International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 2 243 - 249

# Autonomous Vision Based Facial and voice Recognition on the Unmanned Aerial Vehicle

Sanjaa Bold*
Dept. Of Computer Hardware and Network
University of the Humanities
Ulaanbaatar, Mongolia
*Snj_goopy@yahoo.com*

Batchimeg Sosorbaram
Dept. of Computer hardware and Network
University of the Humanities
Ulaanbaatar, Mongolia
*chimeg@humanities.mn*

Bat-Erdene Batsukh
Dept. of Computer software
University of the Humanities
Ulaanbaatar, Mongolia
*b.ba36@yahoo.com*

Seong Ro Lee
Dept. Electronic Engineering
Mokpo national university
Mokpo, South Korea
*srlee@mokpo.ac.kr*

***Abstract—*** The development of human navigation and tracking in the real time environment will lead to the implementation of more advanced tasks that can performed by the autonomous robots. That means, we proposed new intelligent algorithm for human identification using difficult of facial and speech which can substantially improve the rate of recognition as compared to the biometric identification for Robust system development. This project system that can recognize face using Eigenface recognizer with Principal component analysis (PCA) and human voice using the Hidden Markov Model(HMM) and. Also in this paper, combinations of algorithms such as modified Eigenface, Haar-Cascade classifier, PCA and HMM resulted in a more robust system for facial and speech recognition. The proposed system was implemented on AR drone 2.0 using the Microsoft Visual Studio 2015 platform together with EmguCV. The testing of the proposed system carried out in an indoor environment in order to evaluate its performance in terms of detection distance, angle of detection, and accuracy of detection. 500 images of different people were used for face recognition at detection distances. The best average result of 92.22% was obtained at a detection.

***Keywords-*** *PCA, Eigen face recognizer, HMM, OpenCV, VisualStudio2015, AR drone 2.0*

*****

## I. INTRODUCTION

In this paper was talking about interaction between human and robust recognition system. [1] Which means, we are developed to more robust recognition system. This robust system [2] was combined many different recognition algorithms to facial and speech recognition system. These complex algorithms, we called for biometrical robust recognition algorithm. This biometric identity authentication system is based on the biological characteristics of a person, such as face, voice, fingerprint, iris, gait, hand geometry or signature. [3]

So, we are successful to combined this algorithm on the AR drone. An air robot like the Drone UAV's [2] was our choice of such a platform to test our tracking algorithm because of its ability to navigate in most common environments with ease and without much need for obstacle avoidance at human height level [4]. In this project to develop a robust recognition system, we divided the task of recognize in different stages [2] of operation. This means, the Drone is to search its field of view for any human faces using the face detection algorithms that we implemented to determine if there is a person in close range. [2] Therefore, the drone is to search its field before waiting for the human voice command. [5] This means, the drone based recognition algorithm is to show that human-robot face engagement can be used to determine the subject or subjects of verbal commands using indirect speech. [6]

Other important task is Speech recognition and automatic speaker recognition ASR system. The behavioral biometrics, speech is the most convenient parameter. We focus is on text-independent Automatic speaker recognition (ASR) which is considered to be a more challenging problem. Research in the field of speaker recognition traces back to the early 1960s when Lawrence Kersta at Bell Labs made the first major step in speaker verification by computers. [7] He proposed the term voiceprint for a spectrogram, which was generated by a complicated electro-mechanical device. Since then, there has been a tremendous amount of research in the area. Starting from spectrogram comparisons, passing through simple template matching, dynamic-time wrapping, to more sophisticated statistical approaches like Gaussian Mixture Model (GMM), Hidden Markov Model (HMM) and neural networks. [7] Mel Frequency Cepstral Coefficients or MFCC features have also been used for text-independent speaker recognition. See the figure1.1.

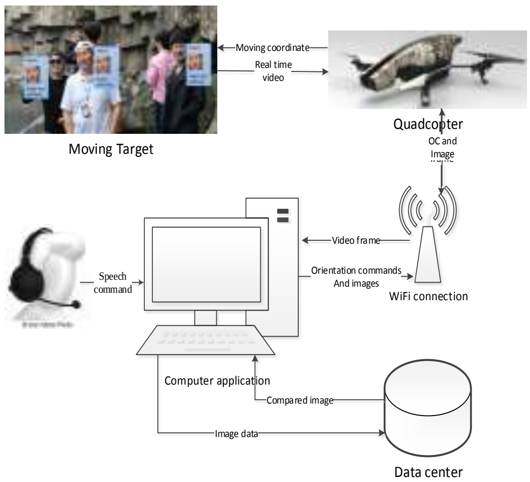


Figure 1.1. The general algorithms of the facial and voice recognizer

## II. ALGORITHMS

### A. *Facial recognition algorithms*

The Eigenface problem is one that aims to provide a means of facial recognition. To begin, a collection of images of human faces needed. Given an input images, we would like to determine whether the image is a face, as well as if the image matches one of the images in the library [8]. Moreover, the Eigen face technique relies on information theory and utilizes PCA [8], the method of reducing dimensionality while preserving the variance of a data set, to recognize facial feature. [9] [10] More specifically, the principal components used in the Eigen face technique are eigenvector of the covariance matrix of face images, while each face is a point in n space where n is the number pf pixel in each image. [11]

The relevant information of a face image needs to be extracted and compared to a previously defined database of images. One convenient measure of facial image information is in the variation of the data of face images. More importantly, this variation will avoid predisposition towards focusing on facial features. We will be seeking the principle components of the face image distribution in the library or training set of face images (recall that shown to the eigenvector of the covariance matrix of the set of data).

Two variables are takes the Eigen recognizer [11]. See fig. 2. The first is the number of components kept for this Principal Component Analysis (PCA). There is no rule how many components that should kept for good reconstruction capabilities.

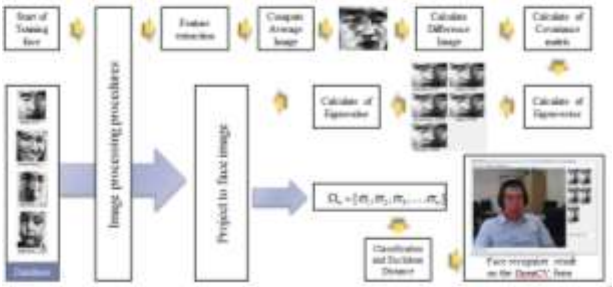

Figure. 2. The Eigen face recognizer based face recognizer algorithm.

It I based on your input data, so experiment with the number. The second variable designed to be a prediction threshold [9] [12], this variable contains the bug as any value above this is considered as an unknown. However, the Eigen recognizer we must use the return distance to provide our own test for unknown. In the Eigen recognize the larger the value returned the closer to a match we have [8] [10]. Then examine the Eigen distance return after recognition, if it is above the threshold we set in the form that it recognized if not then it is an unknown.

This algorithm faces and objects detection and included the drone controller system. In addition, main form can be select for the command and then we can do data analysis. The camera to come from video, this video received to the computer and then this data based the human face detection ability on the video.

### B. *Speech recognition algorithms*

This project development of a voice recognition system has a challenging process. For each country language, you have to create different voice recognition software; hence voice recognition software in different languages has to solve the problems caused by specific features in each language. Despite all these difficulties, an effective voice recognition system can provide great benefits in many areas. For example, it will be making easier to do business with call centers, cargo companies, restaurants, military services and search and rescue teams, also with voice recognition system saves money, time and loss of manpower may be prevented. In addition, the voice recognition system can also be used for text. An advanced voice recognition system allows the machine to give voice as a commands.

## III. SOFTWARE STRUCTURE

### A. *Facial recognition software structure*

This drone at the start of a program using the Eigen face recognizer as the default. In this OpenCV version, face recognizer selection can make through the menu section of the main form. The show as Figure 3.1. The first training made after training data has been added or created is controlled by this selection. The ability to save the face recognizer for exportation to other application was being introduced [13].

Figure. 3. Main form of the face recognition

It is much important to note, that we will need the original data to add an additional face. Fig. 4. shows the trains from allowing for a face to recognized and added individually as the program designed to run from AR drone camera the face is recognize in the same method. A feature to acquire successful faces classification and adds them all or individual ones with the training data have been included. This increases the collection of training data and the amount of image acquire can adjusted in the variable region of in the training form. [14] [15] If different sets of training data, then another contractor carries a string containing the training folder.


Figure. 4. The original data to add additional face

Loading the face recognizer achieved by using the inbuilt load method. The file extension for the recognizer used to determine the constructor required for the face recognizer.

 ISSN: 2321-8169
243 - 249

If we cannot detect unknown face and main form includes a threshold base for calibration the threshold value. This value changes depending on the size or training data. However, to aid in calibration the origin distance is printed next to the name when recognized. The recognize method examine to see if the engine distance is greater than the set threshold. If not an [13] "unknown face" label returned instead of the last person's face name within the database.

### *B. Traing faces*

The training of the face recognizers consistent between all 3 class members of the recognizer. The forum starts of a program using the Eigen face recognizer as the default. The Face recognizer menu selection of the main form. The training made after training data has been added or created is controlled by this form. This form ability to save the Eigen recognizer for exportation to other applications was introduced.it is important to note however, that you will need the original training data to add additional faces.

The training form allows for a face to recognition and added individually as the program is designed to run from an AR drone camera. A feature to acquire 10 successful faces classification and adds them all or individual ones with the training data have been included. This increase the collection of training data and the amount of images acquired can be adjusted in the variable region of Eigen face recognition forms.

So a classifier train class is included, it has two constructors the default takes the standard folder path of application and training faces, also the default save location of the training data. This location folder is our face database. The faces image files type is all of the jpg and size 100x100. [15]

### *C. Greating a new traing faces database*

The image added to the existing training form needed to be edited. This editing was done was cropping the face. This editing allows the Eigenfaces recognition algorithm to focus on extracting features from the face that are robust to illumination, background interference and changes in the subject's hair. A Haar cascade was used to perform this editing. The cropping function also served to reduce the size of the sample images to a standard size of 100x100 pixels that is used in the training form to database. The image cropping process can be seen in Fig. 5.

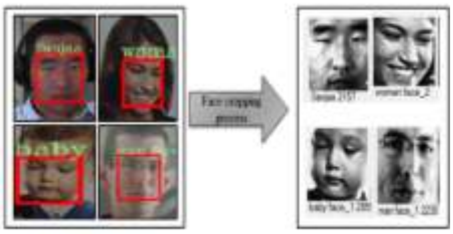

Figure. 5. Faces image cropping on the algorithms

The face image needed to be stored in a specific file structure of the Eigenface algorithm to group samples of the same face as well as to provide a label to be returned when a specific face is recognized. This file structure required all images of the same face to be contained separate folder. See fig. 6. These folders are then required to be contained within a separate folder. This one folder is referenced as the single folder for the Eigenfaces algorithm to model faces. The file structure of the face images is illustrated in figure 5.10


Figure. 6. All faces database.

### *D. Speech recognition software structure*

Voice recognition systems, can be designed by people independently or person-dependent voice recognition Voice commands in the voice recognition system can provide a model for the whole. This model is professional especially suitable for limited speech recognition systems. It will be recognized when it is more than the number of voice and sound are utilized for modeling phonemes. Audio unit with audio/voice recognition and command recognition method is applied continuously. In these systems, the commands are variable and if easy integration of these commands to the system is required continuous speech recognition methods should be preferred. Recently, all models of the voice recognition systems used hidden Markov models (HMM) with Common Vector Approach (CVA). [3] In this study, a subspace classification method for recognizing voice commands which is called CVE (Common Vector Approach) method is also used. CVA results are given close to the performance of HMM. In addition, CVA process has advantages in terms of training and recognition based on HMM. We control the car by giving these commands; "Are you ready", "Take Off", "Fly", "Run", "Stop", "System Deactivate", "Security", "Hover", "Land", "Up", "Down", "Go Left", "Go Right", "Go Back", "Move On", "Turn Left", "Turn Right", Find Human", "Find face", "Stop face detection", "Wait", "Add new face ".

### *E. EMGUCV based drone operation applications*

We are combined to C sharp and Visual basic code created. Also we have to use for OpenCV libraries. The library allows the developer to take control of navigation data and the video stream for the development of the PC based application. See Fig. 7. Usually AR drone visual based control application feature levers and trims for controlling UAV pitch, roll, yaw and throttle.

Input

| | |
|---|---|
| Pitch | +0.000 |
| Roll | +0.000 |
| Yaw | +0.000 |
| Gaz | +0.000 |
| VX | +0.000 |
| VY | +0.000 |
| VZ | +0.000 |

Figure. 7. All of controlling status and value

Basic maneuvers include take off, trimming, hovering with constant altitude, and landing. It generally takes hours for a beginner and many UAV crashes before executing safely these basic maneuvers. The Drone on-board sensor take-off, hovering, trimming and landing are now completely automatic and all maneuvers re completely assisted. See Fig. 8. For examples:

- When landed push *take-off* button to automatically start engines, take-off and hover at a pre-determined altitude [9].
- When flying push landing button to automatically land and stop the engines
- Press turn left button to turn the AR drone automatically to the left at a pre-determined speed. Otherwise the Drone automatically keeps the same orientation [9].
- Press turn right button to turn the Drone automatically to the right. Otherwise the Drone automatically keeps the same orientations
- Press up button to go upward automatically at a predetermined speed. Otherwise the Drone automatically stays at the same altitude.
- Press down button to go downward automatically at a predetermined speed. Otherwise the Drone automatically stays at the same altitude [9].

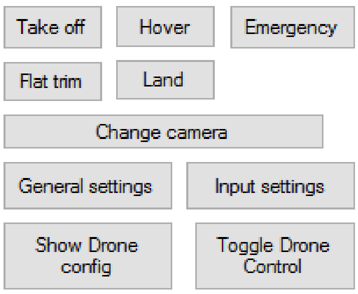


Figure. 8. Some basic function of the maneuvers

## IV. HARDWARE

### A. Motion control model of the Ar drone

In this section, we will describe how the model has been derived from moving the AR drone. Also, we will start by defining the drone as mass M, affected by a gravitational force, and a lifting force $f_a$ caused by the aerodynamics of the propellers. See fig. 9. In the equilibrium scenario with the drone lifted from the ground, these two forces will be equal in size pointing opposite directions.

The move the drone, the internal regulation tilts the body with an angle $\theta$ around the $y-axis$ of the drone, according to the control input. During this, the aerodynamic force will automatically adjust to keep the upward pointing force $f_{up}$ compensate $f_g$ keeping the drone at a fixed height [15]. Those results in a horizontal pointing net force $f_x$. To translate this to a motion model, we use Newton's second law of the motion.

$$f_x = \tan(\theta) * f_{up} \qquad (1)$$

$$f_{net} = M * \alpha \qquad (2)$$

This gives us the acceleration for the motion model:

$$\alpha = \frac{\tan(\theta) * f_{up}}{M} = \frac{\tan(\theta) * (gM)}{M} = \tan(\theta) * g \qquad (3)$$

To complete the motion model we have to know the initial position and velocity $x_0$ and $\upsilon_0$. For this, we will assume that the position and velocity are zero at initial time $t_0$. This gives the velocity and position from the integrals. Eq. (4) and Eq. (5)

$$\upsilon(t) = \int_{t_0}^{t} \alpha(\tau) d\tau + \upsilon_0 \qquad (4)$$

$$x(t) = \int_{t_0}^{t} \upsilon(\tau) d\tau + x_0 \qquad (5)$$

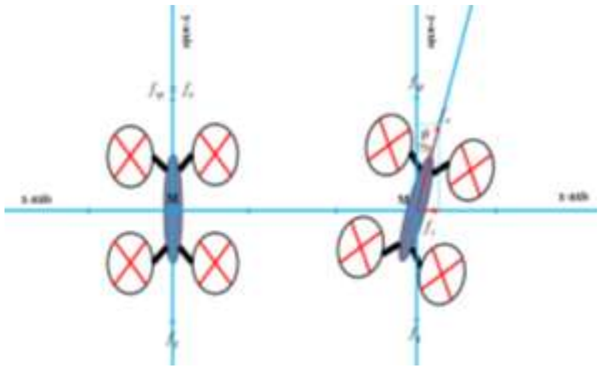

Figure. 9. Forces acting on the body of the AR drone.

Using this, we make the simplification of the horizontal motion, that it is unaffected by any viscous damping proportional to the velocity and only affected by the acceleration, controlled though the pitch angle $\theta$. To control the rotational motion of the AR drone we have the control input $\dot{\psi}$, which is the angular speed of the AR drone, the derivative of the yaw $\psi$. Assuming we have a known initial yaw $\psi_0$ and no viscous damping, the yaw will be following this equation:

$$\psi(t) = \int_{t_0}^{t} \dot{\psi}(\tau) d\tau + \psi_0 \qquad (6)$$

Further, we have the gas and the roll as a control to the drone, the vertical speed and the tilt around the $x-axis$ of the AR drone orthogonal to pitch direction, respectively. However, for now we leave these out of the model for simplicity. The model of motion will bring us to the discrete state model used by the AR drone for detection algorithm to keep track of its own position.

At this point, it should be clear that the acceleration input $\alpha$ to the state space model is calculated from the pitch angle.

In this case, we follow this Eq. (3). To set this pitch angle and hence the horizontal motion, we need a mechanism to calculate optimal values for the controls. For this purpose, we will use a discretized version of a PID controller [50]. That meaning is the proportional part of the PID has actually been used during this project. The error $e_x$ is the signed distance to the target, giving us the target velocity $\upsilon^*$ as an output.

$$\upsilon^*(t) = K_p e_x(t) + K_i \int_{t_0}^{t} e_x(\tau)d\tau + K_d \frac{d}{dt} e_x(t) \qquad (7)$$

As the error, or the signed distance to the target, we have decided to use the point (dot) product between the AR drones direction unit vector ($\hat{e}_x$) and the vector ($\bar{r}$) spanning from the AR drone to just above the target.

$$e_x = \bar{r} * \hat{e}_x \qquad (8)$$

The $x_{drone,m}$ and $y_{drone,m}$ equals the length of the projection of $\hat{e}_x$ onto $\bar{r}$. Intuitively, this gives an error at zero, it the drone is orthogonal to the target $|\bar{r}|$, if the AR drone has its front against the target and -$|\bar{r}|$if the back of the AR drone points against the target. That only allows this PID to bring the AR drone on line with the target according to Fig. 10. For the rotational movement, we will use another PID, directly controlling the angular velocity $\dot{\psi}$, used both for commands to the drone and as input to the recognition algorithm. The calculations for the rotational PID are similar to those for the moving PID [50]. The difference is how the error is evaluated. For the rotational PID, a simple solution could just be to use angle between $\hat{e}$ and, known as $\psi$ in the AR drones local coordinate system, as the error. However, this huge change in the yaw error, if the AR drone is just above the target. To compensate for that, we multiply the yaw by the distance to the target. The PID controller should bring the AR drone to the target, even if they operate independently. [16][17][18]

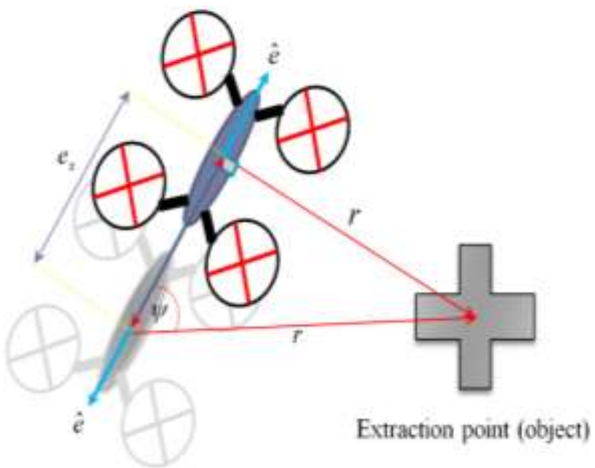


Figure. 10. Minimization of $e_x$ according to the point.

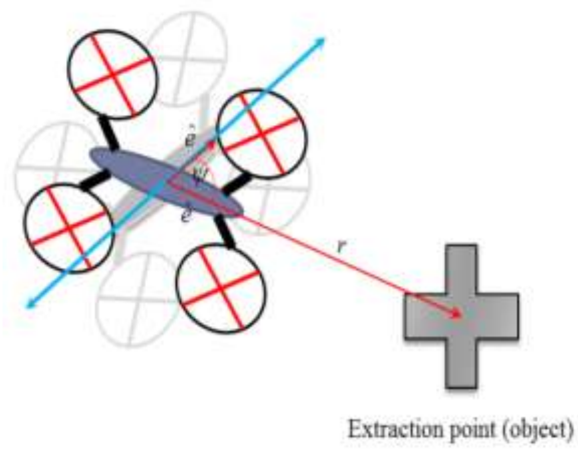


Figure. 11. Minimization of $e_\psi$ according to angle and distance.

## V. DISSCUSTION AND RESULTS

### A. Experiment

It was decided to use the frontal camera to face detection to avoid conflicts with the built in regulation of the AR drone. [**autonomous visual based face recognition UAVs**]. The face recognition seemed to be a good way of recognition objects from the front camera. Especially as a real time face recognition method is provided with EMGUCV library. This method is based Eigen face recognizer and Haar cascade classifier with PCA. Basically our algorithm is based on the Haar-like feature and same of the value determine to the PCA. How used Haarcascade classifier on the Eigen face detection algorithm. See fig. 12.

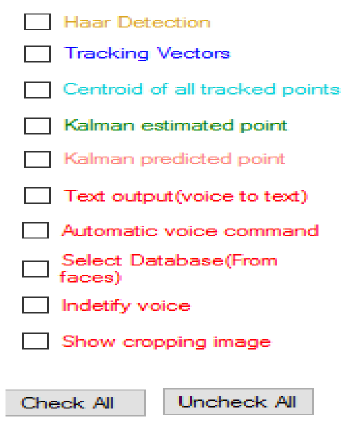


Figure. 12. Select of the detection and recognition and tracking function.

This means, the face recognizes, a cascade Haar classifier, finds an around of each face. Since, the classifier is trained on the frontal faces only, the number of such sub-windows increases when the human is directly looking at the camera. See fig. 13. At this time drone ready to fly. Basically, out drone will be waiting for human operation voice command. Also, this algorithm has a predetermine commands which are "Are you ready", "Take Off", "Fly", "Run", "Stop", "System Deactivate", "Security", "Hover", "Land", "Up", "Down", "Go Left", "Go Right", "Go Back", "Move On", "Turn Left", "Turn Right", Find Human", "Find face", "Stop face detection", "Wait", “Add new face " and also Quadcopter seamlessly detected by computer commands, and in tests of these commands are being applied smoothly. If calls for user to keyword “face detect” is ready to detect face, then drone will be finding for any human faces. See Fig. 14.

Figure. 13. The Eigenface recognition based face recognition algorithm running on the AR drone

Figure. 14. The human face recognition to following for the AR drone.

## VI. CONCLUSIONS

We created a Quadcopter which operates with the use of voice recognition systems. Also this voice recognition system based to applied full autonomous face recognition in the AR drone. The combination of many different algorithms in this paper resulted a great success. This means, algorithms created to perform autonomous function of recognition using the AR drone. Various methods and algorithm were used to implement each fraction and the capability of these algorithms was tested rigorously. Various strengths and limitations of the AR drone and the algorithm used to implement each of the functions were identified. The AR drone was the most suitable aerial vehicle to implement this system as it allowed each of the algorithms to be tested indoors in a controlled environment. So, that drone will automatically orient itself towards the detected object. But before finding, gives to some voice commands. After finding the target, the AR drone moves among human faces at constant speed. The drone basically control has been carried out using the smooth maneuvering algorithm. Since our algorithm is based on fast detection, we needed fast maneuvering algorithm for the AR drone. By this, the detection accuracy of the algorithm was increased. We used Eigenface recognize algorithm which was implemented well on the Eigenface recognition algorithm. It was used in PCA, Eigenface values, Haar-Cascade library, e.t. The result (See Figure 6.1) obtained from testing showed that the Eigenface recognizer algorithm was the most effective in terms of distance capabilities, effective object detection angle, detection and tracking accuracy performance. We have included Haar-Cascade algorithms on the Eigenface detection algorithm. The object is first recognized and the object position is then used as a set direction for the AR drone. In addition to the position, the program also provides some probability to estimate that the found position is correct. The system was successfully able to evaluate the capabilities of various algorithms and demonstrated that a micro UAV such as the AR drone is capable of performing autonomous detection, recognition and tracking during flight.

The testing of the proposed system carried out in an indoor environment in order to evaluate its performance in terms of detection distance, angle of detection, and accuracy of detection. 500 images of different people were used for face recognition at different detection distances. The best average result of 92.22% was obtained at a detection distance of 75cm. Also voice recognition algorithms responsed accuracy 90.2%.

## REFERENCES

[1] Sanjaa Bold , “Autonomous Vision-Based Moving Object Detection for Unmanned Aerial Vehicle” 2014, SCOPUS JOURNAL.Korea .
[2] Sanjaa Bold, Seong Ro Lee “FPGA Based Real Time Embedded Color Tracking Mobile Robot” 2013, ICITCS2013, IEEE Socity
[3] M. A. Goodrich and A. C. Schultz, “Human-robot interaction: A survey,” Foundations and Trends in Human-Computer Interaction, vol. 1, no. 3, pp. 203–275, 2007.
[4] Markus, Z. Tianguang, “Visual tracking and control of a quadcopter using a stereo camera system and inertial sensor,” IEEE Mechatronics and Automation, International Conference (ICMA2009), pp 2863-2869,August 2009
[5] M. Draper, G. Calhoun, H. Ruff, D. Williamson, and T. Barry, “Manual versus speech input for unmanned aerial vehicle control station operations,” in Proc. of the Human Factors and Ergonomics Society Annual Meeting, vol. 47, 2003, pp. 109–113.
[6] Serdar Gökhan Özkartal, “Development of a System for Human Language Commands and Control for a Quadcopter Application”, Journal of Manegement Research, 2015, Vol. 7, No.1.
[7] Bahl, L., Bakis, R., Cohen, P., Cole, A., Jelinek, F., Lewis, B., & Mercer, R. (1981). Speech Recognition of a Natural Text Read as Isolated Words. In Proc. *IEEE International* Conference on Acoustics, Speech, and Signal Processing.
[8] J.Shotton, A.Fitzgibbon, M.Cook, T.Sharp, M.Finocchio, R.Moore, A.Kipman, and Andrew Blake, “Real-time human pose recognition in parts from single depth images,” IEEE Computer Vision and Pattern Recognition, pp.1297-1304, June 2011.
[9] S. Liu, Y. Wang, L. Yuan, J. Bu, P. Tan, and J. Sun, “Video stabilization with a depth camera,” IEEE Conference on Computer vision and Pattern Recognition, pp 89-95, June 2012.
[10] K. E. Wenzel, A. Masselli, A. Zell, “Automatic take off, tracking and landing of a miniature uav on a moving carrier vehicle,” Journal of Intelligent & Robotic Systems, Vol. 61, pp. 221-238, 2011.
[11] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces vs. Fisherfaces:Recognition using class specific linear projection,” IEEE Pattern Analysis and Machine Intelligence, vol. 19, pp.711-720, June 1997.
[12] A. M. Martinez and A. C. Kak, “PCA versus LDA,” IEEE Pattern Analysis and Machine Intelligence, vol. 23, pp. 228–233, Feb 2001.
[13] P. C. Yuen and J. H. Lai, “Face representation using independent component analysis,” Pattern Recognition, vol. 35, pp. 1247–1257, June 2002.
[14] V. Perlibakas “Distance Measures for PCA-based Face Recognition,” Pattern Recognition Letters, vol. 25, No. 6,pp.711-724, April 2004.

ISSN: 2321-8169
243 - 249

[15] Sanjaa Bold "Autonomous Vision-Based Face Recognition System for Unmanned Aerial Vehicles" 2014, SCOPUS JOURNAL, Mongolia, Chinggis Khaan hotel

[16] S. Zhang, "Object tracking in Unmanned Aerial Vehicle (UAV) videos using a combined approach," IEEE International Conference on Acoustics, Speech, and Signal Processing, vol.2, pp 681-684, 2005

[17] P. Theodorakopoulos and S. Lacroix, "Uav target tracking using an adversarial iterative prediction," IEEE International Conference on Robotics and Automation. Kobe, pp. 2866-2871, May 2009.

[18] E. Altug, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," IEEE International Conference on Robotics and Automation, vol.1, pp.72-77, 2002.