Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2007, Article ID 87929, 10 pages
doi:10.1155/2007/87929

Research Article

Telescopic Vector Composition and Polar Accumulated Motion Residuals for Feature Extraction in Arabic Sign Language Recognition

T. Shanableh[1] and K. Assaleh[2]

[1] *Department of Computer Science, College of Engineering, American University of Sharjah, P.O. Box 26666, Sharjah, United Arab Emirates*
[2] *Department of Electrical Engineering, College of Engineering, American University of Sharjah, P.O. Box 26666, Sharjah, United Arab Emirates*

Received 9 January 2007; Revised 1 May 2007; Accepted 2 August 2007

Recommended by Thierry Pun

This work introduces two novel approaches for feature extraction applied to video-based Arabic sign language recognition, namely, motion representation through motion estimation and motion representation through motion residuals. In the former, motion estimation is used to compute the motion vectors of a video-based deaf sign or gesture. In the preprocessing stage for feature extraction, the horizontal and vertical components of such vectors are rearranged into intensity images and transformed into the frequency domain. In the second approach, motion is represented through motion residuals. The residuals are then thresholded and transformed into the frequency domain. Since in both approaches the temporal dimension of the video-based gesture needs to be preserved, hidden Markov models are used for classification tasks. Additionally, this paper proposes to project the motion information in the time domain through either telescopic motion vector composition or polar accumulated differences of motion residuals. The feature vectors are then extracted from the projected motion information. After that, model parameters can be evaluated by using simple classifiers such as Fisher's linear discriminant. The paper reports on the classification accuracy of the proposed solutions. Comparisons with existing work reveal that up to 39% of the misclassifications have been corrected.

1. INTRODUCTION

Although used in over 21 countries covering a large geographical and demographical portion of the world, Arabic sign language (ArSL) has received little attention in sign language recognition research. To date, only small number of research papers has been published on ArSL. Signer-independent recognition of Arabic sign language alphabet using polynomial networks was reported in [1]. More recently, the authors introduced the recognition of Arabic isolated gestures by computing the prediction error between successive images using either forward prediction or bidirectional prediction. The Absolute differences are transformed into the frequency domain. Feature vectors are then extracted from the frequency coefficients [2].

Related work on recognition of non-Arabic using temporal-domain feature extraction mainly rely on computationally expensive motion analysis approaches such as motion estimation. Moreover, since the temporal characteristics are preserved, classification can be done using hidden Markov models (HMMs).

For instance, in [3] the authors proposed to extract spatial and temporal image features. The temporal features are based on the thresholded difference between two successive images. The spatial features are extracted from the skin color and edge information. A logical AND is a binary operation which is known in the literature, AND is then applied to combine the temporal and spatial features. The solution is further enhanced by applying Fourier descriptors to extracted boundaries of hand shapes. Likewise, temporal analysis is enhanced, albeit at a high computational cost, by the use of motion estimation. The temporal features are then extracted from the distribution of the magnitude and phase of the motion vectors. Combining Fourier descriptors with the motion analysis using an HMM classifier resulted in a classification accuracy of 93.5%. Classification based on Fourier descriptors only resulted in 90.5% accuracy. In [4] feature extraction starts by splitting sentences with limited grammar

Table 1: Arabic sign language gestures and their english meanings.

No.	Arabic word	Meaning in English	No.	Arabic word	Meaning in English
1	صـــديق	Friend	13	يأكل	To eat
2	جار	Neighbor	14	ينام	To sleep
3	ضـــيف	Guest	15	يشـــرب	To drink
4	هدية	Gift	16	يســـــتيقظ	To wake up
5	عدو	Enemy	17	يســـمع	To listen
6	عليكـــم السـلام	Peace upon you	18	يســـكت	To stop talking
7	اهلا وسهلا	Welcome	19	يشـــم	To smell
8	شـــكرا	Thank you	20	يســـاعد	To help
9	تفضـــــل	Come in	21	امس	Yesterday
10	عيـــب	Shame	22	يذهب	To go
11	بيـــت	House	23	يأتي	To come
12	انـــا	I/me			

into video gestures. Image segmentation is then used to segment out the hands. This task is very reasonable taking into account the cap-mounted camera pointed downwards towards the hands. The features are then extracted from the following parameters: pixel-wise image differences, angle of the least inertia, the length of the associated eigenvector, and the ratio between the major axis and the minor axis of the enclosing ellipse. Again, HMMs are used for the classification. The reported classification accuracy is 91.9% for a restricted grammar. In [5] similar regions of interest (ROI) across frames are tracked. ROIs are identified through skin color and geometric cues. Motion trajectories are then extracted from the concatenation of the affine transformations associated with these regions. Time-delay neural networks are used for classification. The reported classification accuracy is 96.21% based on 40 American Sign Language gestures.

This work proposes an enhancement of ArSL recognition rates via an assortment of novel feature extraction schemes using the same dataset as the one described in [2].

This paper is organized as follows. Section 2 describes the compiled Arabic sign language dataset. Section 3 introduces the proposed feature extraction schemes. Mainly, they include motion representation through motion estimation, telescopic vector composition, motion residuals, and polar accumulated differences (ADs). Section 4 explains the experimental setup and presents the experimental results. Comparisons against existing solutions are also elaborated upon. Section 5 concludes the discussion.

2. DATASET DESCRIPTION

As the authors reported in [2], Arabic Sign Language does not yet have a standard database that can be purchased or publicly accessed. Therefore, we decided to collect our own ArSL database. We have collaborated with (Sharjah City for Humanitarian Services (SCHS) Sharjah, UAE) [6], and arranged for collecting ArSL data. In this first phase of our data collection, we have collected a database of 23 Arabic gestured words/phrases from 3 different signers. The list of words is shown in Table 1.

Each of the three signers was asked to repeat each gesture 50 times over three different sessions resulting in a total of 150 repetitions of the 23 gestures which correspond to 3450 video segments. The signer was videotaped using an analog camcorder without imposing any restriction on clothing or image background. The video segments of each session were digitized and partitioned into short sequences representing each gesture individually. Note that the proposed feature extraction schemes do not impose any restrictions on the selection of the frame sampling rate.

3. FEATURE EXTRACTION SCHEMES

Two solutions for feature extraction schemes are proposed: motion estimation and motion residuals. Both solutions are discussed with respect to two different extraction scenarios: time-dependent and time-independent feature extraction schemes.

3.1. 1 Motion estimation

In this section the motion of video-based gestures is represented by their motion vectors. Block-based motion estimation between successive images is used to generate such vectors. The input images are divided into nonoverlapping blocks. For each block, the motion estimation process will search through the previous image for the "best match" area within a given search range. The displacement between the current block and its best match area in the previous image is represented by a motion vector.

More formally, let C denote a block in the current image with $b \times b$ pixels at coordinates (m, n). Assuming that the maximum motion displacement is w pixel per frame then the task of the motion estimation process is to find best match area P within the $(b+2w)(b+2w)$ distinct overlapping $b \times b$ blocks of the previous image. An area in the previous image that minimizes a certain distortion measure is selected as the best match area. A common distortion measure is the mean Absolute difference given by

$$M(\Delta x, \Delta y) = \frac{1}{b^2} \sum_{m=1}^{b} \sum_{n=1}^{b} |C_{m,n} - P_{m+\Delta x, n+\Delta y}|, \quad -w \le \Delta x, \Delta y \le w, \tag{1}$$

where $\Delta x, \Delta y$ refer to the spatial displacement between the pixel coordinates of C and the matching area in the previous image. Other distortion measures can be used such as mean-squared error, cross correlation functions, and so forth.

Clearly the motion estimation process is computationally expensive. Many suboptimal algorithms are reported to speedup the computation at the cost of increasing the entropy of the prediction error. In such algorithms, a subset of the $(b+2w)(b+2w)$ overlapping locations are searched, thus no guarantees of finding the best matched area.

An example of fast motion estimation algorithms is 2D logarithmic search with a maximum number of search positions of $2 + 7\log_2 w$ [7]. Other examples are the cross-search algorithm maximum number of search positions of $3 + 2w$ [8]. More recently, a fast block-matching algorithm

called center-prediction and early-termination-based motion search algorithm (CPETS) was proposed [9]. The algorithm reduces 95.67% of encoding time in average compared with the full-search approach yet a negligible loss in peak signal-noise ratio (PSNR) is reported. Further details on motion estimation can be found in [10] and references within.

3.1.1. 1 Motion vector feature extraction schemes

Feature extraction follows the estimation of motion vectors using one of the following approaches: time-dependent and time-independent feature extraction schemes. In the former extraction approach, the temporal dimension of successive images is preserved, while in the latter, the motion vectors of successive images are accumulated into a representative and concise set of feature vectors.

(a) Time-dependent feature extraction

In this approach, the motion vectors of each two successive images are estimated and split into their x and y components. Each motion vector component is then rearranged into an intensity image. The dimensions of such an image are proportional to the motion estimation block size and width and height of the gesture images. In this work, we experiment with a block size of 8×8 and the input images have a dimension of 360×288 pixels. The x and y intensity images are then concatenated into one image f having dimensions $m \times n$ that visually describes the location and intensity of motion between two successive images.

The concatenated image is then transformed into the frequency domain using 2D discrete cosine transformation (DCT) given by

$$F(u,v) = \frac{2}{\sqrt{MN}} C(u)C(v) \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) \times \cos\left(\frac{\pi u}{2M} \cdot (2i+1)\right) \cos\left(\frac{\pi v}{2N} \cdot (2j+1)\right), \quad (2)$$

where $N \times M$ are the dimensions of the input image "f" and $F(u,v)$ is the DCT coefficient at row u and column v of the DCT matrix. $C(u)$ is a normalization factor equal to $1/\sqrt{2}$ for $u = 0$ and 1 otherwise.

An attractive property of the DCT transformation is its energy compaction. Thus, the input concatenated image f having dimensions $m \times n$ can be represented by zonal coding of the DCT coefficients via a zigzag scanned path into an n-dimensional vector [11]. This dimensionality is empirically determined as illustrated in the experimental results section.

The block diagram of the proposed feature extraction approach is shown in Figure 1.

Note that the above feature extraction is repeated for each pair of consecutive images, thus the temporal dimension of the gesture images is preserved. Figure 2 shows an example of applying this feature extraction scheme to gesture 3. The figure shows the vertical concatenation of the MVx and MVy intensity images as a result of the block-based motion estimation processes.

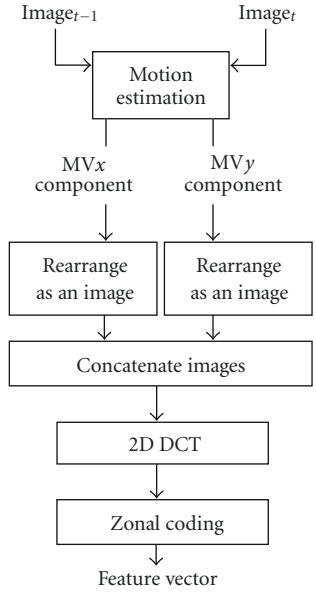


Figure 1: Block diagram of time-dependent feature extraction from motion vectors.

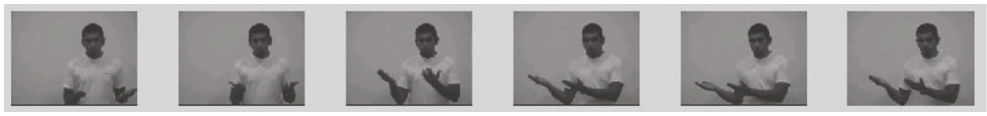
In the experimental results section, hidden Markov models will be used to train and classify such time-dependent feature vectors.

(b) Time-independent feature extraction

On the other hand, the motion vectors of a gesture video can be accumulated into one image prior to feature extraction. This section proposes to compute the vectorial sum of coinciding motion vectors across the motion vector intensity images. We will refer to this block-wise summation of motion vectors as telescopic vector composition (TVC). Note that TVC has been successfully employed in the context of temporal subsampling in digital video transcoding as reported by the author in [12]. The block-wise summed motion vectors are then split into x and y components and rearranged into separate intensity images. Again, the resultant intensity images are concatenated, DCT transformed, and zonal coded. This proposed feature extraction scheme is illustrated in Figure 3.

In this case, the whole video-based gesture is represented by one feature vector only. Figure 4 shows an example of applying this feature extraction scheme to gesture 3 (shown in Figure 2(a)). The figure shows the vertical concatenation of the telescopic vector composition of the MVx and MVy intensity images as a result of the block-based motion estimation processes.

In the experimental results section, simple pattern recognition techniques such as K-nearest neighbor (KNN) and linear classifier will be used to train and classify such time-independent feature vectors.



(a) Input image sequence, gesture 3 (Guest)

(b) Resultant MVx and MVy intensity images

Figure 2: An example of time-dependent feature extraction from motion vectors.

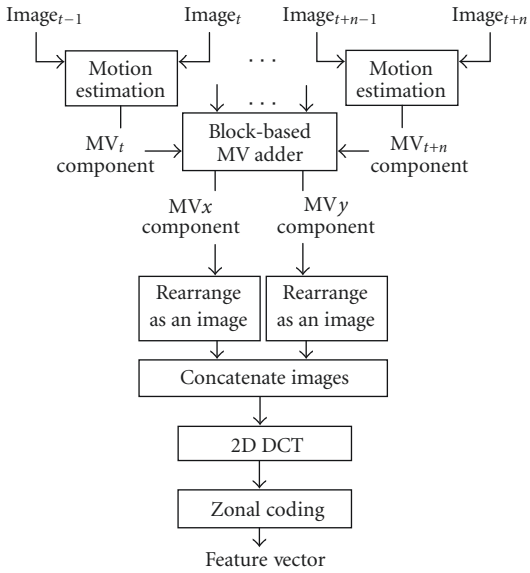


Figure 3: Block diagram of time-independent feature extraction from motion vectors.

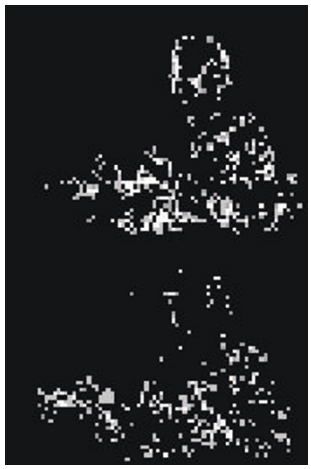


Figure 4: An example of time-independent feature extraction from motion vectors.

3.2. 2 Motion residuals

This section proposes to track the motion by examining the intensity of the motion residuals or prediction error. This is computed from the difference between two successive images without the need for the computationally expensive motion estimation process.

The image difference between two successive images is computed and thresholded. The threshold can be the mean of moving pixels (i.e., mean of nonzero pixel differences), one standard deviation above the mean or zero. Clearly there is a tradeoff between the threshold value and the accurate representation of the motion. Setting it to zero results in treating all pixel differences as motion; setting it to a high value results in discarding some motion information, and so forth. Following [2], the value of the threshold was determined empirically and set to the mean intensity of moving pixels.

Similar to the previous section, we propose two approaches for obtaining the feature vectors using prediction errors, namely: time-dependent and time-independent feature extraction schemes.

3.2.1. Time-dependent feature extraction

In this approach, the image differences between each pair of successive images are computed. Only pixel differences above the threshold are retained and the rest are set to zero. The resultant prediction error is then transformed into the frequency domain using DCT transformation. The feature vectors are then generated by means of zonal coding at a given cutoff. Since this process is repeated for each pair of successive images, then the resultant feature vectors retain the temporal dimension of the video-based gesture.

On the other hand, binary thresholding can be used for a more abstract representation of the prediction error. In this case, the pixel differences above the threshold are set to unity and the rest are set to zero. The resultant prediction error is then transformed to the frequency domain using 2D Walsh-Hadamard transformation (WHT) rather than DCT. The former transformation is known for its simplicity

and suitability for binary images. The WHT has binary basis functions thus has a higher correlation with the binary-thresholded prediction error. The smoothly varying cosine terms of the DCT basis functions on the other hand are not a good choice in this case. The WHT has the following kernel:

$$h(x, y, u, v) = \frac{1}{2^m}(-1)^{\sum_{i=0}^{m=1}\lfloor b_i(x)p_i(u)+b_i(y)p_i(v)\rfloor}, \quad (3)$$

where m is the number of bits needed to represent a pixel value, $b_i(x)$ is the ith binary bit from right to left, and $p_i(u) = b_{m-i}(u) + b_{m-i-1}(u)$. All sums are performed in modulo 2 arithmetic [13].

3.2.2. Time-independent feature extraction

This section introduces the use of polar accumulated difference (ADs) in a first tier of feature extraction. The section also reviews two solutions for a second tier of feature extraction. Lastly, we propose a two tier feature extraction scheme that combines the aforementioned solutions.

(a) First tier of feature extraction

During the first tier of feature extraction, the motion information is extracted from the temporal domain of the input image sequence through successive image differencing. Let $I_{g,i}^{(j)}$ denote image index j of the ith repetition of a gesture at index g. The image formed from the ADs can be computed by

$$\mathrm{AD}_{g,j} = \sum_{j=1}^{n-1} \partial_j\left(\left| I_{g,j}^{(j)} - I_{g,i}^{(j-1)} \right| \right), \quad (4)$$

where n is the total number of images in the ith repetition of a gesture at index g, and ∂_j is a binary threshold function of the jth frame.

While Absolute ADs detect the motion that an object undergoes regardless of its direction, polar ADs, on the other hand, preserve the directionality of that motion. ADs can be categorized into three types: Absolute ($|\mathrm{AD}|$), Positive (AD_+), and Negative (AD_-). These can be defined as follows:

$$\begin{aligned}
&|\mathrm{AD}|(x, y)\\
&= \begin{cases} \mathrm{AD} + 1 & \text{if } \left| f(x, y, t_k) - f(x, y, t_{k-1}) \right| \geq \mathrm{Th}_{(k,k-1)},\\ \mathrm{AD}, & \text{otherwise,} \end{cases}\\
&\mathrm{AD}_+(x, y)\\
&= \begin{cases} \mathrm{AD}_+ + 1 & \text{if } \left(f(x, y, t_k) - f(x, y, t_{k-1}) \right) \geq \mathrm{Th}_{(k,k-1)},\\ \mathrm{AD}_+, & \text{otherwise,} \end{cases}\\
&\mathrm{AD}_-(x, y)\\
&= \begin{cases} \mathrm{AD}_- + 1 & \text{if } \left(f(x, y, t_k) - f(x, y, t_{k-1}) \right) \leq \mathrm{Th}_{(k,k-1)},\\ \mathrm{AD}_-, & \text{otherwise,} \end{cases}
\end{aligned} \quad (5)$$

where (x, y) are the pixel coordinates of the ADs image. The Absolute ADs approach was proposed for sign language recognition by the authors in [2]. Here, we extend this work by experimenting with polar ADs (i.e., AD_+ and AD_-). Note that the latter ADs have been successfully used in the recognition of Arabic handwritten alphabets as reported in [14].

(a) Polar ADs images (b) Absolute ADs image

Figure 5: Examples of ADs images.

Figure 5 shows examples of applying the above ADs approaches to gesture 3 (shown in Figure 2(a)).

(b) Second tier of feature extraction

Once the ADs images are computed, a second tier of feature extraction is applied. Two different approaches are employed: (a) 2D discrete cosine transformation (DCT) followed by zonal coding, and (b) Radon transformation followed by lowpass filtering. Thus, in addition to 2D transformations, we also experiment with image projections through Radon transformation. The pixel intensities of the ADs are projected at a given angle θ using the following equation:

$$R_\theta(x) = \int_{-\infty}^{+\infty} f(x' \cos\theta - y' \sin\theta,\ x' \sin\theta + y' \cos\theta) dy', \quad (6)$$

where f is the input image, and the line integral is parallel to the y' axis, where x' and y' are given by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (7)$$

The resultant projection is then coarsely represented by transforming it into the frequency domain using a 1D DCT followed by an ideal lowpass filter.

(c) Two-tier feature extraction

The aforementioned first and second tiers of feature extraction schemes are merged using either *polar accumulated differences or vectorized polar accumulated differences.*

In the *polar accumulated differences* approach, the Positive and Negative ADs images are concatenated into one image prior to the second tier of feature extraction as shown in Figure 6. The second tier feature extraction follows the methodology used in [2], where either 2D DCT or Radon

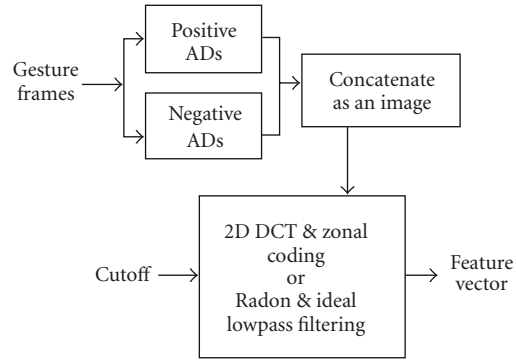


Figure 6: Polar accumulated differences.

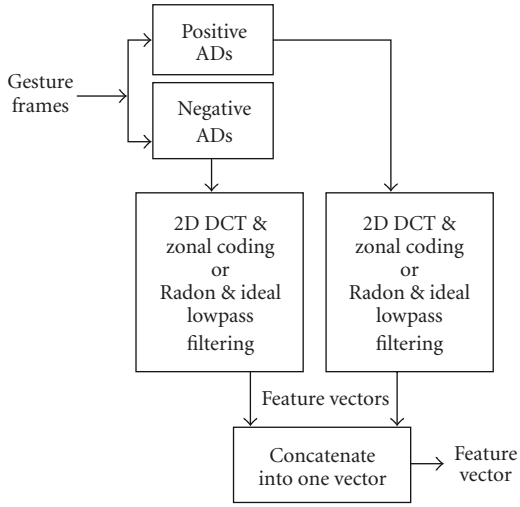


Figure 7: Vectorized accumulated differences with 2D transformation.

transformation is applied to the image formed by the ADs. In case of 2D DCT, the transformed image is zonal coded with different cutoff values. On the other hand, if Radon transformation is applied, then the projected image is 1D DCT transformed followed by ideal lowpass filtering.

On the other hand, in *vectorized polar accumulated differences* approach, the Positive and Negative ADs are computed. A second tier of feature extraction is then applied to each of the ADs images. The concatenation is thereafter applied to the resultant feature vectors. This approach is illustrated in Figure 7.

4. EXPERIMENTAL RESULTS

This section presents the experimental results for the various feature extraction schemes described above. Training is done in an offline mode, and model parameters are uploaded to the recognition stage. Offline training mode is usually done when the training data is large (due to large number of classes or excessive variability within each class) or the recognition is in user-independent mode. The gesture database is divided into training and testing sets. As we mentioned in Section 2, the database is composed of video sequences corresponding

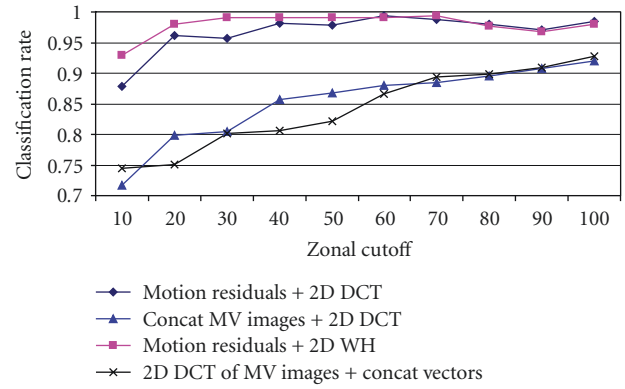


Figure 8: Classification results for the proposed motion estimation versus motion residuals approaches. Hidden Markov models are used.

to 23 different gestures (classes) each of which is repeated 50 times by 3 different signers. In this classification mode, we have used 70% of the data for training and the remaining 30% for testing. The training and testing sets contain mixed samples of all signers. The classification results in the figures to follow show the average classification rate of the 23 gestures.

Additionally, in the following experiments, the motion estimation search range is set to 16×16 pixels and the blocks size is 4×4 pixels. Such parameters are commonly used in digital video compression.

4.1. 1 HMM-based classification

This section classifies the sign language data using hidden Markov models (HMMs). Throughout the experiments, we have used the left to right HMM architecture where a state can only transit to its immediate right neighbor or stay in the same state. The training method applied is the Baum-Welch algorithm and the number of states for the training set is empirically determined to be 2, 3, or 4 according to the complexity of the gesture. Each gesture was visually analyzed to determine the number of the distinct movements that a signer makes while performing that gesture. The number of states was estimated accordingly. As for the number of Gaussian mixtures for the training set, we have experimented with 2, 3, and 4 Gaussian mixtures and obtained slight variations in the recognition rates over the 23 gestures of the validation set. However, we found that 3 mixtures resulted in a slight improvement in the overall recognition rates. Further information on HMMs can be found in [15].

In this approach, the temporal dimension of the input image sequence is preserved. As pointed out previously, the feature extraction step preserves the Absolute motion residuals between successive images without accumulating them into one image. The Absolute motion residuals are then thresholded, binarized, transformed into the frequency domain, and converted into a sequence of feature vectors using zonal coding.

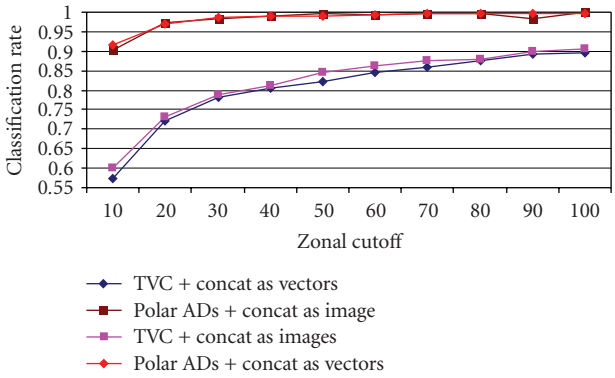


Figure 9: Classification results for the proposed TVC versus polar ADs. 1NN is used for classification.

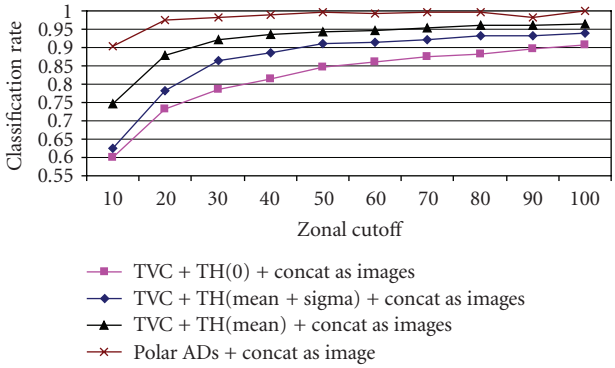


Figure 10: Classification results for the thresholded TVC versus polar ADs. 1NN is used for classification.

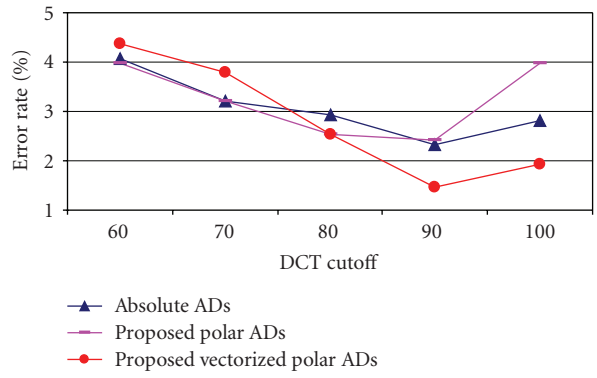


Figure 11: Fisher's linear discrimination with 2D transformation and zonal coding.

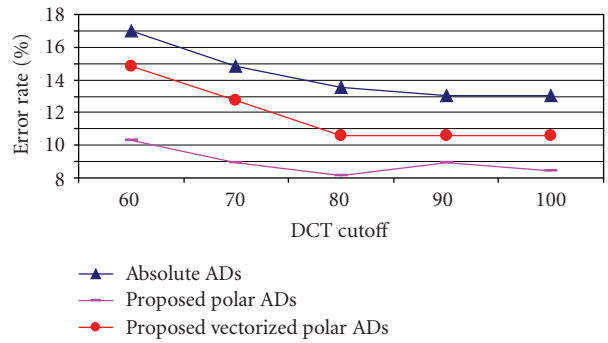


Figure 12: Fisher's linear discrimination with vertical Radon transformation and ideal lowpass filtering.

In Figure 8, a comparison of the classification results of the motion estimation and the motion residual approaches is presented. In the figure, "Concat MV images + 2D DCT" and "2D DCT of MV images + concat vectors" refer to the feature extraction schemes of Section 3.1.1(a). In the former, the intensity images of the MVs are concatenated and transformed using 2D DCT. While in the latter, each MV intensity image is transformed separately. The zonal coefficients of each transformed image are then concatenated into one feature vector.

Despite its simplicity, the latter approach exhibits higher classification results at all DCT zonal cutoffs. Due to its distortion measure, there are no guarantees that the motion estimation approach will capture the true motion in an image sequence. Rather, the motion vectors will blindly point to the location that minimizes the mean Absolute differences or mean-squared differences. Additionally, the maximum motion displacement might exceed the w pixels per frame as illustrated in (1) hence the computed motion vector might not capture the true motion of the sequence.

The figure also shows that concatenating the images of the motion vector components prior to zonal coding outperforms concatenating the feature vectors. Lastly, the figure shows that applying the 2D WHT to the binarized and thresholded motion residuals outperforms the 2D DCT approach. As mentioned previously, the binary basis functions of the 2D WHT, as opposed to the sinusoidal basis functions of the DCT, correlate well with the binarized motion residues hence the more accurate classification rate.

4.2. 2 KNN-based classification

This section presents the experimental results for the proposed time-projections techniques. Here, the whole video sequence of motion vector images or motion residuals is projected into one image which is then 2D DCT transformed and zonal coded. As such, HMMs are no longer needed or even applicable to model estimation and classification rather, simple classifiers like KNN can be used.

Figure 9 compares between the polar ADs and the telescopic vector composition (TVC) techniques. It is shown that the polar ADs of the motion residuals outperforms the TVC approach. The KNN and HMM classification results are quite similar, thus reinforcing the discussion in Figure 8 regarding the differences between the motion residuals and motion estimation solutions. When using KNN classifiers, it is worth mentioning that the projection of the temporal dimension via the polar accumulated differences and the telescopic vector composition schemes yields comparable recognition results to those obtained by HMMs.

Further examination of the motion estimation approach reveals the sensitivity of such a process. Clearly, the block

G#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	42		1				2																
2		44			1																		
3			43																				
4			4	32					5						3					1			
5					42					1													
6			2			42									1								
7			1		1	1	38			2					1					1			
8			2		1	3	1	37													1		
9				1					44														
10					1					44													
11			1		2		3				39												
12			2					1				30	2		3	7							
13													40		5								
14					4									40									
15															45								
16															3	41					1		
17					1										3		41						
18			1										3		3			27	11				
19													2						43				
20			1						1											42			
21			1		1																43		
22															3							42	
23					1					1					4								39

(a) Vectorized Radon transformation of polar accumulated difference

G#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	45																						
2		44				1																	
3			43																				
4				45																			
5					43																		
6						45																	
7							41									4							
8								41								4							
9									44							1							
10										45													
11											45												
12												45											
13												1	42						2				
14														44									
15															45								
16													1			44							
17																	45						
18													1					43	1				
19													1						44				
20																				44			
21																2					43		
22																						44	1
23																							45

(b) Vectorized 2D DCT2 of polar accumulated difference

Figure 13: Confusion matrices for the vecotrized 2D DCT and Radon transformation schemes of Figures 11 and 12.

matching approach minimizes a distortion criterion for all the blocks in a given image. Thus motion vectors might be calculated for blocks that do not represent the motion of a gesture. Such motion vectors can belong to the body, rather than the hands of the signer or can even belong to the background in cases of luminance changes for instance. However, it is observed that such motion vectors have a relatively small magnitude, therefore, can be detected and thresholded for better representation of the motion.

This idea is implemented and its results are shown in Figure 10. We experiment with 3 thresholds: the mean value of nonzero motion vector components, one standard deviation above the mean, and no thresholding. The figure shows that setting the threshold to the mean generates the best classification results. An average increase of more than 10% in classification accuracy is reported. Clearly setting the threshold to one standard deviation above the mean generates lower classification results. This is so because actual motion

information, which is accumulated into one intensity image, is underrepresented by such a rather high threshold. The figure also shows that the thresholded TVC solution approaches the classification results of the polar ADs at high zonal cutoffs.

4.3. 3 Linear discrimination

In the following classification experiments, Fisher's linear discrimination is employed. The proposed polar ADs approaches are compared against the work reported in [2] (thereafter referred to as "Absolute ADs"). For comparison reasons in the following experiments, we plot the classification error rates and elaborate upon the reduction in misclassifications brought by the proposed feature extraction schemes.

In Figure 11, 2D transformations and zonal coding are used for the second tier of feature extraction as explained in Section 3.2.2. The proposed vectorized ADs of Figure 7 outperform the Absolute ADs. The figure also shows that results of concatenating the Positive and Negative ADs images prior to the second tier of feature extraction (as proposed in Figure 6) is comparable to the results of Absolute ADs up to a DCT cutoff of 90 coefficients. In all cases, the figure shows that a cutoff of 90 coefficients minimizes the classification error rate.

On the other hand, the classification gain of the proposed solution is more pronounced with Radon transformation and ideal low pass filtering. Figure 12 shows that both approaches of concatenating ADs images and concatenating the feature vectors outperform the Absolute ADs for all values of DCT cutoff. For instance, at a cutoff of 60, the misclassifications is reduced by 39.4%. The figure also shows that the proposed polar ADs approach maintains stable linear separability even at low DCT cutoffs.

Comparing the classification results of Figures 11 and 12, it is clear that the feature extraction schemes based on 2D DCT are more accurate than the Radon transformation schemes. Recall that in the latter schemes the ADs images are projected at a given angle. Thus ADs images with similar pixel intensities alongside the projection angle will have similar Radon transform coefficients. Such ADs images might or might not belong to the same gesture hence lower classification results. This observation is further clarified in Figure 13 which displays the confusion matrices for both the vectorized 2D DCT approach of Figure 11 and the vectorized Radon transform of Figure 12. For instance, part a of the figure shows that gesture 18 (which translate to "*To stop talking*") is mainly confused with gesture 19 (which translate to "*To smell*"). Figure 13(b) shows that such confusion is alleviated with the 2D DCT approach. Other examples are also evident in gestures 12 and 4.

5. CONCLUSION

In this paper we have proposed a number of feature extraction schemes for Arabic sign language recognition. The proposed schemes are categorized into time-dependent and time-independent feature extractions. In the former, the temporal dimension of the video-based gesture is retained. The gesture's motion is extracted by either motion estimation or motion residuals. Hidden Markov models are then used for model estimation and classification. It was shown that feature extraction through motion residuals is superior to the motion estimation scheme in terms of reducing the computational complexity and achieving higher sign language classification rates.

On the other hand, we have shown that the temporal dimension of the input video gesture can be removed by accumulating either the motion vectors or motion residuals into one or two intensity images. This time-independent approach to feature extraction facilitates the use of simple classifiers such as KNN and linear classifiers instead of HMMs. Lastly, it was shown that preserving the directionality of the motion via the use of polar ADs outperformed the existing solution based on Absolute ADs. It was shown that up to 39% of the misclassifications caused by the use of Absolute ADs have been corrected.

ACKNOWLEDGMENTS

The authors acknowledge Mr. Salah Odeh of the Sharjah City for Humanitarian Services (SCHS) and Mr. W. Zouabi and F. Siam from the American University of Sharjah (AUS) for their invaluable assistance in the facilitation of the ArSL data collection. The authors would also like to thank (AUS) for a research grant in support of this work (2006-2007).

REFERENCES

[1] K. Assaleh and M. Al-Rousan, "Recognition of Arabic sign language alphabet using polynomial classifiers," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 13, pp. 2136–2145, 2005.

[2] T. Shanableh, K. Assaleh, and M. Al-Rousan, "Spatio-temporal feature-extraction techniques for isolated gesture recognition in Arabic sign language," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 37, no. 3, pp. 641–650, 2007.

[3] F.-S. Chen, C.-M. Fu, and C.-L. Huang, "Hand gesture recognition using a real-time tracking method and hidden Markov models," *Image and Vision Computing*, vol. 21, no. 8, pp. 745–758, 2003.

[4] M.-H. Yang, N. Ahuja, and M. Tabb, "Extraction of 2D motion trajectories and its application to hand gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1061–1074, 2002.

[5] T. Starner, J. Weaver, and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, 1998.

[6] Sharjah City for Humanitarian Services (SCHS), http://www.sharjah-welcome.com/schs/about/.

[7] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Transactions on Communications*, vol. 29, no. 12, pp. 1799–1808, 1981.

[8] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Transactions on Communications*, vol. 38, no. 7, pp. 950–953, 1990.

[9] Y. L. Xi, C. H.-Y. Haoa, Y. Y. Fana, and H. Q. Hua, "A fast block-matching algorithm based on adaptive search area and its VLSI architecture for H.264/AVC," *Signal Processing: Image Communication*, vol. 21, no. 8, pp. 626–646, 2006.

[10] M. Ghanbari, *Video Coding: An Introduction to Standard Codecs*, IEE Telecommunication Series 42, Institution Electrical Engineers, London, UK, 1999.

[11] W.-H. Chen and W. Pratt, "Sense adaptive coder," *IEEE Transactions on Communications*, vol. 32, no. 3, pp. 225–232, 1984.

[12] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Transactions on Multimedia*, vol. 2, no. 2, pp. 101–110, 2000.

[13] R. Gonzalez and R. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.

[14] K. Assaleh, T. Shanableh, and H. Hajjaj, "Online video-based handwritten arabic alphabet recognition," in *The 3rd AUS International Symposium on Mechatronics (AUS-ISM '06)*, Sharjah, UAE, April 2006.

[15] L. R. Rabiner, "Tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.