MODEL-BASED COMPRESSION OF NONSTATIONARY LANDMARK SHAPE SEQUENCES

Samarjit Das and Namrata Vaswani

ECE Department, Iowa State University, Ames, IA

ABSTRACT

We have proposed a novel model-based compression technique for nonstationary landmark shape data extracted from video sequences. The main goal is to develop a technique for the compact storage of landmark shape data. We use Nonstationary Shape Activity (NSSA) to model the shape sequences. The shape data is encoded by applying Differential Pulse Code Modulation (DPCM) on the shape velocity coefficients under the NSSA model. We have studied the system performance in terms of compressibility-distortion trade off. NSSA based compression technique has been compared with two other methods based on existing shape modeling techniques namely, Stationary Shape Activity (SSA) and Active Shape Model (ASM). We tested our system with landmark shape data extracted from multiple video sequences of the CMU mocap database. It was found that NSSA outperforms both SSA and ASM in terms of compressibility for a given distortion tolerance. Thus NSSA based compression technique could be very useful in the applications like storage of large volumes of biomedical landmarks' data.

1. INTRODUCTION

The goal of our work is to develop a model-based compression technique for landmark shapes (ordered configuration of feature points of interest). An example of a such shape sequence has been shown in Fig. 1. There are multiple applications where key landmarks of interest are extracted either manually(e.g. by doctors/radiologists in medical imaging applications) or using marker based motion capture technologies(e.g. these are used for human joint motion understanding for biomechanics applications). An example is, the CMU motion capture database ([1], http://mocap.cs.cmu.edu). Now, the question is: Can we model the correlation between temporal landmark shape sequences and use the model for efficient lossy-compression to efficiently reduce the amount of data to be stored? If we can, then it would be a very efficient way of storing large volumes of biomedical landmarks' data. Related work addressing similar questions is [2].

There are quite a few shape models being recently used in the shape modeling literature. Stationary Shape Activity(SSA) [3] assumes a constant mean shape over time and tries to model the shape variations w.r.t a tangent space defined at the mean shape. Active Shape Models(ASM) [4] also assumes a fixed mean shape and models the deviations from the mean shape. Non-stationary Shape Activity(NSSA) [5], however, does not require a constant mean shape and and it models the shape sequences assuming as if the mean shape is changing at each time instant. We use this non-stationary model for shape data compression (details in Sec. 2).

In recent years, there has been a significant amount of work on model based video and shape compression. Quite a few of them are in the field of biomedical imaging [2, 6, 7]. For the shape coding in object-based video sequence, [8] uses a context based arithmetic coding of 2D shape sequences. A low bit-rate video compression technique utilizing compact encoding of motion fields has been proposed in [9]. A lossless and near-lossless compression scheme for 4D volume biomedical image sequences has been proposed in [2].

Paper Organization: We explain the NSSA model and the corresponding shape compression techniques in Sec. 2 and Sec. 3 respectively. Performance evaluations are discussed in Sec. 4 and experimental results are shown in Sec. 5.

2. MODELING SHAPE SEQUENCES

The shape at each time instant is described by an ordered set of k landmark locations. The landmarks are *points of interest* for describing the shape of an object. An example has been shown in Fig. 1.

The configuration of the set of landmark locations, denoted s_t (at time t) is represented as a complex vector (x locations + j y locations) where $j = \sqrt{-1}$. As explained in [10], we first perform translation normalization on s_t to get the centered shape, y_t, i.e. $y_t = C_k s_t$ where C_k is,

$$C_k = I_k - 1_k 1_k^T / k \tag{1}$$

Here, I_k is a $k \times k$ identity matrix and 1_k is a column vector with k rows with all entries as 1. Then we scale normalize y_t to generate the corresponding pre-shape w_t as, $w_t = \frac{y_t}{||y_t||}$. We define the initial shape as $z_0 = w_0$. For $t \geq 1$, we define the rotation alignment of current pre-shape w_t w.r.t the previous shape z_{t-1} as,

$$[z_t, \theta_t] = align(w_t, z_{t-1}) \tag{2}$$

Where, the aligned shape is given as, $z_t = w_t \frac{w_t^* z_{t-1}}{|w_t^* z_{t-1}|}$ and the alignment angle is given as, $\theta_t = angle(w_t^* z_{t-1})$. Here $(.)^*$ denotes conjugate transpose. If the shape variation over a sequence is small, one can compute a single Procrustes mean shape [10], μ, for the entire sequence, project all the z_t's into the tangent space at μ, denoted T_μ, and define an autoregressive (AR) model on the tangent space projections, denoted $v_t(\mu)$. v_t is obtained as follows [10],

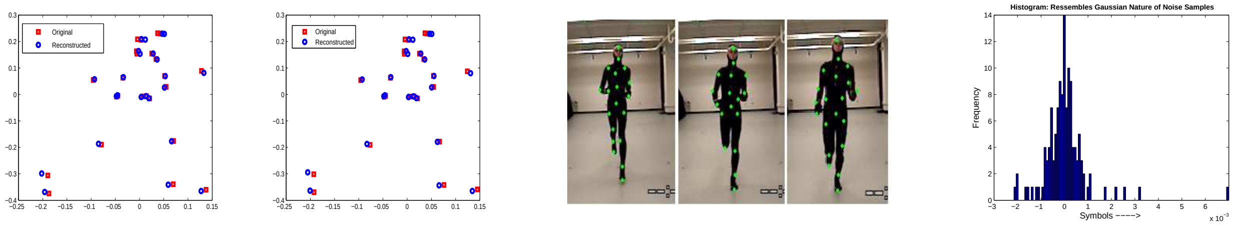
$$v_t(\mu) = [I - \mu\mu^*] z_t \tag{3}$$

The inverse map (projection from tangent space to shape space) is given by [10],

$$z_t = (1 - v_t^* v_t)^{\frac{1}{2}} \mu + v_t \tag{4}$$

SSA [3] used the above and then defined an AR or ARMA model on v_t. ASM (or PDM)[4] assumed both stationarity and the fact that z_t belongs to a Euclidean space and computed $v_t = z_t - \mu$. Statistics of v_t were modeled.

This research was partially supported by NSF grant ECCS 0725849



(a) Frame 1 (b) Frame 2 (c) Landmarks from a video (d) Histogram for $\tilde{n}_t$

Fig. 1. Fig. 1(a) and Fig. 1(b) shows original and reconstructed landmark shapes over two frames of a video sequence of motion activity "Run". Fig. 1(c) shows landmarks corresponding to a video sequence. Fig. 1(d) shows the histogram corresponding to one of the scalar dimension of $\tilde{n}_t$

2.1. Nonstationary Shape Activity(NSSA) model

In non-stationary shape activities, the mean shape can be time varying. We use $\mu_t = z_{t-1}$, i.e. we use the shape at $t-1$ as the pole for the current tangent space projections. Hence, modeling the shape deformation dynamics requires a tangent space to be defined w.r.t the current shape. Thus the tangent space coordinate v_t w.r.t z_{t-1} can be termed as the *Shape Velocity* at time instant t.

The shape velocity, v_t has only $k-2$ independent complex dimensions [5], i.e. it can be rewritten as $v_t = U_t c_t^{complex}$ where $(U_t)_{k \times k-2}$ contains the $k-2$ orthonormal basis directions spanning $T_{z_{t-1}}$ and $c_t^{complex} \in \mathcal{C}^{k-2}$ are the basis coefficients. $c_t^{complex}$ may be interpreted as a "shape speed" vector.

For computing successive U_t's we have used the technique proposed in [11], which fixes a major error in [5]. To obtain an aligned sequence of basis directions over time, we obtain each column of U_t by starting with the corresponding column of U_{t-1}, making it perpendicular to z_{t-1} and applying a Gram-Schmidt orthogonalization procedure on the resulting set of vectors. This procedure can be summarized as follows,

$$\begin{aligned} U_t &= g(U_{old}, z_{old}), \; where \\ g(.)_m &\triangleq [I - z_{old} z_{old}^* - \sum_{j=1}^{m-1} g(.)_j g(.)_j^*] U_{old,m}, \forall \, m \end{aligned} \tag{5}$$

Here, $g(.)_m$ denotes the m^{th} column of $g(U_{old}, z_{old})$. Since the columns of U_t are aligned, it is fair to assume that $c_{t,j}^{complex}$'s are identically distributed. Since they are also correlated, we model them by an AR model. For simplicity of notation, we first convert $c_t^{complex}$ into a $2k-4$ dimensional real vector. We denote this operation by,

$$c_t^{real} = vec(c_t^{complex}) \tag{6}$$

and the inverse operation (obtaining the complex vector) is denoted by $c_t^{complex} = vec^{-1}(c_t^{real})$. We perform Principal Component Analysis(PCA) on $\{c_t^{real}\}$ and retain only M_{pca} number of most significant eigen directions. The corresponding $(2k-4) \times M_{pca}$ dimensional basis vector is denoted by, U_{pca}. Now, the representation of c_t^{real} in the PCA space is given as,

$$c_t = U_{pca}^* c_t^{real} \tag{7}$$

We model $\{c_t\}$ using an Auto-Regressive (AR) model. Thus the NSSA model can be summarized as:

$$\begin{aligned} c_t &= A c_{t-1} + n_t, \; n_t \sim \mathcal{N}(0, Q) \\ c_t^{real} &= U_{pca} c_t \\ U_t &= g(U_{t-1}, z_{t-1}) \\ z_t &= (1 - {c_t^{real}}^T c_t^{real})^{1/2} z_{t-1} + U_t vec^{-1}({c_t}^{real}) \end{aligned} \tag{8}$$

with,
$$z_0 = z_{init}, \; U_0 = basis(C_k), \; c_0 = 0 \tag{9}$$

Starting with $\{w_t\}$ the computation of c_t and θ_t can be summarized as:

1. Set $z_{init} = z_0 = w_0$. Compute, $[z_t, \theta_t] = align(w_t, z_{t-1}) \;\; \forall t > 0$
2. Begin with computing $U_0 = basis(C_k)$ by performing Singular Value Decomposition(SVD) on $[I_k - z_0 z_0^*] C_k$. Then,

$$\begin{aligned} U_t &= g(U_{t-1}, z_{t-1}) \\ c_t^{complex} &= U_t^* z_t \\ c_t^{real} &= vec(c_t^{complex}) \end{aligned} \tag{10}$$

3. Perform PCA on $\{c_t^{real}\}$ and compute U_{pca}. Then, $c_t = U_{pca}^* c_t^{real}$

The AR model parameter A and Q can be computed from $\{c_t\}$ using Yule-Walker equation. These parameters are used while encoding $\{c_t\}$ to perform compression on the shape data.

3. NSSA MODEL-BASED COMPRESSION

We assume that translation and global scale are unimportant and need not be stored. We are only interested in storing the global rotation and the shape change. Rotation is also not needed if it is due to camera motion, but some part of it may be due to actual rotation of the landmark configuration. *Thus our goal is to take the preshape sequence $\{w_t\}$ and compress it so as to achieve the minimum possible shape distortion for a given bit budget.*

We began by first computing the differential entropy of the three possible models: NSSA, SSA and ASM, with model parameters learnt from the data. If the model assumptions are correct, the differential entropy is proportional to the entropy of the quantized data for small enough quantization size [12]. We realized that a measure of the average differential entropy of NSSA was of the order of -250, which is much smaller than that of SSA(-200) or ASM(-195) [11]. This gives a preliminary indication that NSSA will indeed be

Algorithm 1 **Landmark Shape Data Compression/Decompression**

```
Required Inputs: The pre-shape sequence $\{w_t\}$, A, $U_{pca}$ at the transmitter side (compression) and $\{\tilde{n}_t\}$, $\{\tilde{\theta}_t\}$, $U_{pca}, z_{init}, A$ at the receiver side (decompression).
    Initialize: $\tilde{z}_0 = w_0 = w_{init}$, $\tilde{U}_0 = basis(C_k)$, $\tilde{c}_0 = c_0 = \mathbf{0}$. Computation of $basis(C_k)$ is performed as mentioned in Sec. 2.1.
    For $t > 0$,
(a)     $[z_t, \theta_t] = align(w_t, \tilde{z}_{t-1})$. Use the equation (2) for this step.
(b)     $\tilde{U}_t = g(\tilde{U}_{t-1}, \tilde{z}_{t-1})$. Use equation (5) for this step.
(c)     Compute $c_t = U^*_{pca}[vec(\tilde{U}_t^* z_t)]$
(d)     Compute $n_t = c_t - A\tilde{c}_{t-1}$
(e)     Quantize $\tilde{n}_t = Quantize(n_t)$, $\tilde{\theta}_t = Quantize(\theta_t)$. Transmit $[\tilde{n}_t, \tilde{\theta}_t]$
        Decompressor (Implemented at each $t$ at the TX end) :
(f)     Compute $\tilde{c}_t = A\tilde{c}_{t-1} + \tilde{n}_t$
(g)     Compute $\tilde{U}_t = g(\tilde{U}_{t-1}, \tilde{z}_{t-1})$
(g)     Compute $\tilde{v}_t = \tilde{U}_t vec^{-1}(U_{pca}\tilde{c}_t),\ \tilde{z}_t = (1 - \tilde{v}_t^* \tilde{v}_t)^{\frac{1}{2}} \tilde{z}_{t-1} + \tilde{v}_t$
(h)     Compute $\tilde{w}_t = \tilde{z}_t e^{-j\tilde{\theta}_t}$, $\tilde{w}_t^{RX} = \tilde{w}_t$
    end
    Find $p_k(\alpha) = \frac{N(\tilde{n}_{t,k}=\alpha)}{N_{frames}}$, the PMF corresponding to the $k^{th}$ dimension of $\{\tilde{n}_t\}$ for the alphabets $\alpha$'s. A Huffman table can thus be constructed for each scalar dimension. Entropy rate per scalar dimension is given as, $H_k = \sum_\alpha p_k(\alpha)\log_2(\frac{1}{p_k(\alpha)})$.
```

a better model (if model assumptions are valid). Since a large part of the global rotation, θ_t, is often due to camera motion and hence independent of shape dynamics, we compress it separately from the shape sequence, z_t. Consider the NSSA model described in (refer to eq (8)). The AR model prediction error, n_t, is assumed to be independent and identically distributed (iid) gaussian over time. We show in Fig. 1(d) that this assumption is a valid one for our datasets. Hence we propose to compute the sequence, $\{n_t\}$, from the shape sequence, $\{z_t\}$, quantize it and store/transmit the Huffman coded version of quantized n_t.

Now, if the above quantization is done in an open-loop fashion - first compute the $\{n_t\}$ sequence and then quantize it, the reconstruction error at the decompression/receiver end will increase over time. This is because the quantization error in n_t will result in error in the estimate of c_t and hence of z_t, which in turn will propagate to the next time step - the error in z_{t+1} will be both due to error in n_{t+1} and due to the effect of errors in all past n_t's. This is a standard problem in all model-based compression schemes.

We use a standard solution to the above standard problem - we adopt a two-level Differential Pulse Coded Modulation (DPCM) scheme. Thus our encoding scheme involves implementing the receiver at the compression end itself before computing the next n_t, i.e. at each t:

1. Use the quantized version of n_t, denoted $\tilde{n}_t$, to compute $\tilde{c}_t = A\tilde{c}_{t-1} + \tilde{n}_t$
2. Compute the reconstructed shape $\tilde{z}_t$ using (4)
3. Compute $\tilde{U}_{t+1}$ which is the projection matrix for the tangent space perpendicular to $\tilde{z}_t$ (and close to $\tilde{U}_{t-1}$) using Gram-Schmidt given in (5).
4. Compute $c_{t+1} = \tilde{U}^*_{t+1} z_{t+1}$ and $n_{t+1} = c_{t+1} - A\tilde{c}_t$ and quantize it.

The complete stepwise algorithm is summarized in Algorithm 1.

We use simple quantization to encode the rotation angle sequence, θ_t, although if a Markov model were assumed on θ_t then DPCM could be used there as well. We experimented with both approaches, with negligible difference in performance.

4. PERFORMANCE EVALUATION AND COMPARISON

To compare the performance of NSSA-based DPCM, we applied an exactly analogous scheme to the SSA model and to the ASM model. We varied the number of quantization bits per unit time per scalar dimension from 4 to 10 and plotted the mean squared distortion of the preshape against the Huffman-encoded bit rate (R-D plot) per unit time. The Huffman-coded bit rate will always be within one bit of the entropy rate [12] defined by,

$$H(b) = \sum_{k=1}^{M_{pca}} H_b(\tilde{n}_{t,k}) + H_b(\theta) \tag{11}$$

Where, $H_b(\tilde{n}_{t,k})$ is the entropy rate for the k^{th} dimension of $\tilde{n}_t$, given the word-length $\mathbf{b}$. M_{pca} is the dimensionality of n_t and $H_b(\theta)$ is the corresponding entropy rate for $\tilde{\theta}_t$. $H_b(\tilde{n}_{t,k})$ is computed as,

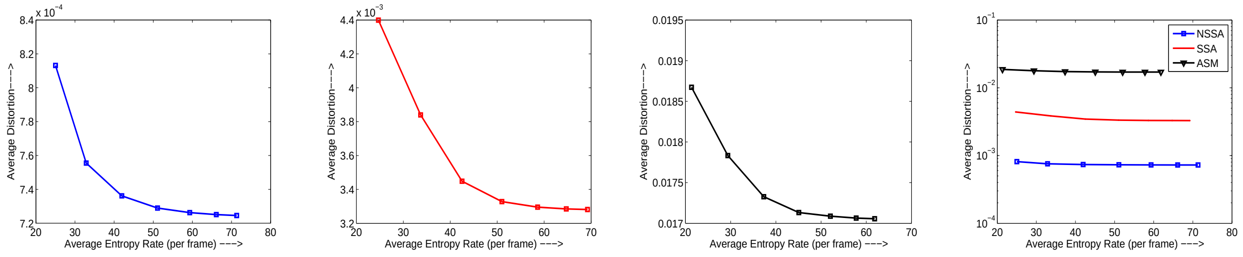
$$H_b(\tilde{n}_{t,k}) = \sum_\alpha p_k(\alpha)\log_2(\frac{1}{p_k(\alpha)}) \tag{12}$$

Where, $p_k(\alpha) = \frac{N(\tilde{n}_{t,k}=\alpha)}{N_{frames}}$, the PMF corresponding to the k^{th} dimension of $\{\tilde{n}_t\}$ for the alphabets α's.

The mean squared distortion is defined as,

$$D = \frac{1}{N_{time}} \sum_{t=1}^{N_{time}} ||w_t - \tilde{w}_t^{RX}||^2 \tag{13}$$

Where, w_t is the original preshape at the transmitter and $\tilde{w}_t^{RX}$ is the reconstructed preshape at time instant t. We compute the distortion



(a) RD plot for NSSA

(b) RD plot for SSA

(c) RD plot for ASM

(d) Performance comparison

Fig. 2. RD plots for compression techniques based on various methods. Fig. 2(d) shows comparison of NSSA,SSA and ASM with log scale on distortion. It is to be noted that the y-axis of the 2(a),2(b) and 2(c) are all in linear scale. Due to order of magnitude differences, they are plotted in separate figures. Distortion for NSSA is of the order of 10^{-4} whereas distortions corresponding to SSA and ASM are of the orders of 10^{-3} and 10^{-2} respectively. The y-axis of the combined plot (i.e. 2(d)) is in log-scale. Note: The bits required for coding the AR model parameters and z_{init} or μ are not considered in the plots as they are common to all the three methods.

per unit time and entropy rate (or Huffman-coded bit rate) for each video sequence and plot their average values over all sequences (a total of 80). The results are shown in Fig. 2.

Note that, in all schemes (NSSA, SSA, ASM), one also needs to accurately quantize and store the Huffman table, the AR matrix, A, the initial shape z_{init} (or the mean shape μ in case of SSA or ASM). These will require the same number of bits for all methods and hence are not compared. Also, this will be a one-time cost and its effect on the total number of bits will become negligible as the length of the sequence increases.

5. EXPERIMENTAL RESULTS

We studied the compression performance of our system over multiple video sequences. In the encoding process, we tried seven different quantization word-length ranging from 4 to 10 bits. For each word-length, we computed average entropy rate per frame and the average distortion over all the video sequences. Thus we got the rate-distortion(R-D) plot of the system. The RD plot characterized the system performance in terms of trade-off between distortion tolerance and compressibility. Lower is the entropy rate(i.e. higher the compressibility), higher is the distortion in the reconstructed shape sequences. We compare the compression performance of NSSA, SSA, and ASM using their corresponding RD curves. The plots are shown in Fig. 2. It can be clearly seen that the performance of NSSA based compression technique is better than that SSA and ASM. NSSA gave much lower distortion for a given compressibility. For similar entropy rates, the NSSA based method gave a distortion of the order of 10^{-4}, while distortions corresponding to SSA and ASM were of the order of 10^{-3} and 10^{-2} respectively.

Another measure of the system performance for a specific word-length can be given by peak signal-to-noise ratio(PSNR). Where,

$$PSNR(b) = 10\log_{10}(\frac{||w_t||^2}{D_{avg}(b)}) = 10\log_{10}(\frac{1}{D_{avg}(b)}) \qquad (14)$$

Since $||w_t|| = 1$ by definition. PSNR values for $b = 4$ are 30 dB for NSSA, 23.5 dB for SSA and 17.2 dB for ASM respectively.

6. CONCLUSION

In this paper, we have proposed an efficient compression technique for landmark shape data extracted from video sequences. It is found that NSSA based compression technique outperforms SSA and ASM based techniques in terms of Compressibility-Distortion trade off. NSSA defines a second order Markov model on shape data while SSA and ASM define a first order model and that is clearly one reason for its superior performance. But note that the reason it is not possible to define a valid second order Markov model in the SSA [3] or ASM [4] framework is because they assume a single mean shape and define dynamics in the tangent space w.r.t. this mean shape.

7. REFERENCES

[1] "The carnegie mellon motion capture database cmu graphics lab, http://mocap.cs.cmu.edu,"

[2] P. Yan and A. Kassim, "Lossless and near-lossless motion-compensated 4d medical image compression," in *IEEE International Workshop On Biomedical Circuits And Systems*, 2004.

[3] A. Veeraraghavan, A. RoyChowdhury, and R. Chellappa, "Matching shape sequences in video with an application to human movement analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 12, pp. 1896–1909, 2005.

[4] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models: Their training and application," *Computer Vision and Image Understanding*, vol. 61, pp. 38–59, January 1995.

[5] N. Vaswani and R. Chellappa, "Nonstationary shape activities," in *IEEE Conf. Decision and Control (CDC)*, 2005.

[6] S. Sclaroff and A.P.Pentland, "On modal modeling for medical images: Underconstrained shape description and data compression," in *Biomedical Image Analysis*, June 1994.

[7] J. L. Su, C. C. Lin, J. R. Duann, and Y. S. Tsai, "Biomedical image compression using vector quantization algorithm," in *IEEE International Conference on Engineering in Medical and Biology Society*, pp. 66–67, Oct 1993.

[8] N. Brady, F. Bossen, and N. Murphy, "Contex-based arithmatic encoding of 2d shape sequences," in *IEEE Intl. Conf. on Image Processing*, pp. 26–29, Oct 1997.

[9] H. Zang and F. Bossen, "Region-based coding of motion fields for low bit-rate video compression," in *IEEE Intl. Conf. on Image Processing*, pp. 24–27, Oct 2004.

[10] I. Dryden and K. Mardia, *Statistical Shape Analysis*. John Wiley and Sons, 1998.

[11] S. Das and N. Vaswani, "Blind submission," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[12] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley Series, 1991.