Robust Lip-Tracking using Rigid Flocks of Selected Linear Predictors

Eng-Jon Ong and Richard Bowden
Centre for Vision, Speech and Signal Processing,
University of Surrey,
Guildford GU27XH, Surrey, UK
e.ong,r.bowden@surrey.ac.uk

Abstract

This paper proposes a learnt data-driven *approach to the accurate, real-time tracking of lip shapes using only intensity information i.e. grey-scale images. This has the advantage that constraints such as* a-priori *shape models or temporal models for dynamics are not required or used. Tracking the lip shape is simply the independent tracking of a set of points that lie on the lip's contour. This allows us to cope with different lip shapes that were not present in the training data and performs as well as other approaches that have pre-learnt shape models such as the AAM. Tracking is achived via linear predictors, where each linear predictor essentially linearly maps sparse template difference vectors to tracked feature position displacements. Multiple linear predictors are grouped into a rigid flock to obtain increased robustness. To achieve accurate tracking, two approaches are proposed for selecting relevant sets of LPs within each flock. Analysis of the selection results show that the LPs selected for tracking a feature point choose areas that are strongly correlated with that of the tracked target and that these areas are not necessarily the region around the feature point as is commonly assumed in LK based approaches. Experimental results also show that this method is comparable in performance to that of AAMs, despite being much simpler, both in the training and tracking phases, without any* a priori *shape information and with minimal training examples.*

1. Introduction

The problem of automatic lip shape tracking is non-trivial since the lip itself is a highly deformable object, and as such, can assume a large variety of shapes. The teeth and tongue add additional complications, since their visibility is highly dependent on the shape of the mouth, and their appearance can cause parts of the mouth region's texture to change dramatically. Consequently, this makes tracking, especially the inner lip shape, very difficult. Problems in lip shape tracking are further compounded by variations in lip colour, lighting and skin colour across different individuals. There exists a number of different methods for lip tracking. One of the most popular methods is the model-based approach using active contours for tracking the both the inner and outer lip contour [2, 13]. This was then improved upon in [1] by coupling this technique with 2D templates. In [12], temporal constraints were also included to improve on tracking. Colour and a markov random field model has also been used to initially segment the lips, before obtaining the final lip shape using active contours [7]. Other methods include active appearance models (AAM) [3] for tracking lip shapes [9], where it is also possible to extract relevant training data from audio [4].

In this paper we proposes a learnt person-specific but importantly, a *data-driven* approach to achieve accurate and real-time tracking of *lip shapes* (both inner and outer lip shapes) using only intensity information. Additional constraints such as *a-priori* shape models or temporal models of dynamics are neither required nor used. Through accurate tracking of independant features, tracking the lip shape is possible by tracking the set of feature points that lie on the lip's contour. This allows it to cope with different lip shapes that were not present in the training data.

To track features, we employ the method of linear predictors as described in Section 2. Each linear predictor essentially provides a mapping from pixel-level information to the displacement vector of a tracked feature (e.g. Lip shape point). A single linear predictor is seldom good enough to provide robust and accurate tracking for points on the mouth. To overcome this, rigid flocks of linear predictors, described in Section 4 are used. Here, a set of linear predictors are grouped together to track a single feature point. The placement and support for a linear predictor is normally randomly placed around the feature. However, given the radical texture changes that occur in and around the mouth during normal speech it is not obvious that this is an appropriate stratergy. To address this, we propose two approaches to

978-1-4244-2154-1/08/$25.00

selecting relevant linear predictors for giving accurate and robust tracking in Section 4. Interstingly, the selection proceedure makes intuitive selctions that are in stark contrast to random selection and results in reliable feature point tracking even for notoriously difficult aspects such as the inner lip shape. We demonstrate the use of the proposed methods for accurately tracking both inner and outer lip shapes and make comparison to AAMs in Section 5 which employ heavy dependancies between features in the form of shape priors. The tracking performance of linear predictor flocks built using the proposed methods is also assessed and analysed. Finally, we conclude in Section 6.

2. Linear Predictors

A Linear Predictor (LP) forms the central component of the proposed tracking mechanism. A LP is responsible for tracking a particular visual feature by means of a linear mapping from an input space of sparse support pixels to a displacement vector space, the motion of the feature point. Recently, linear predictors have been used to efficiently track objects [8]. Along similar lines, Relevance Vector Machines (RVMs) were used to provide displacement predictions [11], however, with the trade-off of additional complexity. More recently, Bayesian Mixtures of Experts coupled with RVMs for more accurate tracking performance have been proposed[10].

The predictor is defined as a set of four components, $L = \{c, H, V, S\}$, where c is a 2D reference point defining the location of the feature, H is the linear mapping to a displacement vector, and V is a V-dimensional vector of base support pixel values. V forms a sparse template for the visual appearance of the feature point. In order to obtain V, $S = s_i{}_{i=1}^{|S|}$ is defined and used, where s_i is the offset relative to c defining the location of a support pixel. The offset positions s_i are obtained as random offsets within a specified radius from the origin. In this paper, the radius was heuristically chosen to be 30. An illustration of a linear predictor can be seen in Figure 1. To use an LP to predict the displacement of its tracked feature (i.e. reference point c), suppose we are given the image $I = I_{ij}^{H,W}$ of dimensions HxW as input. We firstly obtain the difference between the base support pixel values and those from the current image:

$$\delta p = (V_i - V_i^I)_{i=1}^{|S|} \tag{1}$$

where $V_i^I = I_{c+s_i}$ is the pixel value at position $c + s_i$ in the image. The displacement of c, t is then:

$$t = H\delta p \tag{2}$$

2.1. Learning the Linear Mapping

The linear mapping (H) of an LP is learnt using least squares optimisation. As a result, from Eq. 2, we need a set of training data in the form of support pixel differences (δp) and displacement vector pairs. To achieve this, a number of training examples can be synthesised from each single training image.

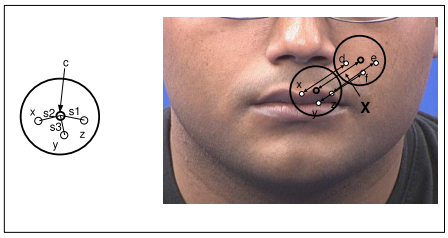

Figure 1. Illustration of a linear predictor(LP). Each LP has a reference point c. Within an area around c, called the support region a set of randomly sampled support pixels $(x, y, z)$with their offsets from c: $s1, s2, s3$. Also shown is the synthesis of training data. X is the artificial translation of c. The corresponding support pixel difference vector is $\delta p = (xd, yf, ze)$.

It is assumed that in each training image, the location of the tracked feature point is ground-truthed, which is also the value of the LP reference point (c). This allows us to extract the base support pixel values (V). Following this, it is possible to synthesise a number of random displacements from c. Along with these displacements, we can also obtain their respective δp vectors by initially translating c by the displacement, obtaining the support pixel values at that position and calculating its difference from the base support pixel values. It should be noted that if we have a set of ground-truthed training images, it is possible to repeat the above for multiple images, thus gathering a wider range of training examples for learning the linear map H. Here, the base support pixel values are fixed as those from the first training image.

The generated examples can then be compiled into the following matrices: T and δP, where T is a $2xN_T$ matrix, of which each column is a displacement vector. Similarly, δP is a $|S|xN_T$ matrix, where each column is the displacement vector's corresponding support pixel difference vector. Using least squares, H can now be obtained as follows:

$$H = T\delta P^{+} = T\delta P^{T}(\delta P \delta P^{T})^{-1} \tag{3}$$

where δP^{+} is the pseudo-inverse of δP.

3. Rigid Flocks of LPs

Using a single linear predictor to determine the displacement of a feature point is insufficient. This is because a single linear mapping between the support pixel difference values to the displacement space is seldom robust to noise, illumination changes and other image warps that may occur on the feature point and its surroundings. This problem

can be addressed by grouping multiple linear predictors together into a *rigid flock* of LPs. In previous work [6, 5], a flock tends to be a loose collection of features or trackers that must lie within an area surrounding a reference point (e.g. feature mean position). Whilst each flock member may move somewhat independently within this area, in general they agree on the general direction of the tracked target. This agreement often cancels out noise present in the individual tracker predictions. However, these trackers are still free to move within an area relative to some reference point (e.g. position of the tracked feature). In our case of a rigid flock, trackers are always fixed to the original offset away from the reference point. Formally, a rigid flock

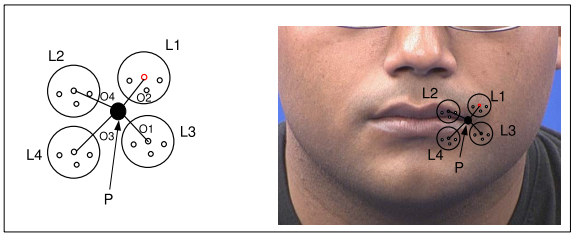

Figure 2. Illustration of a rigid flock of linear predictors, whose position is given by reference point, P. The member LPs are $(L1, L2, L3, L4)$ each with a rigid offset from P: $O1, O2, O3, O4$.

of LPs consists of the following components: a reference point P^F, a set containing L^F number of linear predictors ($L^F = \{L_{f=1}^{|L^F|}\}$) and the 2x$|L^F|$ matrix of linear predictor offsets (O^F) from P^F (Figure 2). We define the displacement predictions obtained from Eq. 2, of each of the member linear sets as $\{t_{f=1}^{|L^F|}\}$. The arrangement allows us to have a reference point offset from the centre of the LPs in the flock, but still be guided by its predictions. This is in contrast to, for example, taking the reference point as the mean of the member LPs, where it is forced to lie in the centre of the flock members. As we will see in Section 5, a rigid flock of LPs combined with carefully selected LPs is critical to increasing the tracking accuracy and fundamentally different to related approaches (e.g [6, 5, 3, 1]).

A naive method of computing the final tracked feature displacement is to use the mean prediction of all LP members in a rigid flock. However, instead of taking the mean of all the members in the flock, we can further increase the prediction robustness by introducing the ability to combine the predictions of different LPs in the flock for different displacement components. In Section 5, we see how different prediction components often require different sets of LPs for providing better tracking accuracy. To achieve this, we define two index sets: X^F and Y^F, each set containing indices of LPs to use in computing the x and y component of the feature point displacement respectively. The rigid flock prediction (x^F, y^F) is defined as:

$$x^F = (1/|X^F|) \sum_{f=1}^{|X^F|} t_{X_f^F}(1) \quad (4)$$

$$y^F = (1/|Y^F|) \sum_{f=1}^{|Y^F|} t_{Y_f^F}(2) \quad (5)$$

where $t_i(1)$ and $t_i(2)$ are respectively the x and y components of the individual LP predicted displacement vectors. This is then used to update the position of the rigid flock reference point as: $P^F = P^F + (x^F, y^F)$.

4. Automatic Selection of LPs

Having defined a rigid flock of LPs in the previous section, we are now faced with the issue of deciding how many and more crucially *where* to place the member LPs. To start, a predetermined number of LPs are randomly scattered within an area around the rigid flock's reference point. It is then possible, with all these linear predictors to use Eq. 5 to predict the displacement of the reference point, given a new input image. This is equivalent to settting $|X^F|$ and $|Y^F|$ to $|L^F|$, each set being $\{1, ..., |L^F|\}$. However, this can be suboptimal, since there may exist many LPs in the flock which will give displacement predictions. The problem is also more subtle, since we often find certain LPs providing wrong predictions for one displacement component (e.g the x-component), whilst simultaneously providing very accurate predictions in other components (e.g y-component). Crucially, we are faced with the problem of identifying meaningful context useful for tracking a particular feature point.

To address this issue, this section proposes two methods for rating and selecting *separate* sets of LPs for accurate and robust predictions. More specifically, these methods aim to estimate the LP index set X^F and Y^F from Eq. 5. These methods will be based on displacement prediction mean errors from training groundtruth data. The first method is simply to remove LPs in the rigid flock with mean prediction errors less than a predefined threshold. The second method aims to optimise the mean error by gradually removing irrelevant LPs.

We first provide a number of definitions. The training set will be a set of N_G images I^G with groundtruth positions for the target feature. The displacement groundtruth dataset is defined as $G = (g_{x,t}, g_{y,t})t = 1^{N_G}$, where each example, g_i, is a 2D displacement vector. Given a rigid flock learnt using a small number of training examples, it is possible to track the feature using Eq. 5 and obtain the predicted displacement vectors for every LP at every frame, which is defined as $(x'^F_{i,t}, y'^F_{i,t})_{i=1}^{N_G}$.

4.1. Individual LP Mean Error

The two displacement prediction component's mean error for each LP in a rigid flock can be computed as follows:

$$\epsilon_{x,i} = (1/|G|) \sum_{t=1}^{N_G} \sqrt{(g_{x,t} x_{i,t}'^F)^2} \quad (6)$$

$$\epsilon_{y,i} = (1/|G|) \sum_{t=1}^{N_G} \sqrt{(g_{y,t} y_{i,t}'^F)^2} \quad (7)$$

For both prediction components, we can then obtain a sorted ascending list of the errors and its sorted indices: $(\epsilon_{x,j}^s, \alpha_{x,j})$ and $(\epsilon_{y,j}^s, \alpha_{y,j})$, where $\epsilon_{x,\alpha_{x,j}} = \epsilon_{x,j}^s$ and $\epsilon_{y,\alpha_{y,j}} = \epsilon_{y,j}^s$. To use this result, we have elected to set the index sets X^F and Y^F to that of a predetermined number (M) LPs with the smallest mean prediction error:

$$X^F = \{\alpha_{x,j}\}_{j=1}^M \quad (8)$$

$$Y^F = \{\alpha_{y,j}\}_{j=1}^M \quad (9)$$

4.2. Mean Error Optimisation

Alternatively, one can also choose to determine the sets X^F and Y^F by iteratively optimising the overall mean error of the *rigid flock's mean prediction vector* from the groundtruth. Note that this is different to the error from *individual LPs to the groundtruth*, which is what the previous section proposed.

We firstly define two index vectors containing the indices for remaining LPs to consider at every optimisation step o: $X_o'^F$ and $Y_o'^F$, for the x and y components of the predicted displacement vector respectively. For each training image, it is possible to use Eq. 5 to obtain the tracked feature's displacement vector, based on the mean of predictions from the considered LPs, as given in $X_o'^F$ and $Y_o'^F$. This results in one displacement prediction vector for each training example which we define as: $(x_{o,t}'^F, y_{o,t}'^F)_{t=1}^{N_G}$. The mean error of the rigid flock's prediction against the groundtruth data at optimisation step o can then be determined using the two functions:

$$\gamma_{x,o}((x_{o,t})_{t=1}^{N_G}) = (1/|G|) \sum_{t=1}^{N_G} \sqrt{(g_{x,t} x_{o,t})^2} \quad (10)$$

$$\gamma_{y,o}((y_{o,t})_{t=1}^{N_G}) = (1/|G|) \sum_{t=1}^{N_G|} \sqrt{(g_{y,t} y_{o,t})^2} \quad (11)$$

which will be used as: $\gamma_{x,o}((x_{o,t}'^F)_{t=1}^{N_G})$ and $\gamma_{y,o}((y_{o,t}'^F)_{t=1}^{N_G})$ respectively.

Now, it is possible to attempt to reduce the error γ by determining which LP when removed will give the greatest reduction in error. The aim of the optimisation process is to iteratively remove elements from $X_o'^F$ and $Y_o'^F$ such that their respective errors in Eq. 11 will be maximally reduced for that step. Formally, we define $\tilde{X}_o^{\rho_o} = X_o'^F \rho_o, \rho_o \in X_o'^F$. Using $\tilde{X}_o^i$, we can recompute the new resulting x-component of the prediction vectors for the training set, which is defined as $(\tilde{x}_{o,t}'^{F,i})_{t=1}^{N_G}$. Now, suppose ρ_o is the element of $X_o'^F$ such that when removed, lead to the greatest decrease in the mean error (Eq. 11):

$$\arg\min_{\rho}(\gamma_{x,o}((\tilde{x}_{o,t}'^{F,\rho})_{t=1}^{N_G}) \quad (12)$$

Similarly, for the $y - component$ we have: $\tilde{Y}_o^{\phi_o} = Y_o'^F \phi_o, \phi_o \in Y_o'^F$. We define the resulting predicted displacement vectors for the training set as: $(\tilde{y}_{o,t}'^{F,i})_{t=1}^{N_G}$ and ϕ as the element in $Y_o'^F$ which, once removed, gives the smallest mean error:

$$\arg\min_{\phi} \gamma_{x,o}((\tilde{x}_{o,t}'^{F,\phi})_{t=1}^{N_G}) \quad (13)$$

The optimisation algorithm is then as follows: It is now possible to set the index sets X^F and Y^F to that of the M LPs with the mean prediction error, where again, M is a predetermined number:

Algorithm 1 Optimise LP Flock Prediction Mean Error

$X_1'^F = \{1, ..., |L^F|\}$
$Y_1'^F = \{1, ..., |L^F|\}$
for $o = 2$ to $N_G 1$ **do**
  Obtain (ρ_o) using Eq. 12.
  Obtain (ϕ_o) using Eq. 13.
  Update used LPs in flock to the prediction's x component: $X_{o+1}'^F = \tilde{X}_o^{\rho_o}$.
  Update used LPs in flock to the prediction's y component: $Y_{o+1}'^F = \tilde{Y}_o^{\phi_o}$.
end for
$remX = \{\rho_{o+1}\}_{o=1}^{N_G-1}$
$remY = \{\phi_{o+1}\}_{o=1}^{N_G-1}$

$$X^F = \{remX_{x,j}\}_{j=N_G-1-M}^{N_G-1} \quad (14)$$

$$Y^F = \{remY_{y,j}\}_{j=N_G-1-M}^{N_G-1} \quad (15)$$

5. Experiments

This section describes the various experiments that were performed on the proposed method using test examples. The experiments carried out were aimed at providing us with the understanding on the following points: firstly on the tracking performance of the proposed method in relation to results from the state-of-the-art AAM method; secondly, how this performance is affected with regards to different parameters, in particular the amount of training examples used for the selection of LPs and the total number of LPs that is used when performing tracking. In order to provide a deeper understanding of tracking performance, the

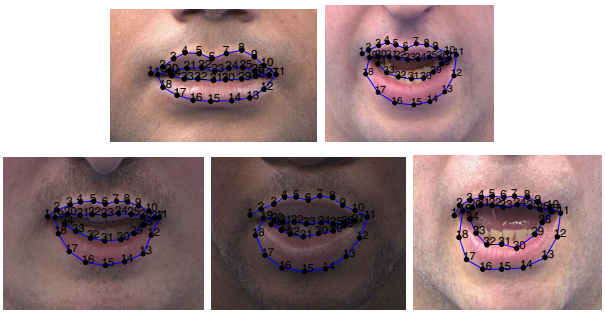

Figure 3. Shown are all 34 tracked points for the mouth of the 5 test subjects.

results are presented separately for different parts of the lips, specifically, the outer upper lip, outer lower lip, inner upper lip and inner lower lip.

5.1. Experimental Setup

For the experiments, five subjects were used (Figure 3). For each subject, two separate sequences were captured, one for training and one for testing. The training sequence only invovled the subject speaking the letter A to D. The test sequence involved reading the letters A to Z with 2 second pauses between each letters. Image acquisition was performed using a single HD-capable camera (Viper Thompson). The capture frame-rate was 25 fps with all images in the standard HD resolution. Both training and test sequences contained 1400 frames, equivalent to 56 seconds of footage. In terms of groundtruthing the data, it was deemed too time consuming to manually label all the lip shape points by hand. As a result, AAMs were used to track the mouth shape and provide the required groundtruth information. However, it is important to note that AAMs do not always work well. For this reason, highly inaccurate results from AAMs were manually removed.

The lip shape consists of 34 points laid out across the inner and outer lip shape contour (Figure 3). In order to track all of these points, 34 independent rigid flocks of LPs were used. Initially, each of the rigid flock had 200 member LPs spread randomly around its corresponding lip shape point. Each LP had a support region of 30 pixels, where 80 support pixels were randomly placed. For training the LPs' linear mapping matrix H, only 9 images were used, originating from the the first half segment where the subject spoke the letter 'A'. For selecting the LPs within each flock, the number of training examples ranged from 50 to 550. The tracking performance across these different training example sizes is described in the next section. For the testing the tracking performance of the different proposed LP selection methods, the entire test sequence (i.e. 1400 frames) was used. It must be noted that only grayscale images were used for training, LP selection and tracking.

5.2. Tracking Test Results

The results of the proposed method for tracking the lip shape is illustrated in Figure 4. This figure shows the various tracked lip shapes in the test sequence using three methods: rigid flocks without any LP selection, LP selection using individual LP errors (Section 4.1) and LP selection by optimisation of the overall prediction mean error (Section 4.2) as well as results using AAMs. Here, the results without any LP selection fares the worst, in particular the tracked inner lip. The tracking results for the other subjects can be seen in Figure 5. The tracking speed using unoptimised C++ code is about 20fps on a standard dual-core processor PC. Despite the fact that the lip shape is tracked with essentially 34 independent tracked points, there was never a tracking failure. Note that no temporal constraints were used as well, the tracking was simply done by updating a rigid flock's reference point with the mean prediction from its member LPs.

In order to quantify the results shown in Figure 4, the mean error of the tracking across all subjects from the groundtruth data was used. The scale of these errors are in terms of pixels. As stated above, 4 mean errors were obtained, corresponding to sections of the inner and outer lip as well as the lower and upper lip. On the whole, we find that the performance of the LP selection method based on optimising the overall prediction mean error is better than that based directly on the error of individual LPs in a flock. This is described in more detail below.

Firstly, we present the results of different tracking mean errors across a range of training data sizes used for selecting LPs. These are shown in Figure 6. Also shown in the figure is a horizontal line, which denotes the best corresponding mean error when no selection was performed on the LPs in the rigid flocks. We note that all the results from rigid flocks with selected LPs are better than those with no selection. We can also see that the increase in training data size is only useful up until a certain degree, whereby further increase in training examples gives little or no improvement in the tracking errors. The second set of experiments was conducted by varying the number of selected LPs used, essentially varying the parameter M(defined in Section 4). This ranged from 10 to 170 LPs. We note that when we increase the LPs to 200, this is equivalent to performing no LP selection. The results are shown in Figure 7. We find that when there are too few LPs selected, the error is relatively large. This is because, certain LPs that are critical for providing correct predictions are not included. This is confirmed by the reduction in error as the number of selected LPs used is increased. However, there comes a point where we start using too many LPs, including those that provide wrong predictions. This is reflected by increasing errors when the number of LPs is large (usually above 80).

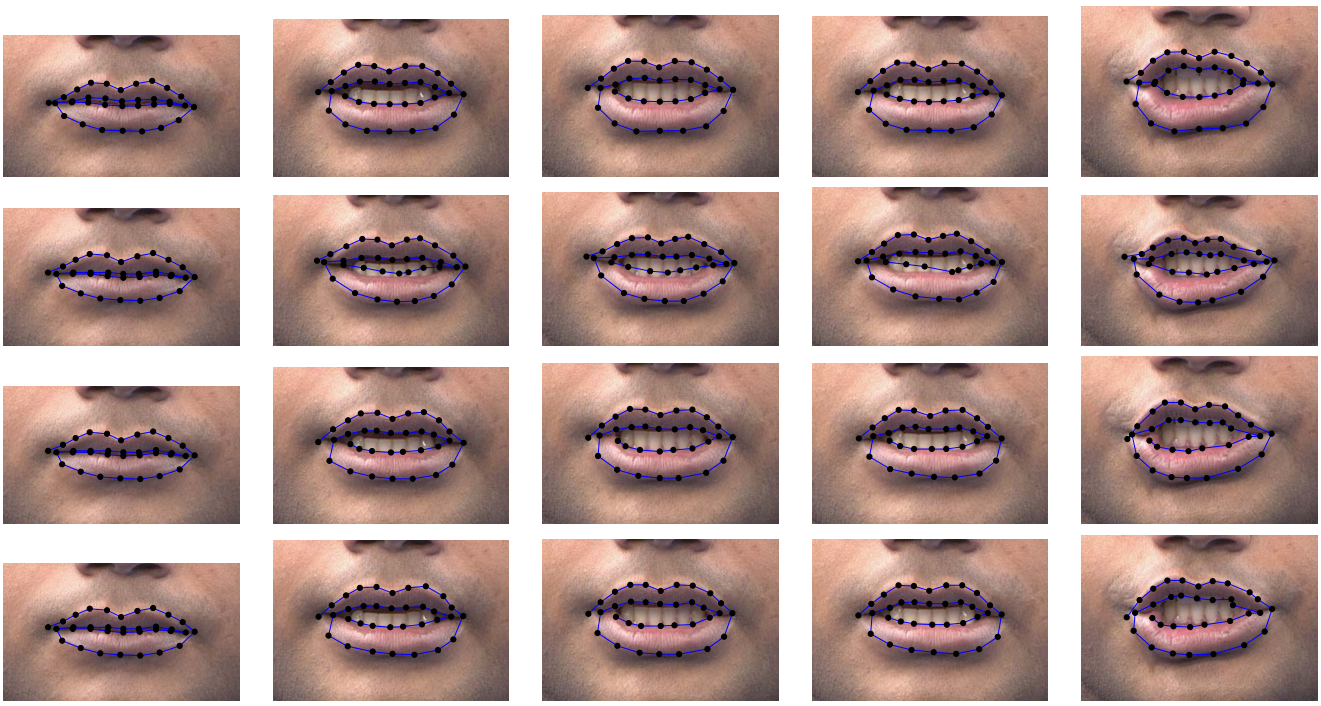

Figure 4. Results of tracking points on the lip shape for different methods. Shown are results for 5 different frames in the tracked sequence. Here, only the mouth region is shown for result clarity, however, for tracking, the image containing the subject's entire head is used. For each frame, the first row (topmost) shows the result of the AAM. The second row shows results using the an LP flock with no selection. The third and fourth row shows results from LP flocks having undergone selections based on the individual member LP's errors and the overall mean error respectively.

Figure 5. Results of tracking points on the lip shape for different subjects using LP flocks with the overall mean error selection.

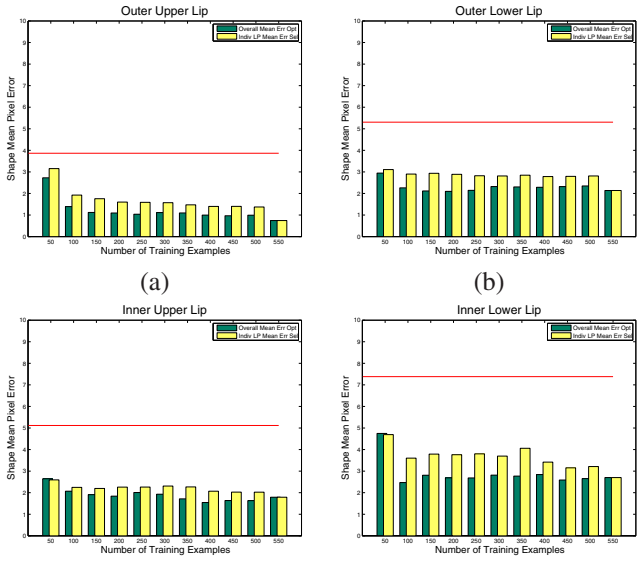

Figure 6. Tracking errors across different training set sizes. Shown are errors for the 4 different sections of the mouth: (a) Outer Upper Lip, (b) Outer Lower Lip, c) Inner Upper Lip, (d) Inner Lower Lip. Horizontal line shows unoptimised LP flock error.

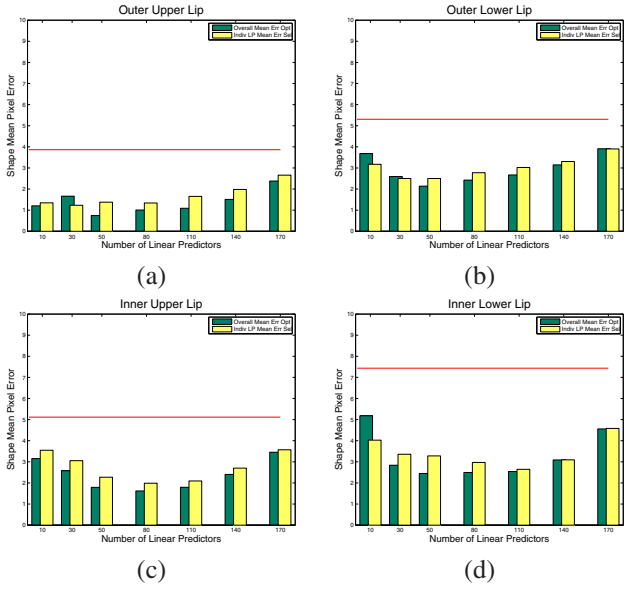

Figure 7. Tracking errors using different numbers of selected linear predictors. Shown are errors for the 4 different sections of the mouth: (a) Outer Upper Lip, (b) Outer Lower Lip, c) Inner Upper Lip, (d) Inner Lower Lip.

5.3. Analysis of Selected LPs

It is known that the outer lip relatively easy to track compared to the inner lip. The main reason is that the outer lip does not suffer from the sudden appearance and disappearance of the teeth or tongue, as is the case for the inner lip. However, from the results obtained using the selected methods, we find that the errors of the inner lip are only slightly lower than those of the outer lip. This is in stark contrast to the errors of the unoptimised LP flock, where the inner lip's tracking performance is much lower than the outer lip.

To understand this further, when analysing the different LPs selected for tracking points on the inner lip, we find that they tend *not* to be located at places near the tracked point, where the variations in appearance greatest and are highly dependent on whether the mouth is open or closed. Instead, most of the LPs used to predict the displacement of inner lip shape points usually lie *away* from the shape point itself (Figure 8). They are usually placed around areas whose movements are strongly correlated with the target shape point, but with less ambiguous appearance information(e.g. the lips and chin for tracking the y-component of the inner lower lip).

6. Conclusions

In this paper, we described the use of rigid flocks of linear predictors to accurately track in real-time different points on the lip contour, both on the inner and outer lip. These points were tracked *independently* with no constraints applied. As a result, this method of tracking the lip shape is fundamentally different to existing model-based approaches using active contours or active appearance models. Furthermore, to achieve accurate tracking performance, two approaches were proposed for selecting relevant sets of LPs within each flock. Analysis of the selection results showed that the LPs selected for tracking a feature point will often lie on places that allow accurate predictions and crucially, whose movements are strongly correlated with that of the tracked target. An example is in the tracking of the inner lip points. LPs lying on the inner lip contour are often not used due to their large variations in appearance. However, points on the lip itself are used instead. Experimental results also show that this method is comparable in performance to that of AAMs, with an average of 2 pixel difference in the lip shape estimations. This was despite using a much simpler approach, both in the training and tracking phases. This was also achieved with minimal amounts of training data. For learning the linear mappings, only 9 images were used, corresponding to half of the video segment of the letter 'A' being said. In the tracking phase, no optimisation was performed. Tracking a lip shape point is done by simply offsetting its LP flock's reference point with its corresponding prediction vector.

Acknowledgements

The investigation reported here has been largely supported by the EPSRC project LILiR and in part by the FP7

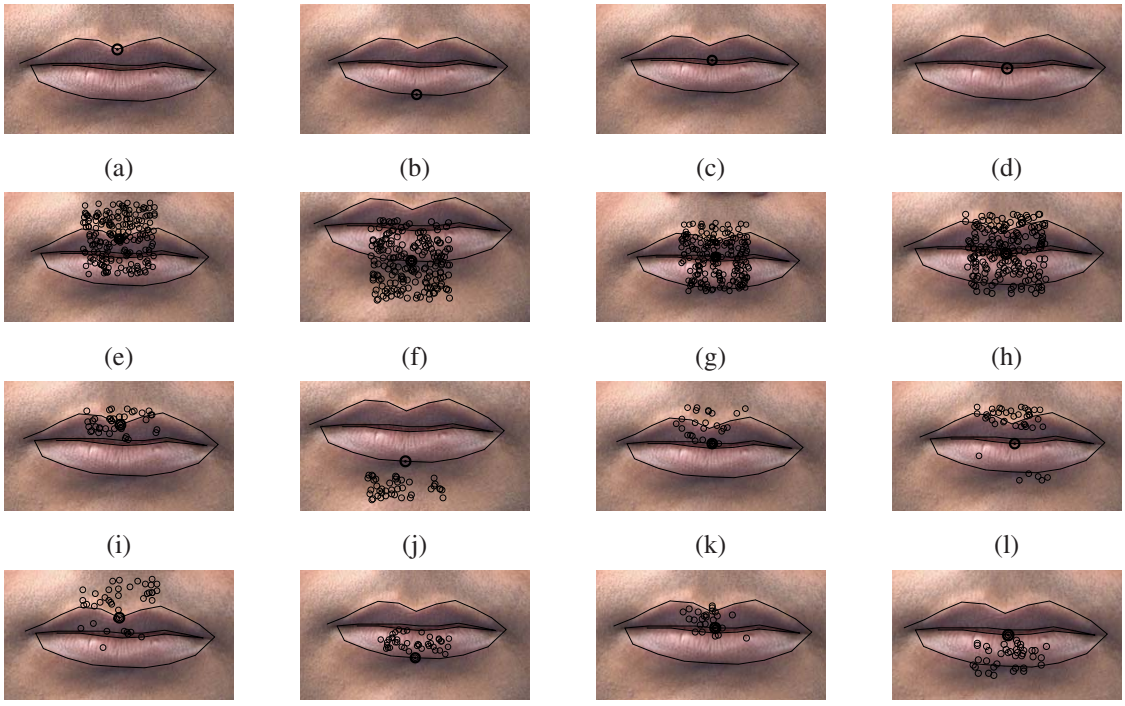

Figure 8. This figure shows the selected LPs for tracking various points on the lip shape. The first row (a,b,c,d) shows the tracked mouth shape point. The second row (e,f,g,h) shows all the member LPs in the flock for a given mouth shape point, highlighted by a thick-lined circle. The middle (i,j,k,l) and bottom (m,n,o,p) rows on this figure shows the selected LPs for the x and y predicted displacement component respectively.

project DIPLECS.

References

[1] M. Barnard, E. Holden, and R. Owens. Lip tracking using pattern matching snakes. In *Proc. of the Fifth Asian Conference on Computer Vision*, January 2002.

[2] C. Bregler and S. Omohundro. Nonlinear manifold learning for visual speech recognition. In *Proc. of Fifth International Conference on Computer Vision*, pages 494–499, 1995.

[3] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[4] P. Daubias and P. Deléglise. Statistical lip-appearance models trained automatically using audio information. *EURASIP J. Appl. Signal Process.*, 2002(1):1202–1212, 2002.

[5] J. Hoey. Tracking using flocks of features, with application to assisted handwashing. In *Proc. of British Machine Vision Conference*, pages 367–376, 2006.

[6] M. Kolsch and M. Turk. Fast 2d hand tracking with flocks of features and multi-cue integration. In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 10*, page 158, Washington, DC, USA, 2004. IEEE Computer Society.

[7] M. Lievin, P. Delmas, P. Coulon, F. Luthon, and V. Fristol. Automatic lip tracking: Bayesian segmentation and active contours in a cooperative scheme. In *Proc. of IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 691–696, July 1999.

[8] J. Matas, K. Zimmermann, T. Svoboda, and A. Hilton. Learning efficient linear predictors for motion estimation. In *Proc. of Fifth Indian Conference on Computer Vision, Graphics and Image Processing*, December 2006.

[9] I. Matthews, T. Cootes, and J. Bangham. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213, 2002.

[10] I. Patras and E. Hancock. Regression tracking with data relevance determination. *IEEE Conference on Computer Vision and Pattern Recognition.*, pages 1–8, June 2007.

[11] O. Williams, A. Blake, and R. Cipolla. Sparse bayesian learning for efficient visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1292–1304, 2005.

[12] Z. Wu, P. Aleksic, and A. Katsaggelos. Lip tracking for mpeg-4 facial animation. In *ICMI '02: Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, page 293. IEEE Computer Society, 2002.

[13] A. Yulle, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.