Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 8, August 2013, pg.211 – 218

RESEARCH ARTICLE

Design and Implementation of Posture Identification System

Indu Mandwi[1], Sofia Pillai[2], Hemant Turkar[3], Yamini Bute[4]

[1]Computer Science and Engineering, DBACER, Nagpur, India

[2]Computer Science and Engineering, RGCER, Nagpur, India

[3]Computer Science and Engineering, RGCER, Nagpur, India

[4]Computer Science and Engineering, DBACER, Nagpur, India

[1] induhrl@gmail.com; [2] pillaisofia@gmail.com; [3] hemantturkar@rediffmail.com,[4] ysbute@gmail.com

Abstract— Robots are, or soon will be, used in such critical domains as search and rescue, military battle, mine and bomb detection, scientific exploration, law enforcement, and hospital care. Such robots must coordinate their behaviors with the requirements and expectations of human team members; they are more than mere tools but rather quasi-team members whose tasks have to be integrated with those of humans. Proposed system will be a robotic vehicle having 360 degree of rotation with camera to take input and processed to calculate input to the system. System will scan for human behavior through camera and try to follow the human object and it will also get the input through human speech and interpret speech to input signal and perform the task as per given to the system.

Key Terms: - Robotics; Embedded System; Markov Model; RSSI

I. INTRODUCTION

This Natural and friendly interface is critical for the development of service robots. Gesture-based interface offers a way to enable untrained users to interact with robots more easily and efficiently. Proposed system presents a human-robot interface system that enables a user to move a robot by moving his hand. System deals with a method to recognize hand-postures in system. The system uses single camera to recognize the user's hand-posture. It is hard to recognize hand-posture since a human-hand is the object with high degree of freedom and there follows the self-occlusion problem, the well-known problem in vision-based recognition area. It can be a possible solution to use multiple cameras. However, when we use multiple images, another problem is arisen, that is, the composition methods of multiple recognition results from each camera to get the final decision system. Here the proposed system is designed to trace and identify the human hand and head gesture using computer vision. The system will help the user interact with robots using human behavior and transform the computer human interaction in to virtual interaction.

The proposed system is mainly designed for controlling robotic hand or an individual robot by merely showing hand gestures in front of a camera. With the help of this technique one can pose a hand gesture in the vision range of a robot and corresponding to this notation, desired action is performed by the robotic system. Simple video camera is used for computer vision, which helps in monitoring gesture presentation. This approach consists of four modules: (a) A real time hand gesture formation monitor and gesture capture, (b) feature extraction, (c) Pattern matching for gesture recognition, (d) Command determination corresponding to shown gesture and performing action by robotic system. Real-time hand tracking technique is used for object detection

in the range of vision Mobile robotics has been used widely in education as a learning tool, as it provides a motivating and interesting tool to perform laboratory experiments within the context of mechatronics, electronics, computer, and control.

The robot innovative platform made it possible for the user to practice and learn many necessary skills. A tracing algorithm about machine vision intelligent tracing System based on CCD [1] [2] camera was described. The proposed system is comprised of a modular robot and the microcontroller as tools to design and implement the intelligent tracing system. The objective of this machine vision system is to implement an intelligent tracing applications running on the embedded systems as well as the RTOS under which they are executed. Document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

II. LITERATURE SURVEY

A Posture Identification is a field where lot of work has been done in an attempt to increase the human robot interaction so that the system can work more efficiently for the work it is designed for. Posture Identification alone with the other system can be used in Gesture Recognition, Gesture Recognition and Pen Computing,

A. Gesture Recognition

RTOSs can be classified in two categories: hard-RTOSs and soft-RTOSs [10][11]. The main difference between the two categories is that a soft- RTOS can tolerate latencies and responds with decreased service quality but the hard-RTOS [4] has to respect its deadlines, because otherwise the tasks fail. RTOSs provide four main types of basic services to the application program. Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand. Current focuses in the field include emotion recognition from the face and hand gesture recognition. Many approaches have been made using cameras and computer vision algorithms to interpret sign language. However, the identification and recognition of posture, gait, proxemics, and human behaviors is also the subject of gesture recognition techniques.

Fig 1.Child being sensed by a simple gesture recognition algorithm

Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse.

Gesture recognition enables humans to interface with the machine (HMI)[5] and interact naturally without any mechanical devices. Using the concept of gesture recognition, it is possible to point a finger at the computer screen so that the cursor will move accordingly. This could potentially make conventional input devices such as mouse, keyboards and even touch-screens redundant.

B. Body posture identification using hidden Markov model with a wearable sensor network

This paper presents a networked proximity sensing and Hidden Markov Model (HMM) [6] [7] based mechanism that can be applied for stochastic identification of body postures using a wearable sensor network. The idea is to collect relative proximity information between wireless sensors that are strategically placed over a subject's body to monitor the relative movements of the body segments, and then to process that using HMM in order to identify the subject's body postures. The key novelty of this approach is a departure from the traditional accelerometry based approaches in which the individual body segment movements, rather than their relative proximity, is used for activity monitoring and posture detection. Through experiments with body mounted

sensors we demonstrate that while the accelerometry based approaches can be used for differentiating activity intensive postures such as walking and running, they are not very effective for identification and differentiation between low activity postures such as sitting and standing. We develop a wearable sensor network that monitors relative proximity using, Radio Signal Strength indication (RSSI) [8] [9], and then construct a HMM system for posture identification in the presence of sensing errors. Controlled experiments using human subjects were carried out for evaluating the accuracy of the HMM identified postures compared to a naïve threshold based mechanism, and its variations over different human subjects.

C. The model-based dynamic hand posture identification using genetic algorithm

This paper proposes a genetic-based hand posture identification system which consists of an efficient model fitting method and a labelling technique. The model fitting method consists of (1) finding the closed form inverse kinematics solution, (2) defining the alignment measure, and (3) developing a genetic-based dynamic posture fitting process. Different from the conventional computation intensive hand model fitting methods, we develop (1) an off-line training process which uses the inverse-kinematics to find the closed-form solution function, and (2) a fast on-line model-based hand'posture identification process. In the experiments, we will illustrate that our genetic-based hand posture identification system is effective and real-time implementable.

D. Two-staged hand-posture recognition method for Softremocon system

This paper deals with a method to recognize hand-postures in Softremocon system. The system uses multiple cameras to recognize the user's hand-posture. It is hard to recognize hand-posture since a human-hand is the object with high degree of freedom and there follows the self-occlusion problem, the well-known problem in vision-based recognition area. It can be a possible solution to use multiple cameras. However, when we use multiple images, another problem is arisen, that is, the composition methods of multiple recognition results from each camera to get the final decision. We show 2-staged recognition process. The first stage gets the results which represent each camera viewpoint. And the second one results out the final decision. This structure gives robustness to the hand-posture recognition against view-point variation.

E. Finger identification and hand posture recognition for human–robot interaction

Natural and friendly interface is critical for the development of service robots. Gesture-based interface offers a way to enable untrained users to interact with robots more easily and efficiently. In this paper, we present a posture recognition system implemented on a real humanoid service robot. The system applies RCE neural network based color segmentation algorithm to separate hand images from complex backgrounds. The topological features of the hand are then extracted from the silhouette of the segmented hand region. Based on the analysis of these simple but distinctive features, hand postures are identified accurately. Experimental results on gesture-based robot programming demonstrated the effectiveness and robustness of the system.

F. Input Devices

The ability to track a person's movements and determine what gestures they may be performing can be achieved through various tools. Although there is a large amount of research done in image/video based gesture recognition, there is some variation within the tools and environments used between implementations.

Depth-aware cameras: Using specialized cameras such as time-of-flight cameras, one can generate a depth map of what is being seen through the camera at a short range, and use this data to approximate a 3d representation of what is being seen. These can be effective for detection of hand gestures due to their short range capabilities.

Stereo cameras: Using two cameras whose relations to one another are known, a 3d representation can be approximated by the output of the cameras. To get the cameras' relations, one can use a positioning reference such as a lexian-stripe or infrared emitters.

Controller-based gestures: These controllers act as an extension of the body so that when gestures are performed, some of their motion can be conveniently captured by software. Mouse gestures are one such example, where the motion of the mouse is correlated to a symbol being drawn by a person's hand, as is the Wii Remote, which can study changes in acceleration over time to represent gestures.

Single camera: A normal camera can be used for gesture recognition where the resources/environment would not be convenient for other forms of image-based recognition. Although not necessarily as effective as stereo or depth aware cameras, using a single camera allows a greater possibility of accessibility to a wider audience.

III. SOFTWARE METHODOLOGY

A. Proposed System

Proposed system figure will show the basic structure of the unit. As per shown in the unit it has different modules like processing unit, Base wheel system, Web camera, etc. This is autonomous robot which is having

own ability to take input process it and take decision and control robots behavior. It will take input from mic and through web camera.

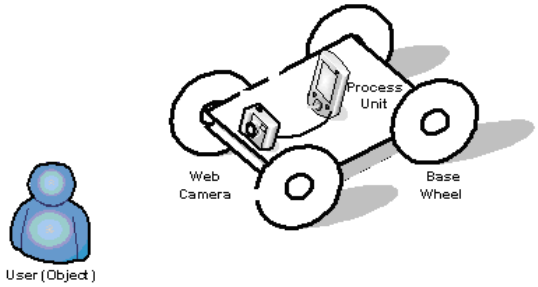

Fig 2. Proposed System Structural Diagram

1) Image acquisition: A digital image is produced by one or several image sensors, which, besides various types of light-sensitive cameras, include range sensors, tomography devices, radar, ultra-sonic cameras, etc. Depending on the type of sensor, the resulting image data is an ordinary 2D image, a 3D volume, or an image sequence. The pixel values typically correspond to light intensity in one or several spectral bands (gray images or colour images), but can also be related to various physical measures, such as depth, absorption or reflectance of sonic or electromagnetic waves, or nuclear magnetic resonance.

2) *Pre-processing:* Before a computer vision method can be applied to image data in order to extract some specific piece of information, it is usually necessary to process the data in order to assure that it satisfies certain assumptions implied by the method.

3) Feature extraction: Image features at various levels of complexity are extracted from the image data. Typical examples of such features are Lines, edges and ridges. Localized interest points such as corners, blobs or points.

4) Detection/segmentation: At some point in the processing a decision is made about which image points or regions of the image are relevant for further processing.

5) *High-level processing:* At this step the input is typically a small set of data, for example a set of points or an image region which is assumed to contain a specific object. The remaining processing deals with, for example: Verification that the data satisfy model-based and application specific assumptions. Estimation of application specific parameters, such as object poses or objects size. Classifying a detected object into different categories.

B. *Research Methodology*

Pattern Matching*:* For vision based input system need to process web camera vision. It will capture the live frames from web camera and the process each frame as a input and calculate the desired value as per the system required once the system get the desired value it will send signal to processing unit and processing unit will control the hardware as per defined.

Vehicle drive motors*:* This is most important part in the system it will move vehicle from one place to another. This is built in such a fashion that it can turn at any angle (like army tanker motions). This rotation feature will give system flexibility to move as per requirements.

IV. IMPLEMENTATION

The Proposed system is divided into following modules. First is video capturing to get the video frames, next is image processing to get the images from frame, next is to get the pixel information from the image, the detection of color from pixel and at last controlling the hardware of our project.

A. *Image Construction and Pixels*

This fig shows the image construction. As we know that image is made up of pixels and each pixel is made up of bits this bit pattern is depending on the file or image format. Each bit of pixel value for color. For example in bitmap file format it use one bit for Red, one bit for Green and one bit for Blue we can see it in this picture.

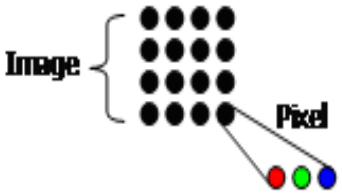

Fig 3. Image Construction

B. *Processing*

First we get the web camera device object then get the frame by frame that is the image. As we get the image then we try to extract the pixels from it and as we get the pixel we try to get the RGB color value from the pixel as pixel made up from the RGB value of the color. As RGB value get we compare the value and last we have the color. Side text show how we can compare the colors if the value of R is 255 and value of green and blue is 0 then it is considered as red color and same for the other. As we are saying that we are capturing camera view and this slide will explain the same how we get the camera view. To access any of the cameras we need camera driver to be installed coz it handle the camera working. And second we need to contact window to provide camera window to user to application after getting the capture window we can load the view in picture box or in any image showing object;

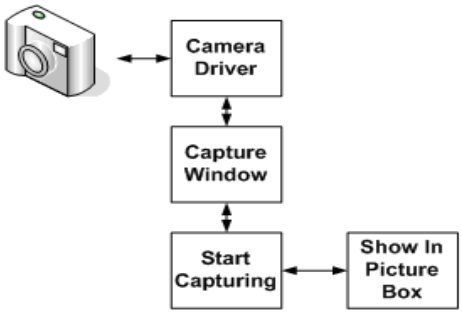

Fig 4 .Working of Camera Capturing

C. *Loading the Driver*

As we are saying that we are gathering the camera but it is important to know how we are doing this programmatically. From this we start explaining the programming terms for capturing and extracting images. System can have more than one web camera connected and the entire camera need camera driver which user will install on the system. These drivers are managed by the O.S. and O.S. gives unique ID to each driver. Even system can have multiple cameras but system can show only one camera view at a time. In VB there is a function "capGetDriverDescription" to detect the entire driver available on the system. This function will return Name of driver and the ID of driver.

D. *Capturing View*

After getting the list of driver we need to use specific driver in the project to start the capturing of the camera view in the picture box. There is a function named "capCreateCaptureWindow" which can be used to get the camera window object. We need to pass the driver detail to this function that is name of the driver and the ID of driver. This function will return a window object on successfully capturing the camera window. Then to actually start the camera view we use "sendmesage" function to the window object and send message Connect we can show it in the slide. This function will actually start the camera view in picture box.

E. *Extracting Frame*

Now we are having the camera view but the problem is that this view is handled by the O.S. and the camera driver can't work on it for processing. So we need to convert the live video in to processing format. So to get the current view as image we used a function "hdcToPicture" This function will convert the view in to image format

which be processed by our project. Next we used a variable which can hold entire image in memory so by using "Dim bm as bitmap" statement we declare an image object which will hold entire image in memory as a variable and to load current view in this bm variable we used a function "getobject" this function will load a picture form picture box to bm variable. After this step we have image in variable which we can used to extract the pixel.

F. *Extracting Pixels*

Now we are having the camera view but the problem is that this view is handled by the O.S. and the camera driver can't work on it for processing. So we need to convert the live video in to processing format. So to get the current view as image we used a function "hdcToPicture" This function will convert the view in to image format which be processed by our project. Next we used a variable which can hold entire image in memory so by using "Dim bm as bitmap" statement we declare an image object which will hold entire image in memory as a variable and to load current view in this bm variable we used a function "getobject" this function will load a picture form picture box to bm variable. After this step we have image in variable which we can used to extract the pixel.

G. *System Architecture*

Now we are having the camera view but the problem is that this view is handled by the O.S. and the camera driver can't work on it for processing. So we need to convert the live video in to processing format. So to get the current view as image we used a function "hdcToPicture" This function will convert the view in to image format which be processed by our project. Next we used a variable which can hold entire image in memory so by using "Dim bm as bitmap" statement we declare an image object which will hold entire image in memory as a variable and to load current view in this bm variable we used a function "getobject" this function will load a picture form picture box to bm variable. After this step we have image in variable which we can used to extract the pixel.

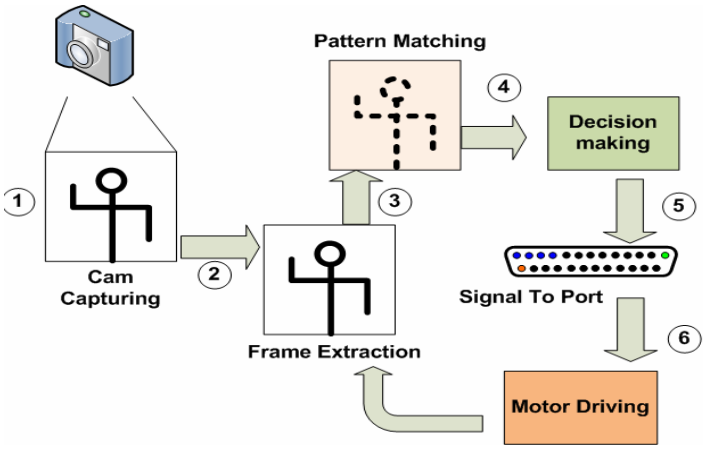

Fig 5. Camera Processing

H. *input32.dll*

This Fig. explains about the working of Inpout32.dll in simple steps, with the help of a flow chart. This could help much if you want to modify the Inpout32 dll source code The outstanding feature of Inpout32.dll is , it can work with all the windows versions without any modification in user code or the DLL itself. This tutorial describes how it is achieved, what programming methods used, what are the APIs used, etc.... The Dll will check the operating system version when functions are called, and if the operating system is WIN9X, the DLL will use _inp() and _outp functions for reading/writing the parallel port. On the other hand, if the operating system is WIN NT, 2000 or XP, it will install a kernel mode driver and talk to parallel port through that driver. The user code will not be aware of the OS version on which it is running. This DLL can be used in WIN NT clone operating systems as if it is WIN9X. The flow chart of the program is given below.

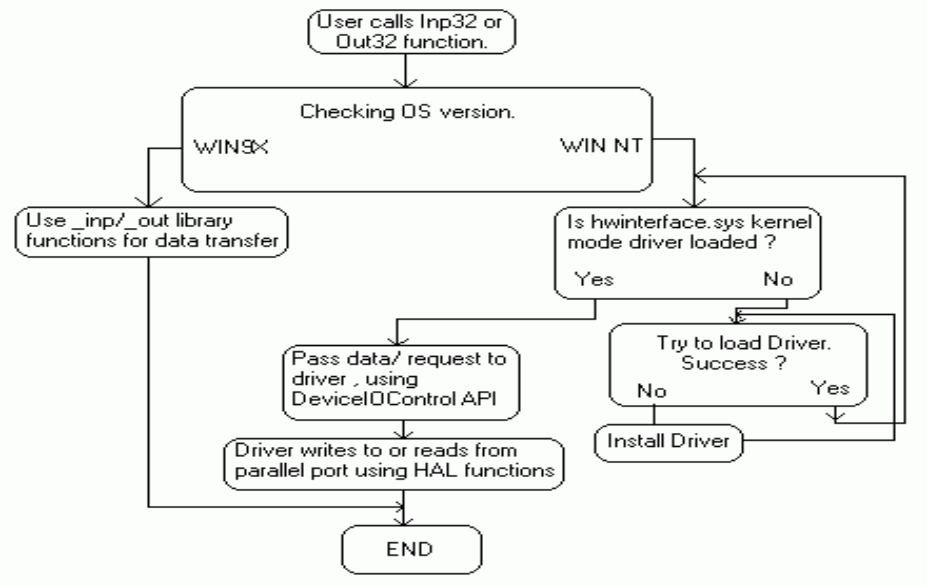

Fig 6 How input32.dll works

The two important building blocks of this program are:First is A kernel mode device driver embedded in the DLL in binary form and DLL itself.

1) *Kernel mode driver Hwinterface.sys*: The source code of Hwinterface.sys kernel mode driver is located in "kernel_mode_driver_source" directory. Where "hwinterfacedrv.c" is the main application source file. Three functions implemented in the driver are: First 'DriverEntry' , Called when driver is loaded. Creates device object and symbolic links. Second 'hwinterfaceUnload', Called when driver is unloaded, performs clean up. 'hwinterfaceDeviceControl', handles calls made through DeviceIOControl API. Performs reading writing to the parallel port according to the control code passed.
2) The DLL Inpout32: The functions in the DLL are implemented in two source files, "inpout32drv.cpp" and "osversion.cpp". osversion.cpp checks the version of operating system. "inpout32drv.cpp" does installing the kernel mode driver, loading it , writing/ reading parallel port etc. 'Inp32', reads data from a specified parallel port register.'Out32', writes data to specified parallel port register the other functions implemented in Inpout32.dll are 'DllMain', called when dll is loaded or unloaded. When the dll is loaded, it checks the OS version and loads hwinterface.sys if needed.'Closedriver', close the opened driver handle. called before unloading the driver.'Opendriver', open a handle to hwinterface driver.'inst' , Extract 'hwinterface.sys' from binary resource to 'systemroot\drivers' directory and creates a service. This function is called when 'Opendriver' function fails to open a valid handle to 'hwinterface' service. 'start' , starts the hwinterface service using Service Control Manager APIs.'SystemVersion' Checks the OS version and returns appropriate code. It is an activex control with same features of Inpout32.dll. It can be used either with VC++ or VB. But it gives great convenience when used with VB. Data can be written to parallel port using Outport method and can be read using Inport method. hwinterface.ocx ActiveX control is an activex control with same features of Inpout32.dll. It can be used either with VC++ or VB. But it gives great convenience when used with VB. Data can be written to parallel port using Outport method and can be read using Inport method.

V. RESULTS

Thus the system that has been made can be used for a wide variety of applications like Industrial robotics, Vision based robots, Computer Human Interaction, Key less communication Mobile robotics has been used widely in education as a learning tool, as it provides a motivating and interesting tool to perform laboratory experiments within the context of mechatronics, electronics, computer, and control. The robot innovative platform made it possible for the user to practice and learn many necessary skills. A tracing algorithm about machine vision intelligent tracing System based on CCD camera was described. The proposed system is comprised of a modular robot and the microcontroller as tools to design and implement the intelligent tracing system. The objective of this machine vision system is to implement an intelligent tracing system. Here the proposed system is designed to trace and identify the human hand and head gesture using computer vision. The

system will help the user interact with robots using human behavior and transform the computer human interaction in to virtual interaction.

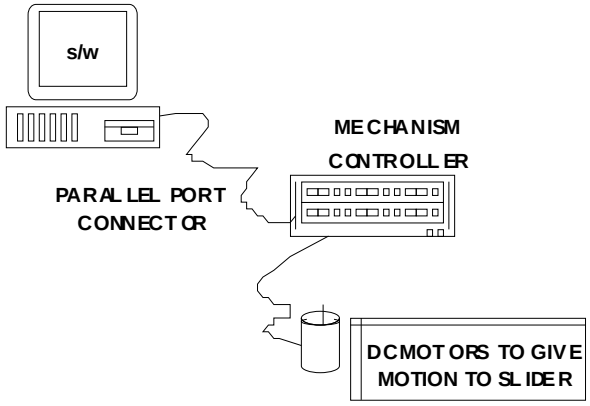

Fig 7. Overall Development of System

VI. CONCLUSION AND FUTURE WORK

Human interaction is very demanding field in all research and development sector and by trying to develop this project which communicate with machine using posture and speech we are planning to take first step in this area. In future development we are tying following things if implemented with bio technology then this can be used for Bio Medical Operations. Using an IC and programming it will remove the parallel port and will become handier .Smooth and more option can be provided.

REFERENCES

[1] L. Aryananda," Recognizing and Remembering Individuals: Online and Unsupervised Face Recognition for Humanoid Robot" in Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002), Vol. 2, pp. 1202-1207, 2009.[Aryananda, 2002]

[2] H. Asoh, S. Hayamizu, I. Hara, Y. Motomura, S. Akaho and T. Matsui, "Socially Embedded Learning of the Office-Conversant Mobile Robot Iijo-2", in Proceeding of 15th International Joint-Conference on Artificial Intelligence (IJCAI'97), pp.880-885, 2008.[Asoh, 1997]

[3] M. F. Augusteijn, and T.L. Skujca, "Identification of Human Faces Through Texture-Based Feature Recognition and Neural Network Technology" , in Proceeding of IEEE conference on Neural Networks, pp.392-398, 1993. [Augusteijn, 2004]

[4] Y. Azoz, L. Devi, and R. Sharma, "Reliable Tracking of Human Arm Dynamics by Multiple Cue Integration and Constraint Fusion", in Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98), pp. 905-910, 1998.[Azoz, 1998]

[5] D. H. Ballard, Christopher M. Brown, Computer Vision, Prentic-Hall, INC., New Jersey, USA, 1982. [Ballard, 1999]

[6] M. S Bartlett, H. M. Lades, and, T. Sejnowski, "Independent Component Representation for Face Recognition in Proceedings of Symposium on Electronic Imaging (SPEI): Science and Technology", pp. 528-539, 1998. [Bartlett, 1998]

[7] C. Bartneck, M. Okada, "Robotic User Interface", in Proceeding of Human and Computer Conference (Hc-2001), Aizu, pp. 130-140, 2001. [Bartneck, 2001]

[8] P.N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, Eigenfaces vs. Fisherfaces:" Recognition

[9] Using Class Specific Linear Projection" IEEE " Transaction on Pattern Analysis and Machine Intelligence (PAMI)" , Vol. 19, pp. 711-720, 1997. [Belhumeur, 1997]

[10] M. A. Bhuiyan, V. Ampornaramveth, S. Muto, and H. Ueno, "On Tracking of Eye For Human-Robot Interface", International Journal of Robotics and Automation, Vol. 19, No,1, pp. 42-54, 2004. [Bhuiyan, 2004] 178 Face Recognition

[11] M. A. Bhuiyan, V. Ampornaramveth, S. Muto, H. Ueno, "Face Detection and Facial Feature Localization for Human-machine Interface" , NII Journal, Vol.5, No. 1, pp. 25-39, 2003[Bhuiyan, 2003]

