SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON
MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 3, ISSUE 6
ISSN 2054-7390

3D HMM-based Facial Expression Recognition using Histogram of Oriented Optical Flow

[1]Sheng H. Kung, [2]Mohamed A. Zohdy and [3]Djamel Bouchaffra
[1,2]Electrical and Computer Engineering, Oakland University, Rochester, MI, USA;
[3]Center for Development of Advanced Technology (CDTA), Baba-Hassen, Algiers, Algeria
shkung@oakland.edu; zohdyma@oakland.edu; dbouchaffra@ieee.org

ABSTRACT

In this paper, we propose a 3D HMM (Three-dimensional Hidden Markov Models) approach to recognizing human facial expressions and associated emotions. Human emotion is usually classified by psychologists into six categories: Happiness, Sadness, Anger, Fear, Disgust and Surprise. Further, psychologists categorize facial movements based on the muscles that produce those movements using a Facial Action Coding System (FACS). We look beyond pure muscle movements and investigate facial features – brow, mouth, nose, eye height and facial shape – as a means of determining associated emotions. Histogram of Optical Flow is used as the descriptor for extracting and describing the key features, while training and testing are performed on 3D Hidden Markov Models. Experiments on datasets show our approach is promising and robust.

Keywords: Human Computer Interaction (HCI); Facial Expression Recognition; Feature Extraction, Optical Flow; Hidden Markov Model; HMM; Three -dimensional Hidden Markov Model; 3D HMM.

1 Introduction

Human Computer Interaction (HCI) is a fast-growing field propelled by advances in computer technology and machine learning theory. It is the study of how we communicate with (or through) machines, be it computers, robots, vehicles, home devices, the internet or artificial limbs. At a physical level, HCI has progressed from a low level of interaction (keyboard, mouse and pen) to a high level one (touch screen, sensor, voice, video and nerve). Beyond the mere physical aspect, however, HCI has also advanced into higher cognitive and affective functions [18]. Looking towards the future, HCI will assuredly evolve one step further to a predictive level, bringing it ever closer to resembling a true human-to-human interaction – with machines mimicking basic human senses of sight, hearing, taste, smell and touch as well as human learning abilities and emotions. Facial expression recognition is a step toward understanding of human emotions.

In simple terms, HCI is about the design of an intelligent interface that eliminates the need for third-party involvement during operation of the machine and expands the capability of that interface. The influence of HCI is far-reaching, finding critical application in areas such as surveillance, biometric security, video games, assistive devices for people with disabilities, robot-assisted surgery, accidence-avoidance and driver-assistance in automobiles, as well as in brain-controlled prosthetic limbs that

DOI: 10.14738/tmlai.36.1661
Publication Date: 27th December, 2015
URL: http://dx.doi.org/10.14738/tmlai.36.1661

explore the neural-machine interface. Clearly, HCI can lead to vast improvement in the quality of life for humans. Facial expression and emotion recognition will raise the compassion level of HCI application.

Much like other pattern recognition problems, the key to determining how difficult facial expression recognition problems are is in the intra-class and inter-class features variations. Intra-class variations, such as age and facial paraphernalia (materials either added-on or attached to faces creating occlusion, such as hair, glasses, beards, moustaches, cosmetics) are not small; inter-class variations, such as appearance, ethnicity, culture background and gender are also quite subtle. This combination of variations will complicate facial expression recognition. In addition, factors such as illumination, pose, viewpoint, scale, shade and noise add even further complexity to the problem. Here, facial expression recognition is defined as using computers in an attempt to automatically identify human facial expressions and infer their underlying meaning (emotion or intention). Facial expression recognition generally consists of facial expression analysis, representation, classification and interpretation.

This research is focused on HCI in the recognition of human facial expression and emotion analysis. We propose a 3D Hidden Markov Model (HMM) approach to recognizing human facial expressions and associated emotions. To the best of our knowledge, this is the first application of 3D HMM and Histogram of Oriented Optical Flow to facial expression recognition. Psychologists classify human emotion as displayed through facial expression into six categories: Happiness, Sadness, Anger, Fear, Disgust and Surprise. They further categorize facial movements based on the muscles that produce those movements using a Facial Action Coding System (FACS). We look beyond muscle movement and investigate facial features – brow, mouth, nose, eye height and facial shape – as demonstrated by motion, and use a Histogram of Optical Flow as a descriptor for extracting and describing those features.

This paper is organized as follows: Section 2 reviews the related work, Section 3 reviews feature representation, Section 4 describes our methodology, based on optical flow and 3D HMM, Section 5 details experiments and analysis, and we conclude with a summary.

2 Related Works

Facial expression research traces back to 1872 when Darwin suggested that facial expressions were innate in his famous book, The Expression of the Emotions in Man and Animals. Over the years, there remains much debate among scientists and psychologist about whether facial expressions are nature or nurture. We think both factors have influences on human facial expressions. There are six universal basic emotional states (e.g., happy, angry) with corresponding facial expressions varying in intensity from culture to culture (i.e. Asian cultures usually suppress facial expressions). There are other built-up and social activity-derived emotional states (e.g. shame, anxiety and embarrassment) whose corresponding facial expressions are heavily influenced by culture and the surrounding environment. These cultural contexts make facial expression recognition even more interesting and challenging.

Not until 1998 when Paul Ekman and Wallace Friesen adopted a system from a Swedish anatomist and published Facial Action Coding System (FACS) [9] did we have a method for measuring and scoring facial behavior. FACS identifies how various facial muscle movements made individually or in groups, affect facial expression. The movements in face and the one or more muscles that cause these movements are described and coded in Action Units (AU). For example, AU 1 is “Raising Inner Brow” and AU 26 is “Dropping the Jaw”, each with its own designated muscle(s) names. AUs are divided between the upper and lower face and also include some non-facial muscle movements concerning the eye and tongue.

AUs can be additive, if they are independent to one other, and ideally all facial expressions can be decomposed into their constituent AUs. FACS quickly became the de facto standard for characterizing facial expressions. Ekman and Friesen later also developed EMFACS (Emotional Facial Action Coding System) and FACSAID (Facial Action Coding System Affect Interpretation Dictionary) to interpret FACS scores in terms of emotion categories. Almost in parallel, Facial Animation Parameters (FAP) was developed in 1998 by Moving Pictures Experts Group (MPEG) as a facial animation specification in the MPEG-4 international standard that provided an alternative way of modeling facial expressions. FAPs – much like AUs – are closely related to muscle actions. There are 68 FAPs (e.g. FAP 3 for “open_jaw”, FAP 5 for “raise_b_midlip and FAP 7 for “stretch_t_cornerlip”).

There are shortcomings in FACS, such as ambiguity and subjectivity in intensity between AUs, to the complexity of representing emotions through multiple AUs (forest/tree syndrome) and reliability issues (agreement between observers). But FACS’ advantage lies in the fact that “a human rater can encode facial actions without necessarily inferring the emotional state of a subject, and therefore one can encode ambiguous and subtle facial expressions that are not categorizable into one of the universal emotions, such as fake smiles” [12]. Researchers including Tian et al. [37], Valstar et al. [41] and Mahoor et al. [24] have had success in using computers to overcome shortcomings in FACS to automatically identify AUs and thus quickly identify emotions.

Ekman and Friesen [9] in essence provided us the framework and the means to analyze facial expressions anatomically.

3 Feature Representation

3.1 Preprocessing

Feature extraction is a starting and important task in any recognition process [44], whether it is a single image-based object, scene, face or facial expression or a video (frame) based facial or human action recognition. It consists of extracting the object or motion cues that are salient and discriminative among the recognized objects or actions. A good feature should be invariant to illumination, occlusion, scale, viewpoint, deformation, and clutter background, as well as affine, rotational and translational transformation of images. In order to reduce sensor noise, the image is usually smoothed by convolving the image with a Gaussian kernel. Illumination variations can be effectively minimized in preprocessing by normalizing the image using a filter; for example, histogram equalization or a discrete cosine transform (DCT) after truncating an appropriate number of coefficients to minimize variations under different lighting conditions.

3.2 Global Feature Extraction

Global feature is also called holistic feature in contrast to component-based local feature. It is a feature which represents image as a whole, for example, the shape, geometric dimension, distance, color etc. Global Feature has the advantage of being simple and low computational complexity. However, it has disadvantage of being generally low accuracy. Depending on the application, a fusion strategy to combine global and local feature may be necessary.

3.2.1 Model-based Approach - Decomposition

One of the model-based approaches is to find a model of bases for images of variable objects and a set of model parameters for each image. It involves conducting Principal Component Analysis (PCA), or, recently, Independent Component Analysis (ICA) to decompose images into a small set of characteristic feature images. It was originated in face recognition [14]. A face image is treated in PCA as a point (or vector) in a very high dimensional space (256x256 = 65,536). Eigenfaces, developed by Sirovich and Kirby [34] and applied to face recognition by Turk and Pentland [39], are eigenvectors derived from PCA and act as a set of orthogonal face basis usually represented in lower dimension. An individual face is then a weighted sum of the eigenfaces by projecting it into the space of face basis. To recognize a face is to compare weights of known individuals. It is generally an approach of nonlinear, generative and parametric model. Since it is based on whole image pixel intensities, the first three eigenfaces are usually discarded to reduce the illumination effect on image variations.

Eigenfaces have shortcomings in robustness to shape, pose and expression variations. To overcome them, Active Appearance Model (AAM), first introduced by Edwards, Taylor and Cootes [7], combined PCA shape model from landmark points and PCA grey-level appearance model from texture (pixel intensities). It can generate almost any face. Matching a face image is to find model parameters to minimize the difference between the image and a synthesized face – an optimization problem similar to template matching.

The basis vectors that ICA produces are more spatially localized and statistically independent than the ones in PCA. The figure below shows the first 8 eigenvectors computed on 500 randomly selected images from FERET gallery (top) and 8 of 200 ICA basis vectors computed (bottom). [9] Uddin et al. [40] proposed an enhanced ICA for facial expression recognition.

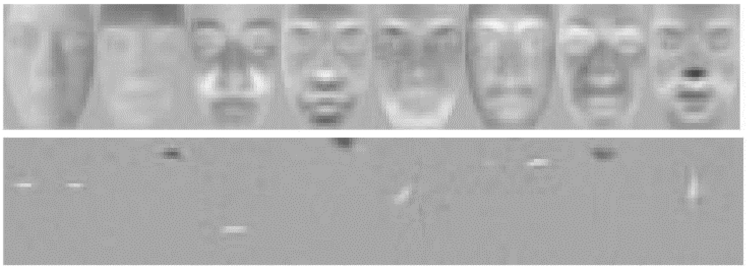


Figure 1. (Top) The first 8 eigenvectors computed on 500 randomly selected images from FERET gallery (Bottom) The 8 of 200 ICA basis vectors computed. (Draper et al. [9])

3.2.2 Model-based Approach - Transformation

Another model-based approach is to transform the image into bases of completely different domain - frequency domain. The most popular transformations are Fourier Transform, its related and faster DCT (Discrete Cosine Transform), and DWT (Discrete Wavelet Transform), all in 1D to 3D. The drawback of Fourier Transform is that it assumes the signal or image frequency is smooth across time or space (i.e., stationary) and, in transforming into the frequency domain the temporal or spatial information - when or where a discontinuous event occurs (frequency pattern changes) - is lost. In an effort to correct this deficiency, STFT (Short-Time Fourier Transform) or Gabor Transform uses a fixed sliding window to localize in time or space some frequency changes. However the dilemma is determining the size of the window – a narrow window resulting in poor frequency resolution and a wide window resulting in poor time/space resolution. Wavelet overcomes the resolution dilemma by trying different frequencies at different windows. Unlike sinusoidal bases of Fourier Transform, Wavelet Transform uses wavelets as bases, which are irregular and asymmetric and, thereby, can very well model the signal or image with bumps, dips and humps. Features are in terms of the coefficients from transform. Pinto et al. [28] used 3D and 3D Wavelet Transforms to extract 3D facial expression features. Hough Transform is to map imperfect geometric shape into parameters space, e.g. a line is mapped to its slope and interception space as a point. After all the feature points are mapped into a space called accumulator, finding the maxima (vote) will determine the original image shape.

Tsai et al. [38] combines Harr wavelet and PCA for human-robot interactive emotion recognition. Global feature extraction is to extract global variation pattern for classification. It could also be used on sub-images for regional feature extraction. Following shows an example of log-Gabor responses extracted from a normalized expressive face. [11]

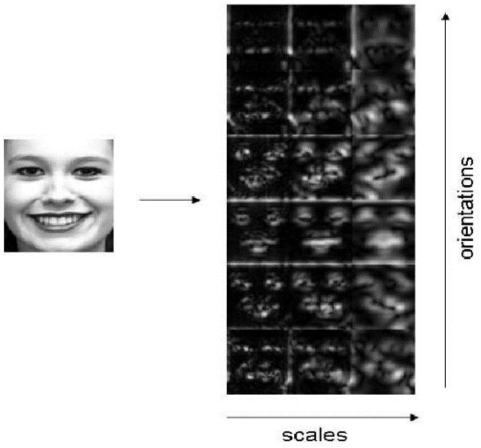


Figure 2. An example of log-Gabor responses (by scales and orientations) extracted from a normalized expressive face on the left. (Fanelli et al. [11])

3.2.3 Template-based Approach

Template is mainly used to detect the components of objects, for example, parts of the body, components of the face - eyes, nose and mouth regions. The template of face is generated by cropping all the facial regions in the training set from the ground truth data and averaging them. It is used to match the facial regions of testing objects in order to detect faces. The template can also be developed by describing the contour or shape of the objects for 2D correlation or distance matchings. The template can be algorithmically deformed to achieve efficient recognition of objects with shape deformation, occlusion and background clutter.

3.3 Local Feature Extraction

Local feature extraction consists of the design of feature detector and feature descriptor on image space. Image space can be spatial (x, y) for single image and spatial temporal (x, y, t) for video. Detector refers to detection of interest points or keypoints in image space for subsequent processing of descriptor. The interest point needs to be clearly defined mathematically in image space, reproducible and rich in salient information. Given the interest point found at a location, descriptor is to describe the image structure in a neighborhood of that location. It needs to be strongly inter-class discriminative, while tolerant of intra-class variation and robust to various image variations.

3.3.1 Detector

Gradient Detector

Gradient is the first order derivative detector. Operators like Prewitt, Sobel, Robinson and Kirsch are all first order derivative filter for detecting edges. Harris corner uses covariance matrix of gradients to determine the corner.

Laplacian Detector

The Laplacian $L(x, y)$ of an image with pixel intensity values $I(x, y)$ is given by

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \ .$$

It is a second order derivative mask, which is also called “zero crossing detector” due to its highlighting of zero passing edges. It is combined with Gaussian smoothing filter of (0, σ) to becomes a Laplacian of Gaussian (LoG) detector. LoG can be approximated by A Difference of two Gaussians (DoG) having different standard deviations. DoG works like a bandpass filter by subtracting one blurred image from another to preserve the spatial information corresponding to some range of frequencies while discarding all others. The extrema (Maxima/Minima) of the Laplacian detector was used by SIFT (Scale-Invariant Feature Transform) to detect interest points in a local patch. However, there are following steps to filter out weak and unstable keypoints, which are low contrast and along the edge.

Hessian Detector

Hessian is also a second order derivative detector. However, it is in a matrix form. It is a square matrix of second-order partial derivatives of a scalar-valued function. The 2X2 Hessian Matrix can be expressed as

$$H(x,y) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \text{ for an image } I(x,y).$$

It describes the local curvature of a function of many variables. The eigenvectors of the matrix give the directions for minimum and maximum curvature while the eigenvalues correspond to the amount of curvature in those directions. SURF (Speed-up Robust Features) [2] use extrema of the determinant of Hessian matrix in a local neighborhood of image to detect interest points. SURF is more efficient than detector and descriptor of SIFT due to the advantage of speed and accuracy of Hessian matrix. Wu et al. compared SIFT with its variants including SURF [42].

Blob Detector

Maximally Stable Extremal Regions (MSER) [8] detect blobs in images, which are robust under perspective transformations. It explores distinguished regions [25], regions that possess some distinguishing, invariant and stable property. It is also called extremal regions because all pixels inside have either higher (bright extremal regions) or lower (dark extremal regions) intensity than all the pixels on its outer boundary. MSER is based on the idea that regions, which stay nearly the same through a wide range of thresholds, must be maximally stable. It is used in face detection and tracking.

Combined Detector

The Harris-Laplace detector combines the traditional 2D Harris corner detector with the idea of a Gaussian scale-space representation in order to create a scale-invariant detector. Harris-Affine detector combined the Harris corner detector with the idea of iteratively applying affine shape adaption algorithm in order to create an affine-invariant detector. Hessian-Laplace and Hessian-Affine detectors are the same. In order to create a scale-invariant detector, a Gaussian pyramid is constructed in SIFT and SURF from the input image by repeated smoothing and subsampling, then a DoG (for SIFT) or Hessian matrix (for SURF) is computed from the differences between the adjacent levels in the Gaussian pyramid.

3.3.2 Descriptor

Local Binary Patterns (LBP)

LBP is a non-parametric local descriptor to capture the texture pattern of an image in a small neighborhood around a pixel. It labels the surrounding pixels with binary 1 and 0 depending on whether the intensity of the pixel is greater than the central pixel. Due to its discriminative power and computational simplicity, LBP has become a popular approach in various applications including facial expression analysis. It could be quantized and grouped into local histograms for classification. Shan et al. [33] proposed a Boosted-LBP for facial expression recognition. Below shows a basic LBP operator.

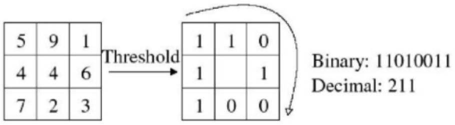


Figure 3. A basic LBP operator. (Shan et al. [33])

Histograms

Histogram of Oriented Gradients (HOG) is used by SIFT [22] for descriptor of the interest points. After SIFT keypoints are detected, a 16x16 neighborhood around the keypoint is taken. As the following diagram shows, it is devided into 16 sub-blocks of 4x4 size. For each sub-block, 8 bin orientation histogram is created. Together all 16 sub-blocks of 8-bin orientation histogram are concatenated to obtain 128 (16*8) dimensional feature vector – the keypoint descriptor. In addition to this, several measures are taken to achieve robustness against illumination changes, rotation etc. The advantage of quantization of gradient locations and orientations is to make the descriptor robust to small geometric distortions and small errors in the region detection.

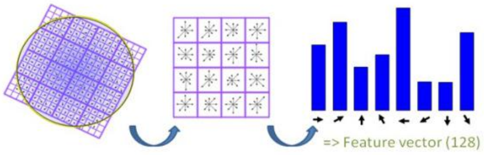


Figure 4. A SIFT descriptor generation process diagram.
(Source: https://gilscvblog.wordpress.com/2013/08/18/a-short-introduction-to-descriptors/)

Soyel et al. [35] proposed a discriminative SIFT (D-SIFT) to match the SIFT keypoints detected from two facial expression images as shown below. On the left are matches from the regular SIFT and on the right from D-SIFT.

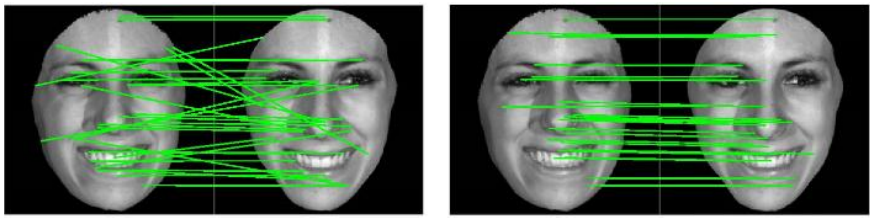


Figure 5. (Left) Face descriptors detected and matched by standard SIFT algorithm. (Right) Face descriptors obtained by KLD-based matching under varying expression intensities. (Soyel et al. [35])

Spatial-frequency

Gabor filter is defined as the product of a Gaussian kernel and a complex sinusoid. It receives attention because the frequency and orientation representations of the Gabor filter are similar to those of mammal's visual cortex. Gabor filter is often used as a filter bank because it has various combination of parameters - θ represents the orientation, λ is the wavelength, and σ1 and σ2 represent scale at orthogonal directions (or σ if the Gaussian is symmetric) [26]. When it is applied the output is equal in size with the original image, and the features extracted have great redundancy. Therefore, dimension reduction is needed. Praseeda Lekshmi et al. [29] and Bashyal et al. [1] used Gabor filter in facial expression recognition.

Gabor wavelets are related to Gabor filter. The family of Gabor wavelets are created by dilation (scale) and shift from the mother wavelet. When Gabor wavelet transformation is applied, Gabor wavelet coefficients are output as features for the neighborhood pixels of the interest point. It is much more powerful than geometric positions. Tian et al. [36] used Gabor Wavelets in facial expression recognition.

SURF [2] calculates Haar wavelet response for descriptor. A 20X20 region is split up into smaller 4×4 square sub-regions. As the following picture shows, for each sub-region, SURF compute a few simple features at 5×5 regularly spaced sample points. The horizontal and vertical Haar wavelet responses dx and dy are calculated and summed. The absolute values of the responses $|dx|$ and $|dy|$ are also calculated and summed for each sub-region. For all 4×4 sub-regions, it forms a vector of length 64 as keypoint descriptor.

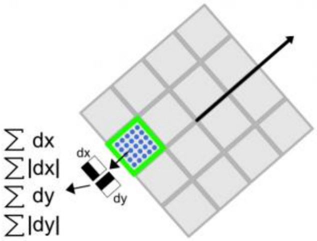


Figure 6. A SURF descriptor generation process diagram.
(Source: http:/www.juergenwiki.de/work/wiki/doku.php?id=public:surf)

4 Methodology

The proposed framework and workflow is shown below.

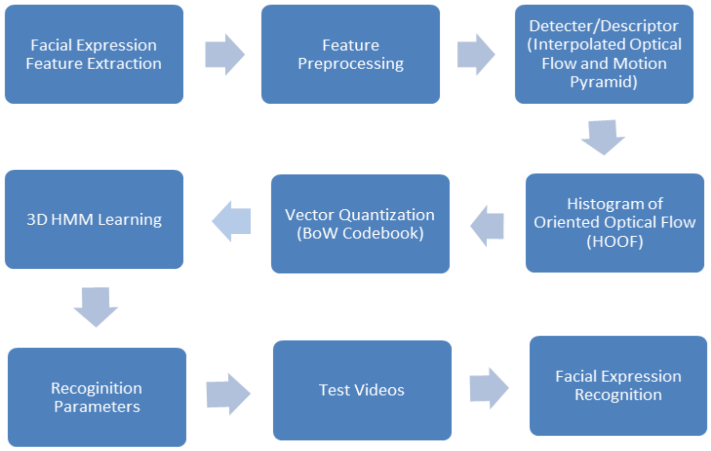

Facial Expression Feature Extraction
Feature Preprocessing
Detecter/Descriptor (Interpolated Optical Flow and Motion Pyramid)
Histogram of Oriented Optical Flow (HOOF)
Vector Quantization (BoW Codebook)
3D HMM Learning
Recoginition Parameters
Test Videos
Facial Expression Recognition

Figure 7. A system flow diagram.

4.1 Optical Flow

Optical flow is the motion estimate of image pixels or regions from one frame to the next. Between two consecutive frames, the following was assumed: 1) Brightness/color constancy between the same pixels, 2) Small displacement (However, image pyramids are to relax this assumption and track larger movements) and 3) Spatial coherence (neighboring points are on the same surface). To cope with the aperture problem, additional assumptions were made in the following Lucas-Kanade method [23] that all neighboring pixels in the patch move at the same speed and in the Horn-Schunck method [13], that a smoothing term was added in the optimization. However, motion discontinuities (e.g. occlusion at motion boundaries, pixels being visible in one frame only) and motion in texture-less regions would make optical flow estimation a challenging task.

4.1.1 Lucas–Kanade Method (Local)

First, Identify distinguished points (e.g., corners detected using Shi-Tomasi algorithm), then obtain image patches surrounding those pixel points, and use a least square optimum fit model to find the flow field between two frames. It is a local estimating method resulting in a sparse flow because it cannot provide flow information in the interior uniform regions of the image. Local method is more robust under noise. First picture below shows an optical flow tracking of the movement of highway vehicles. The second picture shows optical flow of facial expression.

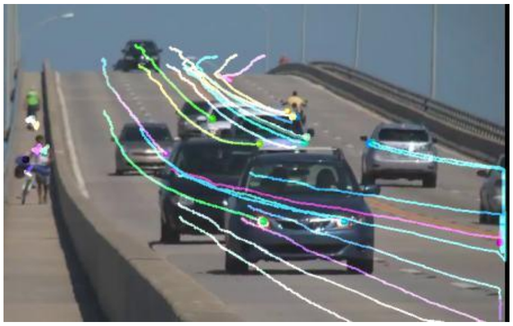


Figure 8. An example of feature-points tracking.
(Source: http://docs.opencv.org/master/d7/d8b/tutorial_py_lucas_kanade.html#gsc.tab=0)

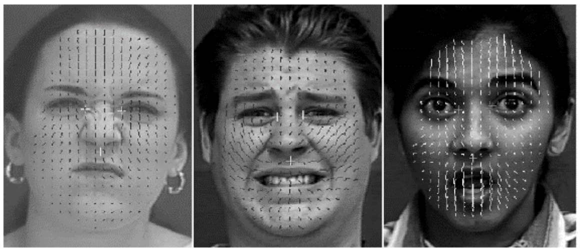


Figure 9. Examples of pixel-wise tracking of face. (Lien et al. [20,21])

4.1.2 Horn–Schunck Method (Global)

Horn-Schunck method is based on Lucas-Kanade method but add a smoothing constraint and integrate over the whole image. Because it is optimizing an integral function of the whole image based on residuals from the brightness constancy constraint, and a specific regularization term expressing the expected smoothness of the flow field, it is a global estimating method. It is resulting in a dense flow because it processes all the pixels. [13]

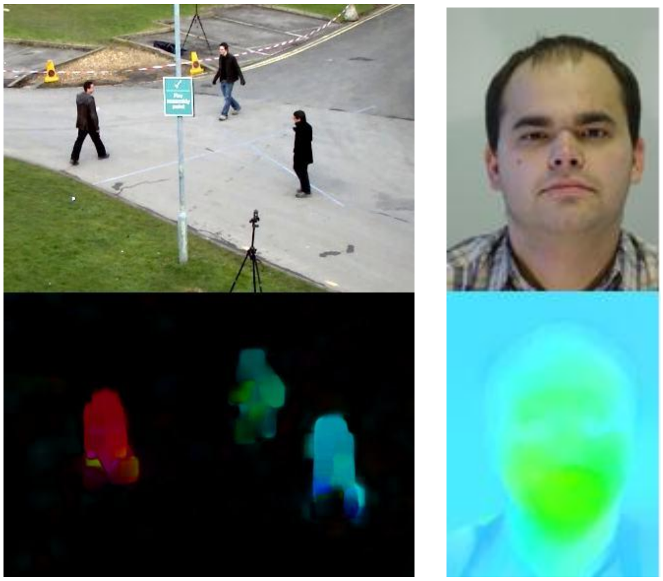


Figure 10. (Left) An example of dense optical flow, (Right) An example of facial optical flow from a pair of head movement frames (Source on the left is same as Figure 8.)

4.1.3 Coarse to Fine Lukas-Kanade Method

Horn-Schunck and Lucas-Kanade optical methods work only for small motion. If object moves faster, the brightness changes rapidly, derivative masks will fail to estimate spatiotemporal derivatives. Therefore, image pyramid is used to computer optical flows at coarser scale as the following diagram indicates. It is also called Multiresolution L-K method or L-K with Pyramid method.

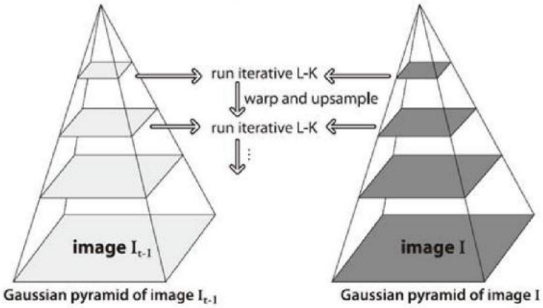


Figure 11. A pictorial coarse-to-fine optimal flow estimation.
(Source: Bradski et al. 2008. *Learning OpenCV: Computer vision with the OpenCV library,* O'Reilly)

4.1.4 Combined Local-Global (CLG) Methods

Bruhn et al. [4] proposed a method combining the robustness of local methods with the density of global approaches. Hence, it is called combined local and global method. It can be considered as a noise-robust generalization of the Horn and Schunck technique because it uses the same concept of integration concept of H-S method. Instead of integrating over the individual pixels, it integrates over the patch of pixels which L-K method users to avoid the aperture problem.

4.1.5 A Proposed Interpolated Lucas-Kanade Method

We proposed an interpolated Lucas-Kanade optical flow method to transform sparse Lucas-Kanade Coarse-to-Fine optical flows to dense flows by adopting a distance-based interpolation method [19]. As the following left figure shows, we use a search circle, whose radius need to be defined beforehand or to be varied, centered at the interpolation point at location x to select data points p_i enclosed in the circle. The weight assigned to each is based on the square of distance from x to p_i. There are other weights can be used. Bigger circle will make the resulting surface smoother but loose the local continuity. Therefore, a good knowledge of the data set will be required to select the radius of the circle.

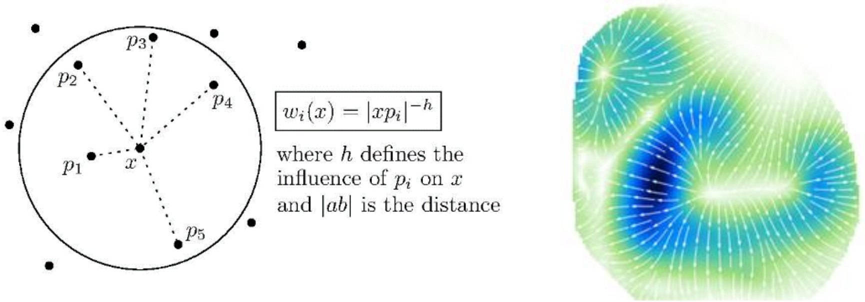

p_3
p_2
p_4
p_1
x
p_5
$w_i(x) = |xp_i|^{-h}$
where h defines the influence of p_i on x and $|ab|$ is the distance

Figure 12. (Left) A pictorial distance-based interpolation circle (Ledoux et al. [19]). (Right) An example of Interpolated L-K optical flows

4.2 HMM

HMM is a tool to model a spatial, temporal or spatio-temporal process when the process is sequential and stochastic in nature. For example, Huang and Kennedy used HMM to uncover hidden spatial states and patterns underlying home sales prices. HMM is also widely used in speech and handwriting recognition, DNA sequencing, and music identification to characterize the spectral properties in time series and in dynamic scene understanding and behavior recognition in video footage to uncover spatio-temporal (space and time) patterns in events.

HMM is a generative graphical model, as well as a simple Dynamic Bayesian Network (DBN). It is based on three assumptions: (1) The first-order Markov chain assumption on the transition probability among hidden states - no influence on current state (t) beyond its predecessor in time ($t-1$), (2) The stationary assumption on transition probabilities – the transition probabilities are time-invariant, and (3) output independence assumption on the probabilities emitted from hidden states to observations - the emission probabilities are time-invariant. Researchers are developing variations of HMM in relaxing these assumptions like the defined state duration or varying the underlying structure or architecture of HMM to cope with the needs of various applications. Some common variants of HMM are continuous HMM with Gaussian or Mixture of Gaussian output, Input-output HMM, Coupled HMM, Factorial HMM, Layered HMM and Hierarchical HMM [3]. This paper is focused on 3D HMM.

HMM is using the observations to find the hidden states following the Bayesian theory of calculating the posterior probability from a prior probability and its likelihood. The model is also in line with the control theory view of the real-world system consisting of state space (unobservable) and output vector (observable). For example, in a stochastic, discrete-time, linear control system represented by

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

$$z_k = Hx_k + v_k$$

where A corresponds to a state-transition matrix, H corresponds to a emission (observation) matrix and B, the input-output matrix, are coefficients, x is the state variable, u is the input variable, z is the output variable, w_k (process noise) and v_k (measurement error) are multivariate Gaussian distributions with covariance q_k, and is called Kalman Filter. The underlying model of Kalman Filter is a Bayesian model similar to HMM, but latent and observed variables are continuous with Gaussian distributions.

Due to the linearity and Gaussian distribution assumptions, the maximum likelihood estimate (MLE) used in Kalman Filter coincides with maximum a posteriori probability (MAP) estimate from expectation maximization (EM) algorithm of HMM.

Here is a summary of HMM algorithm:

Rabiner's* [30] *three basic problems

- What is the probability of an observed sequence, *O*, given a model? **(Evaluation)**
  - *joint probability of observations and state sequence – P(O,S|ϑ)*
  - *as useful as Markov Chain*
- What is the optimal sequence of states that "explains" the observed data? **(Decoding)**
  - *optimality criterion*
- how can one adjust the model parameters to maximize the probability of the observed data given the model **(Learning)**

Problem 1: Evaluation

For a model $\lambda(\theta)$ with parameters θ, what is $P(O|\lambda)$ or more simply $P(O|\theta)$?

- model parameters
  Q a set of states $Q = \{1,2, \ldots, N\}$
  A the state transition matrix $a_{ij} = P(q_{t+1} = j|q_t = i)$
  B the emission probabilities $b_j(k) = P(o_t = k|q_t = j) \;\; 1 \leq k \leq M$
  ω an initial probability distribution $\omega_i = P(q_0 = i) \;\; 1 \leq i \leq N$
  Observations $O = (o_0, \, o_1, \ldots o_T)$
- $P(O|\theta) = \sum_\pi P(O|\pi, \theta) P(\pi|\theta)$
- Assume the observations are independent

$$P(O|\pi, \theta) = \prod_{i=0}^{T} P(o_t \mid q_t, \theta)$$
$$= b_{q0}(o_0) b_{q2}(o_2) \cdots b_{qT}(o_T)$$

The probability of a a particular state sequence (or path), π,
$P(\pi|\theta) = \omega_{q_0} a_{q_0 q_1} a_{q_1 q_2} \cdots a_{q_{T-1} q_T}$

- There are N^T state sequences and $O(T)$ calculations so the brute force complexity is $O(TN^T)$
- *Forward algorithm (Viterbi)*
  - α_t *is the probability of observing the partial sequence* $(o_0, o_1, \ldots, o_t)$ *given that state* $q_t = i$
  - $\alpha_t(i) = P(o_0, o_1, \ldots, o_t, q_t = i|\theta)$
    $\alpha_{t+1} = \left(\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right) b_j(o_{t+1})$ with $\alpha_0(i) = \omega_i b_i(o_0)$
  - $P(O|\theta) = \sum_{i=1}^{N} \alpha_T(i)$
  - *complexity* $O(N^2T)$
- *Backward algorithm*
  - almost the same as forward algorithm

- $\beta_t(i)$ is the probability of observing the partial sequence $(o_{t+1}, o_{t+2}, \dots o_T)$ given that state $q_t = i$

  $\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots o_T | q_t = i, \theta)$ with initial condition $\beta_T(i) = 1$

  $$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \quad t = T-1, \ \dots 0$$

  $$\alpha_{t+1} = \left(\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right) b_j(o_{t+1})$$

  $$\alpha_t(i) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} \, b_j(o_t)$$

Problem 2: Decoding

What is the optimal sequence of states that "explains" the observed data?

- Optimality criteria
  - ○ *the path π that maximizes the correct number of individual states, i.e., the path where the states are individually most likely*
  - ○ *the most probable single path, maximize $P(\pi | O, \theta)$ or equivalently $P(\pi, O | \theta)$ – Viterbi algorithm*

- The optimal path is the path that maximizes $P(q_0, q_1, \dots, q_T | O, \theta)$
  let $\delta_t(i)$ be the highest probability path ending in state i

  $$\delta_t(i) = \max_{q_0, q_1, \dots, q_{t-1}} P(q_0, q_1, \dots, q_t = i, o_0, o_1, \dots, o_t | \theta)$$

  $$\delta_t(j) = \max_i [\delta_t(i) a_{ij}] b_j(o_{t+1})$$

- Keep track of argument that maximizes $\delta_t(j)$ at each position *t*

Problem 3: Learning

- Find the parameters, $\hat{\theta}$, that maximizes $P(O | \theta)$

  $$\theta = \underset{\theta}{\operatorname{argmax}} \left(P(O | \theta) \right)$$

- No analytical solution requires iterative solution (Baum-Welch algorithm)
  1. *initial model θ_0, repeat*
  2. *compute parameters θ_i based on $\theta_{i-1}$and observations O*
  3. *if $\log P(O|\theta_i) - \log P(O|\theta_{i-1}) < \varepsilon$, stop*

     else, accept θ_i, goto 2.

- With Baum-Welch algorithm likelihood is proven to be greater or equal at each step

Training

- Need to update the transition probabilities. The probability of being in state *i* at time *t* and state *j* at time *t+1* is $\xi(i, j)$

$$\xi(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{P(O|\theta)}$$

$$= \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}$$

- The probability of being in state *i* at *time t, given the observed sequence O is* $\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}$ or in terms of ξ, $\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j)$

- Derived quantities
  expected number of times state i is used $\sum_{t=0}^{T}\gamma_y(i)$
  expected number of transitions from state i to state *j* $\sum_{t=0}^{T-1}\xi_t(i,j)$

- Baum-Welch parameter updates
  - ○ $\omega_i = \gamma_0(i)$ *probability of starting in state I*
  - ○ $a_{ij} = \frac{\sum_{t=0}^{T}\xi_t(i,j)}{\sum_{t=0}^{T}\gamma_t(i)}$
  - ○ $b_j(k) = \frac{\sum_{t=0}^{T}\delta_{o_t}\gamma_t(j)}{\sum_{t=0}^{T}\gamma_t(j)}$ *where* $\delta_{o_t} = \begin{cases} 1, & \text{if } o_t = k \\ 0, & \text{if } o_t \neq k \end{cases}$

4.2.1 3D HMM

HMM is exploring a latent statistical model to describe the observed statistical phenomenon, assuming the observed variations of sequence of events were generated from an underlying inherent Markovian stochastic process. The model is used for recognition, classification and interpretation of future sequence of events.

1D HMM is applied to a one dimensional spatial or temporal chain, e.g., speech recognition, musical score analysis, and sequencing problems in bioinformatics. 3D HMM is applied to a two dimensional spaces, e.g., aerial image segmentation, automatic face recognition.

3D HMM is an extension of 3D HMM into a spatio-temporal volume in which consecutive frames of video images are stacked to form a third dimension - time.

Suppose there are M states $\{1,2,\ldots,M\}$ for each node (i,j,k) in a 3D structure, $i = \{1,2,\ldots,I\}$, $j = \{1,2,\ldots,J\}, and\ k = \{1,2,\ldots,K\}$ where I, J are the numbers of row and column of each frame and K is the number of frames in the original image, the feature vector (observation) is $o(i,j,k)$, the corresponding hidden state is $s(i,j,k)$, and the class of the node is $c(i,j,k)$. The transition probability of state $s(i,j,k)$ depends on its adjacent neighboring states in vertical, horizontal and across the frame directions following a predefined lexicographic order as

$$(i',j',k') < (i,j,k) \text{ if } k' < k \text{ or } k' = k, j' < j \text{ or } k' = k, j' = j, i' < i.$$

A graphical illustration for the 3D grid is depicted in Figure 13.

We can define the state transition probability as

$$P\{s_{i,j,k} = l \mid \omega\} = a_{p,m,n,l}$$

where $\omega = \{(s_{i'j'k'}, u_{i'j'k'}) : (i', j', k') < (i, j, k)\}$ is the set of states and feature vectors of all points preceding (i, j, k) in the lexicographic order and their state values are p, m and n. where $p = s_{i,j,k-1}$, $m = s_{i-1,j,k}$ $and\, n = s_{i,j-1,k}$ [16].

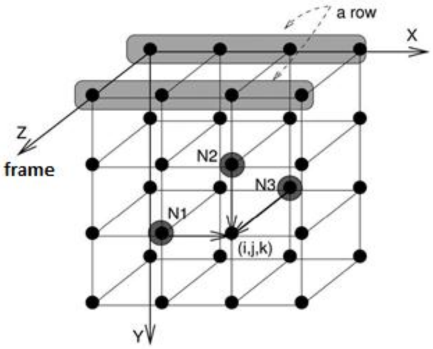


Figure 13. A pictorial 3D HMM structure (Joshi et al. [16]).

We can define the emission probability as normal density distribution. Given the state $s_{i,j,k}$ of a point (i, j, k), the feature vector $o_{i,j,k}$ follows a multivariate Gaussian distribution with a covariance matrix Σ_l and a mean vector μ_l determined by the state $l = \{1, \dots, M\}$ as follows [16]:

$$b_l(o) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_l|}} e^{-(1/2)\,(o-\mu_l)'\,\Sigma_l^{-1}(o-\mu_l)}.$$

If the state $s_{i,j,k}$ of a point *(i, j, k*) is known, its observed vector $o_{i,j,k}$ is conditionally independent of the rest of the points in the 3D grid, we can define a Viterbi training algorithm for estimation of Parameters as follows.

In 1-D HMM a Baum-Welch algorithm, a special case of expectation-maximization algorithm is usually used to derive local maximum likelihood estimate of the parameters of HMM. An approximation to the Baum-Welch algorithm is the Viterbi training, in which each observation and its assigned parameters were used to get the most likely state sequence as shown in (1). Then the sequence was used to re-estimate the hidden parameters as shown in (2) – (4). Instead of maximizing the likelihood of observed data, the Viterbi training is to maximize the probability of the most likely sequence of states and ends up saving significant computational time by sacrificing some accuracy [16].

$$\{s^*_{i,j,k} : (i,j,k) \in C\} = \underset{s_{i,j,k}:(i,j,k)\in C}{\arg\max}\, P(s_{i,j,k} : (i,j,k) \in C \,|\, o_{i,j,k}(i,j,k) \in C; \lambda^{(t)})$$

$$= \underset{s_{i,j,k}:(i,j,k)\in C}{\arg\max}\, P(s_{i,j,k}, o_{i,j,k} : (i,j,k) \in C; \lambda^{(t)}) \quad (1)$$

$$\mu_\ell^{(t+1)} = \frac{\sum_{(i,j,k)\varepsilon C} o_{i,j,k} I(s^*_{i,j,k}=l)}{\sum_{(i,j,k)\varepsilon C} I(s^*_{i,j,k}=l)} \quad (2)$$

Copyright © Society for Science and Education United Kingdom

$$\Sigma_l^{(t+1)} = \frac{\sum_{(i,j,k)\varepsilon C}(o_{i,j,k}-o_l^{(t+1)})(o_{i,j,k}-o_l^{(t+1)})' I(s^*_{i,j,k}=l)}{\sum_{(i,j,k)\varepsilon C} I(s^*_{i,j,k}=l)} \quad (3)$$

$$a^{(t+1)}_{p,m,n,l} = \frac{\sum_{(i,j,k)\varepsilon C} I,(s^*_{i,j,k-1}=p)I(s^*_{i-1,j,k}=m)I(s^*_{i,j-1,k}=n)I(s^*_{i,j,k}=l)}{\sum_{(ij,k)\varepsilon C} I(s^*_{i,j,k-1}=p)I(s^*_{i-1,j,k}=m)I(s^*_{i,j-1,k}=n)} \quad (4)$$

Where $C = \{(i,j,k), 1 \le i \le I, 1 \le j \le J, 1 \le k \le K\}$, s* is the optimal state sequence, $I(\cdot)$ is the indicator function that equals 1 when the argument is true and 0 otherwise, λ is collection of parameters of 3D HMM [16].

In 1-D HMM Viterbi algorithm for finding the states $\{s_t^*\}$ requires $wM(M-1)$ comparisons and wM memory records (assuming the length of HMM is w and M possible states). Hence the computational and storage complexities are $O(wM^2)$and $O(wM)$ respectively. For a 3D HMM there are M^{w^3}possible combinations of states for the entire 3D grid. When Viterbi algorithm is applied to frame-by-frame, it avoids exhaustive search along Z axis, but it still needs to consider all the possible combinations of states in each frame. Therefore, the computational complexity for the optimal set of states is at least $O(wM^{2w^2})$.

Joshi, et al. [16] proposed a locally optimal algorithm to search for the set of states with the MAP probability based on row sequence, then row by row adjustment. It improves the complexity of one iteration of the proposed algorithm to $O(w^3M^2)$ due to 3D grid total of w^2 rows.

Jiten and Merialdo [15] introduced a three dimensional dependency tree HMM (3D-DT HMM) which relaxes the dependencies between neighboring state nodes to a random uni-directional dependency. It allows us to estimate the model parameters along a 3D path linearly through a random decision tree, as the following figure shows. Hence, it maintains a linear computational complexity.

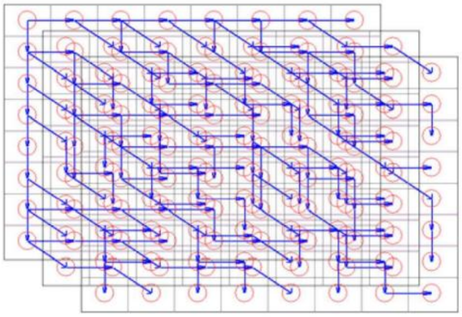


Figure 14. A pictorial 3D-DT (Dependency Tree) HMM structure (Jiten et al. [15]).

5 Experiments and Analysis

The experiments are performed as follows:

- First, select testing video dataset that has variations in the face profile such as sex, age and ethnicity with variety of partial occlusions on their face such as hair, glasses, headscarf etc. (see Figure 15)
- Split video into frames and run interpolated optical flow code to obtain optical flow vectors
- Select block size and run HOOF (Histogram of Oriented Optical Flow) code to obtain histogram for each block of optical flow vectors
- Run VQ (Vector Quantization) code to obtain BoW (bag of words) for all histograms
- Transfer all histogram data to data in terms of BoW and feed them to 3D HMM code for each facial expression to obtain parameters
- Run the parameters against the validation set.

5.1 Data Set

The most widely used database for research on facial expression recognition is the Cohn-Kanade facial expression database [17]. It contains image sequences of approximately 100 subjects, each posing a set of 23 facial displays. They were labeled with FACS codes in addition to basic emotions. There are two main limitations of this database [27]. First, the facial expression recording ends at the apex of the facial motion, instead of full cycle of onset, apex and offset in motion research. Secondly, many recordings have date/time stamp over the chin of the subject making the changes in the appearance of chin difficult to see and track.

MMI facial expression database was developed to overcome the limitations. It was conceived in 2002 by Maja Pantic, Michel Valstar and Ioannis Patras [27]. Initially it was focused on collections of high-quality still images of AUs and AU combinations. It video-recorded 1767 clips of 20 participants (Sessions 1 to 1767). Later data to distinguish the six basic emotions were added and formed the second part of the MMI database. It video-recorded 238 clips of 28 subjects displaying six basic emotions (Sessions 1767 to 2004). Part III is similar to part I, consisting of high quality still images with all AUs and six basic emotions (Sessions 2401-2884). Recently, spontaneous data were added to form Part IV (Sessions 2005 to 2388) and Part V (Sessions 2895 to 2903).

Part II of MMI database was used for this research. Six basic emotions of 20 randomly selected subjects were taken for the study.

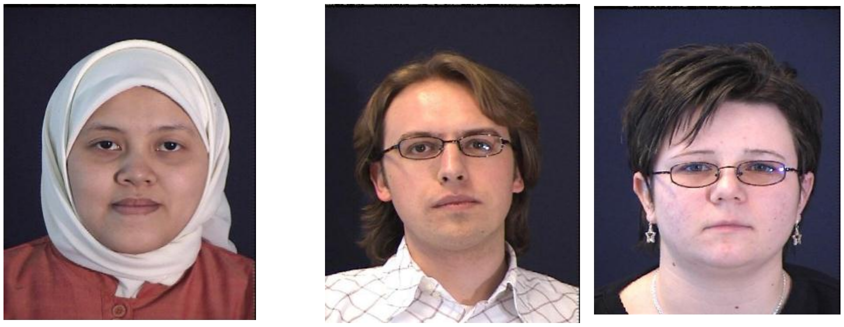


Figure 15. Example of subjects in MMI facial expression database (size 720x576).
(Source: http://mmifacedb.eu/)

5.2 Optical Flow Outputs

After extracting 20 videos of the six basic emotions (Happiness, Sadness, Anger, Fear, Disgust and Surprise), we split each video into 7 frames and feed the frames pairwise to optical flow program to obtain optical flow vectors. Figure 16 - 18 shows the results of the two emotions – Anger and happiness. The first row depict the extracted frames of videos, the second row exhibit the color-coded optical flow diagrams calculated from frame pairs. Figure 19 and 20 display the first six histograms of the optical flow magnitude and orientation from anger data set.

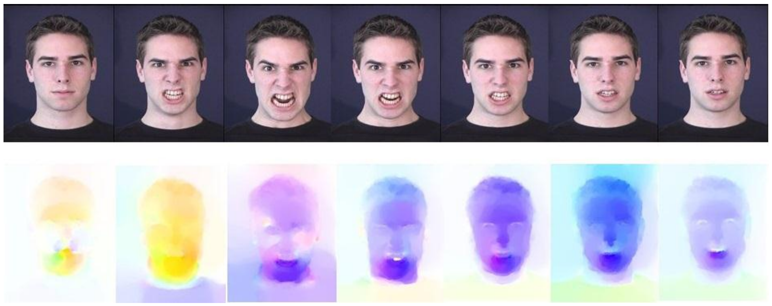


Figure 16. (Top) An example of 7 frames from Anger data set. (Bottom) The optical flows obtained.

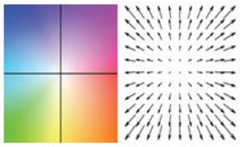


Figure 17. Optical flow color-code (hue for direction and saturation for magnitude).

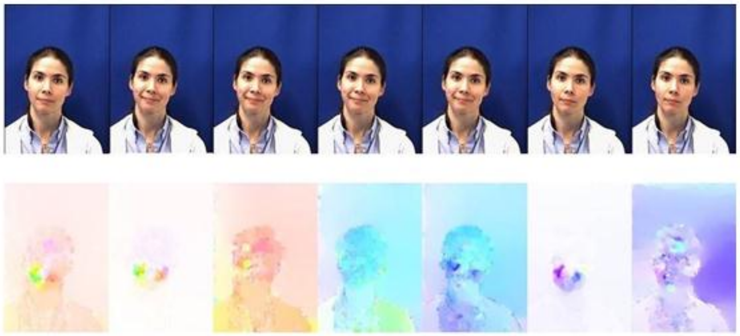
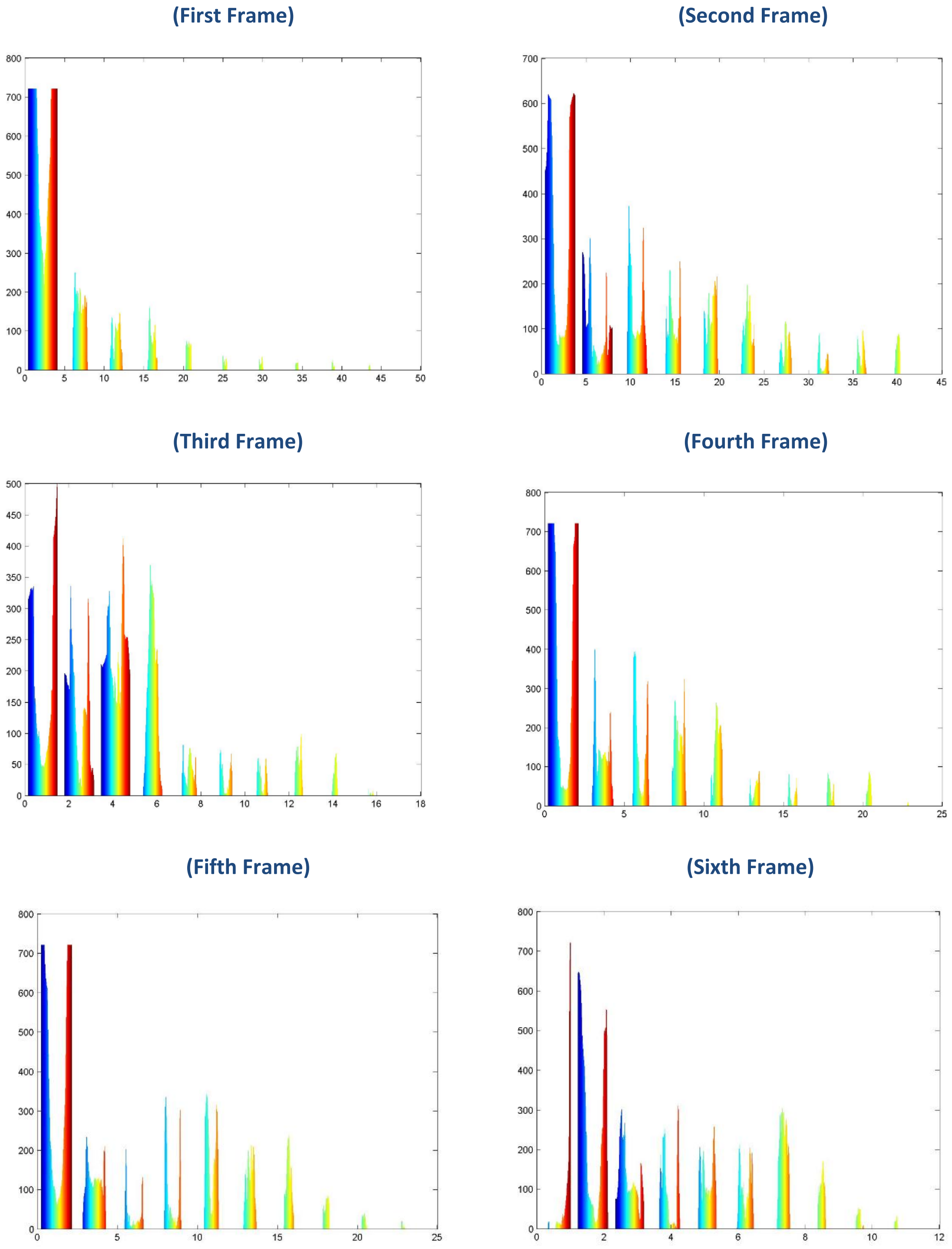


Figure 18. (Top) An example of 7 frames from Happiness data set. (Bottom) The optical flows.


(First Frame)

(Second Frame)

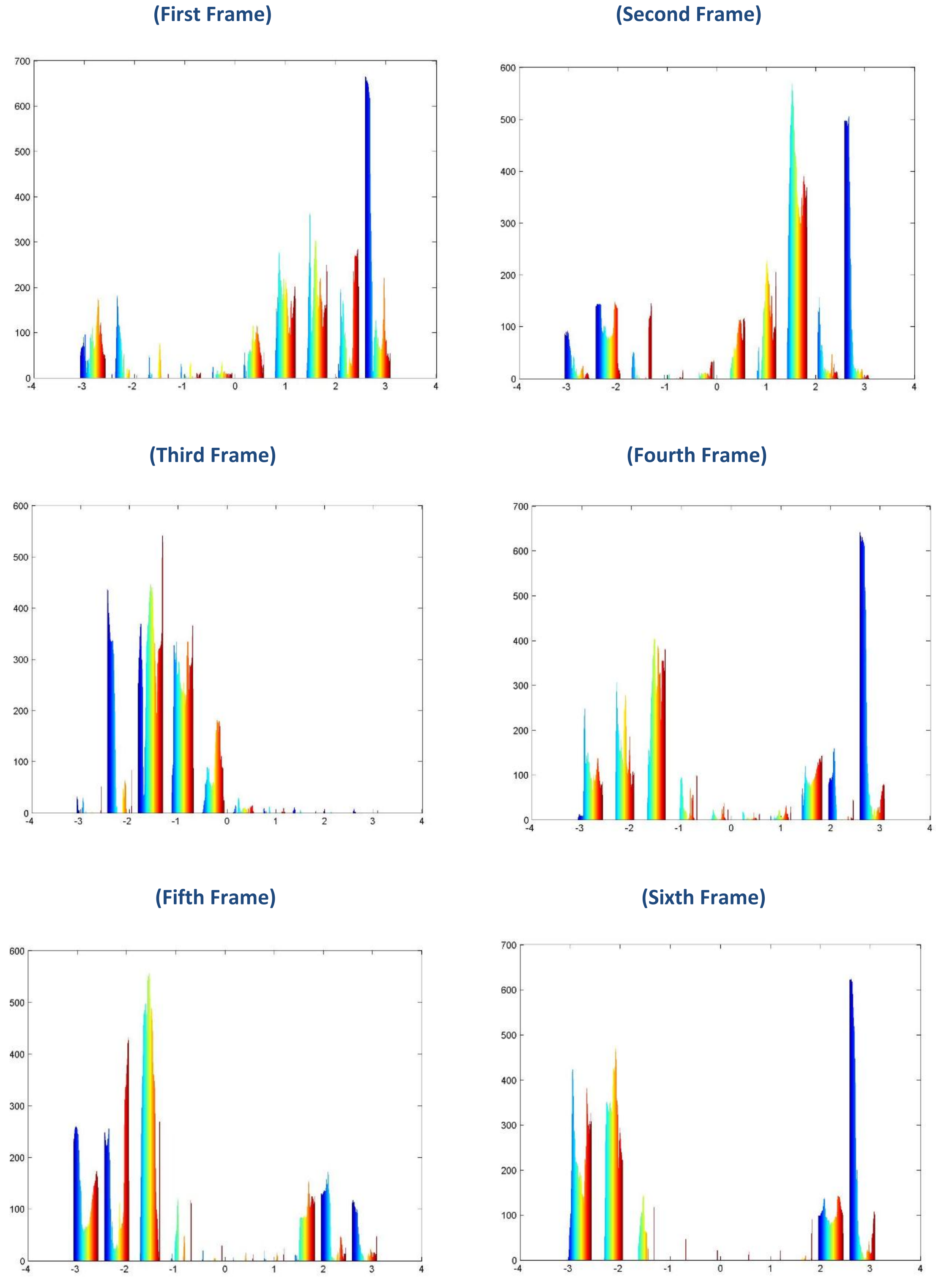
(Third Frame)

(Fourth Frame)

(Fifth Frame)

(Sixth Frame)

Figure 19. Histogram of magnitude of optical flow from Anger data set in Figure 16.



(First Frame)

(Second Frame)

(Third Frame)

(Fourth Frame)

(Fifth Frame)

(Sixth Frame)

Figure 20. Histogram of orientation of optical flow from Anger data set in Figure 16.

5.3 Histogram of Oriented Optical Flow (HOOF) Outputs

We use HOOF algorithm developed by Chaudhry et al. [5]. As indicated earlier using histogram as the descriptor has advantage of being robust to small geometric distortions and small errors in the region detection. Raw optical flows may not be useful as a descriptor if the number of pixels is different in the face and the optical flow computation is prone to background noise, scale variation and direction of movement. No two faces are alike but faces are symmetric. The eyebrow raise on the right half of face is same as the eyebrow raise on the left half of face. The lip stretch to the right is same as to the left with only difference of direction. We need a descriptor from optical flow but is invariance to scale and direction of motion.

First, we computer optical flows at every frame of the video. Each flow vector is binned according to its primary angle from the horizontal axis and it is weighted according to its magnitude. Thus, all optical flow vectors, $f = [u, v]^T$ with direction of $\theta = \tan^{-1}\left(\frac{v}{u}\right)$ in the range

$$-\frac{\pi}{2} + \pi\frac{b-1}{B} \leq \theta < -\frac{\pi}{2} + \pi\frac{b}{B}$$

will have $\sqrt{u^2 + v^2}$ added to the sum in bin $b\ (1 \leq b \leq B)$. There is a total of B bins. B can be varied (Figure 18 shows B = 4). Finally the histogram is normalized to a sum of 1.

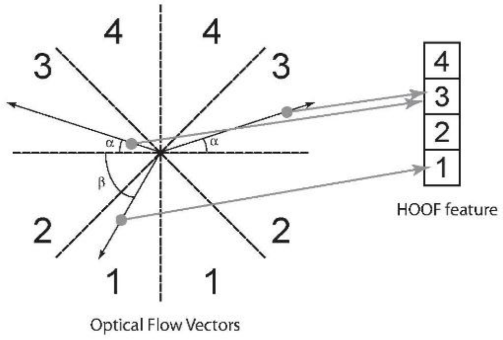
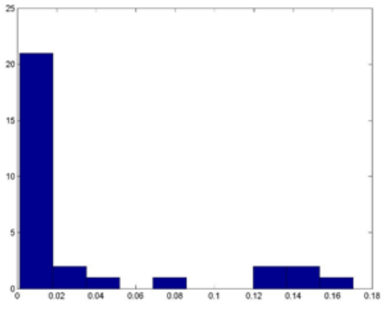
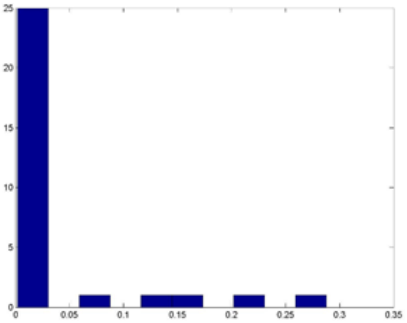


Figure 21. An example of histogram formation with four bins (Chaudhry et al. [5])

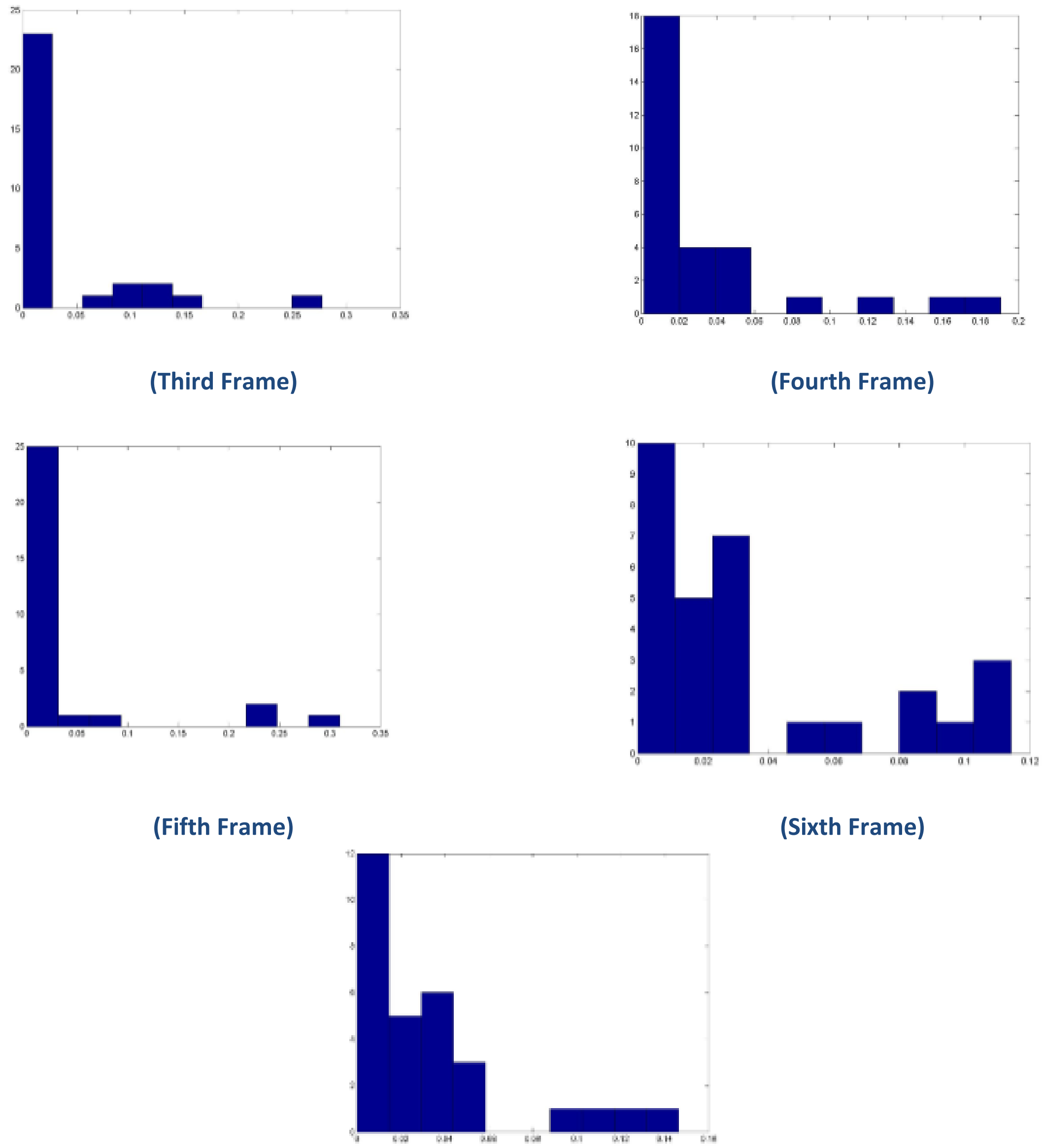
The following figure shows the histogram of the whole image.



(First Frame)



(Second Frame)



(Third Frame)

(Fourth Frame)

(Fifth Frame)

(Sixth Frame)

(Seventh Frame)

Figure 22. Histogram of Oriented Optical Flow from whole image of Anger Data set in Figure 16.

However, in our application we divided the whole image by blocks of size 24x24. There is 720 blocks (30x24) of images for each frame and therefore 720 HOOF histograms for each frame.

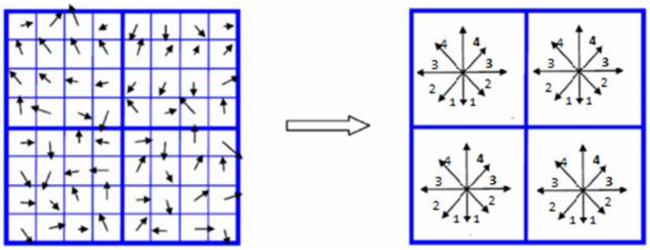


Figure 23. HOOF generated from a 4x4 block of image (our experiment uses blocks of 24x24).

$$\begin{array}{ccc} \textit{Image Blocks with OF } (u,v) & & HOOF\ (B=4) \\ \begin{bmatrix} \begin{bmatrix} (u,v) & \cdots & (u,v) \\ \vdots & \ddots & \vdots \\ (u,v) & \cdots & (u,v) \end{bmatrix} & \cdots & \begin{bmatrix} (u,v) & \cdots & (u,v) \\ \vdots & \ddots & \vdots \\ (u,v) & \cdots & (u,v) \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} (u,v) & \cdots & (u,v) \\ \vdots & \ddots & \vdots \\ (u,v) & \cdots & (u,v) \end{bmatrix} & \cdots & \begin{bmatrix} (u,v) & \cdots & (u,v) \\ \vdots & \ddots & \vdots \\ (u,v) & \cdots & (u,v) \end{bmatrix} \end{bmatrix} & \rightarrow & \begin{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} & \cdots & \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} & \cdots & \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \end{bmatrix} \end{array}$$

Figure 24. HOOF generation in matrix form.

5.4 Bag of Words (BoW) Outputs

After extracting 20 videos of the six basic emotions (Happiness, Sadness, Anger, Fear, Disgust and Surprise), we obtain observation symbol sequence via vector quantization. Because a discrete 3D HMM is employed to decode the temporal variations of the facial expression features and the discrete HMMs are usually trained and tested using sequence of symbols, the feature vectors need to be transformed to symbols by comparing with the codeword vectors of a codebook. To obtain the codebook, vector quantization algorithm is performed on the feature vectors from the training datasets. A clustering algorithm (e.g., K-means) is generally used for codebook generation. It selects the initial centroids and splits the whole dataset based on the centroids. Then, it recalculate the centroids from the split datasets and continues to split the dataset according to the codeword size. The optimization is obtained when the distortion between two consecutive centroids is minimized. As shown in figure 25 each centroid vector is a codeword of a code book (similar to word dictionary).

The main idea is to quantize the extracted feature vector of each patch/block into one of visual words, and then represent the observations of 3D HMM by sequences of the visual words. A codeword can be considered as a representative of several similar patches/blocks.

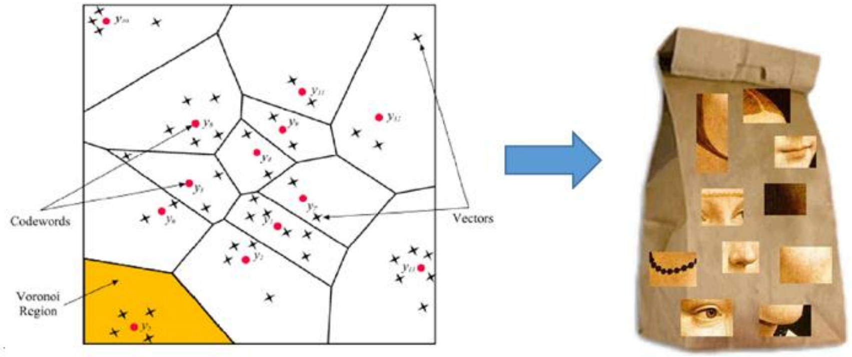


Figure 25. Left shows VQ (Vector Quantization) codewords in 2-dimensional space with input vectors (x), codewords (●) and Voronoi regions. Right shows a pictorial bag of words (BoW) from an image of a face. (Source: http://www.mqasem.net/vectorquantization/vq.html and http://people.csail.mit.edu/fergus/iccv2005/bagwords.html)

5.5 3D HMM Classification Result

Table 1. 3D HMM classification confusion matrix

	Happiness	Sadness	Anger	Fear	Disgust	Surprise
Happiness	92.7%	0.5%	1.6%	3.7%	1.0%	0.5%
Sadness	0.8%	87.8%	3.2%	2.4%	5.0%	0.8%
Anger	0.5%	5.3%	87.3%	3.2%	3.5%	1.2%
Fear	2.0%	3.1%	4.2%	89.2%	1.0%	0.5%
Disgust	0.8%	2.1%	4.2%	0.5%	91.8%	0.6%
Surprise	0.5%	0.5%	4.2%	0.4%	0.5%	93.9%

6 Conclusion

In this work we have shown the viability of doing facial expression recognition by characterizing facial actions in terms of optical flows, transforming magnitude and direction of optical flow to a histogram and subsequently into bag of words for classification of 3D HMM. A regular HMM has limited dimension of either time or space but not both. It was first applied to facial expression recognition by Yamato et al. [43]. We used a 3D spatio-temporal HMM to apply to facial expression recognition and showed how the extension of dimension is more intuitive to track the motion or behavior in a video. Future extensions include exploring other spatio-temporal features and extending the current 1D or 2D HMM applications to the spatio-temporal domain. Schmidt et al. [32] compared three different features (principal component analysis, orientation histograms and optical flow estimation) on emotional recognition with HMM. Using features detected at multiple scales and segmentation of video frames at shorter interval should improve performance at the cost of computer time. Another possible direction of future work is to incorporate a parallelism in running of 3D HMM using a massively parallel hardware architecture and high performance of floating point arithmetic and memory operations on GPU (Graphics Processing Unit).

ACKNOWLEDGEMENT

We thank Jasser Jasser of ECE Department of Oakland University for assistance with development and running of the experiment and sharing of his pearls of wisdom with us during the course of this research.

REFERENCES

[1]. Bashyal, Shishir, and Ganesh K. Venayagamoorthy. 2008. Recognition of facial expressions using gabor wavelets and learning vector quantization. *Engineering Applications of Artificial Intelligence* 21 (7): 1056-64.

[2]. Bay, Herbert, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. 2008. Speeded-up robust features (SURF). *Computer Vision and Image Understanding* 110 (3): 346-59.

[3]. Bouchaffra, Djamel. 2010. Conformation-based hidden markov models: Application to human face identification. *Neural Networks, IEEE Transactions on* 21 (4): 595-608.

[4]. Bouchaffra, Djamel, and Abbes Amira. 2008. Structural hidden markov models for biometrics: Fusion of face and fingerprint. *Pattern Recognition* 41 (3): 852-67.

[5]. Bruhn, Andrés, Joachim Weickert, and Christoph Schnörr. 2005. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision* 61 (3): 211-31.

[6]. Chaudhry, Rizwan, Arunkumar Ravichandran, Georg Hager, and René Vidal. 2009. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. Paper presented at Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, .

[7]. Cohen, Ira, Nicu Sebe, Ashutosh Garg, Lawrence S. Chen, and Thomas S. Huang. 2003. Facial expression recognition from video sequences: Temporal and static modeling. *Computer Vision and Image Understanding* 91 (1): 160-87.

[8]. Donoser, Michael, and Horst Bischof. 2006. Efficient maximally stable extremal region (MSER) tracking. Paper presented at Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, .

[9]. Draper, Bruce A., Kyungim Baek, Marian Stewart Bartlett, and J. Ross Beveridge. 2003. Recognizing faces with PCA and ICA. *Computer Vision and Image Understanding* 91 (1): 115-37.

[10]. Ekman, Paul, and Wallace V. Friesen. 1977. Facial action coding system.

[11]. Elramsisi, AM, MA Zohdy, and NK Loh. 1991. A joint frequency-position domain structure identification of nonlinear discrete-time systems by neural networks. *Automatic Control, IEEE Transactions on* 36 (5): 629-32.

[12]. Hamm, Jihun, Christian G. Kohler, Ruben C. Gur, and Ragini Verma. 2011. Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders. *Journal of Neuroscience Methods* 200 (2): 237-56.

[13]. Horn, Berthold K., and Brian G. Schunck. 1981. Determining optical flow. Paper presented at 1981 Technical symposium east, .

[14]. Jafri, Rabia, and Hamid R. Arabnia. 2009. A survey of face recognition techniques. *Journal of Information Processing Systems* 5 (2): 41-68.

[15]. Jitén, Joakim, and Bernard Merialdo. 2007. Video modeling using 3D hidden markov model. Paper presented at VISAPP (1), .

[16]. Joshi, Dhiraj, Jia Li, and James Z. Wang. 2006. A computationally efficient approach to the estimation of two-and three-dimensional hidden markov models. *Image Processing, IEEE Transactions on* 15 (7): 1871-86.

[17]. Kanade, Takeo, Jeffrey F. Cohn, and Yingli Tian. 2000. Comprehensive database for facial expression analysis. Paper presented at Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, .

[18]. Karray, Fakhreddine, Milad Alemzadeh, Jamil Abou Saleh, and Mo Nours Arab. 2008. Human-computer interaction: Overview on state of the art.

[19]. Khan, Aftab Ali, and MA Zohdy. 1997. A genetic algorithm for selection of noisy sensor data in multisensor data fusion. Paper presented at American Control Conference, 1997. Proceedings of the 1997, .

[20]. Ledoux, H., and CM Gold. 2005. Interpolation as a tool for the modelling of three-dimensional geoscientific datasets. Paper presented at Proceedings ISPRS

[21]. Lien, James J., Takeo Kanade, Jeffrey F. Cohn, and Ching-Chung Li. 1998. Automated facial expression recognition based on FACS action units. Paper presented at Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on, .

[22]. Lien, Jenn-Jier James. 1998. *Automatic Recognition of Facial Expressions using Hidden Markov Models and Estimation of Exprssion Intensity*.

[23]. Lowe, David G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60 (2): 91-110.

[24]. Lucas, Bruce D., and Takeo Kanade. 1981. An iterative image registration technique with an application to stereo vision. Paper presented at IJCAI, .

[25]. Mahoor, Mohammad H., Steven Cadavid, Daniel S. Messinger, and Jeffrey F. Cohn. 2009. A framework for automated measurement of the intensity of non-posed facial action units. Paper presented at Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on, .

[26]. Matas, Jiri, Ondrej Chum, Martin Urban, and Tomás Pajdla. 2004. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing* 22 (10): 761-7.

[27]. Moreno, Plinio, Alexandre Bernardino, and José Santos-Victor. 2005. Gabor parameter selection for local feature detection. In *Pattern recognition and image analysis.*, 11-19Springer.

[28]. Pinto, Sílvia Cristina Dias, Jesús P. Mena-Chalco, Fabrício Martins Lopes, Luiz Velho, and Roberto Marcondes Cesar Jr. 2011. 3D facial expression analysis by using 2D and 3D wavelet transforms. Paper presented at Image Processing (ICIP), 2011 18th IEEE International Conference on, .

[29]. Praseeda Lekshmi, V., and M. Sasikumar. 2009. Analysis of facial expression using gabor and SVM. *International Journal of Recent Trends in Engineering* 1 (2): 1-43.

[30]. Rabiner, Lawrence R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77 (2): 257-86.

[31]. Sánchez, A., José V. Ruiz, Ana Belén Moreno, Antonio S. Montemayor, Javier Hernández, and Juan José Pantrigo. 2011. Differential optical flow applied to automatic facial expression recognition. *Neurocomputing* 74 (8): 1272-82.

[32]. Schmidt, Miriam, Martin Schels, and Friedhelm Schwenker. 2010. A hidden markov model based approach for facial expression recognition in image sequences. In *Artificial neural networks in pattern recognition.*, 149-160Springer.

[33]. Shan, Caifeng, Shaogang Gong, and Peter W. McOwan. 2009. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing* 27 (6): 803-16.

[34]. Sirovich, Lawrence, and Michael Kirby. 1987. Low-dimensional procedure for the characterization of human faces. *Josa a* 4 (3): 519-24.

[35]. Soyel, Hamit, and Hasan Demirel. 2012. Localized discriminative scale invariant feature transform based facial expression recognition. *Computers & Electrical Engineering* 38 (5): 1299-309.

[36]. Tian, Ying-li, Takeo Kanade, and Jeffrey F. Cohn. 2002. Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity. Paper presented at Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on, .

[37]. ———. 2001. Recognizing action units for facial expression analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23 (2): 97-115.

[38]. Tsai, Ching-Chih, You-Zhu Chen, and Ching-Wen Liao. 2009. Interactive emotion recognition using support vector machine for human-robot interaction. Paper presented at Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, .

[39]. Turk, Matthew, and Alex P. Pentland. 1991. Face recognition using eigenfaces. Paper presented at Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on, .

[40]. Uddin, Md Zia, JJ Lee, and T-S Kim. 2009. An enhanced independent component-based human facial expression recognition from video. *Consumer Electronics, IEEE Transactions on* 55 (4): 2216-24.

[41]. Valstar, Michel, and Maja Pantic. 2006. Fully automatic facial action unit detection and temporal analysis. Paper presented at Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on, .

[42]. Wu, Jian, Zhiming Cui, Victor S. Sheng, Pengpeng Zhao, Dongliang Su, and Shengrong Gong. 2013. A comparative study of SIFT and its variants. *Measurement Science Review* 13 (3): 122-31.

[43]. Yamato, Junji, Jun Ohya, and Kenichiro Ishii. 1992. Recognizing human action in time-sequential images using hidden markov model. Paper presented at Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on, .

[44]. Zohdy, M., D. Bouchaffra, and J. Quinlan. 2001. Optimal mapping from chromosome space to feature space for solving sequential pattern recognition problems. Paper presented at Circuits and Systems, 2001. MWSCAS 2001. Proceedings of the 44th IEEE 2001 Midwest Symposium on, .