Automatic Facial Expression Recognition using Bags of Motion Words

Liefei Xu
http://www.cs.stevens.edu/~lxu1/
Philippos Mordohai
http://www.cs.stevens.edu/~mordohai/

Department of Computer Science,
Stevens Institute of Technology
Hoboken, NJ, USA

Abstract

We present a fully automatic approach for facial expression recognition based on a representation of facial motion using a vocabulary of local motion descriptors. Previous studies have shown that motion is sufficient for recognizing expressions. Moreover, by discarding appearance after optical flow estimation, our representation is invariant to the subjects' ethnic background, facial hair and other confounders. Unlike most facial expression recognition approaches, ours is general and not specifically tailored to faces. Annotation efforts for training are minimal, since the user does not have to label frames according to the phase of the expression, or identify facial features. Only a single expression label per sequence is required. We show results on a database of 600 video sequences.

1 Introduction

The human visual system is highly specialized in recognition tasks related to people and, in particular, faces. We are very adept at identifying discriminating features in people's faces and sensitive to the emotional states manifested by their facial expressions. Arguably, we are less sensitive in similar discriminative tasks for animals and objects. Computer vision research has followed a similar path by developing specialized techniques for face and expression recognition and more general methods for object recognition. Here, we investigate whether a general approach for expression recognition based on motion is feasible.

Facial expression recognition has justifiably attracted a lot of attention resulting in a large corpus of publications, surveyed in [17, 29, 36, 46]. A fundamental study of human perception of expressions was conducted by Bassili [3]. In the study, subjects performed expressions with their faces darkened and covered with white dots. When these videos were played back so that only the white dots were visible, moving displays of the six basic emotions were recognized more accurately than static displays of the white spots at the apex of the expressions. An additional advantage of an approach that relies on motion features is that it achieves desirable invariances to the subjects' ethnic background, facial hair, make up and to some degree spectacles, hats and other accessories. These act as distinctive features for optical flow estimation, rather than obstacles for recognition.

Our approach differs from previous work by not being tailored to faces; we use general motion features, instead of highly detailed models of the human face, eyes and mouth. Since

BMVC 2010 doi:10.5244/C.24.13

we do not employ face-specific models, we do not rely on FACS coding, but only use a single expression label per sequence for training. FACS is the Facial Action Coding System of Ekman and Friesen [14] that objectively encodes expressions in terms of a number of pre-specified Action Units (AUs), which are the smallest visibly distinguishable changes in facial display. FACS addresses the lack of specificity when expressions are described in terms of the emotional state that is being projected. Here, we present recognition results on videos of the six universal expressions: happiness, sadness, fear, surprise, anger and disgust. Our method could also be trained on data labeled in other ways, for instance according to AU activations, without modification.

An additional constraint we imposed on our method, besides minimal annotation, is that it should be applicable to videos captured by a single unknown camera. Face and expression recognition can benefit greatly from the availability of additional modalities such as 3D or infrared, but this restricts them to processing data captured by a particular sensor or at a specific location. We show results on a the 3D database BU-4DFE [44] provided to us by its authors, but *we only use a single video stream for testing and training*. The 3D models were not used in any way to augment the images.

Finally, we require our method to be fully automatic. The user should not have to identify neutral faces or the apex of the expression in the sequence, and more importantly the user should not have to localize the eyes or other features in the first frame in order to align a model with the input. Manual initialization is rather common in top-performing systems, probably because some of the key points on the face do not have unique and distinctive appearance, while other features need to be localized with very high precision. Our approach tolerates low repeatability in the placement of our dense set of descriptors. This is due to the SIFT-style encoding [25] we employ and to the overlaps between nearby descriptors.

These requirements are met with the use of a bag of words representation, as in [33], where the words encode optical flow in local patches. The motion from frame to frame is represented by histograms of these words, which enable fast queries in a database containing frames from labeled sequences. Once labels for the frames of the unknown sequence have been estimated, the entire sequence can be assigned one of the labels.

2 Related Work

The literature on facial expression recognition is too voluminous to review here. We refer readers to surveys [17, 29, 36, 46] and focus on methods that rely on motion cues. Mase [26] was among the first to use local optical flow estimates to infer expressions. Yacoob and Davis [43] tracked rectangles around the facial features and used their motion as a representation for expressions. Black and Yacoob [6] defined parametric motion models for the eyes and mouth. The motion parameters were estimated from optical flow and used as inputs to a rule-based classifier. Tian et al. [35] used multi-state face and facial component models that take into account permanent and transient (furrows, wrinkles) features. The parameters of these models are the inputs to neural networks that recognize AUs.

Essa and Pentland [15] developed geometric and anatomical models of the human face and a temporal extension to FACS. Using these models and control theory, they were able to estimate muscle actuations from optical flow fields. Lien et al. [24] tracked feature points and principal components of dense optical flow to provide inputs to an HMM that recognizes AUs. Donato et al. [12] evaluated different motion-based techniques and concluded that Gabor wavelets for local representation coupled with ICA gave the best results on short (six-frame) sequences.

Cohen at al. [7] evaluated different Bayesian network classifiers for motion-based frame classification and HMMs for sequence classification. The underlying features were derived from the deformations of a mesh-based face tracker. A similar approach was undertaken by Kotsia and Pitas [22] who place the Candide grid on the face and use SVMs for classification according to the displacement of the vertices from a neutral expression to the apex.

All these methods require manual labeling of feature points or regions in the first frame of each sequence. The difference between fully automatic and manually assisted recognition rates appears to be fairly significant, when reported.

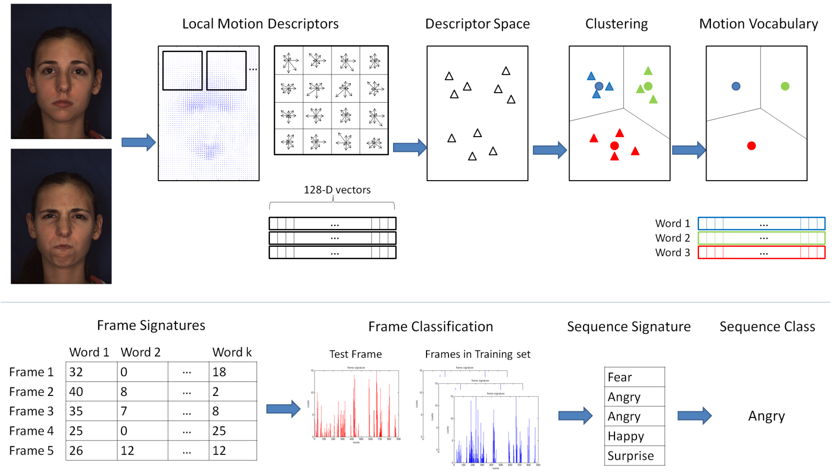
Fully automatic methods have been made possible in the last few years benefiting from progress in facial feature detection. A geometric method that relies on tracking 20 feature points was presented by Valstar and Pantic [38]. Tong et al. [37] learned the time-varying relationships between AUs using dynamic Bayesian networks operating on Gabor wavelets. Wang and Lien [41] proposed a feature-tracking method that explicitly models rigid and non-rigid head motion using a 3D head model. HMMs classify expressions based on the motion of the tracked feature points. An approach similar to ours that uses a large number of generic descriptors was presented by Zhao and Pietikäinen [47]. Spatiotemporal local binary patterns are computed on the volume formed by stacking the input frames. AdaBoost is used for feature selection and SVMs for final classification. Anderson and McOwan [1] introduced a region and appearance based approach in which detected faces are divided into a pre-specified set of regions. Then, an SVM recognizes expressions based on the averaged optical flows inside these regions. Koelstra et al. [21] extended feature-based methods to take appearance information into account. They address the problem as dynamic texture recognition and use boosting to train frame classifiers and HMMs to classify sequences.

A different path, compared to the above highly specialized models, is typically taken when general motion is analyzed. Zelnik-Manor and Irani [45] used the magnitudes of spatiotemporal gradients to form descriptors of actions in video. Spatiotemporal gradients, however, depend on the appearance of the actors. This was alleviated by Efros et al. [13] who relied strictly on motion and recognized frames by finding maxima of correlation of global frame descriptors over time.

Our approach also borrows from the concept of the visual vocabulary [8, 33] that enables efficient retrieval in large databases. This concept has been extended to spatio-temporal inputs (videos) via a number of methods that detect and describe spatio-temporal features [11, 23, 27]. Due to lack of space, we refer readers to the survey by Wang et al. [40]. What is noteworthy is that these descriptors have been applied for human activity recognition and not for facial expression analysis. We conjecture that this is due to the fact that optical flow fields of faces are smooth and do not contain many regions that would be considered features by the above methods. We overcome this by placing descriptors on a regular grid.

3 Overview of the Approach

This section provides an outline of our approach. We assume that the videos contain one face that covers a large part of the image. Each video is labeled according to the displayed expression, but individual frames do not need to be labeled, fiducials do not need to be marked on the faces and the sequence does not need to begin or end with a neutral expression. A consequence of these minimal supervision requirements is that there is no prior information on which frames depict the onset, apex or offset of the expression.



	Word 1	Word 2		Word k
Frame 1	32	0	...	18
Frame 2	40	8	...	2
Frame 3	35	7	...	8
Frame 4	25	0	...	25
Frame 5	26	12	...	12

Figure 1: Overview of the main steps of our algorithm. See text for details.

For each video sequence, optical flow is computed using an implementation of Black and Anandan's method [4]. Since motion between two consecutive images of the sequence may be at sub-pixel levels, we concatenate multiple optical flow fields and trace the combined motion of each pixel to generate optical flow fields with larger motions.

On these motion fields, we compute a dense set of local descriptors of the motion vectors following the encoding of the SIFT descriptor [25], but without scale detection or rotation invariance. We, then, cluster the descriptors to generate a *motion vocabulary* in which the words are the cluster centroids [33]. During testing, we compute motion descriptors in the same way and assign each descriptor to one of the words. Thus, we obtain a histogram of word frequencies in each frame, which we use as the frame signature. We assign a label to each frame based on its signature. Finally, we classify sequences according to the labels assigned to the frames that comprise them. The processing steps are summarized in Fig. 1. We have compared several alternatives for the three main classification tasks: the assignment of descriptors to words, frame classification and sequence classification.

Before going into details, let us introduce the notation used in the remainder of the paper.

- A *frame* is one of the optical flow fields generated by concatenating optical flow estimates. A frame contains a 2D motion vector per pixel. The term frame here always refers to a motion vector field and not to an image.
- s_i denotes a local motion *descriptor*.
- A *word* is a cluster of descriptors represented by its centroid w_i.
- k is the total number of words in the vocabulary.
- A *frame signature* is a k-D histogram of word counts in a frame denoted by f_i.
- A *sequence signature*, denoted by e_i, is a 6D histogram that counts the number of frames in a sequence that have been labeled as displaying a particular expression.

4 Local Motion Descriptors and Motion Vocabulary

As mentioned above, the inputs to our method are video sequences on which optical flow is computed using [4]. The motion between two consecutive images is typically small. Therefore, to avoid low signal to noise ratios, we concatenate multiple optical flow fields to generate new fields with large motion estimates for each pixel. We, then, select the 20 frames with the largest motions from each sequence and discard the remaining frames. This is done to remove uninformative frames with very little motion when the face is in a neutral expression or is stationary at the apex of an expression.

4.1 Motion Descriptors and Motion Words

We have chosen an encoding scheme inspired by those of SIFT [25] and the Histogram of Oriented Gradients (HOG) [9]. SIFT and HOG features are computed on gradient maps, which are also 2D vector fields; thus, we can directly use the same encoding. The optical flow vector at each pixel votes for an orientation bin in a square cell of the image. In our implementation, we use 8 orientation bins defined over $0-360^\circ$ and 4×4 non-overlapping cells in the descriptor (as in SIFT), but we do not perform dominant orientation alignment or extrema detection in scale-space. Rotation invariance is not desirable for our purposes. We have experimentally verified that the use of 8 orientation bins provides sufficient invariance to small rotations due to head motion. Scale is fixed, but ideally should be provided by a face detector. Square cells are more invariant than log-polar cells to small translations of the descriptor and are thus preferable.

Descriptors are computed on a dense regular grid as in [9]. The placement of descriptors at dense grids has been shown to be more effective for recognition than placing them at detected feature locations [18, 28]. The resulting descriptors are 128D vectors, which are clustered to form a vocabulary of motion words, as in [33]. We use two different strategies for obtaining large vocabularies. In both cases, clustering is done without considering the labels of the sequence that produced each descriptor.

The first strategy is the use of two-level hierarchical k-means clustering. Initially, k-means clustering is applied on a large set (about 1M) of descriptors extracted from several sequences. To overcome memory limitations, we first partition the data into typically 128 clusters. On the second level, we do not subdivide each cluster into an equal number of sub-clusters. Instead, the number of subdivisions is set so that on average the same number of descriptors would be assigned to each second-level cluster. That is, a cluster with 3,000 descriptors will be divided into 30 sub-clusters, if the desired average cluster size is 100. Tests using sequence recognition rate as the criterion have shown this to be more effective than imposing an equal number of subdivisions on the first-level clusters.

We have also used the radius-based clustering (RBC) technique of Jurie and Triggs [20] following the approximation of Van Gemert et al. [39]. RBC sequentially detects new clusters by finding local maxima of density and clustering all descriptors within a preset radius. The local density maximum becomes the cluster's centroid and all previously unclustered descriptors within the radius are assigned to the new cluster. Instead of using mean-shift to detect modes in descriptor space, we select the descriptor with the maximum number of neighbors within the radius as the centroid of the new cluster and proceed to find the next centroid [39]. The loss of accuracy is negligible, while speed gains are large. For efficiency, density estimation is done on random subsets of the training set, as in [20]. During testing, descriptors outside all cells are assigned to the nearest word.

4.2 Describing Frames using the Vocabulary

Once a vocabulary has been computed, all frames are encoded in it. Each descriptor s_i takes the label of the nearest word w_j. We have evaluated the use of the L_2 (Euclidean) and the

Mahalanobis distance for this task using k-means clustering. For the latter, the covariance matrix Σ_s of the descriptors is computed before the vocabulary is generated. Before training and testing, all descriptors are pre-scaled by the square root of the inverse of the covariance matrix $\Sigma_s^{-\frac{1}{2}}$ as in [32, 33]. k-means clustering is applied on the resulting descriptors. As shown in Section 7, both distance functions provide very similar recognition performance (at the sequence level) with L_2 having a slight advantage. Therefore, the L_2 distance is used in the remainder to measure the dissimilarity between descriptors and/or words.

Recent results in the bag of visual words literature [30, 39] suggest that *soft assignment* of descriptors to words may be preferable to assigning each descriptor to the nearest word. Under this approach, each descriptor votes for a few neighboring words, with the votes weighted according to distance. The result is that descriptors that are close to the Voronoi boundaries between words contribute almost equal votes to the relevant words reflecting this ambiguity and reducing the sensitivity of the output to small perturbations of the descriptors. Having generated the vocabulary using standard k-means (L_2 distance), we adopt the method of [30] in which each descriptor votes for the r nearest words with weights that decay with distance, according to:

$$v_{ij} = e^{-\frac{d(s_i, w_j)^2}{2\sigma^2}}, \tag{1}$$

where $d()$ is the L_2 distance between the descriptor s_i and word w_j and σ is set such that the votes to the $(r+1)^{th}$ neighbor are small. The vector of weights $\vec{v}$ is L_1-normalized. Soft assignment outperforms hard assignment for small vocabularies (see Section 7).

For RBC, we only show results using L_2 which achieves the best recognition results.

In all cases, the output of this stage is a set of *k-D* frame signatures f_n for the frames of the training set that measure how many descriptors of frame n were assigned to word j. During testing, frame signatures are generated for the unlabeled frames as in training.

5 Frame Recognition

In this section, we present the second stage of classification in which frames are assigned expression labels. The inputs are a training set of labeled frames represented by *k-D* histograms of word counts and unlabeled queries of the same form. This is a challenging classification task because each of the six labels does not represent a compact cluster but rather encompasses frames from the onset, apex and offset of potentially different manifestations of the same emotion. (For example, one can easily recognize multiple different manifestations of fear in the BU-4DFE database.) While it is possible that finer clustering within the set of frames with the same expression label can be obtained in an unsupervised manner enabling the use of advanced classifiers, this is far from trivial and not pursued here.

We have evaluated several distance functions in the context of a nearest neighbor classifier that assigns to each frame the label of the nearest neighboring frame in the training set. In all cases, sequence classification accuracy is used as the evaluation criterion. Varying the number of nearest neighbors considered did not have a significant impact on accuracy, as long as that number was small. Therefore, test frames are assigned the label of the nearest neighbor throughout. We have evaluated the L_1, L_2 and Mahalanobis distances, as well as a quadratic form distance function described below. Mahalanobis distances are now computed by estimating the $k \times k$ covariance matrix of frame signatures Σ_f. Frame signatures are pre-multiplied by $\Sigma_f^{-\frac{1}{2}}$ before classification.

The quadratic form distance function [16] can be viewed as an alternative implementation of soft assignment or an approximation to the Earth mover's distance [31]. The motivation for using quadratic form distance functions, which were developed to measure distances between color histograms, is that we are measuring the distance between two vectors (frame signatures) that are defined on a non-orthogonal coordinate system, in which the basis vectors are the words. We can account for the similarity between the basis vectors by measuring the distance using the following quadratic form, where f_n and f_m are two frame signatures:

$$d_q(f_n, f_m)^2 = (f_n - f_m)^T A (f_n - f_m), \qquad a_{ij} = e^{-\frac{d(w_i, w_j)}{\sigma}}. \tag{2}$$

where the entries of the matrix A are functions of the L_2 distance between words w_i and w_j. We opted for the exponential form of the weights a_{ij}, as in [2]. We also tried a Gaussian variation, which performs slightly worse on our data.

Finally, we used a multi-class *Support Vector Machine* [6]. We train 15 one-against-one binary classifiers, using an RBF kernel, that decide between each pair of labels. The final label is produced by combining the 15 binary decisions.

The nearest neighbor classifier on frame signatures is efficient, but it is completely agnostic of the geometric relationships between the descriptors. This fact should be contrasted with all methods reviewed in Section 2 in which the spatial configuration of features or regions is of primary importance. We implemented *geometric verification* as template matching of the concatenated motion fields. For each query frame, we find the nearest M neighbors in the database as candidate frames. Then, we translate and scale the query frame in the range of 80% and 120% to find the best match among the candidates.

6 Sequence Classification

Each sequence is represented by 20 frames, selected as those with the largest motion, which have been assigned expression labels according to Section 5. As a result, a sequence is represented by a 6D histogram with 20 elements. We use a simple majority-based classification rule and assign to each sequence the most popular label among its constituent frames. We also trained multi-class SVMs on these 6D histograms. The motivation was that sequences from the weaker classes may be misclassified because the correct label was not the majority, but the distribution of votes may reveal the correct class. For example, if videos of fear typically have 6 frames labeled as fear and 8 as anger, an SVM could learn to label them correctly. As shown in Section 7, however, this assumption was not correct.

Note that at this stage, temporal coherence is not enforced, even though this would undoubtedly be beneficial for performance. The main reason for this is that the sequences are weakly labeled and non-trivial unsupervised clustering would be required to detect the onset, apex and offset, as well as frames of little or no activity.

7 Experimental Results

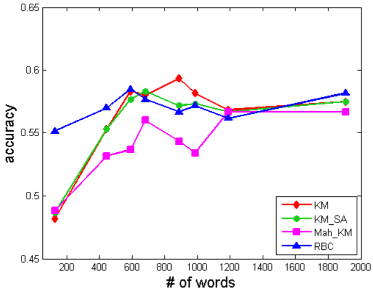
We show results on the BU-4DFE database provided to us by Binghampton University [44]. It is a 3D dynamic database containing videos of 101 subjects performing each of the six expressions. Here, we *only use single camera video sequences* which are also provided with the database. No 3D information is used in any form in the experiments described below. Moreover, no manual steps, such as initialization, are required. All experiments are performed using ten-fold cross-validation on the expressions of the first 100 subjects. The sequences are divided into training and test sets, so that all the expressions of the same person are used either for testing or training.

The images in BU-4DFE are 1392 × 1040 captured at 25 frames per second and the sequences are approximately 100 frames long. The subjects are almost centered in the images, facing the camera, with small head motions and scale differences. We downsampled all images by a factor of 2. After optical flow computation, 12 flow fields are concatenated to form one *frame*. Descriptors are 200 × 200 pixels and the spacing between adjacent grid locations is 21 pixels. Thus, each frame includes 384 descriptors.

The radius used for RBC is $R = 1.4 \cdot 10^4$. If soft assignment is enabled, each descriptor votes for its three nearest neighbors ($r = 3$) and σ is set to 10^4, so that votes to the $(r+1)^{th}$ neighbor are small. The parameter for the quadratic form distance function is $\sigma = 1.4 \cdot 10^4$. The ten nearest neighbors of a frame are used as hypotheses during geometric verification which is performed over multiple translations and five scales ranging from 80% and 120%.

Figure 2(a) shows a comparison on sequence classification accuracy among different methods for generating the vocabulary. After frame signatures have been obtained using these methods, frames are classified using the nearest neighbor classifier according to L_1 distance and the sequences are classified according to the majority of the frame labels. The methods compared in Fig. 2(a) are: standard k-means (using L_2 distance), k-means with soft assignment, k-means using Mahalanobis distance and RBC clustering using L_2 distance. The L_2 distance is superior and is used for all remaining results. The highest accuracy obtained by k-means is 59.33% compared to 58.5% by RBC.

Figure 3 shows sequence classification results using different options for classifying frames. Descriptors are assigned to words using hard assignment according to the L_2 distance and sequence labels are obtained by voting in all cases. Figure 3(a) shows results for k-means using L_1, L_2, Mahalanobis and quadratic form distances and RBC using L_1 distance. (Note that Mahalanobis distance is computed on frame signatures in this case, as in Section 5.) Figure 3(b) shows results using the SVM frame classifier and geometric verification us-

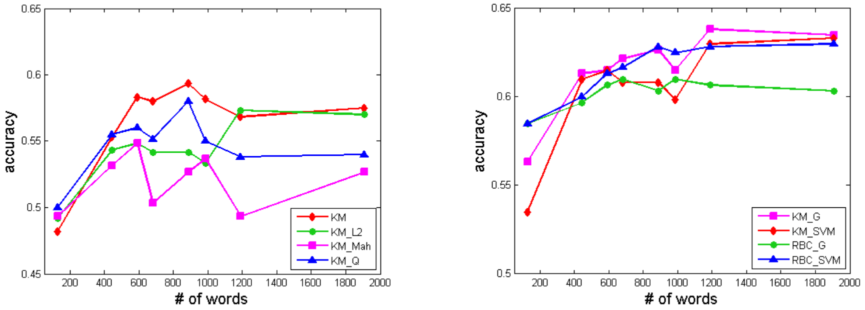


(a) Comparison of clustering methods

	Anger	Disgust	Fear	Happy	Sadness	Surprise
A	0.60	0.11	0.04	0.01	0.20	0.04
D	0.10	0.55	0.08	0.09	0.11	0.07
F	0.15	0.15	0.27	0.13	0.17	0.13
H	0.02	0.02	0.01	0.91	0.04	0.00
Sa	0.15	0.11	0.10	0.01	0.58	0.05
Su	0.00	0.01	0.03	0.00	0.04	0.92

(b) Confusion matrix for KM_G (1200 words)

Figure 2: (a) Sequence classification accuracy as a function of vocabulary size. The following clustering methods are shown: KM: k-means (using L_2), KM_SA: k-means with soft assignment, Mah_KM: k-means using Mahalanobis distance between descriptors and RBC clustering. Frames are classified using the L_1 nearest neighbor classifier. (b) confusion matrix for KM_G with 1200 words (see Fig. 3(b)). The accuracy is 63.83%.



(a) Comparison of frame classifiers (b) Comparison of advanced frame classifiers

Figure 3: Sequence classification accuracy as a function of vocabulary size using different frame classifiers and voting to classify sequences. (a) KM, KM_L2, KM_Mah, KM_Q: k-means vocabulary with the nearest neighbor frame classifier according to L_1, L_2, Mahalanobis and quadratic form distance between frames; RBC vocabulary using the L_1 nearest neighbor frame classifier. (b) KM_G, RBC_G: k-means or RBC vocabulary followed by geometric verification on the ten nearest frames according to L_1; KM_SVM, RBC_SVM: SVM frame classifier on frame signatures generated by k-means and RBC.

ing the L_1 distance to obtain ten hypotheses per frame. (SVMs only return a label per frame and therefore geometric verification is not possible.) The confusion matrix for KM_G with 1200 words is shown in Fig. 2(b). Recognition accuracy is 63.83%.

We also trained SVMs for classifying sequences according to frame labels as described in Section 6. The recognition rates we obtained, however, range from 51.7%-58.2% for RBC followed by the SVM frame classifier. The performance of this classifier is consistently lower than that of the voting-based options. (For instance, RBC followed by the SVM frame classifier and voting for a sequence label achieves accuracy in the range of 58.5%-63%.) We do not show additional results in the interest of space.

Our experiments show that the L_2 distance with hard assignment is superior to alternatives for assigning descriptors to words. Similarly, the L_1 distance is superior for classifying frames according to the nearest neighbor rule. Majority-based sequence classification consistently outperforms SVM sequence classifiers, possibly due to having only 90 sequences of each type in the training set. RBC outperforms k-means for small vocabulary sizes. The highest accuracy for k-means is obtained using L_1 to retrieve 10 candidate frames, followed by geometric verification. An SVM classifier on frame signatures computed on k-means vocabularies is somewhat worse, but considerably faster. This relationship is reversed for RBC vocabularies, for which SVMs are more accurate than geometric verification.

Our results are competitive with those of the most similar published experiment on the BU-4DFE database. Our highest accuracy is 63.83% compared to 66.95% obtained by the “dynamic 2D" method of [34], which classifies a video stream according to the motion of 83 feature points but requires manual initialization and correction. It should be emphasized that the recognition tasks are not defined equivalently in [34] and here.

8 Conclusions and Future Work

Our approach achieves satisfactory performance in expression recognition without requiring manual intervention. Among its advantages are that it can be transferred to other domains, such as activity recognition, with minor modifications and that it offers the potential of a real time implementation. This is possible since optical flow can be computed in real time on current GPUs [42] and all steps, except geometric verification, have low computational cost.

The most promising future direction is the use of a Hidden Markov Model (HMM) to enforce temporal coherence on the predicted sequence of frames. We also plan to address issues that we neglected while developing the core of our algorithm. These include the use of a face detector [19] combined with temporal stabilization of its output to estimate scale consistently and reduce the effects of head motion. A system must also be able to segment the expressions in time [7, 10] before being deployed. These additional steps do not require user interaction and thus meet our requirement for an automatic system.

Acknowledgements

This work was funded in part by the NSF CreativeIT Award IIS-0742440 and a Honda Initiation Grant, 2009. The authors are grateful to H. Quynh Dinh and Klaus Mueller (SUNY-Stony Brook) for their help in the initial stages of this effort.

References

[1] K. Anderson and P.W. McOwan. Real-time automated system for the recognition of human facial expressions. *IEEE Trans. Systems, Man and Cybernetics*, 36(1):96–105, 2006.

[2] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *International Symposium on Spatial Databases*. Springer, 1999.

[3] J. N. Bassili. Emotion recognition: The role of facial movement and the relative importance of upper and lower areas of the face. *Journal of Personality and Social Psychology*, 37:2049–2058, 1979.

[4] M.J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow-fields. *Computer Vision and Image Understanding*, 63(1): 75–104, 1996.

[5] M.J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.

[6] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[7] I. Cohen, N. Sebe, A. Garg, L.S. Chen, and T.S. Huang. Facial expression recognition from video sequences: Temporal and static modeling. *Computer Vision and Image Understanding*, 91(1-2):160–187, 2003.

[8] G. Csurka, C. Dance, J. Willamowski, L. Fan, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision*, 2004.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages I: 886–893, 2005.

[10] F. de la Torre, J. Campoy, Z. Ambadar, and J.F. Cohn. Temporal segmentation of facial behavior. In *Int. Conf. on Computer Vision*, pages 1–8, 2007.

[11] P. Dollar, V. Rabaud, G. Cottrell, and S.J. Belongie. Behavior recognition via sparse spatio-temporal features. In *PETS*, pages 65–72, 2005.

[12] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski. Classifying facial actions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(10):974–989, 1999.

[13] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Int. Conf. on Computer Vision*, pages 726–733, 2003.

[14] P. Ekman and W.V. Friesen. *The Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, 1978.

[15] I.A. Essa and A.P. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):757–763, 1997.

[16] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, 1994.

[17] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36(1):259–275, 2003.

[18] Li. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 524–531, 2005.

[19] M.J. Jones and P.A. Viola. Robust real-time face detection. In *Int. Conf. on Computer Vision*, page II: 747, 2001.

[20] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Int. Conf. on Computer Vision*, pages I: 604–610, 2005.

[21] S. Koelstra, M. Pantic, and I. Patras. A dynamic texture based approach to recognition of facial actions and their temporal models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2010.

[22] I. Kotsia and I. Pitas. Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Trans. Image Processing*, 16 (1):172–187, 2007.

[23] I. Laptev and T. Lindeberg. Space-time interest points. In *Int. Conf. on Computer Vision*, pages 432–439, 2003.

[24] J.J.J. Lien, T. Kanade, J.F. Cohn, and C.C. Li. Automated facial expression recognition based on facs action units. In *Automatic Face and Gesture Recognition*, pages 390–395, 1998.

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

[26] K. Mase. Recognition of facial expression from optical flow. *IEICE*, E74(10):3474–3483, 1991.

[27] J.C. Niebles, H.C. Wang, and F.F. Li. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79:299–318, 2008.

[28] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *European Conf. on Computer Vision*, pages IV: 490–503, 2006.

[29] M. Pantic and L.J.M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, 2000.

[30] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[31] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[32] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[33] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Int. Conf. on Computer Vision*, pages 1470–1477, 2003.

[34] Y. Sun and L.J. Yin. Facial expression recognition based on 3d dynamic range model sequences. In *European Conf. on Computer Vision*, pages II: 58–71, 2008.

[35] Y.L. Tian, T. Kanade, and J.F. Cohn. Recognizing action units for facial expression analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.

[36] Y.L. Tian, T. Kanade, and J. F. Cohn. Facial expression analysis. In S.Z. Li and A.K. Jain, editors, *Handbook of Face Recognition*. Springer-Verlag, 2005.

[37] Y. Tong, W.H. Liao, and Q.A. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(10):1683–1699, 2007.

[38] M.F. Valstar and M. Pantic. Combined support vector machines and hidden markov models for modeling facial action temporal dynamics. In *IEEE Workshop on Human Computer Interaction*, pages 118–127, 2007.

[39] J.C. van Gemert, C.J. Veenman, A.W.M. Smeulders, and J.M. Geusebroek. Visual word ambiguity. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2010.

[40] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, 2009.

[41] T.H. Wang and J.J.J. Lien. Facial expression recognition system based on rigid and non-rigid motion separation and 3d pose estimation. *Pattern Recognition*, 42(5):962–977, May 2009.

[42] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. In *British Machine Vision Conference*, 2009.

[43] Y. Yacoob and L.S. Davis. Recognizing human facial expressions from long image sequences using optical-flow. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(6):636–642, 1996.

[44] L.J. Yin, X.C. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *Automatic Face and Gesture Recognition*, 2008.

[45] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages II: 123–130, 2001.

[46] Z.H. Zeng, M. Pantic, G.I. Roisman, and T.S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009.

[47] G.Y. Zhao and M. Pietikainen. Boosted multi-resolution spatiotemporal descriptors for facial expression recognition. *Pattern Recognition Letters*, 30(12):1117–1127,, 2009.