Mapping and Manipulating Facial Dynamics

Andrew J. Aubrey, Vedran Kajić, Ivana Cingovska, Paul L. Rosin and David Marshall

Abstract— **This paper describes a novel approach to building models of temporal dynamics for facial animation with applications in performing perceptual testing of trustworthiness. A vital component of the system is a method to bring two image sequences into temporal alignment. Our approach is to project the two sequences into face space (built using shape models [1]) and apply dynamic time warping (DTW). However, the variability in the sequences causes the standard DTW algorithm to perform poorly on our data, and so we have overcome this by extending DTW in the following ways: 1) the signal magnitudes are augmented by incorporating derivatives [2], and a scheme for estimating weights in the cost function is proposed, 2) the set of sequences is used to build a graph, with nodes representing sequences and edges indicating the cost of applying the extended DTW to align pairs of sequences; better alignments between sequences can now be found by traversing the minimum cost path through the graph.**

Once all signals are aligned to a common temporal reference it is straightforward to map the temporal dynamics from one face to another. A remapped face is synthesised using the new trajectory in face space to drive an active appearance model [1]. Furthermore, the common temporal reference allows us to build a statistical model of the dynamics. This can be used to both identify dynamics of interest and also to manipulate the dynamics, e.g. to reduce or exaggerate facial dynamics.

I. INTRODUCTION

Facial animation is becoming an increasingly important area in computer graphics as various media (movies, television, video games) look to produce more and better computer generated animation. Much of the early work focused on generating a face that looked as realistic as possible, but recent emphasis is directed at making these faces move in a realistic manner.

Due to recent advances in computer graphics it is becoming increasingly common for psychologists to use computer generated stimuli to conduct perceptual experiments since they can be more easily controlled and manipulated than direct video footage.

The goal of this work is to enable psychologists to analyse the trustworthiness of a speaker as determined by the perception of their facial dynamics. For this a system is required that allows the dynamics of a speaker to be modified. In our work we require a system that allows automatic manipulation of a speaker's dynamics. We also require a system that both retains the shape and appearance characteristics of a subject, and also modifies the speech of a subject so that after modification the audio and video remain synchronised.

The necessity for life-like animation requires the use of performance driven animation [3]. Current techniques do not easily allow manipulation of the timing of facial dynamics, nor can they be easily used to synchronise the audio and video data.

Previous work on performance driven animation has included work by Vlasic et al [4], Zhang et al [5] and Chang and Ezzat [6]. Statistical models such as the Active Appearance Model (AAM) are widely used due to their ability to encode a high level of information using only several parameters. Cosker et al [7] describe how AAMs can be used to transfer expressions between AAMs and morph-target based models by identifying key regions of the face and manipulating these regions independently of one another. Theobald et al [8] and De la Hunty et al [9] both describe how AAMs can be used for real time expression transfer. In [8] the authors drive the AAM mean of one model with the modes of variation from another model. While this produces very realistic animations, it does not allow any control over the dynamics. The method in [9] is an extension of an early version of the method in [8] and suffers the same drawback.

This paper addresses those issues using a framework based on dynamic time warping (DTW) and shape and appearance models. Firstly, we explain our novel approach for extending existing DTW methods. We then, using shape models to capture the face dynamics, show how the trajectories of shape models from a reference video and target video are aligned using DTW. The aligned shape trajectories are used to re-synthesise a new video sequence with the desired dynamics. Our method differs from previous work as it allows us to manipulate the temporal dynamics in a more sophisticated manner. As well as simply mapping temporal dynamics, we can exaggerate them, compute mean dynamics, identify typical and atypical dynamics etc. The alignment found using DTW can also be used to warp the audio to ensure synchrony with the video. It should also be pointed out that none of the methods cited above allow simultaneous warping of the speech as our method does.

II. DYNAMIC TIME WARPING

Dynamic time warping (DTW) is a method to find the optimal non-linear alignment between two time series signals. It achieves this by warping (compressing or expanding) the time base of two signals. The classic DTW algorithm is as

This work was supported by One Wales Research Institute of Visual Computing (RIVIC)

A. Aubrey, P. Rosin and D. Marshall are with the Visual Computing Group, School of Computer Science and Informatics, Cardiff University, Wales, UK. {A.J.Aubrey, Paul.Rosin, Dave.Marshall} @cs.cardiff.ac.uk

V. Kajić is with the School of Optometry and Vision Sciences, Cardiff University, Wales, UK. KajicV1@cardiff.ac.uk

I. Cingovska took part in this work while at the School of Computer Science and Informatics, Cardiff University as part of the International Association for the Exchange of Students for Technical Experience (IAESTE) program.

follows [10]. Given two sequences s_1 and s_2 of length M and N respectively, a warping path W can be found that defines a mapping (alignment) between s_1 and s_2. To align the two sequences an M by N distance matrix must first be constructed where the (i^{th}, j^{th}) element of the matrix contains the distance between the i^{th} point in s_1 and the j^{th} point in s_2. Typically the Euclidean distance is used.

The warping path W is a contiguous set of matrix elements defined as:

$$W = w_1, w_2, \ldots, w_k, \ldots, w_K \quad \max(M,N) \leq K \leq n+m-1 \tag{1}$$

where K is the length of the warp path and $w_k = (i,j)_k$. The aim of DTW is to find the path W through the distance matrix with the least cost. To this end the following constraints are employed:

- Monotonicity: the indices i and j are monotonically spaced in time.
- Continuity: if $w_k = (i,j)_k$ then $w_{k+1} = (i',j')_{k+1}$ where $i'_{k+1} - i_k \leq 1$ and $j'_{k+1} - j_k \leq 1$. This limits (in time) the distance between the aligned points.
- Boundary: The warping path must start and finish at diagonally opposite corners of the distance matrix, $w_1 = (1,1)$ and $w_K = (M,N)$.

There are many warping paths that satisfy the above constraints, but the path of interest has the minimum warping cost:

$$C_{DTW} = \sum_{k=1}^{K} D(i_k, j_k) \tag{2}$$

Using dynamic programming [10], the path with the minimum cost can be found by building an accumulated cost matrix C which is defined as follows:

$$C(i,j) = D(i,j) + \min \left\{ \begin{array}{l} C(i,j-1) \\ C(i-1,j) \\ C(i-1,j-1) \end{array} \right\} \tag{3}$$

The above equation controls the step size of DTW, and modification of (3) produces a different alignment.

While the classic DTW algorithm performs well for signals that are similar except for local accelerations and decelerations in the time axis, Keogh and Pazzani [11] describe how it is prone to failure when the two sequences differ in amplitude. Other typical errors are when a small portion of one signal maps to a large portion of the other signal or when a rising edge is incorrectly mapped to a falling edge because they are close in time.

A. DTW variation

To improve the accuracy of the warping path there have been many proposed variations to the classic algorithm; several are discussed by Muller [10]. Typically, one or more of the following variations are used.

Globally constraining the domain of the warp path can speed up the DTW computation. Also, it will limit the degree of certain failures but may not prevent them. The Sakoe-Chiba band [12] and the Itakura parallelogram [13] are the two most widely used global constraints. A weighting factor can be introduced to (3) to bias the direction of the warp path. This can prevent the many to one mapping situation, but a bad choice of weights can force an incorrect path to be taken. The step pattern (3) can also be modified and is one of the more popular methods of varying the DTW algorithm due to the flexibility it allows.

The above DTW variations are not always effective, and it may be necessary to incorporate additional information. To this end Keogh and Pazzani [11] proposed the Derivative DTW (DDTW) algorithm. They calculate the first derivative of the sequences which provides shape information of the signals. Their experiments show that the DTW and DDTW have similar performance with signal distortion confined to the time axis. However, DTW performs poorly when there is a distortion in the amplitude of the signals, whereas the DDTW algorithm performs significantly better.

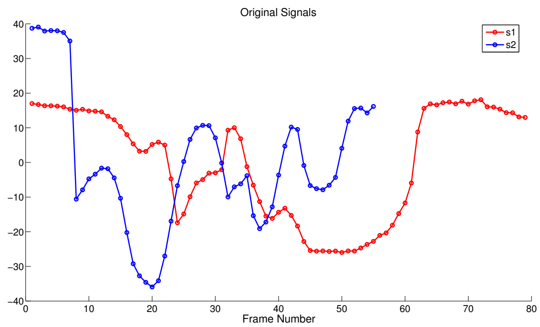
According to Benedikt et al [2], DDTW is highly sensitive to noise, so they expand on the work in [11] by proposing a Weighted Derivative DTW (WDTW) algorithm. The WDTW is a weighted combination of classic DTW and DDTW, and they also use the local second derivative of the signals. The use of the second derivative allows information on acceleration to be included in the calculation of the alignment. Their experiments show the WDTW produces a better alignment than relying on just the signal magnitudes or derivatives alone. However, the values of the weights were carefully chosen manually as the magnitudes of the derivatives and the original signal are not in the same range.

Figure 1 shows the results of applying the classic DTW algorithm and the WDTW algorithm to a pair of signals. The original signals are shown in Fig 1(a), and they are taken from our data set to highlight why existing methods may not give the desired results. Fig 1(b) is the result of applying the classic DTW algorithm to the signals, and Fig 1(c) is the result of applying the WDTW algorithm. It is clear that both methods have to some degree aligned the two signals, however, this has been achieved at the cost of creating periods where there is no amplitude variation. This repeating of a value for several samples may not be an issue for some applications. However, as we want to resynthesise video with realistic dynamics this a problem as the face will appear frozen for several frames which appears unrealistic.

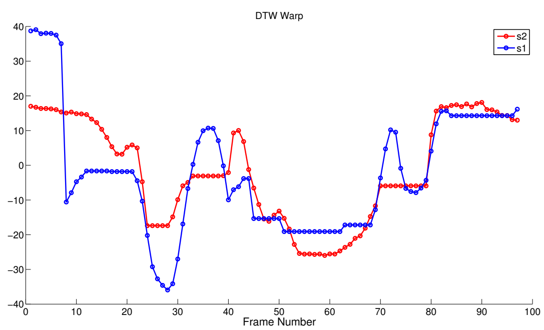
In our work we use the WDTW algorithm as the basis of our DTW framework but contribute the following novel extensions. The first is a simple method for automatically calculating the weight values. Secondly, we describe a graph based DTW algorithm that, given a set of signals to be aligned, can find the best mapping between two signals by using intermediate steps of first mapping to other signals in the set and then to the target signal.

B. Automatic Weight Selection

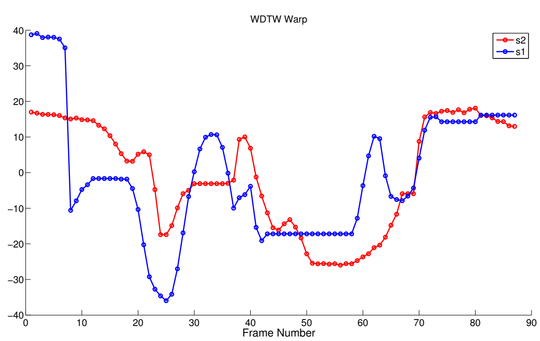
Weighted derivative DTW (WDTW) is simply a weighted combination of the DTW and DDTW methods. It is also possible to include higher derivatives, but as in [2] we limit ourselves to using first and second derivatives only. Mathematically this is expressed as:



(a) Original signals



(b) Signals warped using DTW



(c) Signals warped using WDTW

Fig. 1. Plots of, (a) two signals from our dataset, (b) the signals warped using the classic DTW algorithm and (c) the signals warped using the WDTW from [2].

$$C_W = w_0 \times C_0 + w_1 \times C_1 + w_2 \times C_2 + \ldots + w_q \times C_q \qquad (4)$$

where C_0 is the distance matrix found using the classic DTW method, C_q is the distance matrix found using the q^{th} derivative and w_q is the weight associated with the q^{th} derivative.

Each pair of signals to be warped requires the weight values for each of the distance matrices in (4), the wrong values of these weights can lead to a bad alignment by including too much or too little of the derivatives. Without loss of generality, the weight w_0 is chosen to be 1.

To find values for w_1 and w_2 the following is used. Given two signals to warp s_1 and s_2 of length n and m respectively, treat each signal independently, in this case work with s_1 first:

1) First calculate s_1' and s_1'', the first and second order derivatives respectively.
2) Randomly permute s_1, s_1' and s_1'' to obtain s_{1P}, s_{1P}' and s_{1P}''.
3) Find w_1 and w_2 using the following ratios:

$$w_1 = \frac{\sum(s_{1P} - s_1)^2}{\sum(s_{1P}' - s_1')^2} \qquad (5)$$

and

$$w_2 = \frac{\sum(s_{1P} - s_1)^2}{\sum(s_{1P}'' - s_1'')^2} \qquad (6)$$

This results in a set of weights for sequence s_1, $\{w_{s_1 0}, w_{s_1 1}, w_{s_1 2}\}$. The same procedure can be applied to sequence s_2 to obtain the set of weights $\{w_{s_2 0}, w_{s_2 1}, w_{s_2 2}\}$. Using either of these sets of weights risks biasing the WDTW algorithm in favour of one of the signals. To counter this we find the mean set of weights:

$$w_0 = w_{s_1 0} = w_{s_2 0} = 1 \qquad (7)$$
$$w_1 = E(w_{s_1 1}, w_{s_2 1}) \qquad (8)$$
$$w_2 = E(w_{s_1 2}, w_{s_2 2}) \qquad (9)$$

where E is the expectation value.

The above is easily extended for multi-dimensional signals, by summing the numerator and denominator of (5) and (6) over each dimension. For example, for (5) we would have:

$$w_1 = \frac{\sum \sum_{r=1}^{R} (s_{1rP} - s_{1r})^2}{\sum \sum_{r=1}^{R} (s_{1rP}' - s_{1r}')^2} \qquad (10)$$

where R is the number of dimensions. A similar expression is obtained for (6).

III. Graph based DTW (GWDTW)

The signals we wish to warp are not guaranteed to have a high correlation in shape and will have different amplitudes, thus it is quite possible for the previously mentioned DTW methods to fail to align the signals correctly. As mentioned in [11] the amplitude difference will have a significant impact on the "correctness" of the warp. If however there exists a set of sequences $S = \{s_1, s_2, \ldots, s_Y\}$, it may be possible to correctly warp s_1 to s_x by first warping via s_y where $1 \leq x, y \leq Y$.

If the signals are treated as nodes and the warping between two signals as edges the set of signals and their respective warping can be treated as a graph (see Figure 2). This may be viewed as a complete graph as all nodes are connected to each other. It is therefore possible to align any two signals in the set. Each pair of aligned signals will have an associated error (the mismatch between the resulting aligned signals) and this is used as a weight value for the graph edges. Figure 2 is an example graph containing four signals represented as nodes and the associated edge weights given by the signal mismatch error values. The total error of the warping path W is used as the edge weights for pairs of signals. i.e $e = w_K$.

To find the best warping path between two signals it is simply a case of finding the shortest path through the graph. As the error values are non negative, Dijkstra's shortest path algorithm [14] can be applied. The path with the smallest total error e is chosen as the warping path. Typically in our small dataset we find path lengths of two or three.

IV. Face Appearance Model

To transfer the facial dynamics from one face sequence to another we first capture the shape and appearance of each face using the standard active appearance model (AAM) developed by Cootes et al [1]. To obtain the shape information, salient feature points on the face must be chosen and tracked through each frame in the video sequence. This is a highly time consuming process if done manually, therefore, an automatic method is sought. It is possible to use make up or markers placed on the subject's face prior to capture to track facial features, however, we wish to use the captured textures for re-synthesis, so it is preferable to avoid post processing the images to remove unwanted make up and markers. There are methods that can automatically track landmarks, but while they are adequate for rigid motion, they are still not fully reliable for highly non-rigid movement, particularly in tracking the lips as the lips are highly deformable.

A. Landmarking a video sequence

To automatically landmark a sequence of images it is typically required to manually landmark an image in the sequence (usually the first frame) and then automatically track these landmarks through the image sequence. However, this can prove to be unreliable for the lips, especially the outer and inner edge of the lower lip due to the nature and amount of deformation they undergo whilst speaking or performing facial expressions. To overcome this we use a semi-automatic framework based around groupwise registration of the frames in each sequence. To landmark a video sequence the following steps are taken. First the groupwise registration algorithm of Sidorov et al [15] is applied which performs non-rigid alignment of every image in a sequence. The result of this is a mean image, where the sharper the mean image is the better the registration and a set of deformation maps $\{\mathscr{D}_n; n = 1, \ldots, N\}$, where N is the length of the sequence. The deformation map describes the pixel mapping from a single image to the mean. Therefore, we only need to landmark a single frame (the mean image) and use the set of deformation maps to reverse map the landmarks onto the original images. Figure 3 contains example mean images from groupwise registration of one sequence in our dataset.

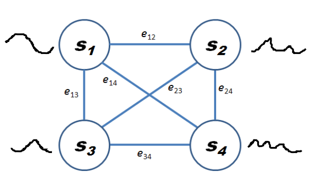


Fig. 2. Example graph, where nodes $s_1, \ldots, s_4$ are the signals and the edges e_{ij} are the error values obtained from warping signals i and j.

Fig. 3. Mean images from the first (left) and last (right) iteration of groupwise registration from one sequence from the dataset

The left image is the mean image from the first groupwise iteration. The blurring around the edge of the face and the mouth region highlights the amount of respective rigid and non-rigid motion in the sequence. The right image is the final groupwise output. The image is visibly sharper, hence, landmarks placed in this image will be mapped to the correct positions in the original set of images with little to no manual correction.

While we have found this technique significantly reduces the manual placement of landmarks, we find that some frames occasionally need to be manually corrected. This is because groupwise registration requires the deformation in the image set to be diffeomorphic. When working with faces, blinking or mouth opening will violate this rule. The sequences we use have the subjects speaking, and through all sequences we find that only several frames require manual correction (where the sequence length varies between 60 and 100 frames), so there is still a significant saving in manual landmarking.

B. Active Appearance Models

Active appearance models (AAMs) [1] are a joint statistical model of shape and colour values (texture), where a single appearance parameter defines a corresponding texture and shape vector. The shape model is obtained from the Cartesian coordinates of the landmarks. For a single image, $\Phi(j)$ is the vector of landmark coordinates, and the collection of vectors $\{\Phi(j)\}, 1 \leq j \leq N$ describe the shape variation over the set of images N. As we wish to re-synthesise colour video, the textures from within these shapes are described by the texture model Γ. By applying principal component analysis (PCA) to the shape and texture data separately, the statistical shape and texture models are obtained. The shape model can be expressed as:

$$\Phi = \overline{\Phi} + \mathbf{P}_\Phi \mathbf{b}_\Phi \tag{11}$$

and the texture model is expressed in a similar format:

$$\Gamma = \overline{\Gamma} + \mathbf{P}_\Gamma \mathbf{b}_\Gamma \tag{12}$$

where $\overline{\Phi}$ and $\overline{\Gamma}$ are the mean shape and texture vectors, $\mathbf{P}_\Phi$ and $\mathbf{P}_\Gamma$ are matrices formed from eigenvectors, and $\mathbf{b}_\Phi$ and $\mathbf{b}_\Gamma$ are the corresponding eigenvalues, typically referred to as shape and texture parameters respectively. Varying the shape and texture parameters $\mathbf{b}_\Phi$ and $\mathbf{b}_\Gamma$ allows any of the original images to be approximated, or indeed new images to be synthesised. The advantage of using shape and appearance models is that the shape and texture parameters allow the high dimensional space within which the images lie to be represented by only a few parameters.

To construct a full AAM we combine $\mathbf{P}_\Phi$ and $\mathbf{P}_\Gamma$ and apply PCA once more to obtain a combined shape and appearance parameter. The number of shape and texture parameters retained can be chosen by either retaining the several highest shape and texture values, or by retaining a percentage of their total energy.

To align the dynamics of a reference and target sequence we first align the trajectories of the shape parameters of the reference to the target. A new sequence is resynthesised by using the shape parameters of the aligned reference sequence to correctly manipulate the appearance model of the reference sequence.

V. Data Collection and Experiments

The data used in the following experiments was recorded using an interlaced digital video camera at 25fps and the subjects were filmed under the same lighting conditions. Each subject was asked to recite the sentence “Once upon a time”, and to also elongate one word in the sentence. We recorded seven sequences and denote these as $s_1, s_2, \ldots, s_7$.

To build the shape and appearance models we used the semi-automatic procedure described in Section IV to annotate each frame with 49 landmarks. For each subject we retained 95% of the variance in the shape model and for the appearance model we retained 99% of the total variance.

The purpose of this work is to transfer the dynamics of one subject’s facial actions onto another subject who is performing a similar set of facial actions. The idea being that one subject inherits the timing of the dynamics of the other subject. Each of the shape parameters will have a trajectory through the image sequence, so for actors performing similar actions there should be similarities between the shape trajectories. This is shown in Figure 4, which depicts the trajectories of the principal shape component from the actors reciting the same phrase.

It should be noted that even though the shape parameters are multi-dimensional, it is trivial to adapt the classic DTW, its variants and our GWDTW technique to be used with such signals.

A. Experiments

One issue with using shape models is ensuring the principal components (modes of variation) of different subjects are correlated. By this we mean that the first t modes correspond to the same features in all subjects. It would be useless to have the first 2 modes of subject one corresponding to vertical and horizontal motion of lips and the first 2 modes of subject two corresponding to eye motion. To ensure correspondence we manually inspect the first t modes of each subject to ensure they correspond.

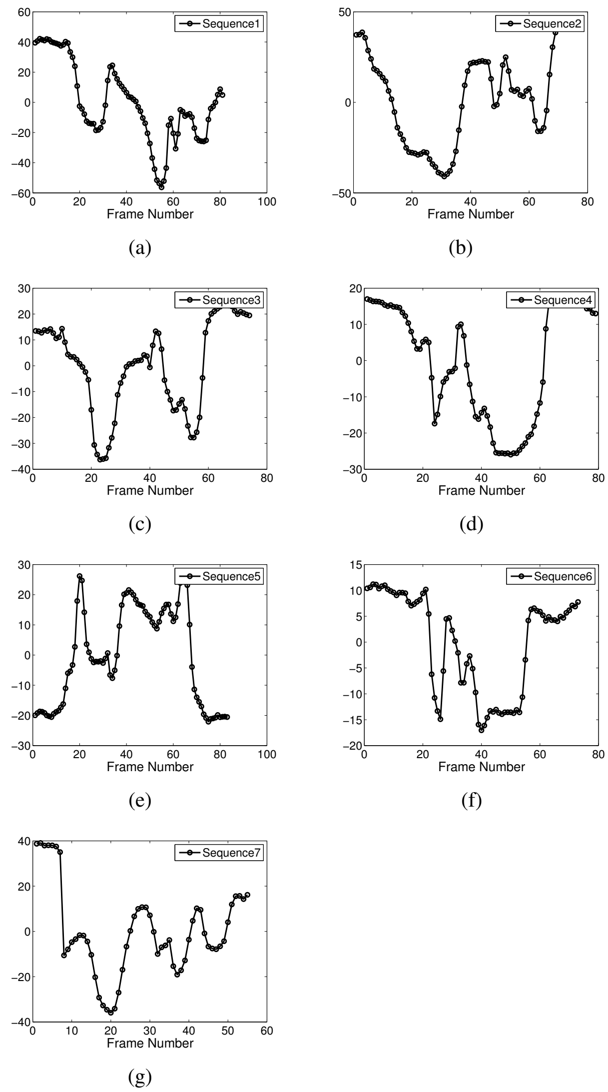


Fig. 4. Plots of the first mode of variation of the shape model from each of the subjects in the dataset. The plots (a) to (g) are the sequences 1 to 7 respectively.

In the following experiments we do not show the results of using the classic DTW algorithm to align the sequences, but only consider the WDTW and our proposed GWDTW method. To test the accuracy of the method we start by first comparing the result of warping sequence seven (s_7) to sequence three (s_3) with the weights $w_0, w_1, w_2 = 1$. Figures 5 and 6 show the results of aligning the signals using the WDTW and GWDTW methods respectively. In Figure 5 s_3 is represented by the blue curve and s_7 by the red. It can be seen that until around frame 30, the WDTW method appears to have aligned the signals correctly, but from thereon the trajectories are not aligned.

Using our GWDTW method we find that there is not a satisfactory direct warp from s_7 to s_3. Instead s_7 is first warped to s_4 and then warped to s_3. Figure 6 depicts the result of warping s_7 to s_4 with the blue curve and the result of warping from here to s_3 with the red curve, which is in

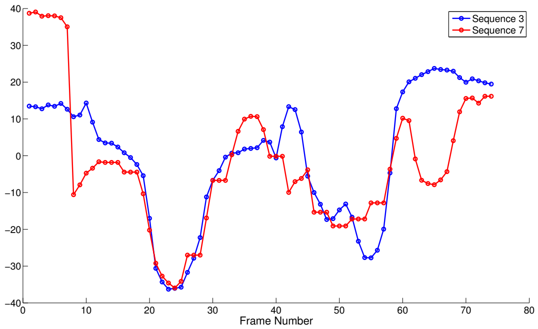


Fig. 5. Sequence s_7 warped to s_3 using WDTW with the weights fixed at $w_0 = w_1 = w_2 = 1$.

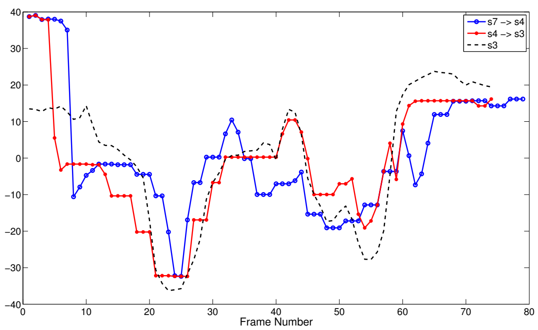


Fig. 6. s_7 warped to s_3 using our GWDTW method with the weights fixed at $w_0 = w_1 = w_2 = 1$.

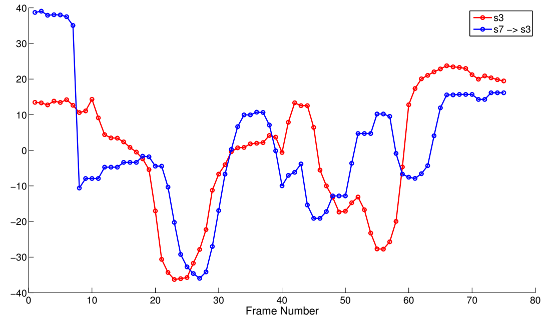


Fig. 7. s_7 warped to s_3 using WDTW with the weights w_1 and w_2 calculated automatically.

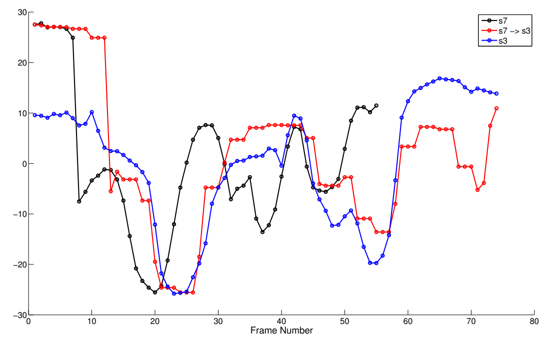


Fig. 8. s_7 warped to s_3 using our GWDTW method with the weights w_1 and w_2 calculated automatically.

fact s_7 but with the time base of s_3. The dashed black curve is s_3 shown for comparison with the warped version of s_7. It is clear from these results that in this instance a better warp is achieved by using the intermediate step of warping via s_4.

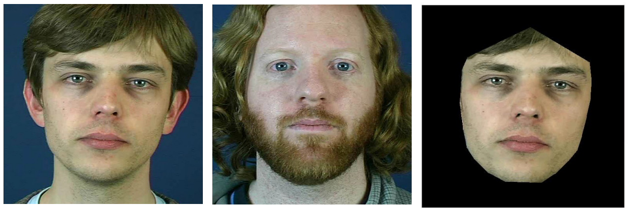
To test the automatic weight selection scheme described in Section II-B we automatically calculate the weights as described and then apply them to the WDTW and GWDTW methods. The results are shown in Figures 7 and 8 respectively. We find that the GWDTW method still provides a better alignment than the WDTW. Also it should be noted that the path GWDTW takes through the graph has now been altered (because of different weights) and now maps directly from s_7 to s_3.

Figure 9 contains example frames from two original sequences, s_7 and s_2, and the resulting resynthesised sequence when s_7 is warped to s_2. The actors are saying the phrase "Once upon a time", however, s_2 is elongating the word "once" while s_7 is not emphasising any words. Because of this the sequences are of different lengths, s_7 has 55 frames and s_2 has 68 frames. Applying our GWDTW method to the shape parameters of s_7 and s_2 allows us to produce a resynthesised sequence for s_7 which now has the same temporal dynamics as s_2 and is also of the same length. Frame 15 (Figure 9(b)) indicates the opening of the mouth for the word "once" in both, but by frame 33 (Figure 9(c)) s_7 is beginning to pronounce the word "upon". In the resynthesised sequence, the dynamics of s_7 have now been aligned with s_2.

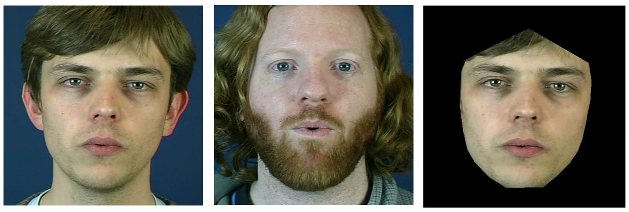
VI. Conclusion

In this paper we have described a novel approach for building models of temporal dynamics. Our application is to generate stimuli, consisting of videos of talking heads, for experiments to investigate how the perceived trustworthiness of individuals is affected by alterations to their temporal dynamics. Image sequences were represented as trajectories in face space, and aligned with GWDTW, our improved version of dynamic time warping. GWDTW was shown to perform better than the standard DTW, and could successfully map the dynamics of one speaker onto another.

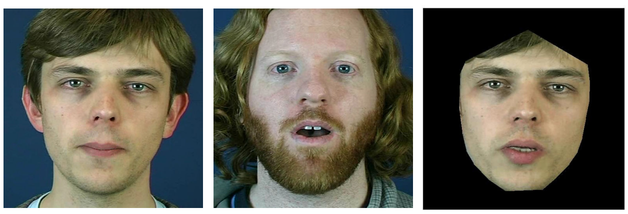
Current work is looking at the warping of audio, using the mappings determined from the alignment in face space. Early work has shown promising results. Our method will then be used to generate stimuli for the initial perceptual experiments. Finally, we plan to build and exploit PCA models of temporal dynamics for further perceptual experiments.



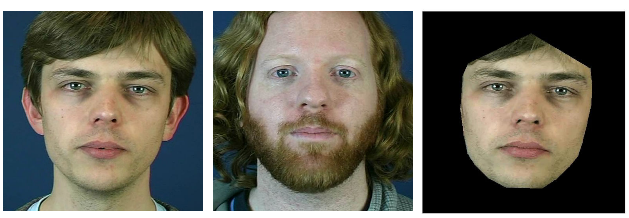
(a) Frame 1



(b) Frame 15



(c) Frame 33



(d) Final frame

Fig. 9. Frames from two original sequences and the resulting resynthesised sequence. Left column s_7, Middle column s_2, Right column the resynthesised sequence of s_7 warped to s_2.

REFERENCES

[1] T. Cootes, G. Edwards, and C. Taylor., "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[2] L. Benedikt, D. Cosker, P. L. Rosin, and D. Marshall, "Assessing the uniqueness and permanence of facial actions for use in biometric applications," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 40, no. 3, pp. 449–460, 2010.

[3] L. Williams, "Performance-driven facial animation," in *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, 1990, pp. 235–242.

[4] D. Vlasic, M. Brand, H. Pfister, and J. Popović, "Face transfer with multilinear models," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 426–433, 2005.

[5] Q. Zhang, Z. Liu, B. Guo, D. Terzopoulos, and H. Shum, "Geometry-driven photorealistic facial expression synthesis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 1, pp. 48–60, 2006.

[6] Y. Chang and T. Ezzat, "Transferable videorealistic speech animation," in *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2005, pp. 143–151.

[7] D. Cosker, R. Borkett, D. Marshall, and P. L. Rosin., "Towards automatic performance-driven animation between multiple types of facial model," *Computer Vision, IET*, vol. 2, no. 3, pp. 129 –141, Sept 2008.

[8] B. Theobald, I. Matthews, M. Mangini, J. Spies, T. Brick, Z. Ambadar, J. Cohn, and S. Boker., "Mapping and manipulating facial expression," *Journal of Language and Speech*, vol. 52(2/3), pp. 369–386, 2009.

[9] M. de la Hunty, A. Asthana, and R. Goecke., "Linear facial expression transfer with active appearance models," in *2010 International Conference on Pattern Recognition*, 2010.

[10] M. Muller, *Information Retrieval for Music and Motion*. Springer, 2007.

[11] E. Keogh and M. Pazzani, "Derivative dynamic time warping," in *First SIAM International Conference on Data Mining*, 2001.

[12] H. Sakoe and S. Chiba., "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26(1), pp. 43–49, 1978.

[13] F. Itakura., "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 23(1), pp. 67–72, 1975.

[14] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Math*, vol. 1, pp. 269–271, 1959.

[15] K. Sidorov, S. Richmond, and D. Marshall., "An efficient stochastic approach to groupwise non-rigid image registration," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, Miami, USA, June 2009, pp. 2208–2213.