A Vision based Hand Gesture Interface for Controlling VLC Media Player

Siddharth Swarup Rautaray
Indian Institute of Information Technology
Allahabad

Anupam Agrawal
Indian Institute of Information Technology
Allahabad

ABSTRACT

Human Computer Interaction can acquire several advantages with the introduction of different natural forms of device free communication. Gestures are a natural form of actions which we often use in our daily life for interaction, therefore to use it as a communication medium with computers generates a new paradigm of interaction with computers. This paper implements computer vision and gesture recognition techniques and develops a vision based low cost input device for controlling the VLC player through gestures. VLC application consists of a central computational module which uses the Principal Component Analysis for gesture images and finds the feature vectors of the gesture and save it into a XML file. The Recognition of the gesture is done by K Nearest Neighbour algorithm. The theoretical analysis of the approach shows how to do recognition in static background. The Training Images are made by cropping the hand gesture from static background by detecting the hand motion using Lucas Kanade Pyramidical Optical Flow algorithm. This hand gesture recognition technique will not only replace the use of mouse to control the VLC player but also provide different gesture vocabulary which will be useful in controlling the application.

Keywords

VLC player, recognition, gesture, human computer interface.

1. INTRODUCTION

WIMP (windows, icons, menus, pointers) prototypes, together with the keyboard and the mouse, have been definitive in providing the flexibility for use of computers machine. It provides users a clear objective model of what task, instructions to perform and their possible outcomes. These paradigms permit a user a sense of achievement and obligation about their interaction with computer application. [1]. By the underlying prototype, users express their significance to the computer user's using their hand to perform button clicks, positioning the mouse and key presses. This is a rather unnaturally a restrictive way of interaction with end user systems. In our everyday life, computers are comely more and more pervasive. It is highly worthy that the interaction with the systems does not essentially differ from the natural interaction taking place between different users. Perceptual User Interfaces (PUI) is the basis in which they are interested with extending Human Computer Interaction (HCI) to use all modalities of human perception.

Early development of PUI, it uses vision-based interfaces which perform online hand gesture recognition and also one of the finest approaches. High precision and speed is the major advantages of hand gesture. The most successful tools like mice, joysticks and keyboards are capable for HCI, as they have been thoroughly certified. Humans learn easily how to perform them, accomplish the most divers and complex tasks. These interfaces based on computer vision techniques are also modest and economical, making them perfect.

Traditionally HCI uses different types of hardware devices like instrumented gloves, sensors, actuators, accelerometers for integrating gestures as an interface for interaction. But these devices do not provide flexibility for interacting in real time environment. However, in HCI a number of applications related to hand gesture recognition exist. The applications designed for gesture recognition generally requires restricted background, set of gesture command and a camera for capturing images. A number of applications related to gesture recognition are designed for presenting, pointing, virtual workbenches, VR etc. Gesture input can be categorized into different ways [2]. One of the types is deictic gestures which refer to pointing an object or reaching for something. Accepting or refusing an action for an event is termed as mimetic gestures. It is utile for language representation of gestures. An iconic gesture is way of defining an object or its features. Pavlovic *et. all* [3] concludes in this paper that the gestures perform by the users should be logically explainable for designing the human computer interface, as the cutting edge in the domain of computer vision based techniques for gesture recognition is not in a state of providing a acceptable solution to this problem. A major challenge evolves is the complexity and robustness linked with the analysis and evaluation for recognition of gestures. Different researchers have proposed different pragmatic techniques for gesture as an input for human computer interfaces. Liu and Lovell [4], proposed a technique for real time tracking of hand capturing gestures through a web camera and Intel Pentium based personal computer without any use of sophisticated image processing techniques and hardware.

In this paper we presents an application which is designed for human computer interaction which uses different computer vision techniques for recognizing hand gestures for controlling the VLC media player. The aim and objectives of this application is to use a natural device free interface, which recognizes the hand gestures as commands. The application uses a webcam which is used for image acquisition. To control VLC media player using defined gesture, the application focuses on some function of VLC which are used more frequently. Figure 1 shows the defined function. The rest of paper is organized under following sections: architecture

design of the application is shown in section 2.Section 3 covers overview of computer vision techniques used in the application. Section 4 shows the methodology designed for the application. Application results are highlighted in section 5. Section 6 shows the testing and analysis of the. Conclusion in section 7 with future work in section 8 is discussed. References used by the application are shown in section 9.

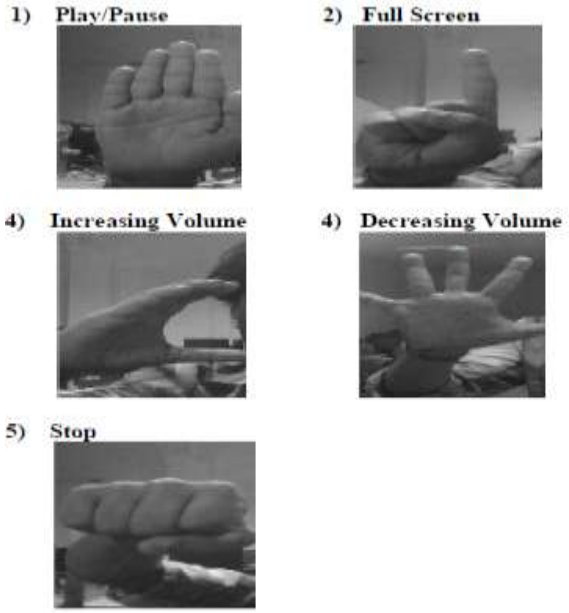

Figure 1. Gesture defined functions.

2. ARCHITECTURE DESIGN

The application uses a hybrid approach for hand gesture recognition. It recognizes static hand gestures. Figure 2 shows the architecture design of VLC control player. Images are captured from camera and passed through following phases/algorithms:

Making of Efficient Training Image: Aim of this phase is to increase information of the object of interest (gestures) in captured images by following steps:

- Detection of hand from streaming video by using Lucas Kanade Pyramidical Optical Flow [5], [6] algorithm. It detects moving points (hand) in image.
- It passes the above moving points to K-MEAN [7], [8] algorithm to find center of motion which is equivalent to the center of moving hand.
- Generate a rectangle around this motion center and crop the region within this rectangle.
- After cropping save image to a specific location for learning or directly use for recognition.

Learning Phase: After getting efficient images from above operations these are used for training. Principle Component Analysis [9], [10] algorithm is used for training. This gives a feature of images which is saved in a XML File.

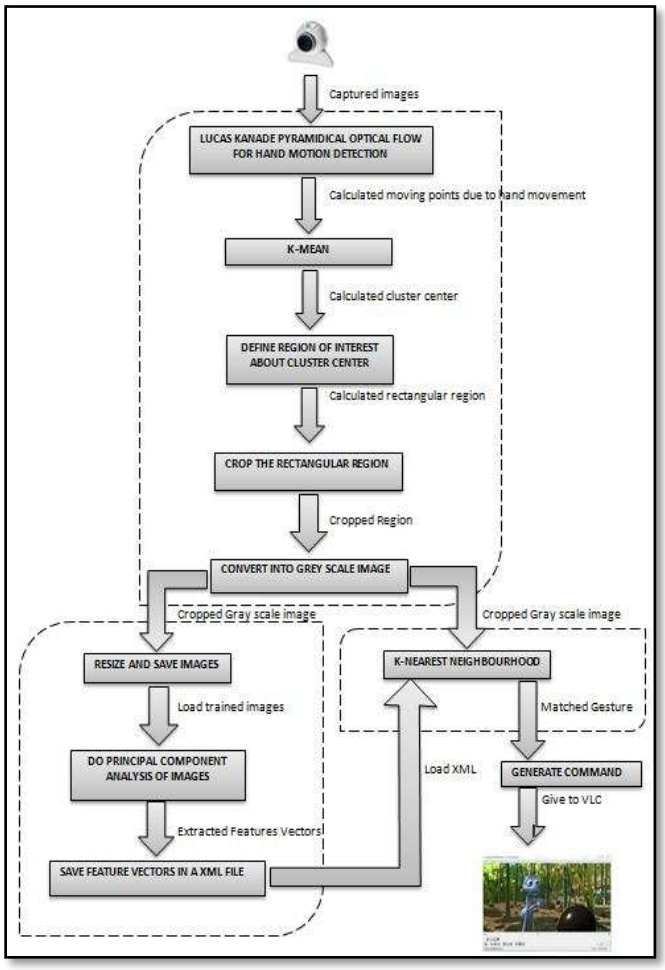

Figure 2. Architecture design.

Recognition Phase: Taking efficient images from a camera and passing directly to K-Nearest Neighbourhood [11] for matching with previous stored gesture database.

VLC Interaction: Now according to recognized gesture sending a pre-defined command to VLC to perform appropriate action.

3. COMPUTER VISION TECHNIQUES

Tools and Techniques for VLC application: The application uses a hybrid approach for hand gesture recognition which recognizes static hand gestures. The images are captured from camera and then passed to different algorithms for learning and recognition. The computer vision techniques used for the application are discussed below:

Pyramid Lucas-Kanade Optical Flow: This is used for recognition of gestures. Hand detection is done using two techniques i.e. skin color and motion tracking. Motion tracking is done using Lucas-Kanade Optical Flow algorithm. Figure 3 shows the optical flow field generated by optical algorithm.

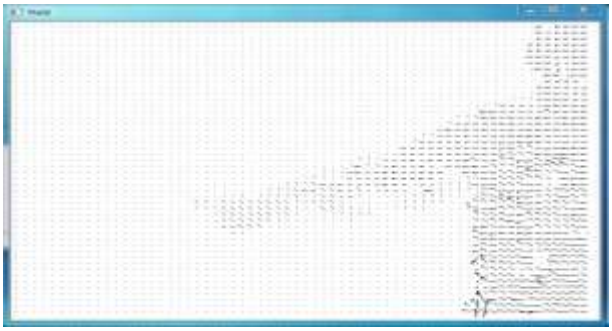

Figure 3. Optical Flow Field generated by Optical Flow Algorithm.

K-Mean Algorithm: Optical flow generates a vector of moving point. These moving points are arranged in clusters for further processing like cropping, resizing etc. K-Mean [7] is used for clustering. Its process can be defined as if there are N given points, where each point is a *d*-dimensional , then *k*-means clustering partitions the N points into *k* sets ($k < n$) $\mathbf{S}=\{S_1, S_2, \ldots, S_k\}$ so as to reduce the within-cluster sum of squares:

$$arg_s \min \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

where μ_i is mean of S_i

In this application the input for the algorithm is the x, y co-ordinate of the points generated by optical flow. There are two vectors x1 and y1. x1 has the x co-ordinate of points and y1 has y co-ordinate. At the output K-Mean returns the cluster center that is used for clipping etc. Figure 4 shows the generated cluster.

Figure 4. Generated cluster

Principal Component Analysis: This algorithm is used for extracting common features of all images and further reducing its dimension. Following are the steps involved in PCA technique:

Calculating the empirical mean [9]

- Find the empirical mean along each dimension $m = 1...M$.
- Place the calculated mean values into an empirical mean vector **u** of dimensions $M \times 1$.

$$u[m] = \frac{1}{N}\sum_{i=1}^{N} X[m,n]$$

Calculating the deviations from the mean

- Store mean-subtracted data in the $M \times N$ matrix **B**.

$$B = X - uh$$

Where h is identity matrix

Finding the covariance matrix

$$\mathbf{C} = \frac{1}{N}\sum \mathbf{B}.\mathbf{B}$$

Find the eigenvectors and eigenvalues of the covariance matrix

- Compute the matrix **V** of eigenvectors which diagonalize the covariance matrix **C**:

$$V^{-1}CV = D$$

Where **D** is the diagonal matrix of eigenvalues of **C**.

- Sort the columns of the eigenvector matrix V and eigenvalue matrix D in order of decreasing eigenvalue.
- Remove the smaller eigenvalue.

The input for the algorithm is matrix of image size (M*N) containing information about each pixel of image. The output matrix is of common features with reduced dimensions. These features are saved in an XML file.

K-Nearest Neighbourhood: This algorithm is used for recognition which takes the input image and recognizes the class from which it belongs. The K-NN algorithm can summarized as follows:

K-nearest neighbors algorithm (*k*-NN) is a technique for classifying objects which is based on selecting closest training examples in the featured space [11].

- An arbitrary 0instance is represented by ($a^1(x)$, $a^2(x)$, $a^3(x)$,.., $a^n(x)$)
- $a^i(x)$ denotes features
- Euclidean distance between two instances
  d (x^i, x^j)=sqrt (sum for r=1 to n ($a^r(x^i)$ - $a^r(x^j)$)^2)

The input used for the algorithm is in the matrix form of the eigenvalues. When an input frame passes it calculate eigenvalues of this image and pass it to algorithm as an input parameter. In the output the function returns an integer value which indicates from which gesture the image is matching. Figure 5 shows the Communication between training and testing phase.

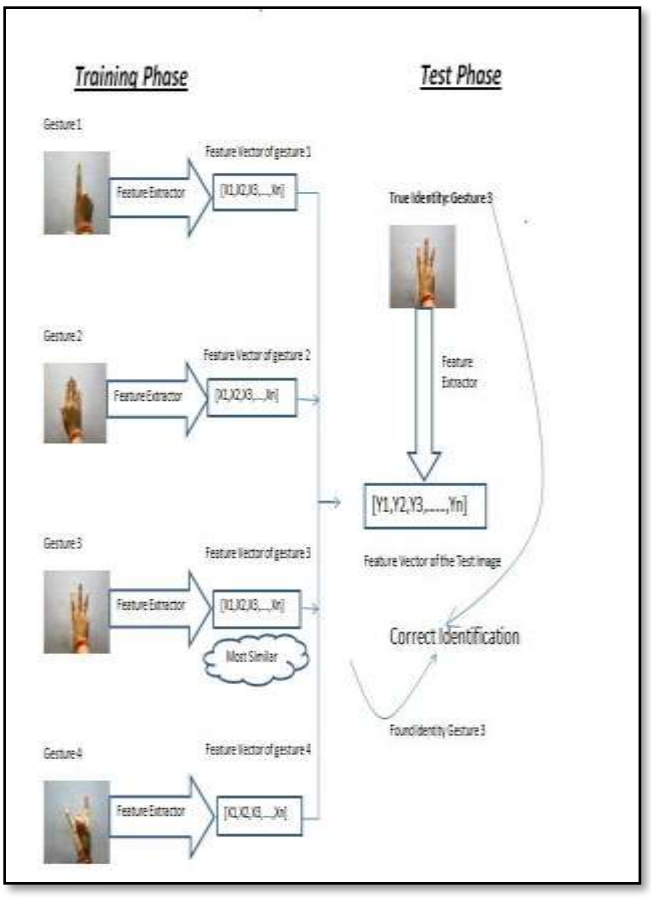

Figure 5. Communication between training and testing phase. KNN lies in between both phases

4. APPLICATION METHODOLOGY

The methodology used for the application as follows:

Hand Segment: The task is to convert camera input frame into an image which has more information. Following are the steps:

Changing Image to Gray Scale: In this image is first transformed to gray scale color. In optical flow consist of three assumptions, one of which is Brightness constancy**.** For maximizing brightness constancy image needs to be converted to gray scale color.

Detecting Moving Points: For detection of hands generally two strategies are followed:

- Skin color
- Tracking of moving hand

This application uses the second method. The hand is moved in front of camera in a non-moving background when the user uses it or at the time of feature learning. Lucas Kanade Optical flow technique is applied for tracking the moving points from a streaming video by comparing previous frame with current frame.

Making Clusters: After detecting moving points which are done through optical flow the points needs to be clustered which generates more information about the image. K-Mean algorithm is used for this purpose.

Cropping of Images: The clusters are cropped and stored in a different image. The cropped image moves the background, noise etc which generates more percentage of information than previous image. First creating a rectangle around the clusture and clipping that rectangle to a new image.

Saving Images: After cropping the image is saved for learning process. During learning it reads all saved image and apply algorithms.

Learning Segment: Learning phase is divided into two parts:

- **Extracting Features:** The application uses 15 positive images for each gesture used. All the images are loaded from their corresponding address where the PCA is applied for extracting features.Figure 6 shows the input image to be train.

Figure 6. Input Image to be train

- **Saving of Features:** We have save features extracted by PCA. And also save some important information.

Recognition Phase: Recognition phase is divided two parts:

- **Loading of XML Document:** For recognizing, the application loads the xml document.

- **Matching:** Matching of input images is done with the loaded xml data to decide which gesture it matches. Matching is done by using KNN.

VLC Interaction: After recognition phase an integer value is obtained. The value of gesture matches with the input gesture. Further generation of a equivalent keyboard event [8] of the hotkey that is predefined to perform user's intended action.

Defining Own Gestures: The application provides flexibility to users for defining his gestures for controlling VLC functions. The figure 7 shows an interface the learning process in the following steps:

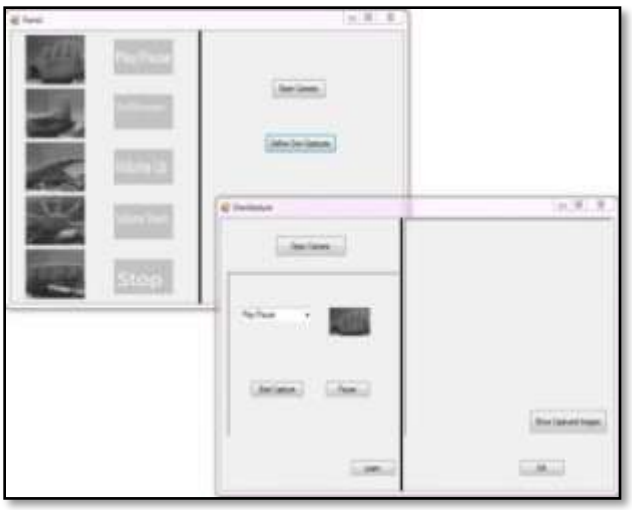

Figure 7. Interface for defining own gestures

Step 1: First click on Open Camera button and then select which gesture wants to redefine and then click on start capture. Figure 8 shows while defining own gestures select gesture and click start capture.

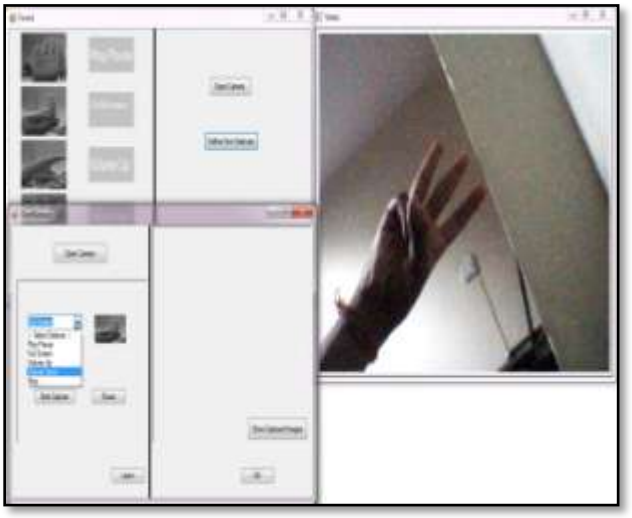

Figure 8. While defining own gestures select gesture and click start capture.

Step 2: After 15 images a pop-up message will appear with message that 15 images captured. After that captured images are displayed by clicking on "Show Captured Imaged" as shown in figure 9.

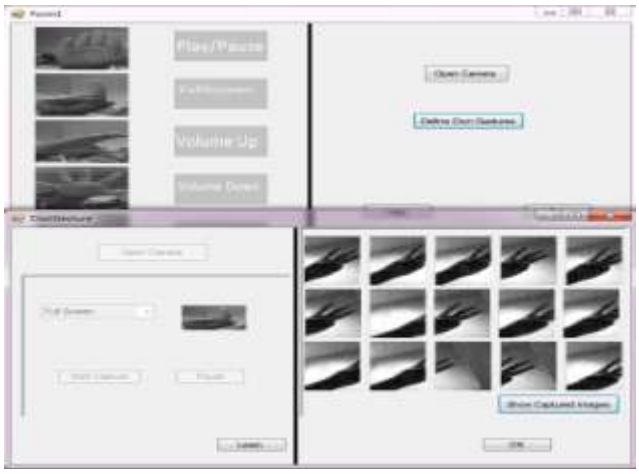

Figure 9. After capturing 15 images you can see all by clicking "Show Captured Images"

Step 3: In the last step for learning process click on "learn". Once the process will complete a message box will pop with message that Learning Complete.

5. RESULTS

Following figures shows the results obtained of different gestures used to control the VLC player.

Figure 10. Play/Pause

Figure11. Full Screen

Figure12.IncreseVol.

Figure 13 .Decrease Vol.

Figure 14. Stop

6. TESTING AND ANALYSIS

Testing of Learning Phase: For increasing efficiency the application captures 15 images of each gesture. At the time of recognition all gestures are recognized with less robustness. This

shows that the features of all gestures are present in XML file. This makes the learning phase successfully tested.

Testing of Recognition Phase: Table 1 shows the hand gesture recognition results obtained from the test images stored in the database.

Table 1. Hand Gesture Recognition Results

Gesture	No. of images stored	No. of hits	No of misses	Recognition rate (%)
Play/Pause	15	15	0	100
Full Screen	15	15	0	100
Increase Vol.	15	11	4	73.33
Decrease Vol.	15	14	1	93.33
Stop	15	13	2	86.67

Figure 15 compares the recognition rate of the gestures recognized with the images saved based on the number of hits and misses of different gesture commands used for controlling the VLC application. Due to the noisy background and gesture shapes performance of some gesture decreases. For increasing performance of the application more number of test images needs to be stored and taking decision for functions of VLC according to max recognized gesture.

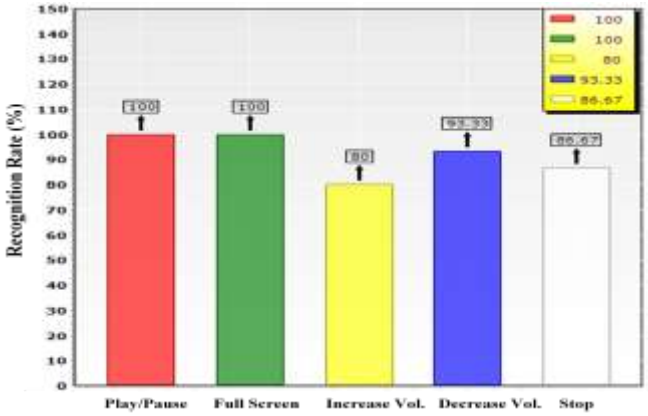

Figure 15. Comparison of different gestures recognition rate.

7. CONCLUSION

In current world many facilities are available for providing input to any application some needs physical touch and some without using physical touch (speech, hand gesture etc.).But not many applications are available which are controlled using current and smart facility of providing input which is by hand gesture .By this method user can handle application from distance without using keyboard and mouse. This application provides a novel human computer interface by which a user can control media player (VLC) using hand gesture. The application defines some gesture for controlling the functions of VLC player. The user will provide gesture as an input according to interested function. The application provides a flexibility of defining user interest gestures for specific command which make the application more useful for physically challenged people, as they can define the gesture according to their feasibility.

8. FUTURE WORK

The present application is less robust in recognition phase. Robustness of the application can be increased by applying some more robust algorithms to reduce noise and blur motion.

For controlling VLC, presently the application uses global keyboard shortcut in VLC and making keyboard event of that global shortcut with keybd_event () function. It's not the smart way of controlling any application. Inter-process communication technique can be applied for this. By applying inter-process communication then VLC can be replaced with other application very easily.

9. REFERENCES

[1] Turk, M. and Robertson, G. 2000. Perceptual user interfaces. *Communications of the ACM*, 43(3), (March 2000).

[2] Liu, J., Pastoor, S., Seifert, K. and Hurtienne, J. 2000. Three-dimensional pc: toward novel forms of human-computer interaction. *In Three-Dimensional Video and Display: Devices and Systems SPIE* CR76, (2000).

[3] Pavlovic, V., Sharma, R. and Huang, T. S. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. on Pattern Analisys and Machine Inteligence (PAMI)*, 7(19):677–695.

[4] Liu, N and Lovell, B. 2001. Mmx-accelerated realtime hand tracking system. In *IVCNZ*, (Nov. 2001), pp. 26–28.

[5] Ki-Sang, K. and Dae-Sik, J. 2007. Real time face tracking with pyramidal lucas-kanade feature tracker, Computational science and its applications. *ICCSA* (2007). 4705: 1074–1082.

[6] Z. Vamossy, Z., Toth, A. and Hirschberg, P. 2004. PAL Based Localization Using Pyramidal Lucas-Kanade Feature Tracker. In *Proceedings of the imposium on Intelligent Systems.* (2004), 223-231.

[7] Kang, M and Kim, J. 2007. Real Time Object Recognition Using K-Nearest Neighbor in Parametric Eigenspace," *Lecture Notes in Computer Science*, Vol. 4688/2007, (2007), 403-411.

[9] Kim, J, Heo, J, Yang, H, Song, M, Park, S and Lee, W. 2006. Object Recognition Using K-Nearest Neighbor in Object Space," *Lecture Notes in Computer Science*, Vol. 4088/2006, (2006), 781-786.

[10] Smith, L. 2002. *A tutorial on Principal Components Analysis.*

[11] Shamaie, A, Hai, W and Sutherland, A. 2001. Hand gesture recognition for HCI", *ERCIM News (on line edition),http://www.ercim.org/publication/Ercim News*, no. 46, (2001).

[12] Yeung,C.M.A, Gibbins, N and Shadbolt, N. A. 2008. k-nearest-neighbour method for classifying web search results with data in folksonomies. *International Conference on Web Intelligence and Intelligent Agent Technology*, (2008). 70–76.