From Videos to Verbs: Mining Videos for Activities using a cascade of dynamical systems (Supplemental Material)

Pavan K. Turaga, Ashok Veeraraghavan, Rama Chellappa
Department of Electrical and Computer Engineering and Center for Automation Research, UMIACS
University of Maryland
College Park, MD 20742
{pturaga,vashok,rama}@umiacs.umd.edu

1. Generative power of Cascade of LTI

A useful test for a representational model is to synthesize from it, and see how well the synthesized samples resemble real-world phenomenon. In this section, we show a few synthesis results obtained using the learnt models. In the first experiment, we used one walk sequence from the USF gait gallery data to learn one walk pattern. We modeled the entire walk sequence using just one LTI model. Then, we used the learnt model to generate the sequence. A few frames from the generated sequence are shown in figure 1.

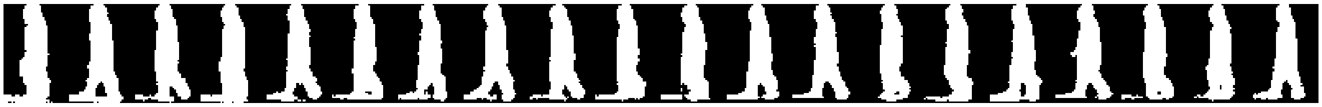

Figure 1. Generated Gait Sequence from learnt model

In the next experiment, we generated the Bending sequence. During the learning stage, the sequence was segmented automatically into 3 segments by the proposed segmentation technique. A model was learnt for each segment. To synthesize the activity, we generated sequences from each of the models, and switched from one model to the other according to the discovered cascade. The dwell time in each segment was sampled from the learnt distributions. The generated sequence is shown in figure 2.

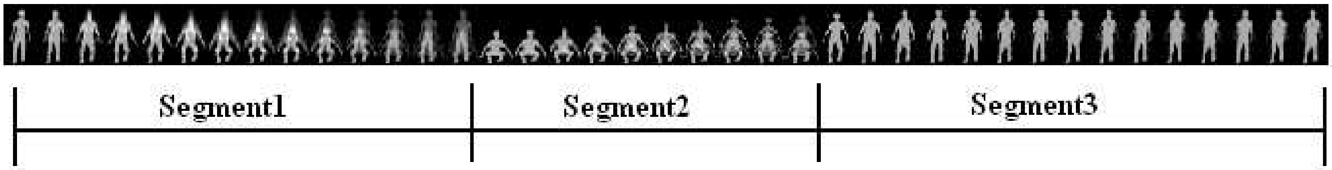

Figure 2. Generated Bending Sequence from learnt cascade of LTI

We see from both these experiments that the sequence of LTI is indeed a rich model which can be used to represent several activity classes.

2. Temporal Segmentation

In this section, we show some segmentation results obtained on actual video sequences of a person performing 5 different activities. We show segment boundaries for the activities as seen from two different views in figures 3 to 7.

We see that the videos are segmented at the same pose consistently in both views. This indicates that our algorithm indeed finds semantically meaningful segment boundaries.

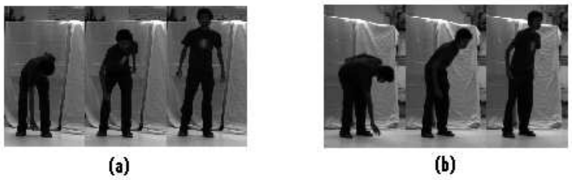

Figure 3. Bending (a) View 1, (b) View 2

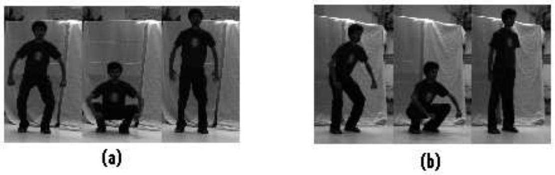

Figure 4. Squatting (a) View 1, (b) View 2

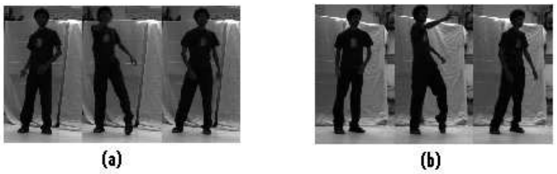

Figure 5. Throwing (a) View 1, (b) View 2

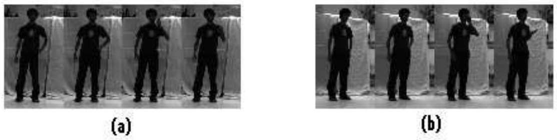

Figure 6. Pick Phone (a) View 1, (b) View 2

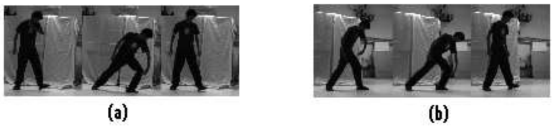

Figure 7. Batting (a) View 1, (b) View 2

2.1. Effect of Boundary Improvement

We suggested a scheme for tweaking the segment boundaries based on the learnt models, to take care of the sub-optimality of the segmentation scheme, in section 3.1 of the main paper. In most cases, temporal segmentation based on affine parameters gave reasonable results. But, in cases where this segmentation did not give good results, we observed improved segmentation results after tweaking the boundary according to the proposed scheme. We show one such example in figure 8.

3. View Invariance

In this section, we shall discuss in more detail some assumptions of section 4.1 of the main paper.

3.1. Application to View Invariance

It was stated in section 4.1 of the main paper, that for the case of a 2-D homography given by $H = [h_{ij}]$, under small changes in view-point $h_{31}, h_{32} << h_{33}$. We will justify this statement here.

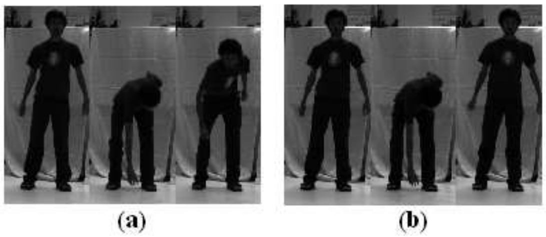

Figure 8. Bending boundaries (a) Before tweaking, (b) After tweaking

An argument involving camera rotations is given below. Let, the transformation between the coordinate frame of the first camera and that of the second camera be given by a rotation and translation. Then, the homography induced by a plane π, between the two views is given by [3]

$$H = M'(R + \frac{Tn^T}{d_\pi})M^{-1} \tag{1}$$

where R and T are the rotation matrix and translation vector respectively, n is the normal to the plane π and d_π is the distance of the plane π from the origin, M and M' are the transformation from the image plane to the camera coordinate system for the two cameras. In the simplest case, we can take $M = M' = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}$,

where f denotes the focal length of the camera, and x_0, y_0 is the origin of the image plane. When the two views are close to each other, we can approximate $T = [\epsilon_x, \epsilon_y, \epsilon_z]'$ and R using small rotations as [2]

$$R \approx \begin{bmatrix} 1 & -n_3\theta & n_2\theta \\ n_3\theta & 1 & -n_1\theta \\ -n_2\theta & n_1\theta & 1 \end{bmatrix} \tag{2}$$

where, θ is the rotation angle, n_1, n_2, n_3 are the directional cosines of the axis of rotation, hence, related by $n_1^2 + n_2^2 + n_3^2 = 1$. On substituting these quantities and the plane normal $n = [n_x, n_y, n_z]$, in (1) and simplifying, we obtain the following relations between the required elements of $H - h_{31}, h_{32}, h_{33}$,

$$\frac{h_{31}}{h_{33}} = \frac{a/f}{-ax_0/f - by_0/f + c} \tag{3}$$

$$\frac{h_{32}}{h_{33}} = \frac{b/f}{-ax_0/f - by_0/f + c} \tag{4}$$

where $a = -n_2\theta + \frac{\epsilon_z n_x}{d_\pi}, b = n_1\theta + \frac{\epsilon_z n_y}{d_\pi}, c = 1 + \frac{\epsilon_z n_z}{d_\pi}$. In the limit, when $\theta \rightarrow 0$ and $\epsilon_x, \epsilon_y, \epsilon_z \rightarrow 0$, we obtain $a \rightarrow 0, b \rightarrow 0, c \rightarrow 1$.

$$\lim_{\theta, \epsilon_x, \epsilon_y, \epsilon_z \rightarrow 0} \frac{h_{31}}{h_{33}} = 0 \tag{5}$$

$$\lim_{\theta, \epsilon_x, \epsilon_y, \epsilon_z \rightarrow 0} \frac{h_{32}}{h_{33}} = 0 \tag{6}$$

Thus, $h_{31}, h_{32} << h_{33}$.

We conducted an experiment to test the performance of recognition across views. The setup is similar to the previous experiment. There are two cameras looking at the person performing the activities with about a 20° angle between their optical axes, which is a significant camera shift. Models were built on one view, and tested on another. Recognition performances are shown in table 1.

	Baseline		Our Method	
	Exemplars		Exemplars	
Activity	1	10	1	10
1	30	37.5	30	37.5
2	60	62.5	70	75
3	0	12.5	0	12.5
4	50	62.5	30	37.5
5	30	25	40	62.5
6	40	75	20	62.5
7	70	100	80	100
8	0	50	0	50
9	40	25	50	37.5
10	10	50	10	62.5
Average	33	50	33	53.5

Table 1. Recognition accuracies for two schemes

4. Model Order Selection

A practical issue in learning the LTI model parameters is to choose an appropriate value for the hidden state dimension d. The answer to this is tied to the domain, and there is no general selection rule. The number d represents the number of basis vectors to project the data on to (the number of principal components). Usually, the higher the dimension d, the more accurate the representation will be. But, the higher the d, the more the data required for robust estimation of the parameters and the higher the computational cost. One needs to make a tradeoff between these issues based on domain knowledge. To see the effect of varying d, we conducted recognition experiments on the USF dataset using $d = 5, 10, 15$ on Probes A-G. Results are shown in figure 9. We see that the recognition accuracies show an increasing trend as d increases, but the increase from $d = 10$ to $d = 15$ is only marginal. In general, criteria such as Akaike Information Criteria (AIC) [1], Bayesian Information Criteria (BIC) [4], etc may also be used to get the optimal number of free parameters (in our case d). In our experiments, we empirically found that using $d = 10$ gives good results across various domains and activity classes.

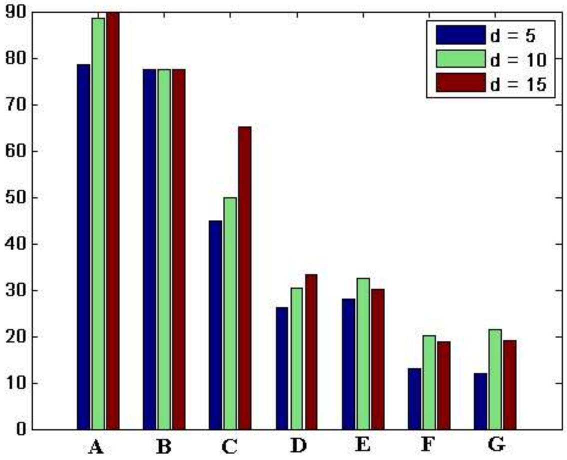

Figure 9. Recognition Accuracies on USF data

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 1974.

[2] J. Q. Fang and T. S. Huang. Solving three-dimensional small-rotation motion equations. *In Proc. IEEE CVPR*, 1983.

[3] B. Rousso, S. Avidan, A. Shashua, and S. Peleg. Robust recovery of camera rotation from three frames. *In Proc. APRA IU Workshop, 1996*, 1996.

[4] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 1978.