

Key Frame-Based Activity Representation Using Antieigenvalues*

Naresh P. Cuntoor and Rama Chellappa

Center for Automation Research, University of Maryland,
College Park, MD, 20742, USA
{cuntoor, rama}@cfar.umd.edu
<http://www.cfar.umd.edu/users/cuntoor>

Abstract. Many activities may be characterized by a sequence of key frames that are related to important changes in motion rather than dominant characteristics that persist over a long sequence of frames. To detect such changes, we define a transformation operator at every time instant, which relates the past to the future states. One of the useful quantities associated with numerical range of an operator is the eigenvalue. In the literature, eigenvalue-based approaches have been studied extensively for many modeling tasks. These rely on gross properties of the data and are not suitable to detect subtle changes. We propose an antieigenvalue-based measure to detect key frames. Antieigenvalues depend critically on the turning of the operator, whereas eigenvalues represent the amount of dilation along the eigenvector directions aligned with the direction of maximum variance. We demonstrate its application to activity modeling and recognition using two datasets: a motion capture dataset and the UCF human action dataset.

1 Introduction

The scope of modeling human activities has expanded from recognizing simple activities such as walking, running and making hand gestures, to more complex ones that involve an underlying structure. While statistical techniques have been applied in the case of simple activities ([1],[2]), primitive - based approaches that rely on domain knowledge have been proposed for complex ones ([3], [4]). We attempt to provide an unsupervised key frame based representation for human activities by focusing on changes in motion properties rather than a sequence of dominant features that form primitives.

In many activities, the relevant information is contained in a few key frames. These frames may be significant due to certain changes in the data, such as direction, speed and deviation from a known behavior. As an illustration, consider the trajectory traced by a hand when opening the door. The shape of the trajectory depends on the person opening the door, the initial position of the hand,

* This work was sponsored by the Advanced Research and Development Activity, a U.S. Government entity which sponsors and promotes research of import to the intelligence community.

the camera’s viewing direction, etc. Modeling these variations is neither easy nor relevant for the activity of opening. The opening action occurs within a few frames when the hand makes contact with the door. The sequence of key frames - extending the hand, grabbing the handle and opening the door - is a sufficient representation. Similarly, we may say that walking is a sequence of events or key stances including the rest stance when the feet are closest to each other and the swing stance when the feet are maximally apart. Jogging may be represented by a similar set of freeze frames, but the changes from frame to frame are different from those of walking.

The theory of antieigenvalues is based on changes in the data. It is sensitive to how much a data vector is turned from a known direction, rather than the direction of persistence [5]. On the other hand, eigenvectors represent the direction of maximum spread of the data and the eigenvalues are proportional to the amount of dilation. We propose an antieigenvalue-based approach for detecting key frames by investigating properties of operators that transform past states to observed future states.

The paper is organized as follows. Section 2 motivates the key-frame based representation for activities. Section 3 gives a brief overview of antieigenvalue theory. Section 4 describes the proposed approach. Section 5 demonstrates the proposed method using two datasets: the MOCAP database and the UCF human action database. Section 6 concludes the paper.

1.1 Prior Work

Aggarwal and Cai [6] present a comprehensive review of human motion and human activities. Ivanov and Bobick [7] propose a two-step procedure where primitives are modeled using HMMs and a sequence of primitives is parsed using stochastic grammar. Hamid et al. [3] present a dynamic Bayesian network framework for tracking and recognizing complex multi agent activities. Vaswani et al. regard a sequence of moving points engaged in the activity as a shape using Kendall’s shape space theory [8]. Nevatia et al. [9] present an Event Representation Language (ERL) that captures the ontological structure of activities using events. Rao et al. [10] detect *dynamic instants*, which are defined as points of maximum curvature along a trajectory. Parameswaran and Chellappa [11] compute view invariant representations for human actions in both 2D and 3D. State-space approaches have been used by many researchers. For example, Brand et al. [1] use coupled HMMs to model human actions that involve multiple parts such as hands and the head. Eigenvalue (and singular value)-based methods have been used extensively in many modeling tasks including face, gait and activities ([12], [13], [14]).

2 Key Frame Representation

As we argued through examples of opening a door, walking, etc., many activities can be represented using key frames instead of the entire video sequence. Generally, there are three ways to decide on what constitutes a key frame. We

may use domain knowledge in a top-down fashion. It requires an extensive model for the activity, which may be tedious. It relies on our ability to detect the key frames across variations in the data that occur due to structural changes and noise [3]. We may hypothesize that the important characteristics of the activity are present in the persistent and dominant frames [14]. This makes it difficult to detect subtle changes, since it may be difficult to distinguish them from noise. We may look for key frames that are a result of certain changes in the data. In other words, changes in the activity may be more useful than the absolute values of a dominant feature in representing the activity. We present an unsupervised approach for detecting key frames based on changes in the data.

Let the past state vector \mathbf{x}_- be transformed by an operator A_t to a future state vector \mathbf{x}_+ . If motion properties do not change appreciably, then \mathbf{x}_+ may be related to \mathbf{x}_- by an identity transformation modulo translation. Such a transformation may be less interesting compared to the case where A_t turns the state \mathbf{x}_- . We show how antieigenvalues may be used to detect such changes and to identify the key frames. In contrast, eigenvalues are tuned to detecting identity-like transformations. It is important to point out that these quantities are of intrinsic interest in their own right. As the term denotes, however, it may be easier to gain an insight into antieigenvalues by contrasting with eigenvalues and eigenvectors of the operator.

The motion trajectories are associated with two quantities: the antieigenvalue sequence, which is the sequence of antieigenvalues for the operator A_t for every time t , and the location of the key frames detected using minima in the average antieigenvalue sequence. Both the extent of change as given by the antieigenvalues and the location of key frame are useful for recognition. If viewing conditions change, we may expect the time instants of occurrence of key frames to be more useful since the extent of change depends on viewing direction. On the other hand, if the viewing direction is fixed, antieigenvalues may be used in comparing two activities. We illustrate both these cases in our experiments.

3 Mathematical Preliminaries: Antieigenvalues

We present a brief description of antieigenvalues before discussing its application. A detailed discussion of antieigenvalues may be found in [5] or [15].

For a square matrix A , a non-zero vector \mathbf{x} is said to be an eigenvector if $A\mathbf{x} = \lambda\mathbf{x}$, and λ is called the eigenvalue. Equivalently, we may state the condition as $\cos\theta = 1$, where θ is the angle between \mathbf{x} and $A\mathbf{x}$. Geometrically, we may think of eigenvectors as those that dilate A but do not turn at all. The eigenvalues represent the amount of dilation. On the other hand, antieigenvectors are critical to the turning of A . Instead of seeking $\cos\theta = 1$ or $\theta = 0$, antieigenvectors minimize $\cos\theta$, or equivalently, maximize θ . The n^{th} antieigenvalue is defined variationally [5] as

$$\mu_n(A) = \inf_{A\mathbf{x}_n \neq 0} \frac{\Re\langle A\mathbf{x}_n, \mathbf{x}_n \rangle}{\|A\mathbf{x}_n\| \|\mathbf{x}_n\|}, \quad (1)$$

where the n^{th} antieigenvector $\mathbf{x}_n \perp \{\mathbf{x}_1, \dots, \mathbf{x}_{n-1}\}$. It has been shown [15] that all antieigenvectors for 2×2 matrices are of the form

$$\mathbf{x} = \left(\frac{\pm \sqrt{\lambda_j}}{\sqrt{\lambda_i + \lambda_j}}, \frac{\sqrt{\lambda_i}}{\sqrt{\lambda_i + \lambda_j}} \right), \quad (2)$$

where i, j index all eigenvalues. For example, let

$$A = \begin{pmatrix} 9 & 0 \\ 0 & 16 \end{pmatrix} \quad (3)$$

The eigenvalues of A are $\lambda = 9, 16$. Using (2), the first antieigenvector is $\mathbf{x}_1 = (\frac{-4}{5}, \frac{3}{5})$. The antieigenvalue may be calculated by substituting the value of x_1 in (1). The first antieigenvalue is $\mu_1(S) = \frac{\langle A\mathbf{x}_1, \mathbf{x}_1 \rangle}{\|A\mathbf{x}_1\|} = 0.96$. The second antieigenvector is $\mathbf{x}_2 = (\frac{3}{5}, \frac{4}{5})$ and the corresponding antieigenvalue is 0.97.

The first total antieigenvalue is defined as $|\mu_1(A)| = \inf_{A\mathbf{x} \neq 0} \frac{|\langle A\mathbf{x}, \mathbf{x} \rangle|}{\|A\mathbf{x}\| \|\mathbf{x}\|}$. The higher total antieigenvalues are similarly defined.

The total antieigenvalues for matrices of size greater than 2×2 may be calculated as follows (theorems 2.1 and 2.2 in [5]). Let A be a normal operator with eigenvalues $\lambda_i = \beta_i + i\delta_i$, $i = 1, \dots, n$. Then the first total antieigenvalue is either 1 or the smallest number in the set of values

$$G = \left\{ \frac{\sqrt{(\beta_i|\lambda_j| + \beta_j|\lambda_i|)^2 + (\delta_i|\lambda_j| + \delta_j|\lambda_i|)^2}}{(|\lambda_i| + |\lambda_j|)\sqrt{|\lambda_i||\lambda_j|}}, \quad (4)$$

where $i \neq j, 1 \leq i \leq n, 1 \leq j \leq n$. If $|\mu_1(A)| = 1$, then the first total antieigenvector is $\mathbf{z}_1 = (z_1, z_2, \dots, z_n)$ with $|z_j| = 1$ for some j and all other $z_i = 0$. If $|\mu_1(A)|$ is one of the values in G , then the components of \mathbf{z}_1 satisfy $|z_i|^2 = \frac{|\lambda_j|}{|\lambda_i| + |\lambda_j|}, |z_j|^2 = \frac{|\lambda_i|}{|\lambda_i| + |\lambda_j|}$, all other $z_k = 0$. Further, all higher total antieigenvectors take their value from the set G and the corresponding higher total antieigenvectors possess the same component structure as the first total antieigenvector.

4 Key Frame Detection Using Antieigenvalues

In this section, we describe the proposed antieigenvalue-based key frame detection procedure. The key frames are used to compare two activities.

4.1 Feature Selection

We obtain trajectories of the moving object and compute its apparent velocities. The tracking procedure for the different datasets is outlined in section 5. The state of a moving object is said to be the tuple $(x(t), y(t), \dot{x}(t), \dot{y}(t))$, where $(x(t), y(t))$ represents the instantaneous position. We assume that the state undergoes certain important changes at the key frames. We are interested in detecting these changes, rather than modeling the entire sequence of frames. Let

$A_t : H \rightarrow H$ be an operator that relates the past state $\mathbf{x}(t_-)$ into the future state $\mathbf{x}(t_+)$, where H is the Hilbert space domain. There are two estimation tasks here. We need to estimate the past and future states $\mathbf{x}(t_-)$ and $\mathbf{x}(t_+)$. For robust estimation, we assume that the state of the system remains constant for a short interval of time. The other estimation tasks involves optimizing the parameters of the operator A_t . If there is no change in the state from t_- to t_+ , we may expect A to be the identity matrix (modulo translation).

4.2 Computing the Transformation Operator

We assume that the speed remains approximately constant for W frames. The value of W depends on the type of data. For instance, it may be reasonable to assume $W = 25$ or 1 second in far field surveillance data. On the other hand, we may assume $W = 3$ or 0.1 second for short-term human actions (e.g. opening the door, picking up an object, etc.) performed in an office environment. Using W frames of the data, we estimate the state variables $\mathbf{x}(t_-)$ and $\mathbf{x}(t_+)$. Assume that the two states are related by a linear transformation, i.e., $x(t_+) = A_t x(t_-)$. We estimate the parameters of the operator A_t using least squares technique and W frames each for $\mathbf{x}(t_-)$ and $\mathbf{x}(t_+)$.

For two vectors $\mathbf{x}, \mathbf{b} \in \mathcal{R}^n$, let A be the transformation operator such that $A\mathbf{x} = \mathbf{b}$, where $A = [a_{ij}]$, $i, j = 1, 2, \dots, n$. This can be rewritten as $X\mathbf{a} = \mathbf{b}$, where $\mathbf{a} = (a_{11}, a_{12}, \dots, a_{1n}, a_{21}, \dots, a_{2n}, \dots, a_{n1}, \dots, a_{nn})$ and X is a matrix that consists of rows of the form $(0, 0x_1, x_2, \dots, x_n, 0, 0, \dots, 0)$. Suppose $A\mathbf{x} = \mathbf{b}$ holds for W vector pairs $(\mathbf{x}_1, \mathbf{b}_1), \dots, (\mathbf{x}_W, \mathbf{b}_W)$, we can write

$$\begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_W \end{pmatrix} \mathbf{a} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_W \end{pmatrix} \quad (5)$$

We use least squares technique to solve for \mathbf{a} in (5) and recombine the vector \mathbf{a} into the matrix A .

4.3 Numerical Range of the Operator

The numerical range of an operator A is defined as the set $W(A) = \{\langle A\mathbf{x}, \mathbf{x} \rangle, \mathbf{x} \in H, \|\mathbf{x}\| = 1\}$, where H is the Hilbert space. For example, consider an operator defined by the matrix $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$. Let $\mathbf{x} = (p, q)$. For simplicity, assume $\|\mathbf{x}\| = |p|^2 + |q|^2 = 1$. Then $A\mathbf{x} = (q, 0)$ and $\langle A\mathbf{x}, \mathbf{x} \rangle = qp$. A simple calculation shows that $W(A) = \{\mathbf{x} = (p, q) : |p|^2 + |q|^2 \leq \frac{1}{2}\}$ or the half disk. Closely related to the numerical range, we can define the angle of the operator $\cos A$ and the antieigenvalues of the operator A as discussed in section 3.

4.4 Choosing Key Frames

We compute antieigenvalues of A_t using (4) and use the mean antieigenvalue as a measure of relative significance of the frame in representing the activity.

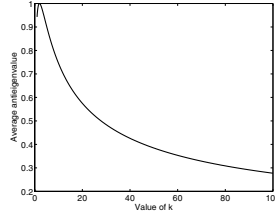


Fig. 1. Average antieigenvalue for A in (6) as a function of increasing k , the change in velocity.)

A small value of the mean antieigenvalue indicates that the minimum $\cos A_t$ is small or that the turning angle is large. This indicates a larger relative change in the state vector and hence significant for representing the activity. We illustrate the use of antieigenvalues in detecting key frames through a few examples in the 1-D case. The state of the moving object is the pair $(x(t), \dot{x}(t))$. Suppose the transformation operator is given by

$$A = \begin{pmatrix} 2 & 0 \\ 0 & k \end{pmatrix}. \quad (6)$$

For differing values of k , this means that the change in the state of the object is due to a changing speed, while the position remains constant (modulo translation). Figure 1 shows the variation of the average antieigenvalue as the value of k is increased. We observe that, as expected, the average antieigenvalue varies inversely as the extent of change in the state of the moving object.

4.5 Matching Two Sequences

We compute the similarity score between two video sequences by comparing the sequences of key frames. Clearly, an activity need not be repeated with the same timing scale from one instantiation to the next and the location of key frames may change slightly. To allow for non-linear time normalization while matching, we use dynamic time warping (DTW) [16]. The similarity score is computed by traversing the warping path, which gives the correspondence of the frames in the reference and probe sequences.

To place the proposed approach in context, we compare this to the eigenvalue based methods. Various approaches in the literature have used eigenvalue-based ideas to model activities in two main ways: for pre-processing or filtering the data and for extracting the dominant characteristics for representation. The basic hypothesis in all these approaches is that the dominant characteristics of the signal are important. Also, the main characteristics are assumed to be highly structured and stationary. In such a setting, the eigenvectors capture the dominant characteristics and the eigenvalues represent the relative contribution of the eigenvectors for representation. For example, eigenfaces capture the dominant characteristics for face recognition [12]. By reconstructing the signal using the top few eigenvectors, it induces a smoothing operation on the original signal

[13]. Zhong et al. [14] use this idea for activity classification where they cluster the sequence of frames into prototype classes.

4.6 Algorithm Overview

- Pre-processing: Extract object trajectory from video and smooth it.
- For every time t , compute the state $\mathbf{x}(t) = (x(t), y(t), \dot{x}(t), \dot{y}(t))$. For computing $\dot{x}(t), \dot{y}(t)$, we use finite differencing over W frames of data.
- Compute the least squares estimate of the operator $A_t : \mathbf{x}(t_-) \rightarrow \mathbf{x}(t_+)$.
- Compute the antieigenvalues of the operator A_t using (4). Compute its mean.
- Recognition: compare the key frames detected from the average antieigenvalue sequence for the training using DTW.

5 Experiments

We demonstrate our approach to activity recognition using the MOCAP action dataset and the UCF human action dataset.

5.1 Motion Capture (MOCAP) Dataset

The MOCAP dataset available from Credo Interactive Inc. and Carnegie Mellon University consists of motion capture data of subjects performing different activities including different kinds of walking, jogging, sitting and crawling. The system tracks 53 joint locations and the tracks are stored in the bvh format. Since not all the 53 points are relevant, we use only a few of the trajectories. For example, trajectories of the different fingers and toes may not be as informative as the location of the arms, legs or hip for activities such as walking or sitting. We choose 5 regions of the 53 locations to demonstrate activity classification. This dataset allows us to test the efficacy of the proposed method in the absence of noise and errors due to low-level issues. There are 9 activities in the dataset and approximately 75 sets of observation overall. The tracks for an activity such as walking consists of multiple cycles of the activity. We divide the sequence into individual walking cycles and treat each half-cycle as an observation. Half-cycle refers to the part of the walking cycle starting from the standing pose, right (or left) leg forward, reaching the swing pose, and withdrawing the right (or left) leg to the standing pose. The number of observations is increased to 365 by treating similar trajectories of nearby locations as multiple samples, i.e., 2 locations near the abdomen are treated as multiple samples of the same location. To ensure that there is no bias due to the displacement, we use mean-subtracted trajectories for all locations.

We compute the state vector for every time instant and estimate the transformation operator A_t as described in section 4. We compute the antieigenvalue and use its mean as a signature for the activity. The antieigenvalue sequences are matched using DTW. All the activities were correctly recognized. Table 1 summarizes the activities that were the closest matches following the top match. We

Table 1. MOCAP dataset: Closest-matching activities based on comparing event probability sequences. All activities were correctly recognized. Table shows the matches following the top match.

Test activity	Match #2	Match #3
Blind-walk	Normal walk	Normal walk
Prowl-walk	Jog	Exaggerated walk
Broom	Sit	Exaggerated walk
Crawl	Broom	Sit
Exaggerated walk	Sad walk	Normal walk
Jog	Jog2	Normal walk
Sit	Sit1	Neutral
Normal walk	Normal walk	Sad Walk
Sad walk	Exaggerated walk	Normal walk

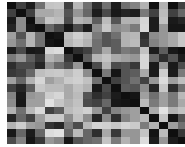


Fig. 2. Confusion matrix for activities in the MOCAP dataset

observed that the different types of walking resembled each other while the similarity scores corresponding to *sitting*, *sweeping with a broom* were significantly larger. Figure 2 shows the confusion matrix across all activities. It may not be straightforward to associate a physical meaning to the detected key frames for activities such as walking, etc. other than saying a key frame was detected at the stance when the feet are maximally apart, and so on. In the UCF action dataset described below, the key frames are more readily apparent.

5.2 UCF Human Actions Dataset

The UCF dataset consists of 60 trajectories of common activities. We divide these into 7 classes: open door, pick up, put down, close door, erase board, pour water into cup and pick up object and put down elsewhere. The hand trajectories are obtained after initialization using a skin detection technique. The resulting trajectories are smoothed out using anisotropic diffusion. A detailed description of the dataset, tracking and smoothing operations are available in [10].

The average anti-eigenvalue sequence was computed as outlined in section 4.6. The key frames were identified by finding the minima in the average anti-eigenvalues. Figure 3 shows the key frames identified for some of the activity trajectories. The dots marked along the trajectory denote the key frames detected along the trajectory. Figure 3(a) shows the key frames for opening a door. In figure 3(b), the trajectory for picking up an object from the desk and putting it on the floor shows two key frames detected, one of which is the result of a sharp change in

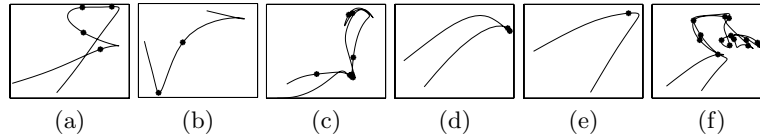


Fig. 3. Sample trajectories from UCF dataset showing key frames detected

direction and the other a gradual change. The second sharp change is not detected due to boundary effects. In the case of erasing a white board, we observe a key frame when the eraser is picked up, and several key frames at the left side of the erasing back-and-forth action of the hand (figure 3(c)). This means that each back and forth action of the hand may be considered as the past and future states separated by the key frames. Figures 3(d) and (e) show trajectories of picking up objects. They each have one key frame detected at approximately the instant the object is picked up. Figure 3(f) shows the trajectory of a random action. The lack of structure in the data is reflected by a large number of changes leading to the detection of several key frames.

Comparison with the UCF method[10]: Rao et al. treat activities as a sequence of *dynamic instants* that are defined as the points of maximum curvature along the trajectory [10]. The key frames in the proposed approach are detected based on changes in the data including changes in direction and changes in speed. The comparison of recognition rates are given in figure 4.

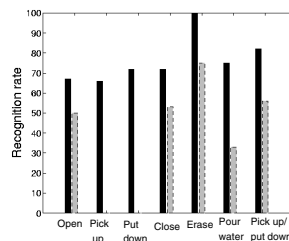


Fig. 4. UCF dataset: Comparing recognition rates. Solid black bar represents proposed method, dashed gray bar are the rates reported in [10].

6 Summary

We have presented a key frame based activity representation using the largely unexplored theory of antieigenvalues. We have argued that key frames should be related to changes in the data, rather than dominant, persistent properties. This allows a natural way to detect both subtle and sudden changes, which are often more interesting than the portions of the data that are normally observed. As part of future work, we will investigate the measures to compare antieigenvalues. It may be useful to obtain more efficient ways of calculating antieigenvalues.

References

1. Brand, M., Oliver, N., Pentland, A.: Coupled hidden markov models for complex action recognition. In: Proc. CVPR. (1996) 949–999
2. Wilson, A.D., Bobick, A.F.: Parametric hidden markov models for gesture recognition. IEEE Trans. PAMI **21** (1999) 884–900
3. Hamid, R., Huang, Y., Essa, I.: Argmode - activity recognition using graphical models. In: Proc. CVPR. Volume 4., Madison, WI, USA (2003) 38–43
4. Ghanem, N., Dementhon, D., Doermann, D., Davis, L.: Representation and recognition of events in surveillance video using petri nets. In: Proc. IEEE Workshop on Event Mining. (2004)
5. Gustafson, K.: Antieigenvalues. Linear Algebra and Appln. **208** (1994) 437–454
6. Aggarwal, J., Cai, Q.: Human motion analysis:a review. Computer Vision and Image Understanding **73** (1999) 428–440
7. Ivanov, Y.A., Bobick, A.F.: Recognition of visual activities and interactions by stochastic parsing. IEEE Trans. PAMI **23** (2000) 852–872
8. Vaswani, N., Chowdhury, A.R., Chellappa, R.: Activity recognition using the dynamics of the configuration of interacting objects. In: Proc. CVPR. (2003)
9. Nevatia, R., Zhao, T., Hongeng, S.: Hierarchical language-based representation of events in video streams. In: Proc. IEEE Workshop on Event Mining. (2003)
10. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. International J. Comput. Vision **63** (1989) 257–285
11. Parameswaran, V., Chellappa, R.: View invariants for human action recognition. In: Proc. CVPR. (2003)
12. Turk, M.A., Pentland, A.: Face recognition using eigenfaces. In: Proc. CVPR. (1991)
13. Kale, A., Rajagopalan, A.N., Sundaresan, A., Cuntoor, N., Roy-Chowdhury, A.K., Kruger, V., Chellappa, R.: Identification of humans using gait. IEEE Trans. Im. Processing. (2004) 1163–1173
14. H. Zhong, J.S., Visontai, M.: Detecting unusual activity in video. In: Proc. CVPR. (2004) 819–826
15. Gustafson, K., Seddinghin, M.: Antieigenvalue bounds. J. Math. Anal. and Appln. **143** (1989) 327–340
16. Juang, B.H.: On the hidden markov model and dynamic time warping for speech recognition - a unified view. Technical Journal **63** (1984) 1213–1243