp" flag from Platform/Linux/Build/Common/Platform.ARM file

- chmod a+x RedistMaker
- ./RedistMaker
- cd ../Redist/Sensor-Bin-Linux-Arm-v5.1.0.41/
- chmod a+x install.sh
- sudo ./install.sh

2.2.6 Download the NITE ARM library

- cd nite-arm
- sudo ./install.sh

Other essential libraries installed on the board include:
- build-essential,
- checkinstall git,
- cmake,
- libfaac-devlibjack-jackd2-dev,
- libmp3lame-dev,

- libopencore-amrnb-dev,
- libopencore-amrwb-devlibsdl1.2-dev,
- libtheora-dev libva-dev,
- libvdpau-dev,
- libvorbis-dev,
- libx11-devlibxfixes-dev,
- libxvidcore-dev,
- texi2html,
- yasm zlib1g-dev,
- libgstreamer0.10-0,
- libgstreamer0.10-dev,
- gstreamer0.10-toolsgstreamer0.10-plugins-base,
- libgstreamer-plugins-base0.10-devgstreamer0.10-plugins-good,
- gstreamer0.10-plugins-ugly,
- gstreamer0.10-plugins-badgstreamer0.10-ffmpeg x264,
- ffmpeg gtk libjpeg v4l.
-

The OpenNI library comes with samples:

- cd /OpenNI/Platform/Linux/Build
- make
- cd ../Bin/Arm-Release

However, during the samples running on the Beagleboard-xM, some problems were encountered, e.g.:

- Error: VFH float — soft
  Solution: change Common/CommonMakefile and add mfloatabihard into CFLAGS
- Error: armangstromlinuxgnueabi does not exist
  Solution: cp Platform.x86 Platform.Arm
- Error: InitFromXml failed: Failed to set USB Interface
  Solution: sudo rmmod gspca_kinect

2.3 Robot navigation system design

The hand detection is a prior step of the gesture recognition process. The installed Asus Xtion Live sensor driver and NITE middleware provides system functionalities for detecting and tracking a hand centroid point utilized in the designed system. The authors then created and implemented an algorithm for human gesture recognition in motion based on the hand centroid point coordinates. This algorithm will be used to provide robot platform movements commands in directions corresponding to the hand motion. The authors developed a hand motion recognition method by detecting the hand centroid point velocity [6].

Once the user makes a sign gesture, i.e. a wave motion, the camera starts tracking the hand centroid as shown in Fig. 2. The pool of collected hand centroid points p is set from the image stream. Each p in has its own coordinate in the cartesian coordinate system $(p(x, y, z))$. The pool is filled at each frame update. When the pool is filled less than 30 (

Figure 2: A depth image with human hand centroid tracking.

about 1 sec of video at 30FPS), there is enough information to calculate the velocity v of the current point p. The velocity v is calculated through the current node

$$x(t), y(t), z(t)$$

and the previous node

$$x(t-\alpha), y(t-\alpha), z(t-\alpha),$$

where α is number of the past nodes.

The value of α depends on the video update speed, and must be chosen very carefully. If the value of α is too small the difference between current and the node under consideration will be too small to detect hand velocity changes. If α is too big the system will respond slowly. α must be in the range $0 < \ < 30$. Assuming the estimation that a simple gesture of a human takes about 0.33 sec, the camera will be able to detect the gesture in 10 frames update. Therefore, $\alpha = 10$ is chosen.

Next, the difference between nodes are computed taking difference of point coordinates:

$$\delta(x) = x(t-\alpha) - x(t) \tag{1}$$
$$\delta(y) = y(t-\alpha) - y(t) \tag{2}$$
$$\delta(z) = z(t-\alpha) - z(t). \tag{3}$$

The difference is taken at each direction x, y, z, such that any movement of the hand towards each side can be tracked. If δ becomes larger than the mean value μ, it indicates that the change of hand velocity is occurred. μ is the sum of the values of hand point coordinates taken in very slow hand motion effect, and in the rest position of the hand divided by the number of values. Thus, if $\delta(x)$ or $\delta(y)$, or $\delta(z)$ is larger than the mean value μ, then the hand gesture to the corresponding direction has been detected by the system and the corresponding signal is sent to the robot, which changes the robot direction to the matching coordinate in the area using the following robot commands: "Left", "Right", "Forward", "Backwards", "Stop".

Table 1: Accuracy of hand gesture recognition

Hand gesture	Training	Successful	Missing	Recognition rate
Left	50	46	4	92
Right	50	47	3	94
Forward	50	45	5	90
Backwards	50	45	5	90
Stop	50	46	4	92

3 Results and Discussion

The proposed hand gesture recognition algorithm uses the hand velocity data to detect the hand motion. To experimentally evaluate the algorithm implemented on the designed embedded system, a training set of 250 different hand motion sequences including each 50 of the gestures "Left", "Right", "Forward", "Backwards" and "Stop" was created. The gestures were acquired in real-time by the Asus Xtion Live depth camera and processes using the algorithm on the Beagleboard-xM. The following formula is utilized for calculating the gesture recognition rate:

$$RecognitionRate\% = \frac{\#Successfull}{\#Training\ Set} * 100\%. \tag{4}$$

Table 1 presents the statistical analysis of performed test experiment. As clear from the table the recognition rate is in the range of $90 - 94\%$ that is very satisfactory in this initial stage of the system implementation. Subsequently, the serial connection between the BeagleBoard-xM and robot platform will be tested through Receiver/Transmitter ports. Sending commands "1" or "2" will correspond to the Left and Right commands of the robot platform.

After all the installation was ported to the BeagleBoard-xM, the 3D image viewer sample was tested and succeeded as shown in Fig. 3. Unfortunately, during the system development, the authors encountered a significant problem. Due to the BeagleBoard-xM architecture the OMAP driver showed incompatibility, which can be explained with that OpenGL API is not applicable for the GPU architecture used in the Beagleboard Xm version (PowerVR SGX530, Series 5). Therefore, it was impossible to run the OpenGL GUI functions on that processor. It was decided to implement the evaluation of the algorithm with the cross-compilation tool Qt, and compile the OpenGL part of the algorithm on PC with advanced architecture, and develop a console version of the developed software to run on the designed embedded system. As the result, the authors are working on implementing another experiment to test the performance of console interaction system.

4 Conclusion

In this paper, the authors present a human-robot interaction embedded system based on hand motion recognition. The system allows the interaction with a mobile robot, standing in front of it in real time. The system captures 3D hand trajectory via 3D hand tracking algorithm based on the Asus Xtion Live sensor. The velocity data of the hand motion is extracted, which shows the hand motion direction. The test of the proposed in the paper

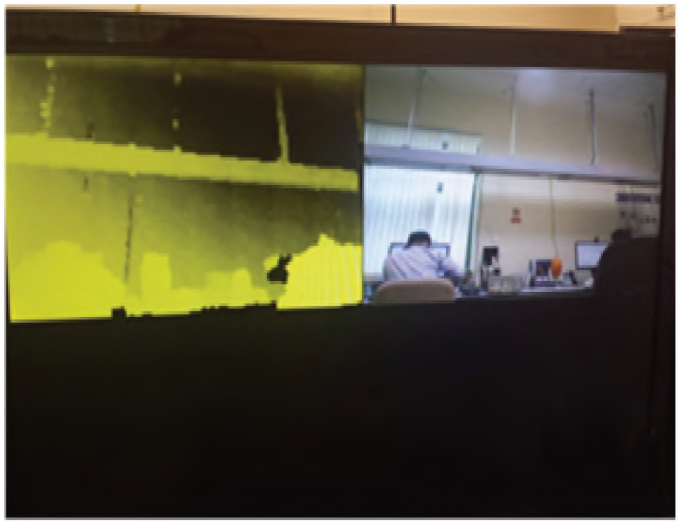

Figure 3: 3D image viewer running on BeagleBoard Xm.

algorithm showed the satisfactory recognition rate of $90-94\%$. Due to the incompatibility of BeagleBoard-xM hardware with the sensor library the authors would like to build the console version of the algorithm without replacing the hardware. Besides, the Hidden Markov Model method for more complex gesture recognition will be incorporated in the system.

References

[1] R. S. a. C. Meyer. The need for an intuituve teaching method for small and medium enterprises. In *Proceedings of the 2nd I*PROMS Virtual International Conference*, Amsterdam, Netherlands, July 2006.

[2] A. Kendon. *Gesture: Visible action as utterance.* Cambridge University Press.

[3] P. K. Aby. Implementation and optimization of embedded face detection system. In *Proceedings of the Internation Conference on Signal Processing, Communication, Computing and Network Technologies (ICSCCN)*, Thuckafay, 2011.

[4] Beagleboard-xm rev c2 system reference manual.

[5] P. N. Villaroman. Teaching natural user interaction using openni and the microsot kinect sensor. In *SIGITE Proceedings of the 2011 Conference on Information Technology Education (ICSCCN)*, New York, USA, 2011.

[6] X. W. Dan Xu. Real-time dynamic gesture recognition system based on depth perception for robot navigation. In *SIGITE Proceedings of the International Conference on Robotics and Biometrics*, Guangzhou, China, 2012.

Ainur Begalinova, Almas Shintemirov*
Department of Robotics and Mechatronics
Nazarbayev University, Astana, Kazakhstan

Email: ainur.begalinova@nu.edu.kz, ashintemirov@nu.edu.kz
*Corresponding author

Received 15 Nov 2014, Accepted 20 Dec 2014