Facial Recognition Using Human Visual System Algorithms for Robotic and UAV Platforms

Nicholas Davis
Tufts University
Nicholas.Davis@tufts.edu

Francesco Pittaluga
Tufts University
Francesco.Pittaluga@tufts.edu

Karen Panetta
Tufts University
Karen@eecs.tufts.edu

***Abstract*— In this paper we present a low-cost facial recognition system using a commercial off-the-shelf (COTS) unmanned aerial vehicle (UAV) platform to capture images and video. Our novel approach produces real-time accurate detection and recognition of key features allowing the system to be used in real world security applications. We present recognition algorithms that enable the system to perform quality recognition with a minimal training set. The system can be used in other robotic platforms. It is highly modular and adaptable to other systems.**

Keywords— Human Visual System, Object Detection, Facial Recognition, Robotic Vision

I. INTRODUCTION

First responders' ability to respond rapidly to emergency situations is limited by a lack of real time intelligence. To ensure the safety of the responders, the situation must first be evaluated for dangerous conditions including life-threatening hazards. Live visual feeds let remote experts gauge the safety levels and assess damages of a situation, but do not perform adequately when images are captured in poor lighting or in harsh environmental conditions. Our novel approach leverages HVS-based (Human Visual System) object detection in combination with low-cost COTS UAVs, to deliver efficient real time image enhancement and detection. This approach enables our system to deliver timely information in low visibility environments making it ideal for aiding first responders in their search for critical objects such as wounded victims, human bodies and threat objects.

II. BACKGROUND INFORMATION

A. HVS-LBP

Human Visual System (HVS) based adaptive thresholding methods for image decomposition have been used for image enhancement by Wharton, Panetta and Agaian [1] [2]. This method utilizes different types of image enhancement algorithms and has been applied selectively to the four different HVS regions to achieve a better overall enhancement.

We apply this approach to the problem of facial recognition by combining the classical Local Binary Pattern (LBP) [3] [8] feature descriptors with image processing in the logarithmic domain and the Human Visual System. The HVS logarithmic image processing framework used in this work more accurately characterizes the non-linearity of computer image arithmetic and is also consistent with the non-linear characteristics of the human visual system [11]. Here, we use these algorithms in conjunction with robotic platforms, such as a COTS UAV for developing a low-cost recognition system.

B. Human Visual System (HVS) Based Image Decomposition

HVS based image decomposition aims at emulating the way in which human eyes respond to visual stimulus [5]. Information received by the human eye is characterized by attributes like brightness, edge information, color shades etc. Brightness is actually a psychological sensation associated with the amount of light stimulus entering the eye. Due to the great adaptive ability of the eye, the human eye cannot measure the absolute brightness rather it measures the relative brightness [6]. Weber's Contrast Law quantifies the minimum change required for the human visual system to perceive contrast, however this only holds for a properly illuminated area. The minimum change required is a function of background illumination, and can be closely approximated with three regions, as shown in figure 1 [11]. The first is the De Vries-Rose region, which approximates this threshold for under-illuminated areas and can be represented using the linear equation given by, $\log\Delta B_T = \frac{1}{2} * \log B + \log K_2$. The second region is the Weber region, which models this threshold for properly-illuminated areas represented by $\log\Delta B_T = \log B + \log K_1$. Finally, there is the saturation region, which approximates the threshold for over-illuminated areas and is represented by, $\log\Delta B_T = 2 * \log B + K_3$. Here K_1, K_2 and K_3 are the constants.

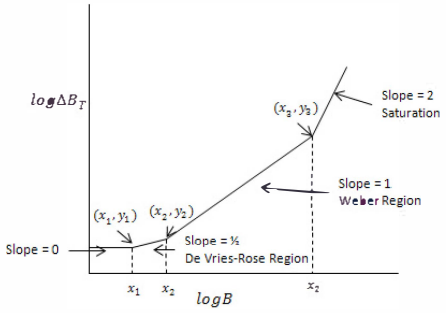

Fig. 1: The Buchsbaum Curve: Approximation of the Increment Threshold $\log\Delta B_T$ as a function of Reference Intensity logB

The value of the parameter B_t which is the maximum difference in the image can be obtained by, $B_t = \max(X(x,y)) \ominus \min(x(X,Y))$ and let us assume that the value of B corresponding to $\log B = x_i$ is B_{x_i} for $i = 1,2,3$.

978-1-4673-6225-2/13/$31.00 ©2013 IEEE

$$B_{x_i} = \alpha_i B_t,\ i = 1,2,3 \quad (2)$$

Here $0 < \alpha_1 < \alpha_2 < \alpha_3 < 1$ and the values of these parameters are based on the 3 regions of the human visual response. The values of the constants are given by:

$$K_1 = \frac{\Delta B_T}{B} = \frac{\beta}{100}\left(\frac{\Delta B_T}{B}\right)_{max} \quad (3)$$

$$K_2 = K_1\sqrt{B_{x_2}} \quad (4)$$

$$K_3 = \frac{K_1}{B_{x_3}} \quad (5)$$

The background intensity image B can be obtained by taking the local mean at each and every point in the image and is given by [2],

$$B(x,y) = \Big[\ p \otimes \Big(\frac{p}{2} \otimes \sum_Q X(i,j) \oplus \frac{q}{2} \otimes \sum_{Q'} X(k,l)\Big) \oplus X(x,y)\Big] \otimes p \quad (6)$$

Here $B(x,y)$ represents the background intensity at each pixel and $X(x,y)$ is the input image. Q represents all the pixels that are directly left, right, up and down of the pixel and Q' is all of the pixels diagonally one pixel away. Also p and q are constants and ΔB_T can be represented by any standard gradient detection algorithm. For our experiments we have used the Sobel operator to estimate the gradient image$X'(x,y)$. The value of the parameter β is selected as 0.02.

Using this information, the image is first partitioned into the different regions of human visual response. These different regions are characterized by the formula for the minimum difference between two pixel intensities for the human visual system to register a difference. Next, these three regions are thresholded, removing the pixels which do not constitute a noticeable change for a human observer, placing these in a fourth image. The formulae are:

$$Im1 = X(x,y) \text{ for } \quad B_{x_2} \geq B(x,y) \geq B_{x_1} \ \&\ \frac{X'(x,y)}{\sqrt{B(x,y)}} \geq K_2 \quad (7)$$

$$Im2 = X(x,y) \text{ for } \quad B_{x_3} \geq B(x,y) \geq B_{x_2} \ \&\ \frac{X'(x,y)}{B(x,y)} \geq K_1 \quad (8)$$

$$Im3 = X(x,y) \text{ for } \quad B_{x_3} \leq B(x,y) \ \&\ \frac{X'(x,y)}{B(x,y)^2} \geq K_3 \quad (9)$$

$$Im4 = X(x,y) \text{ for all remaining pixels} \quad (10)$$

Often it is seen that the De Vries-Rose and the Saturation regions represented by Im1 and Im3 do not contain significant information. Hence these regions are often fused with the Weber region. In this paper we'll refer to the fused image as the "Weber Image". The features extracted from the Weber image is fused with that from the original image and used for recognition.

C. *Local Binary Pattern Feature Descriptors [9]*

The local binary pattern (LBP) operator, introduced by Ojala et al. [7], is a powerful local descriptor for describing image texture and has been used in many applications such as industrial visual inspection, image retrieval, automatic face recognition and detection. It is a widely used texture operator because of its robustness to gray level changes and high computational efficiency. Basic LBP is a window based feature extractor where the texture descriptor is computed based on the neighborhood. It assigns a binary value to every neighboring pixel by thresholding it with respect to the central pixel. We assume that our input is an NxM 8-bit gray-level image, which can be represented as an NxM matrix I, each of whose elements satisfy $0 <= I(x,y) \leq 2^8$. The LBP operator assigns a label to every pixel of an image by thresholding a 3x3-neighborhood of each pixel with the center pixel value and considering the result as a binary number. At a given pixel position (x, y), the LBP operator is defined as an ordered set of binary comparisons (comparison implies calculation of the distance between the pixel intensities calculated by taking the arithmetic difference of pixel intensities) between the center pixel (x, y) and its surrounding pixels. Then the histogram of the labels can be used as a texture descriptor [9]. The neighborhoods of the LBP operator can be various sizes. The local neighborhood can be defined as a set of sampling points evenly spaced on a circle centered at the pixel to be labeled. Bilinear interpolation is used when a sampling point does not fall in the center of a pixel. The neighborhoods are represented using the notation (P, R) where P denotes the set of points on a circle of radius R. Figure 2 shows some variants of circular neighborhood.

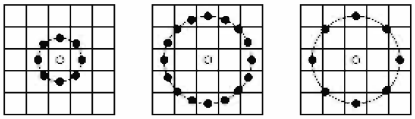

Fig. 2: Circular (8, 1), (16, 2) and (8, 2) neighborhood

An extension to the original LBP operator is the uniform local binary patterns. A local binary pattern is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. For example, the patterns 00000000 (0 transitions), 01110000 (2 transitions) and 11001111 (2 transitions) are uniform whereas the patterns 11001001 (4 transitions) and 01010011 (6 transitions) are not. In the computation of the LBP histogram, uniform patterns are used so that the histogram has a separate bin for every uniform pattern and all non-uniform patterns are assigned to a single bin. This is due to the fact that uniform patterns account for about 90 % of all patterns when using the (8, 1) neighborhood and for around 70 % in the (16, 2) neighborhood. This enables significant space savings when building LBP histograms. To indicate the usage of two-transition uniform patterns, the superscript u2 is added to the LBP operator notation.

Hence the LBP operator with a 2 pixel radius, 8 sampling points and uniform patterns is known as $LBP_{8,2}^{u2}$.

A histogram of the labeled image $f_l(x, y)$ can be defined as:

$$H_i = \sum_{x,y} I\{f_l(x, y) = i\}, i = 0, \dots, n-1 \quad (11)$$

Where n is the number of different labels produced by the LBP operator and

$$I\{A\} = \begin{cases} 1, & A \text{ is true} \\ 0, & A \text{ is false} \end{cases}$$

This histogram contains information about the distribution of the local micro-patterns, such as edges, spots and flat areas, over the whole image. For more efficient face representation however, the facial image is divided into regions and feature descriptors are extracted from each region independently. Therefore if the image is divided into m regions, m local histograms are obtained. Let n be the size of each local LBP histogram, and then the combined LBP feature vector has the size $m \times n$. Thus with the spatially enhanced histogram, a description of the image on three different levels of locality is obtained. The LBP labels for the histogram contain information about the patterns on a pixel-level, the labels are summed over a small region to produce information on a regional level and the regional histograms are concatenated to build a global description of the face.

D. *Classifiers [6]*

An object detection and recognition system typically consists of 3 phases: training, testing and classification. The face recognition problem involves identifying a face from the database. The training facial images are preregistered in the system. Feature vectors are extracted from the training facial images and stored in the system. In the testing phase, feature vectors are extracted from the test image and compared with the feature vectors existing in the database. The test facial image is identified based on its similarity with images in the database. Usually, in face recognition, there are a number of face classes (representing individual person) and a few training images per class. Hence instead of using sophisticated classifier, a nearest-neighbor classification approach is used. The different types of dissimilarity measures that can be used are [8]:

Histogram Intersection	Log − likelihood statistic	Chi square statistic (χ^2)
$D(S, M) = \sum_i \min(S_i, M_i)$	$L(S, M) = -\sum_i S_i \log M_i$	$\chi^2(S, M) = \sum_i \frac{(S_i - M_i)^2}{S_i + M_i}$

Table 1: Dissimilarity Measures

Here S and M are the normalized histograms to be compared. For most of our experiments, the chi square distance statistic has been used since it is an effective measure of similarity between a pair of histograms. Since the face is divided into smaller regions, these regions can be weighted differently depending on their contribution. Psychophysical findings indicate that some features like the eyes play more important role in human face recognition than others in terms of extra-personal variance [10]. The weighted chi square statistic is thus,

$$\chi^2(S, M) = \sum_{i,j} w_j \frac{(S_{i,j} - M_{i,j})^2}{S_{i,j} + M_{i,j}} \quad (12)$$

Where the indices i and j refer to the $i-th$ bin corresponding to the $j-th$ local region and w_j is the weight for the region j. The weights can be determined by the methodology described in [8]. Here the training set is classified using one of the sub regions of the image at a time and the weights were assigned based on the rate of recognition. That is if the sub-region yielded a greater rate of recognition, the weight associated with that region is selected to be higher than the others. The weights are however selected without utilizing an actual optimization procedure.

E. Using the HVS-LBP Algorithm for UAV Platforms

Our HVS-LBP algorithm, proceeds as follows: The algorithm begins with a pre-training phase in which each facial image in the training database is decomposed based on the four HVS regions: Dark, Devries-Rose, Weber, and Saturation. The images are decomposed into four constituent images each of which is then converted into a feature vector. The latter three feature vectors are then fused together into a single optimized feature vector in a process we refer to as HVS Decomposition and Fusion.

Once the pre-training phase is completed, the image capture and recognition phase begins. Captured images, which we refer to as test images, are processed by HVS Decomposition and Fusion in real time. Once a test image has been processed, the Chi-Squared distance between its feature vector and each feature vector in the pre-training database is computed to determine which feature vector in the pre-training database is closest to the test image feature vector. These two vectors are then defined as a match. The algorithm considers a match correct if the two vectors correspond to an image of the same person's face. Otherwise; the algorithm considers the match a false positive.

F. AR.Drone

An AR.Drone 1.0 was used as our image-capturing platform. The AR.Drone 1.0 is a widely available and

Fig. 3: AR.Drone 1.0

reliable low-cost COTS UAV. ARDrone-Control.NET, an open source AR.Drone control and navigation system was

used to control the drone. Images taken by the AR.Drone 1.0 have a maximum resolution of 640x480 pixels.

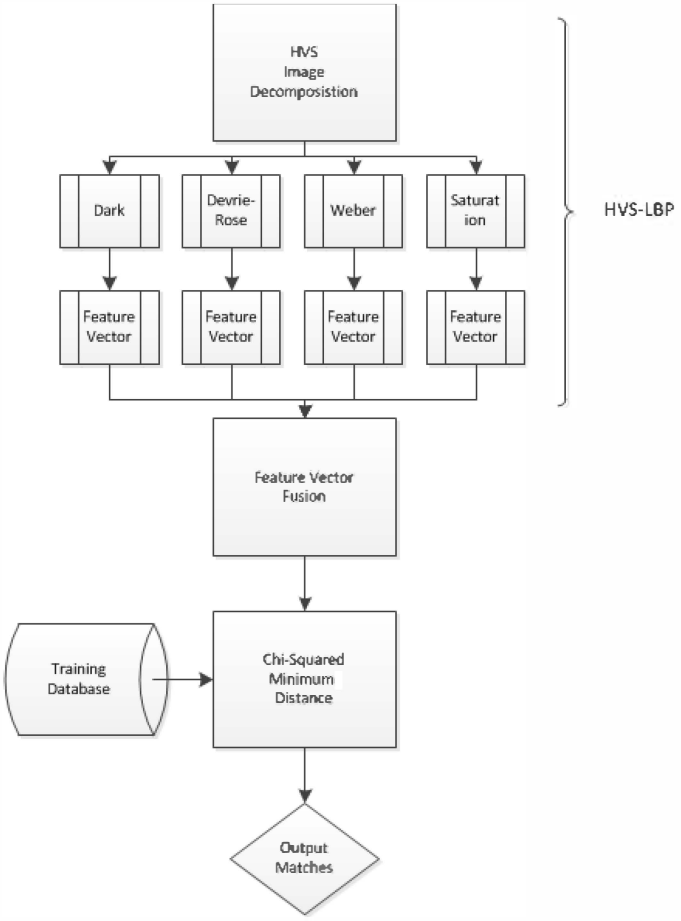

Fig. 4: HVS-LBP Algorithm Flow Chart

III. Experimental Results

In order to pre-train our algorithm with a large database, we used the AT&T Database of Faces [12] which includes 10 different pictures of 40 people. This database was used alongside our own database of images, which were captured using the AR.Drone 1.0. As we will show, the accuracy of our system depends on the number of training images per test individual.

Number of Training Images per Test Individual	Recognition Accuracy
5	100%
4	97.78%
3	91.11%
2	82.23%
1	77.78%

Table 2: Accuracy of Results Based on Number of Training Images per Test Individual

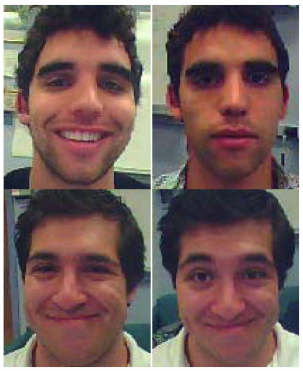

Figure 5: Match of Test Image (Right) and Training Image (Left)

Our experimental results were obtained by testing our facial recognition system with a varying number of training images per test individual. Nine test individuals were photographed using the UAV system. The results are displayed in Figure 4, and Table 2.

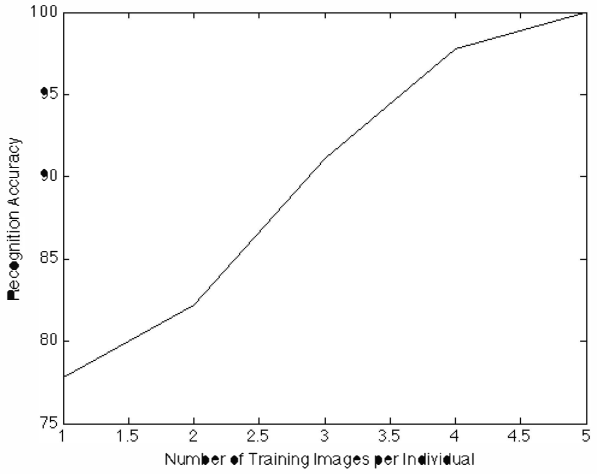

Figure 6: Recognition Accuracy Plot

IV. Conclusion and Future Work

Since our system is highly modular, adaptable, and accurate it can be modified for use with any COTS robotics. Furthermore, the system can be propagated across multiple platforms to implement a network of systems with different sensor packages including chemical sensors, thermal imaging, Geiger counters, etc.

Future additions to this system can include an automated facial cropping system as well as object detection. The motivation to integrate an automated facial cropping system is reducing noise and error in the recognition process, as well as increasing the response time of the system. The motivation to integrate object detection into the system is to expand the range of scenarios in which the system can be deployed.

ACKNOWLEDGMENT

ARDrone-Control.NET was written by Thomas Endres, Stephen Hobley, and Julien Vinel.

REFERENCES

[1] E. Wharton, K. Panetta, and S. Agaian, "Human Visual System Based Multi-Histogram Equalization for Non-Uniform Illumination and Shoadow Correction," *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on* vol. 1, pp. I-729-I-732, April 2007.

[2] [11] K. Panetta, E. Wharton, and S. Agaian, "Human Visual System-Based Image Enhancement and Logarithmic Contrast Measure," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 38, pp. 174-188, Feb 2008.

[3] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Recognition with Local Binary Patterns," *Computer Vision - ECCV 2004,* pp. 469-481, 2004.

[4] K. Etemad and R. Chellappa, "Discriminant analysis for recognition of human face images," *Journal of the Optical Society of America, vol. 14,* pp. 1724-1733, 1997.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 19*, pp. 711-720, July 1997.

[6] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on* pp. 84-91, 1994.

[7] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 971-987, July 2002.

[8] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 28*, pp. 2037-2041, 2006.

[9] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Recognition with Local Binary Patterns," *Computer Vision - ECCV 2004*, pp. 469-481, 2004.

[10] M. K. Kundu and S. K. Pal, "Thresholding for edge detection using human psychovisual phenomena," *Pattern Recognition Letters 4,* pp. 433-441, December 1986.

[11] G. Buchsbaum, "An Analytical Derivation of Visual Nonlinearity," *IEEE Transactions on Biomedical Engineering,* vol. BME-27, May 1980.

[12] Samaria,F.S. "Parameterization of a stochastic model for human face indentification," *Applications of Computer Vision, 1994., Proceedings of the Second IEE Workshop,* pp. 140, December 1994.