VIDEO ACTION RECOGNITION USING RESIDUAL VECTOR QUANTIZATION AND HIDDEN MARKOV MODELS

Salman Aslam, Christopher Barnes, Aaron Bobick

Georgia Institute of Technology

ABSTRACT

In this paper, we discuss usage of a multi-stage Residual Vector Quantization (RVQ) strategy for human action recognition. To the best of our knowledge, this is the first reported application of multi-stage RVQ to any form of video processing. The RVQ is used to generate a single byte descriptor per image. The evolution of these descriptors is analyzed using a bank of HMM models, one per action to be classified. Correct classification is implemented by maximizing the posterior class conditional density. We report comparable classification results to state of the art methods, i.e. above 90%, while using less than half the available training set.

Index Terms— Action recognition, residual vector quantization, σ-tree classifier, pattern recognition, machine learning, source coding, compression.

1. INTRODUCTION

Understanding human action in video is a form of high level vision processing that has received considerable interest over the past few years. Being able to automatically detect human action can have several advantages, in areas such as surveillance, child and elderly care, crowd analysis, law enforcement and many other areas. Two basic approaches have been adopted in this regard. The first is related to understanding human motion trajectories [1] [2] [3] while the other is related to human pose estimation [4]. In this paper, we focus on the former approach.

Our approach is two fold. In the first step, we use Residual Vector Quantization to represent each image in the video sequence as a single byte. A Hidden Markov Model is then used to track the evolution of this descriptor over time. [5] reports the first usage of RVQ for image analysis. We extend that work to video analysis. Usage of HMMs for inference in video analytics has been used by many authors [6] [7].

We first introduce some theory and notation regarding RVQ and HMMs.

1.1. σ-tree Classifier and Residual Vector Quantization

In this paper, we use the Image Driven Data Mining approach (IDDM)introduced in [5]. This approach is based on the σ-tree for data warehouse similarity searching within the pixel-block space of feature classes, and for extracting labels from knowledge base collections or ”aggregates”. These collections, in this paper are foreground blobs corresponding to tracked targets. A key difference is that the data warehouse is built on-line, and is updated every few frames as new observations of the targets under interest arrive. Another difference is that temporal correlations are used for unsupervised learning of target labels. To better understand the σ-tree classifier, we relate it with five well-known areas of information theory, pattern recognition and machine learning.

First, the σ-tree classifier is a type of variable rate vector quantizer [8]. Second, Truncel *et al* explore the relationship between rate-distortion theory and content-based retrieval in high dimensional databases [9]. Third, the σ-tree design problem parallels the joint codebook design problem addressed in the source coding literature for multiple stage vector quantization [10]. Fourth, the σ-tree classifier can be compared to dimensionality reduction methods such as Principal Components Analysis (PCA). PCA seeks to reorient the basis vectors in $\mathbb{R}^k$ and achieves compression by ignoring projected data components with least variances. Even though a σ-tree classifier can be used to achieve lower dimensionality, it primarily seeks to partition $\mathbb{R}^k$ like other well known classifiers such as neural networks and support vector machines [5]. Further, note that neural networks partition the decision space with hyperplanes or hypersurfaces, depending on whether or not hidden layers are used. Support vector machines also do partition the decision space, but with hyperplanes in a higher dimension space that separate the data with maximum margin. The σ-tree classifier tesselates the decision space $\mathbb{R}^k$ with k Voronoi cells, the geometric dual of delaunay triangulation in $\mathbb{R}^2$. The fifth and final comparison is with the well-known k-means algorithm, or the Linde Buzo Gray (LBG) algorithm as it is more commonly referred to in the source coding literature. It is also called the Generalize Lloyd Algorithm (GLA). Most vector quantization methods use this algorithm to create cluster centroids [11]. Indeed, the σ-tree classifier centroids can also be designed using this algorithm. For a concrete and simple example of a σ-tree, see [5].

The mechanism that allows a σ-tree to be built is multi-stage Vector Quantization (VQ), also called Residual Vector Quantization (RVQ) or Direct Sum Successive Approximation (DSSA) Vector Quantization. These three terms will be used interchangeably in this paper. Juang and Gray first proposed the RVQ structure [12] and suggested that RVQ stages be designed by sequential application of GLA. The main shortcoming of this approach is that each stage codebook is generated while considering only the error due to the previous stages (the *causal error*); the error due to subsequent stages (the *anticausal error*) is ignored. A joint design approach on the other hand takes into account both the causal and anticausal errors to reduce the *total error*. Various techniques have been proposed for joint optimization. We now introduce notation and define ”joint optimality.”

The pth stage of a P-stage RVQ, with N codevectors per stage, is a k-dimensional VQ defined by the mapping

$$Q_p : \mathbb{R}^k \mapsto C_p \tag{1}$$

Here, $\mathbb{R}^k$ is the k-dimensional input space,

$$C_p = \{y_p(0), y_p(1), \ldots y_p(N_p - 1)\} \tag{2}$$

is the pth-stage codebook and $y_p(i_p) \in \mathbb{R}^k$ is a pth-stage *codevector*. Residual quantizer stage mappings Q_p are collectively equivalent to a single mapping,

$$Q : \mathbb{R}^k \mapsto C \tag{3}$$

Here, C is the *direct sum codebook* and is the direct sum of the stage codebooks $C = C_1 + C_2 + \ldots C_p$. In practice, the stage mappings $Q_p(.)$ are realized as a composition of a stage encoder mapping $E_p = \mathbb{R}^k \mapsto I_p$, and a stage decoder mapping $D_p : I_p \mapsto C_p$, that is $Q_p(x_1) = D_p[E_p(x_1)]$, where x_1 is a realization of X_1, which is the random source output.

An RVQ is said to be jointly optimal if a local or global minimum value of the average distortion $d(E, D) = E[m(X_1, D(E(X_1)))]$ is achieved, where $m(., .)$ is a distortion metric, and E[.] is the expectation operator. This shows that an RVQ is a product code VQ with a direct sum codebook structure [13]. Also notice that computation (search) and memory (storage) requirements are proportional to kNP. So, the cost functional grows linearly with classifier dimensionality k, branch multiplicity N, and the number of tree levels P. However, the number of indexing options is N^P. This computational advantage makes it possible to use RVQ in real time tracking, even though the data dimensionality is very high.

1.2. Hidden Markov Model

A comprehensive overview of HMMs can be found in [14]. The goal is to analyze a sequence of random variables. We relax the constraint of independence at the cost of sacrificing the factorization of the joint distribution going from a product of marginal distributions to a product of conditional distributions. An HMM (Figure 1)is characterized by the following:

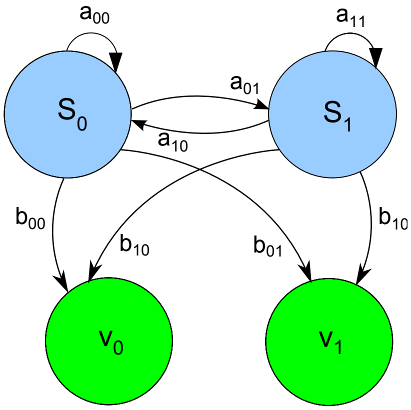

Fig. 1. Hidden Markov Model.

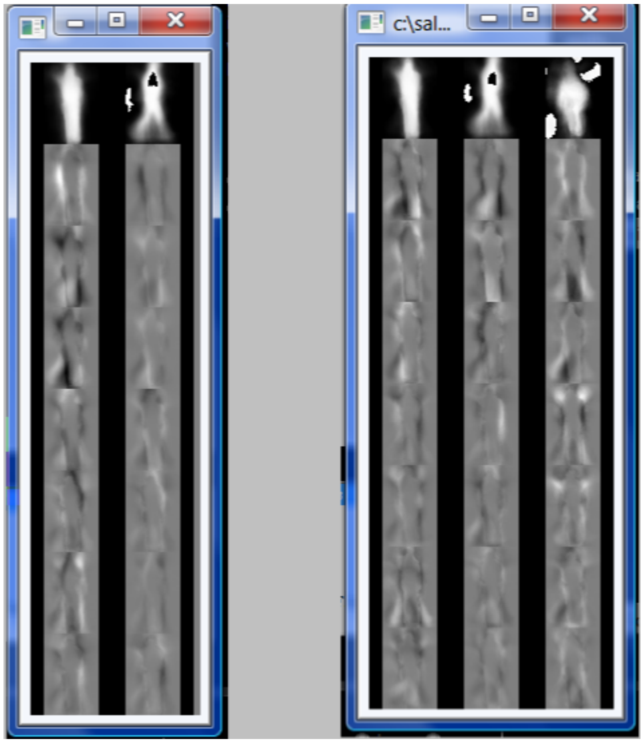

Fig. 2. RVQ codebooks. With 2 templates per stage, and 8 stages, a total of 256 leaf nodes are possible. With 3 templates per stage, a total of 6561 leaf nodes are possible.

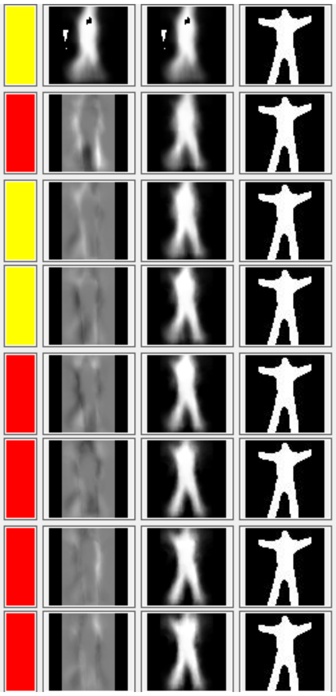

Fig. 3. Image reconstruction using direct sum successive approximation RVQ.

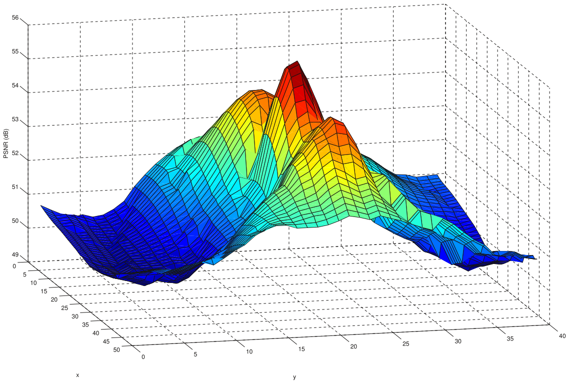

Fig. 4. PSNR values of reconstruction computed at every point of the test image.

1. **Number of states in the model, N**: We denote the individual states as $S = S_1, S_2, ...S_N$ and the state at time t as q_t.

2. **Number of distinct observation symbols per state, M**: This is the discrete alphabet size. We denote the individual symbols as $V = v_1, v_2, ...v_M$.

3. **State transition probability distribution**. For the special case where every state can reach every other state, $a_ij > 0$.

$$a_{ij} = p(q_{t+1} = S_j | q_t = S_i) \quad 1 \leq i, j \leq N \tag{4}$$

4. **Observation symbol probability distribution**.

$$b_{jk} = p(v_k | q_t = S_j) \quad 1 \leq j \leq N, 1 \leq k \leq M \tag{5}$$

5. **Initial distribution**.

$$\pi_i = P(q_1 = S_i) \quad 1 \leq i \leq N \tag{6}$$

An HMM, defined by N, M, A, B, π, can be used to generate a sequence of observation symbols, $O_1, O_2, ...O_T$, and as a model for how a given observation was generated by an appropriate HMM. For convenience, we use

$$\lambda = (A, B, \pi) \tag{7}$$

to indicate the complete parameter set of the model. Each observation O_t is one of the symbols from V and T is the number of observations in the sequence.

In HMMs, there are 3 basic problems: (a) Given an observation sequence, what is the probability of a particular model generating that sequence. This is solved efficiently using the Forward Backward Algorithm. (b) Training the model. This is done using the EM Algorithm. (c) Finding the most likely sequence of state evolution. This is solved using the Viterbi algorithm. In our case, we are interested in the first application, i.e. given a sequence of observing human actions, what is the most probable label for that action. Mathematically, what is the likelihood of observing the given sequence,

$$\begin{aligned} p(\mathbf{O}_n | \mathbf{Q}_n) &= \frac{p(\mathbf{O}_n, \mathbf{Q}_n)}{p(\mathbf{Q}_n)} \\ &= p(o_n|q_n)p(o_{n-1}|q_{n-1})...p(o_2|q_2)p(o_1|q_1) \end{aligned} \tag{8}$$

2. EXPERIMENTS

Our approach, as mentioned earlier is two fold. The first step is generating a descriptor for every image using RVQ. The second step involves tracking the evolution of this descriptor using an HMM. We use the Weizmann database (Figure 5) reported in [2] and compare our results with other researchers

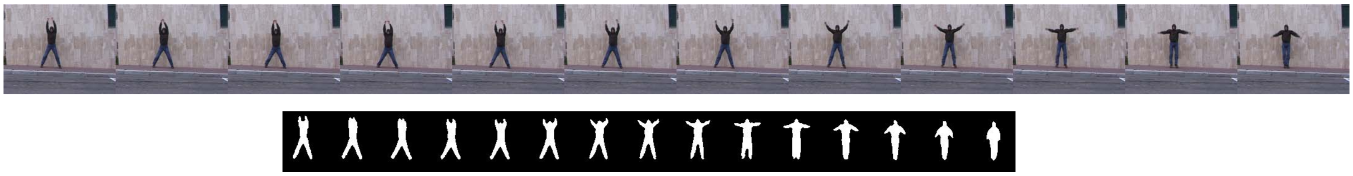

Fig. 5. Sample images from the Weizmann dataset.

who have used this database as well for similar purposes [2] [15] [3]. [2] uses the solution to the Poisson equation to extract a wide variety of useful local shape properties. [15] proposes a statistical measure between video sequences based on their behavioral content. [3] uses kinematic features of optical flow of human motion to understand human action.

We separate the dataset into training and test images. An RVQ codebook is generated using the training dataset (Figure 2). This one time offline step takes approximately 6 minutes on a Core2 Duo 2.5 GHz laptop for a total of 2400 images. An image reconstruction using direct sum successive approximation with this codebook is shown in Figure 3. Generating descriptors for the images takes another 3 minutes, and this is also an offline step. This time includes GUI related instructions as well as accessing an SQL Server database. These steps are accomplished using the C language. Figure 4 shows the PSNR value of image reconstruction for every point inside the image. Notice that the highest PSNR is at the center, essentially showing that this is the centroid of the human at that point in the sequence. Training an HMM takes around 700 msec in Matlab. The online task of computing the class conditional probabilities takes 1.1 msec per frame in Matlab.

We are trying to identify 10 actions, walk (a1), run (a2), skip (a3), jumping jacks (a4), jumping (a5), in place jumping (a6), sideways motion (a7), waving with one hand (a8), waving with two hands (a9) and bending (a10). There are a total of 8 users. To test the actions of particular user, we train our system on all the other users. This process is repeated for all users one at a time.

3. RESULTS

Our results, and a comparison with other researchers is given in Table 6. It is worth mentioning that we use around 40% of the images for training. Since the dataset is based on 50 fps video, this is about half a second of training video per action. Yet, Table 6 shows that our classification accuracy is comparable with other methods that use the entire database. Notice that the greatest error is in classifying jumping jacks as waving with two hands, since these two actions are very similar. Moreover, since we use very few images, this is understandable.

4. CONCLUSIONS

We demonstrate the first reported application of multi-stage residual vector quantization to any form of video processing. In this paper, the application we focus on is human action recognition using HMMs as an inference engine over an RVQ generated sequence of image descriptors. We demonstrate comparable performance with other state of the art methods, but using less than half the number of training images. Our future work will focus on temporal evolution of the codebooks for time adaptive analysis of human motion evolution.

5. REFERENCES

[1] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 3, pp. 257–267, 2001.

[2] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 12, pp. 2247–2253, 2007.

[3] Saad Ali and Mubarak Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 2, pp. 288–303, 2010.

[4] L. Sigal, S. Bhatia, S. Roth, M.J. Black, and M. Isard, "Tracking loose-limbed people," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004.

[5] C. F. Barnes, "Image-driven data mining for image content segmentation, classification, and attribution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, pp. 2964–2978, 2007.

[6] A. Hervieu, P. Bouthemy, and J. P. Le Cadre, "A hmm-based method for recognizing dynamic video contents from trajectories," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 2007, vol. 4.

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
a1	87.5						12.5			
a2		100.0								
a3		12.5	87.5							
a4				75.0					25.0	
a5			12.5		87.5					
a6						87.5		12.5		
a7							100.0			
a8								100.0		
a9									100.0	
a10										100.0

(a)

Overall accuracy: 92.5%
40% of dataset used

a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
100.0									
	98.0	2.0							
	2.9	97.1							
			100.0						
		10.8		89.2					
					100.0				
						100.0			
			0.9		0.9		94.8	3.5	
			0.9				1.9	97.2	
								0.9	99.1

(b)

Overall accuracy: 97.8%

a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
82.4	0.8	2.4		14.4					
2.0	34.7	51.0		10.2		2.0			
1.4	40.6	43.5		8.7		5.8			
			95.5		0.8		0.8	1.5	1.5
13.8	13.8	26.2		29.2		16.9			
			12.8		84.9			1.2	1.2
3.2	4.8	23.8		17.5		50.8			
			0.9		7.0		29.6	62.6	
			0.9		0.9		44.3	51.9	1.9
			4.5				3.6	5.4	86.6

(c)

Overall accuracy: 58.9%

Details not provided

(d)

Overall accuracy: 95.75%

Fig. 6. Action confusion matrix. We compare classification performance of our approach on 10 actions, walk (a1), run (a2), skip (a3), jumping jacks (a4), jumping (a5), in place jumping (a6), sideways motion (a7), waving with one hand (a8), waving with two hands (a9) and bending (a10) with other researchers using the same Weizmann dataset. (a) Our method, produces accurate results despite using less than half the training data. (b) Method reported in [2]. (c) Method reported in [15] (d) Method reported in [3].

[7] Xie Lexing, Xu Peng, Chang Shih-Fu, A. Divakaran, and H. Sun, "Structure analysis of soccer video with domain knowledge and hidden markov models," *Pattern Recognition Letters*, vol. 25, pp. 767–75, 2004.

[8] Riten Gupta, Alfred Hero, Alfred, and O. Hero, "High rate vector quantization for detection," *IEEE Transactions on Information Theory*, vol. 49, pp. 1951–1969, 2003.

[9] E. Tuncel, P. Koulgi, and K. Rose, "Rate-distortion approach to databases: storage and content-based retrieval," *Information Theory, IEEE Transactions on*, vol. 50, no. 6, pp. 953–967, 2004.

[10] C. F. Barnes, S. A. Rizvi, and N. M. Nasrabadi, "Advances in residual vector quantization: a review," *Image Processing, IEEE Transactions on*, vol. 5, no. 2, pp. 226–262, 1996.

[11] Allen Gersho and Robert Gray, *Vector Quantization and Signal Compression (The Springer International Series in Engineering and Computer Science)*, Springer, 1991.

[12] Juang Biing-Hwang and Jr. Gray, A., "Multiple stage vector quantization for speech coding," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '82.*, 1982.

[13] C. F. Barnes and R. L. Frost, "Vector quantizers with direct sum codebooks," *IEEE Transactions on Information Theory*, vol. 39, no. Copyright 1993, IEE, pp. 565–80, 1993.

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

[15] L. Zelnik-Manor and M. Irani, "Event-based analysis of video," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 8-14 Dec. 2001*, 2001.