Robust Facial Feature Tracking using Shape-Constrained Multi-Resolution Selected Linear Predictors

Eng-Jon Ong and Richard Bowden *Senior Member, IEEE,*

Abstract—This paper proposes a learnt *data-driven* approach for accurate, real-time tracking of facial features using only intensity information. The task of automatic facial feature tracking is non-trivial since the face is a highly deformable object with large textural variations and motion in certain regions. Existing works attempt to address these problems by either limiting themselves to tracking feature points with strong and unique visual cues (e.g. mouth and eye corners), or by incorporating a-priori information that needs to be manually designed (e.g. selecting points for a shape model). The framework proposed here largely avoids the need for such restrictions by automatically identifying the optimal visual support required for tracking a single facial feature point. This automatic identification of the visual context required for tracking, allows the proposed method to potentially track any point on the face. Tracking is achieved via linear predictors which provide a fast and effective method for mapping pixel-intensities into tracked feature position displacements. Building upon the simplicity and strengths of linear predictors, a more robust *biased* linear predictor is introduced. Multiple linear predictors are then grouped into a rigid flock to further increase robustness. To improve tracking accuracy, a novel probabilistic selection method is used to identify relevant visual areas for tracking a feature point. These selected flocks are then combined into a hierarchical multi-resolution LP model. Finally, we also exploit a simple shape constraint for correcting the occasional tracking failure of a minority of feature points. Experimental results show that this method performs more robustly and accurately than AAMs, with minimal training examples on example sequences that range from SD quality to Youtube quality. Additionally, an analysis of the visual support consistency across different subjects is also provided.

Index Terms—Facial Feature Tracking, Learning, Linear Predictors, Multiple Resolution, Probabilistic Selection.

1 Introduction

The task of automatic facial feature tracking is non-trivial since the face is a highly deformable object. For example, the lip is highly deformable and can assume a large variety of shapes. This difficulty is compounded by the potential appearance and disappearance of the teeth and tongue during speech causing the inner lip's texture to change dramatically. Other parts of the face can contain extremely fast movements, for example, the eye shape can change from an open eye to a closed eye in the period of a single frame. There are also areas of the face that are challenging to track directly, in particular, points on the cheek where the texture can be homogeneous.

In this paper, we approach the above issues by proposing a learnt, person-specific but importantly, a *data-driven* approach to achieve accurate and real-time tracking of independent facial features using only pixel intensities. A crucial component of our method is the ability to automatically locate visual support that is optimal for tracking a particular feature point. This allows us to potentially track any facial feature. Importantly, this includes points on regions where the visual complexity is high due to potential texture changes (e.g. inner lip) and facial features that are challenging due to the lack of texture (e.g. points on cheek). However, this does not discount the usefulness of shape constraints. Indeed, in this work, we show that the robustness of the trackers can also be improved by means of simple shape constraints. It is important to note, however, that our use of shape functions merely as a simple constraint, allowing the correction of occasional tracking errors in a minority of feature points. Additional constraints such as temporal models of dynamics are neither required nor used.

• *E.Ong and R.Bowden are with the Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, UK. E-mail: {e.ong, r.bowden}@surrey.ac.uk*

1.1 Related Work

There exists a number of different methods for facial feature tracking. One class of popular methods treats locating facial features as a detection problem. The locations of various facial features (e.g. eyes) are semi-reliably detected, followed by the elimination of invalid facial feature configurations using various statistical methods [4], [34], [12]. Whilst it is possible to detect the locations of facial features with strong and consistent visual cues (e.g. eyes, nose, mouth, etc...), it is unclear whether these methods will be accurate enough to be used for tracking more specific points on the face, for example a particular point on the eyelid, or a set of points around the inner mouth. Certainly, such methods avoid tracking

Digital Object Indentifier 10.1109/TPAMI.2010.205

less subtle feature points such as those on homogeneous regions on the face (e.g. cheeks).

Other methods fall into the category of model-based methods. An example of which is active contours. Examples of active contours applied to tracking lip contours are [3], [35]. This was improved by Barnard et al. [2] by coupling this technique with 2D templates. In [32], temporal constraints were also included to improve tracking. It is possible to exploit colour information of specific parts of the face (e.g. lips) by initially performing segmentation using colour and markov random field models before obtaining the final shape using active contours as proposed by Lievin et al. [15]. One disadvantage of active contours are their complexity in time and computation.

A widely used method for facial feature tracking is the Active Shape Model (ASM), originally proposed by Cootes et al [7]. Here, a statistical model (usually a single or mixture of Gaussians) of the shape of various facial features is initially built. This statistical model is then used as a generative mechanism to constrain and produce plausible facial shapes. Models of local visual information around each facial point are then constructed. A generated shape is then iteratively deformed to fit the face in an observed input image in such a way that the local model best fits the observed information in the input image. In the original work of Cootes et al[7], the intensity profile along the normal of the shape's curve was used to model a feature point's local visual information. Recently, machine learning methods have been used to produce classification approaches for locating feature points. Examples of such learning mechanisms range from Boosting methods [36], [8], [9], to neural networks [26], [5]. The work by Ding et al[9] is particularly related to the proposed approach. Here, the surrounding visual context is automatically selected and used as negative for building feature detectors was proposed. The linear prediction approach proposed here differs from the machine learning approaches by tracking facial features in terms of estimating feature displacements using a regression approach as opposed to detecting facial features.

Pose invariance can be obtained by using a mixture pose-specific ASMs as proposed in [1]. However, one main disadvantage of the above ASM methods is the requirement for a large training database of different facial shapes, where the number of examples can range from hundreds to thousands. This is necessary in order to produce an accurate statistical model of valid shapes capable of generating plausible facial configurations. Although there exists approaches for automatically landmarking the shape information from images, as proposed by Milborrow et al[21], results depend on the quality of the data and the landmarks characteristics (e.g. a point on the inner lip may not be easy to automatically label consistently).

The large training datasets required by ASMs can be reduced by building person-specific trackers using Active Appearance Models (AAM) by Cootes et al [6]. One advantage of the AAM over the ASM is that it allows for the learning of person specific feature trackers using a much smaller training dataset. This is made possible by the use of the shape-constrained face appearance information during the tracking stage. AAMs have previously been used for tracking lip shapes by Matthews et al[20]. Pose invariance can also be achieved by coupling AAMs with an underlying 3D model, for example by Xiao et al [33], followed by Dornaika et al [11] and more recently by Sung et al [27]. However, whilst person-specific AAMs produce excellent results and can be optimised to be fast, as demonstrated by Matthews and Baker[19], the non-linear optimisation step during the tracking phase can still be complicated and they are not necessarily robust, particularly in the presence of large movements or image noise.

Recently there has been a great deal of research aimed at producing person-independent AAMs. For example, the adaptation of an existing AAM to a novel subject was proposed by Dornaika et al[10]. Generalisation of AAMs can also be obtained by learning a person-independent boosted classifier that produces an alignment error score [16]. The performance of these AAMs was improved further by Nguyen et al [22] by learning a suitable cost functions that is quasi-concave. Alternatively, Wu et al [31] approached this task as a machine learning problem, where Boosting is used to select the best visual features that produce a suitable cost function [31]. However, the generalisation obtained, again comes at the cost of a relatively large training set. Additionally, these methods are used for addressing the face alignment problem, where only a limited set of facial feature configurations are considered. Consequently, it is still an open question as to how accurate they will be when applied to the problem of accurately tracking facial features.

1.2 Novel Contributions

In order to track features, the method of linear predictor flocks is used. Each Linear Predictor (LP) provides a mapping from sparse template differences to the displacement vector of a tracked facial feature [38]. Multiple LPs can then be grouped into rigid flocks to track a single feature point with greater robustness and accuracy. The following novelties are then introduced to produce a state of the art facial feature tracking framework. Firstly, the framework proposed can be used for general facial feature tracking. This is an important attribute, as it provides tracking for a general set of facial feature points i.e. we are not limited to points that have consistent and strong appearance such as the corners of the eyes and mouth. The relevant visual context for tracking any facial point can be established during training. Secondly, the linear predictors are extended into a full linear regression function by introducing a bias factor, resulting in *biased-LPs* (Section 2). Thirdly, a novel LP selection method based on probabilistic selection is proposed, removing

the need for a heuristically defined threshold for how many LPs to retain in a flock. Instead, the proposed selection method migrates member LPs into optimal positions, essentially automatically identifying the visual context for tracking a particular feature point (Section 3). Fourthly, the tracking robustness is further increased by integrating the selected LP flocks into a hierarchical multi-resolution framework (Section 4).

The performance of the method is quantified and compared to AAMs using convergence curves. These convergence tests have the additional benefit of measuring how robust a method is under different initialisation conditions, as detailed in Section 6.

A preliminary version of the work described in this manuscript appeared in [24]. In this manuscript, we further improve on the above with the following contributions: Firstly, a simple shape constraint is added to the existing tracking framework (Section 5). It was found that this addition contributed to a significant improvement in both the accuracy and robustness of the method, as is documented in the experiments section.

Secondly, a more rigorous experimental evaluation of the proposed method is provided (Section 6 and 7). These include new convergence experiments on LP flocks with and without the shape constraints and further analysis on the performance of AAMs when the entire face was used for tracking (Section 6). Additionally, tracking performance comparisons are also made with AAMs by using an existing database of American Sign Language video sequences (provided by authors of [9]) where tracking groundtruth is available. These sequences are challenging due to the variations in pose and facial expressions present, as well as significant occlusions to the face from the subject's hands when performing different signs. As such, it allows us to evaluate the tracking accuracy of the proposed method in the presence of the above factors and compare it against existing results using AAMs. It also allows the effect of training set size on performance to be quantified. Further quantification of tracking performances on even more challenging video sequences obtained from Youtube is also made (Section 7). These explore how well tracking copes with factors that are present in the ASL database, but with more extreme variations and under significant image noise (e.g. motion blur, compression artefacts). Furthermore, analysis is also performed on the consistency of the selected visual information across different individuals for tracking the same facial feature (Section 6.8) before concluding in Section 8.

2 Biased Linear Predictors

A Linear Predictor (LP) forms the central component of the proposed tracking mechanism. An 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 for efficient constrained tracking of planar objects[38]. Along similar lines, Relevance Vector Machines (RVMs) have also been used to provide displacement predictions [30]. More recently, Bayesian Mixtures of Experts coupled with RVMs have been proposed for tracking [25]. However, compared with LPs, both the above approaches are of high complexity.

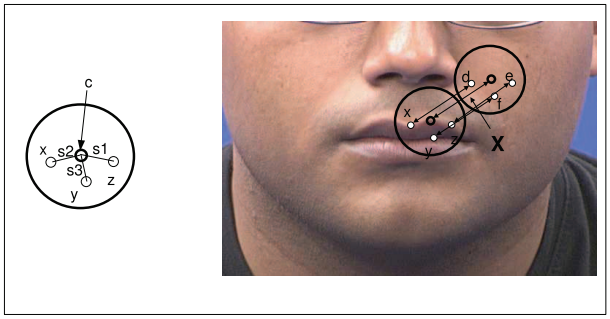


Fig. 1: Illustration of a linear predictor(LP). Each LP has a reference point c. Within an area around c, called the support region a set of randomly sampled support pixels $(x, y, z)$with their offsets from c: $s1, s2, s3$. Also shown is the synthesis of training data. X is the artificial translation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

A linear predictor (LP) is defined as a set of five components, $L = \{c, H, V, S, b\}$, where c is a 2D reference point defining the location of the feature, H is the linear mapping to a displacement vector, S is a set of 2D offsets for positioning the support pixels, $|S|$ is the number of support pixels and V is a $|S|$-dimensional vector of base support pixel values. V forms a sparse template for the visual appearance of the feature point. Additionally, we improve on the original work on LPs[38] by adding a bias factor b to the linear mapping H. As can be seen in Section 6.2, this simple but effective addition significantly increases the tracking accuracy. We will refer to this new type of linear predictor as a *biased LP*.

In order to obtain V, $S = \{s_i\}_{i=1}^{|S|}$ is defined and used, where s_i is the offset relative to c. The offset positions s_i are obtained as random offsets within a specified radius from the origin. In Section 4, we show how LPs of different sizes can be combined together into multi resolution LPs for improved robustness. An illustration of a linear predictor can be seen in Figure 1. To use a biased LP to predict the displacement of its tracked feature (i.e. reference point c), 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 biased displacement of c, t is then:

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

2.1 Learning the Biased 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 examples in the form of support pixel differences (δp) and displacement vector pairs (t). To achieve this, a number of training examples can be synthesised from each single training image.

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. This process is then repeated for all training images, allowing us to gather a wide range of training examples for learning the linear mapping H. The base support pixel values are then set as their respective mean intensities across all training images.

The generated examples are then 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. Additionally, we learn the bias for the linear model by adding an additional column of 1s to the end of δP, giving: $\delta P' = (\delta P, [1])$, where $[1]$ denotes a column vector of rows N_T. Using least squares, H can now be obtained as follows:

$$H = T\delta P'^{+} = T\delta P'^{T}(\delta P'\delta P'^{T})^{-1} \quad (3)$$

where $\delta P'^{+}$ is the pseudo-inverse of $\delta P'$.

2.2 Rigid Flocks of Biased 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 [14], [13], 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. In our case of a rigid flock, the LP trackers are similar to the above flocks, in that they are constrained to lie within some area around a reference point. The difference from the above work is that they are fixed by an original offset away from the reference point. In order to further improve on the robustness and accuracy of the tracking predictions, separate flocks are used for predicting the individual x and y displacement components. To disambiguate between these two types of flocks, the superscript x and y are used respectively. Since both of these LP flocks are similar in form, for the following LP flock definitions, we use a "*" to act as a placeholder that can take either the superscript x or y. This removes the need to repeat the formal definitions twice.

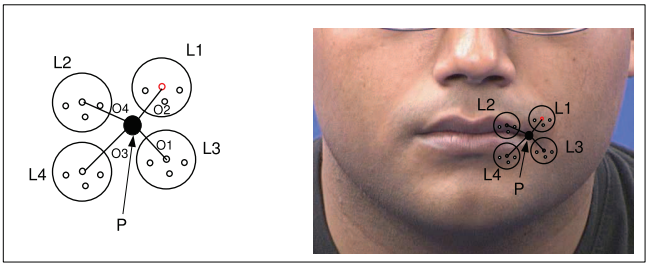


Fig. 2: Illustration of a rigid flock of linear predictors, whose position is given by reference point, P. The member LPs are $(L1, L2, L3, L4)$ each with a rigid offset from P: $O1, O2, O3, O4$.

Thus, a rigid flock 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 $\{t^*\}_{f=1}^{|L^{F*}|}$. This 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 6, 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 [14], [13], [6], [2]).

The 2D image displacement prediction of a feature point modeled with 2 LP flocks (L^{Fx}, L^{Fy}) is given as:

$$x^F = (1/|L^{Fx}|) \sum_{f=1}^{|L^{Fx}|} t_f^x(1) \quad (4)$$

$$y^F = (1/|L^{Fy}|) \sum_{f=1}^{|L^{Fy}|} t_f^y(2) \quad (5)$$

where $t_f^*(1)$ and $t_f^*(2)$ are respectively the x and y components of the f^{th} LP's predicted displacement vectors. From this, we find that $t_f^x(1)$ is the x-component of the f^{th} x-coordinate LP, and $t_f^y(2)$ is the y-component of the f^{th} y-coordinate LP. This is then used to update the position of the rigid flock reference point for both x and y LP flocks: $P^{Fx} = P^{Fx} + (x^F, y^F)$ and $P^{Fy} = P^{Fy} + (x^F, y^F)$.

3 Probabilistic Selection of LPs

Having defined a rigid flock of LPs in the previous section, we are now faced with the crucial issue of deciding *where* to place the member LPs. To start, for both sets of LP flocks (L^{Fx}, L^{Fy}), a predetermined number of LPs are randomly scattered within an area around the rigid flocks' reference point. It is then possible, for all linear predictors to use Eq. 4 and 5 to predict the displacement of the reference point, given a new input image.

However, this can be suboptimal, since there may exist many LPs in a flock that will give incorrect displacement predictions. We are now faced with the problem of identifying meaningful context useful for tracking a particular feature point and crucially, this context may not be the local support region as would be assumed for traditional template-based tracking approaches (e.g. the LK tracker[18], [29]). To address this issue, this section proposes an iterative and probabilistic method for selecting *separate* sets of LPs for accurate and robust predictions. More specifically, this method will be based on iteratively selecting new sets of LPs based on their displacement prediction mean errors from training groundtruth data. In earlier work [23], a naive method of removing LPs in the rigid flock with mean prediction errors less than a predefined threshold was used. Here, with an iterative scheme, a threshold is no longer needed.

To continue, a number of definitions are firstly given: 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. 4 and 5 and obtain the predicted displacement vectors for every LP at every frame, which is defined as $(x'^F_{i,t}, y'^F_{i,t})_{i=1}^{N_G}$.

3.1 Probabilistic Re-selection using the Individual LP Mean Error

The biased LP selection method proposed here is similar in essence to factored sampling, particularly the first step of particle selection. The proposed method is essentially an iterative method, where at each iteration, we draw a new set of location offsets for member LPs in a flock with respect to weightings based on an inverted form of training error. Following this, it is possible to relearn the linear mappings for this new set of LPs and reiterate the whole process to progressively locate better locations for placing LPs resulting in greater tracking accuracy.

To start, we calculate the two displacement prediction component's mean error for each LP in a rigid flock for iteration step α as follows:

$$\epsilon^{\alpha}_{x,i} = (1/|G|) \sum_{t=1}^{N_G} \sqrt{(g_{x,t} - x'^F_{i,t})^2} \quad (6)$$

$$\epsilon^{\alpha}_{y,i} = (1/|G|) \sum_{t=1}^{N_G} \sqrt{(g_{y,t} - y'^F_{i,t})^2} \quad (7)$$

Having obtained the individual LP mean errors, the next step is to transform them into a selection probability value. This essentially involves inverting the error values. There are a wide variety of methods to achieve this, however, we have found that given a set of errors $\epsilon^{\alpha}_{*,i}$ where $*$ can either be x or y, the errors can be inverted to selection probabilities $(\beta^{\alpha}_{*,j})_{j=1}^{M}$ using:

$$\beta^{\alpha}_{*,j} = \frac{\max(\epsilon^{\alpha}_{*,i}) - \epsilon^{\alpha}_{*,i}}{\sum \epsilon^{\alpha}_{*,i}{}_{i=1}^{M}} \quad (8)$$

Following this, a cumulative probability graph can be formed:

$$\gamma^{\alpha}_{*,j} = \sum \beta^{\alpha}_{*,i}{}_{i=1}^{j} \quad (9)$$

From this, we can probabilistically select the next batch of LPs from the existing set in an iterative manner. Here, it is possible to iterate for a fixed number of times, or until the error has converged to a preset value; T^{ϵ}. The selection algorithm is given in Algorithm 1. An illustration of how the positions of the members LPs in a flock changes as the algorithm progresses can be seen in Figure 3.

Algorithm 1 Probabilistic Selection Algorithm

$\alpha = 1$
$\delta\epsilon = T^{\epsilon} + 1$
Start with the initial LP flock ($L^{F*,\alpha}$) whose LP members' offset positions ($O^{*,\alpha}$) were initialised randomly around its respective feature point P^*.
while $\alpha < N^{\alpha}$ or $\delta\epsilon < T^{\epsilon}$ **do**
  Create a LP offset matrix, whose elements are 0: $O^{F*,\alpha}$
  for $i = [1...M]$ **do**
    $\alpha = \alpha + 1$
    Generate random number $R^{\alpha} \in [0, 1)$
    Obtain index for new LP:$\text{argmin}_j\, \gamma^{\alpha}_{*,j} < R^{\alpha}$
    Get the offset position of the selected LP: $(o_x, o_y) = O^{F*,\alpha-1}_j$
    Add a small random noise offset (n_x, n_y) to it: $(o_x, o_y) += (n_x, n_y)$
    Set the position of the LP in the new set as (o_x, o_y): $O^{F*,\alpha}_i = (o_x, o_y)$
  end for
  Relearn the linear mappings for the current iteration's LP set ($L^{F*,\alpha}$) using the newly obtained offsets $O^{F*,\alpha}$
end while

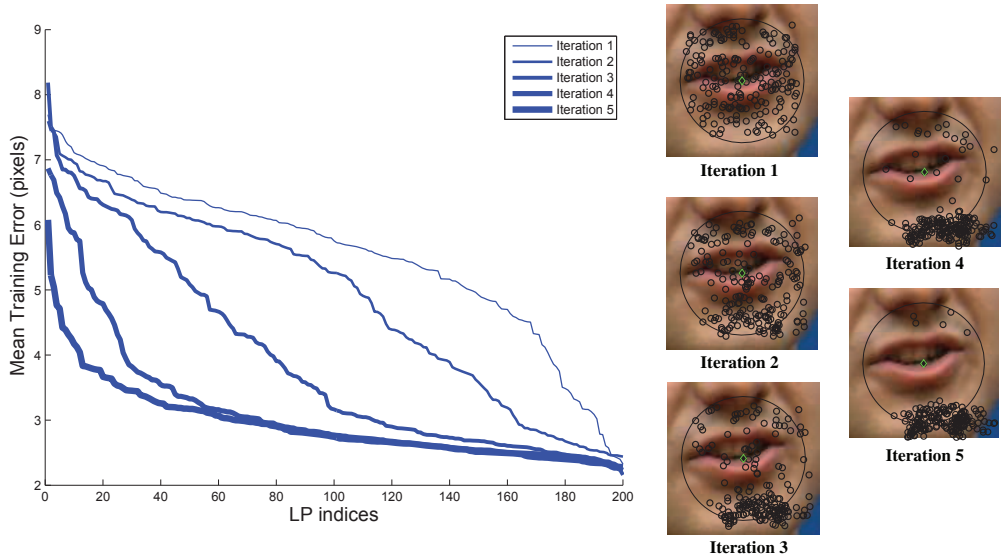
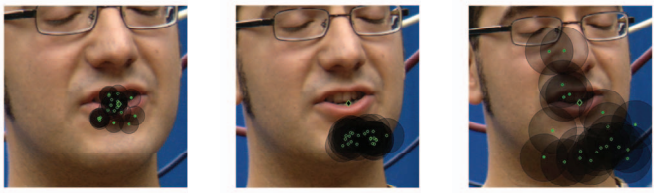


Fig. 3: Shown are the training errors of 200 individual LPs in a flock ranked in desceding order over different iterations of the probabilistic selection. The error graph on the left shows that the errors of a majority of LPs are reduced through iterative selection. Shown on the right are the locations of the different LPs for tracking a point on the lower inner lip at different iterations of the selection process. The large circle on the figures on the right shows the area where member LPs were initially randomly placed.

4 Chain of Multi-Scale Selected LP Flocks

It is known from previous work on LPs [38] that increasing the support region size and the range of training displacements result in greater robustness to large displacements from a feature's location, with reduced accuracy as a trade off. In order to obtain tracking results that are *both* robust and accurate, a chain of LPs of decreasing sizes can be used. The largest LP is first used to predict a feature's location. The result is then passed to the next LP in the chain that is less robust but more accurate, and repeated until we have reached the end of the LP chain, with an accurate location of a displaced feature.

We have found that the above argument also holds true for our case of selected LP flocks. As a result, we employ a similar strategy for obtaining robust and accurate feature tracking by chaining together a sequence of LP flocks of decreasing size. Here, we define the *size* of an LP flock by the radius of feature displacements that it has been trained to cope with. This also affects the offset range for member LPs as well as the support regions of the individual member LPs (see Figure 4). The exact sizes of the support regions, training displacement ranges and LP member offset radius will be given in Section 6. Formally, suppose we have trained N_S number of different sized LP flock pairs for tracking a feature point. It must be noted that the sizes of the member LPs of each pair are the same. More specifically, an LP flock pair of size θ is defined as: $L^{\theta} = (L^{Fx,\theta}, L^{Fx,\theta})$. Flocks can be sorted in descending order based on their sizes and used to form an ordered set to represent the multi-scale LP flock chain: $\{L^{\theta_i}\}_{i=1}^{N_S}, \theta_{i+1} < \theta_i$, where the N_S flock sizes are defined as the set $\{\theta_i\}_{i=1}^{N_S}$.



(a) (b) (c)

Fig. 4: Shown here are the different sized LP flocks that form a multi-scale LP flock, where (a) is the smallest to (c) being the largest. Predicting the feature displacement cascades from the largest (c) to the smallest (a) LP flock.

We can now use the multi-resolution LP flock chain to predict a feature's position. Suppose our input image is given as I. The initial starting position for prediction is given by the reference point position (Section 2.2) of the largest LP flock. A prediction is then made on the location of the tracked feature using Eq. 5. The resulting feature position, given by the updated reference point P^{F*,θ_1} is used to initialise the reference position of the next flock in the chain, that is: $P^{F*,\theta_2} = P^{F*,\theta_1}$. This process is then repeated until we have reached the last scale θ_{N_S}. The updated reference position of the last LP flock $P^{F*,\theta_{N_S}}$ is used as the tracked position of the feature of interest.

5 Shape Constraints

In order to further improve on the tracking performance in terms of accuracy and robustness, a simple shape constraint is introduced. Here, the shape constraint is modelled as a Point Distribution Model (PDM) [7] built using the labelled training set that was used for learning the LP flocks. Our shape constraint consists of a mean shape, set of basis vectors and a corresponding set of constraint extents.

To obtain the above, we start by creating a shape training dataset. Suppose, we have we have N_I number of training images, each labelled with N_F number of facial feature positions. It is then possible to construct a shape vector by concatenating all the coordinates of each feature point into the vector $k_i \in \mathbb{R}^{2N_F}, i = \{1...N_I\}$. A mean shape vector is also obtained from this set and defined as $\bar{k}$.

A covariance matrix can be obtained from the training shape vectors and the shape basis vectors are extracted by performing Principal Component Analysis (PCA) on the training covariance matrix. The resulting shape basis vectors are defined as: $E = \{e_i\}_{i=1}^{N_E}$, where the N_E largest eigenvectors account for 95% of the variations present in the data. Additionally, PCA also provides us with the extent in which the data varies in the direction of a particular shape basis, given by the eigenvalues of the covariance matrix. These are then used to form a set of constraints as follows: $\Lambda = \{\lambda_i\}_{i=1}^{N_E}$.

In order to use the above shape constraint model, at every frame, we obtain the tracked positions of each facial feature point of interest using the selected LP flocks. These positions are then concatenated into an input shape vector in a similar manner to that used to form the shape training set: k_I. Following this, a robust affine transform [37] using the Least Median of Squares Regression method[1], is used to align the input shape to the mean shape ($\bar{k}$) giving: k'_I. This aligned input shape vector is then projected onto each basis vector giving the eigenspace coefficients: $c_i = k'_I e_i, i = 1...N_E$. Finally, the constraints Λ are applied to c_i before being used to reconstruct a constrained shape vector (k''_I) containing the corrected positions of the facial feature points:

$$k''_I = \sum_i^{N_E} c'_i e_i, \text{where, } c'_i = \begin{cases} \boldsymbol{c_i} & \text{if } |\boldsymbol{c_i}| < \boldsymbol{\lambda_i} \\ \boldsymbol{\lambda_i} & \text{if } \boldsymbol{c_i} >= \boldsymbol{\lambda_i} \\ -\boldsymbol{\lambda_i} & \text{if } \boldsymbol{c_i} <= -\boldsymbol{\lambda_i} \end{cases} \quad (10)$$

5.1 Relation of Shape-Constrained LP Flocks to ASMs and AAMs

In this section, we will discuss the relation between the proposed method and that of ASMs and AAMs. The proposed selected LP tracker method is similar to that of ASMs in the use of the shape model. Both methods initially use some form of feature specific trackers that provides a hypothesis of the facial feature configuration. Subsequently, a shape model is then used to correct inaccuracies in this given hypothesis. However, there are important differences. Firstly, ASMs rely heavily on the shape model to generate a fairly accurate hypothesis. This hypothesised shape is then refined using local feature trackers. These feature trackers rely on the fact that they are initially placed (using the hypothesised shape) near their corresponding features. Our proposed method is effectively the opposite, with minimal reliance on the shape constraints. Most of the robust and accurate tracking of facial features is effectively performed by independent LPs flocks. Should a minority of these LPs lose track (e.g. due to severe occlusions), the shape constraint serves to "pull" them back to their region of convergence, where they can continue to track in the next frame.

In relation to AAMs, we find that the proposed LPs are person specific and work mainly on texture information (i.e support pixel values and differences). Additionally, the proposed shape constrained LPs require only a small training set, usually less than 20 examples as described in Section 6. Incidentally, it is these attributes that make the LP method closer to AAMs, which are also generally person-specific, rely more on texture for tracking and require a relatively small training set.

1. The homest package[17] provided the C++ implementation of this method.

6 Convergence Experiments

This section describes experiments carried out to further understand the following points: firstly how the proposed method's robustness is affected with regards to different parameters, in particular the usefulness of the bias component in an LP (Section 6.2), the number of member LPs used in a flock (Section 6.3) and the role of different sized LPs in a multi-scale LP flock model (Section 6.4); secondly, a comparison of the tracking performance of multi-scale LP flocks is compared to those from an AAM method [20] (Section 6.7). AAMs were chosen due to their known performance in tracking facial features without requiring a large number of training examples.

In order to quantify the robustness of the proposed method at different parameter settings and to compare against AAMs, a test measuring convergence to groundtruth from displaced data was used. The results effectively provides information on how robust a method is to initialisation conditions that are of increasing distance from the groundtruth location across all frames in the test sequences. To achieve this, for each test image in the test sequence, all the tracked points are initially displaced from their groundtruth positions. These displacements range from 1 to 60 pixels in radius (Figure 5c,d). A tracking method (e.g. proposed LP method or AAMs) is then applied to these corrupted positions. A count of how many points eventually fall within a nominal radius of the groundtruth data is made. Here, we have set the convergence radius to 5 pixels. It is then possible to obtain the percentage of points across all test

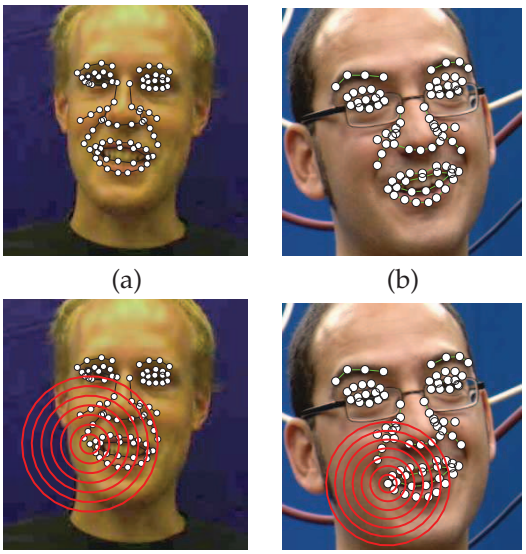


Fig. 5: Shown are all 83 tracked points for the webcam data(a) and the SD camera data(b). The displacement error sizes (up to 60 pixels) with respect to the face is illustrated in (c) and (d) as red circles of different sizes.

images and tracked points that have fallen within the convergence region for a given range of displacement noise.

6.1 Experimental Setup

For the convergence experiments, two classes of data were captured: a standard definition (PAL) resolution camera; and a low-cost 640x480 resolution webcam. Using the SD camera, two separate video sequences from each of 3 different subjects engaged in general discussion with another person were captured. Similarly, using a webcam, two separate video sequences from each of 4 different subjects discussing the contents of pictures with another person were captured. For each subject, one sequence was retained for training purposes and the remainder for testing. The test sequences contained approximately 800 to 1000 frames. In order to groundtruth the data, semi-automatic labelling was used, since hand labelling every frame would have been too time consuming. In total, 83 points on the face were tracked as shown in Figure 5a,b. In order to track all of these points, 83 independent multi-resolution LP flocks were used. For training the LPs' linear mapping matrix H, around 13-15 images extracted from the training sequence was used. It is important to note that the training size is very small in comparison to the size of the test data ($< 0.15\%$). Each LP had a support region of half the size of its associated scale, with 80 support pixels randomly placed within this area. A visualisation of various converged results for image frames from both webcam and SD images are shown in Figure 11. The results in Figure 11 show lines that join together different facial features. These lines are

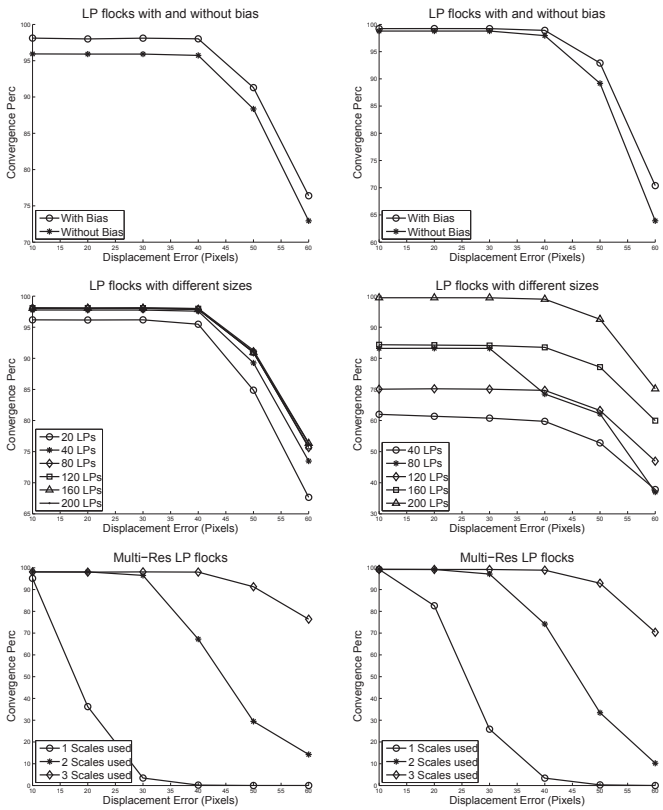


Fig. 6: The convergence graphs for the different LP parameters settings for webcam sequences (a) and SD camera sequences (b). For more details, refer to Section 6.

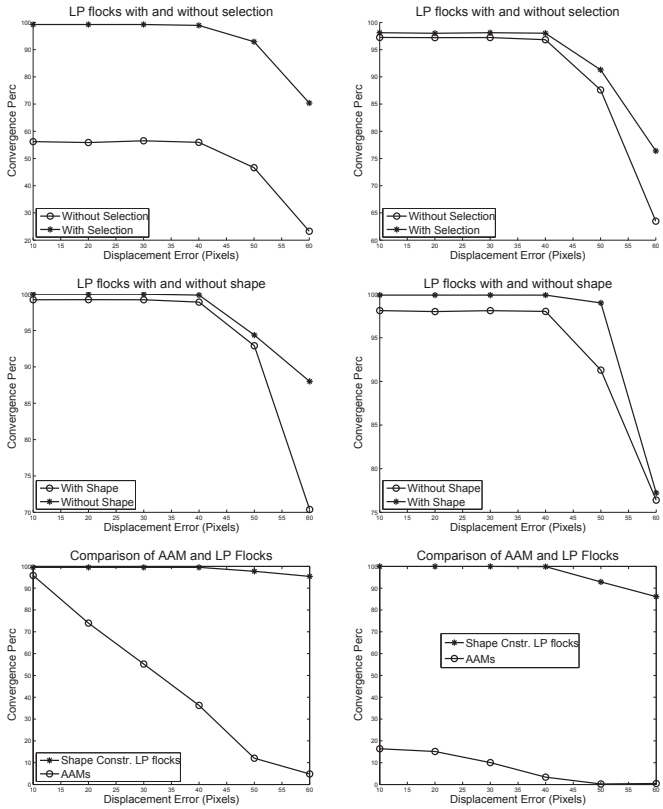
purely for visualisation purposes and play no role in the tracking process.

6.2 Biased LPs vs Original LPs

In order to analyse the improvements due to the addition of a bias in the LPs, convergence tests were performed on selected LP flocks with 3 scales with and without the bias term. All LP flocks have 200 members each. The results can be seen in the first row of Figure 6. We see that for the webcam data, there is a consistent improvement across the entire error range. In the SD dataset, we find that the bias provided increased robustness, especially when the noise is large.

6.3 LP Flock Size

The next set of experiments were used to determine the effect of increasing the number of members in an LP flock. To this end, tests were performed using LP flocks with the following number of members: 20,40,80,120,160,200. These flocks underwent the selection process. Additionally, 3 scales were used as before: 10, 40 and 60 pixels. The results can be seen in the second row of Figure 6. For the webcam database, it was



(a) (b)

Fig. 7: The convergence graphs for the different LP parameters settings for webcam sequences (a) and SD camera sequences (b). For more details, refer to Section 6.

interesting to note that increasing the flock size beyond 80 members did little to improve the robustness of the method. However, the case was different in terms of the SD dataset, whereby increasing the sizes resulted in continuous improvement in robustness.

6.4 Role of Different LP Flock Scales

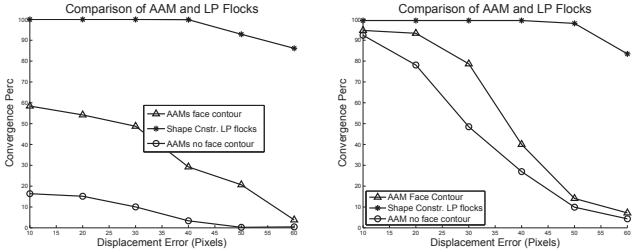
In order to analyse the role of the different LP sizes in a multi-resolution LP flock, tests were performed with LP flocks with an increasing number of scales. Here, a maximum of 3 scales are used, of sizes 10, 40 and 60 pixels respectively. Each individual scale has selected flocks with 200 members. Tests were firstly performed with LP flocks with LPs of size 10 pixels. Next, LP flocks of size 40 were added and the convergence tests repeated. Finally, the LPs of size 60 were added and the results compiled. The obtained convergence graphs are shown on the third row of Figure 6. It can be seen that adding the larger scales provide significant improvements on the convergence scores when the noise level is high (i.e. 60 pixels radius).

6.5 LP Selection

To better understand how much improvement is obtained from the LP probabilistic selection method, convergence tests with the selection method switched on and switched off were performed. For these experiments, 200 LPs were used per flock, and 3 scales were used: 10,40 and 60 pixels. The results can be seen in the first row of Figure 7. Here, it can be seen for both SD and webcam quality data that the selection process results in LP trackers that are significantly more accurate and robust than those with LP flocks whose members were randomly placed.

6.6 Role of Shape Constraints

In order to analyse the benefits of using shape constraints, convergence tests were performed using multi-resolution LP flocks with and without shape constraints. These LP flocks are multi-resolution with 3 scales (10, 40 and 60 pixels) and their members consist of biased-LPs. The results can be seen in the second row of Figure 7. The results indicate that adding shape constraints further improves the robustness and accuracy of the proposed method across the entire range of displacement errors. We can see the biggest improvement in the form of robustness to large amounts of noise from the SD camera data. Here, when the noise displacement is 60 pixels, there is a 17% improvement in the convergence results. This improvement is due to the shape constraint essentially "pulling back" the few unstable LP flocks that would not have converged to the solution when noise is large. However, in the case of the webcam images, we do not see a significant improvement. One reason for this is due to the size of the faces in the webcam sequences. At 60 pixels displacement error, we find that many facial feature trackers are initialised outside the face region, on background clutter. When this happens, the linear mappings in the LP flocks can often be invalid, since the background's pixel values can be arbitrary. As a result, these LP trackers can often fail to converge to their corresponding feature.



(a) (b)

Fig. 8: Convergence results comparing AAMs with and without face contour information against the proposed LP tracker for (a) SD camera sequences and (b) webcam camera sequences.

6.7 Comparison with AAMs

To compare the proposed method against AAMs, we trained separate AAMs for each subject using exactly the same training data used for the LP flocks. The implementation of AAMs used was provided by the authors of [28]. The convergence scores using the AAM method was then obtained and compared against multi-resolution LP selected-flocks with 3 scales and 200 member LPs (third row of Figure 7). We find that the AAM performs poorly when the displacement error is high. In the webcam data, when the displacement error is small (< 10 pixels), the performance of the AAM is comparable to the proposed method. However, it is surprising that the AAMs performed poorly when the image quality was higher for the SD dataset. This was despite it using both shape and texture constraints that were employed in a multi-resolution optimisation process. To further investigate the performance of the AAMs, an additional experiment was performed where the face contour was also included, thus providing the AAM with information from the entire face. The convergence results are shown in Figure 8. It can be seen that there is significant improvement in the performance of the AAMs. However, even with this improvement, performance remains lower than that of the LP flocks.

6.8 Analysis of Selected Features

In this section, we analyse consistency of visual support for tracking across different subjects. This can be achieved by looking at the average distances of support pixels for an LP flock of a single facial feature point across different subjects. Specifically, we firstly extract and group all the support pixels offsets of a subject's LP flock for a particular facial feature point. This is then repeated for each subject. Suppose we have S number of subjects, we have S sets of support pixel offsets. We then calculate the average minimum Euclidean distance of each subject's support pixel offset to that of the other subjects. From this, we can quantify the degree of similarity of a facial feature's LP flock across different subjects.

The above analysis is then performed separately on the LP flocks for the SD data and webcam data used in the convergence tests of Section 6. For both of these datasets, each LP flock has 200 members. Each member in turn has 30 support pixels, resulting in each flock having a total of 6000 support pixels. The results can be seen in Figure 14 and Figure 15 for the SD and webcam sequences respectively. We find that the LP flocks for the SD data are remarkably similar, with all average distances of support pixel distances across subjects being less than 1.6 pixels (Figure 14a). This can also be visually seen by directly showing the LP flocks' support pixels for different subjects (Figure 14c). The results from the webcam data, however, are less consistent and have a higher average inter-subject support pixel distances (Figure 15a). An interesting observation is on the middle point on the lower inner-lip. Here, the differences in the LP flocks is greatest, with the average support pixel distance being 3.7 pixels. These differences in the LP flocks can also be visualised by looking at Figure 15c.

No. Train Images	Mean Err	Std Dev
5	5.8	6.6
10	4.6	4.5
20	4.3	4.3
40	4.2	4.5
AAM (unknown) [9]	5.5	1.8
SubAdaBoost (none) [9]	7.8	1.9

TABLE 1: Table showing average tracking error in pixels for the ASL sequences (Section 7.1) using the proposed method trained with different training set sizes. For comparison, the errors for the AAM and SubAdaBoost methods [9] are also shown.

7 Tracking Experiments

In order to evaluate the tracking performance of the proposed method, two sets of experiments were performed. The first set uses an existing American Sign Language (ASL) nonmanual database that was also used in [9] where tracking groundtruth information was available. There are significant amounts of occlusions of the face from the hands in this database and thus allows us to evaluate how well the LP flock method will cope with such an issue. The second set of experiments attempts to track three challenging sequences obtained from Youtube (Section 7.2). These sequences allow for the evaluation of tracking performance in the presence of various factors such as significant noise artefacts as well as extreme facial expressions and poses.

7.1 ASL Database

This experiment uses video sequences of ASL nonmanuals (facial displays). In total, there are 35 sequences, with approximately 77 frames for each sequence. There are significant occlusions of the face from the hand as the subject perform various signs. Additionally, there are also large variabilities in facial expressions and pose. All sequences were captured at the resolution of 720x480 pixels, of which the face occupies an area that is approximately 300x250 pixels. There are a total of 7 subjects in this database with 5 video sequences for each subject.

For the ASL sequences, a total of 98 facial features were tracked. In terms of training, a separate LP tracker was trained for each subject. For each subject, approximately 17 training frames were extracted from two sequences. Tracking was then performed on the remaining three sequences. Since groundtruth information obtained through manual labelling, was available for 34 video sequences, it is possible to obtain statistics on the tracking error. It was reported in [9] that the tracking error of AAMs on the test sequences are as follows: average error of 5.5 pixels with the standard deviation of 1.8 pixels. However, as the precise number of training data used

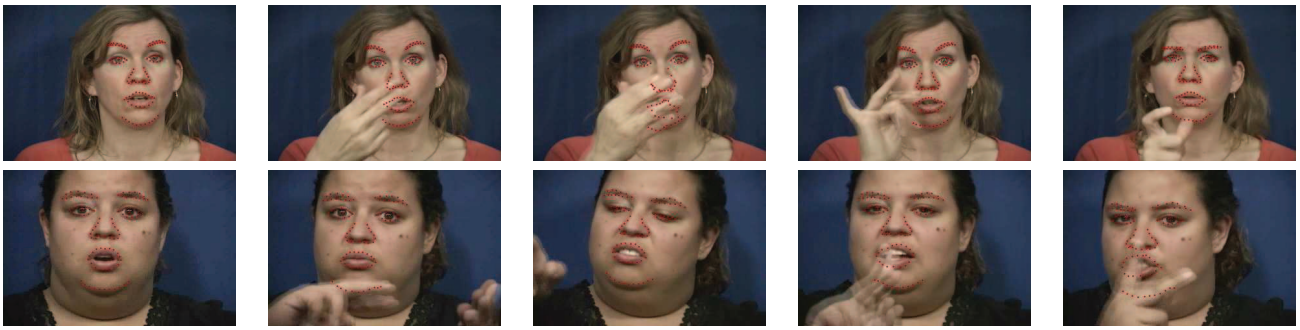


Fig. 9: Examples of frames from the ASL sequences described in Section 7.1.

Youtube Sequence 1 Youtube Sequence 2 Youtube Sequence 3

Fig. 10: Examples of frames from the Youtube test tracking sequences described in Section 7.2.

there is not known, a direct comparison with this error is difficult.

The tracking performance of the LPs on the ASL database is then evaluated across a set of training sets using increasing numbers of training examples. Specifically, training sets of 5, 10, 20 and 40 images were used to train the LPs. These LPs were then used to track all 35 sequences in the database and the tracking errors statistics were extracted. Some examples of the tracking results from the proposed LP flocks can be seen in Figure 9. The resulting tracking errors can be seen in Table 1. The results confirms the intuition that the larger the training set, the lower the tracking errors. This is to be expected as a small set of training images will not provide enough variability in facial configurations for learning an accurate linear mapping in the LPs. However, we can also see that after 20 images, adding more training images does not result in significant improvements in the training performance. This is because, adding redundant images and thus training information only results in an overcomplete system when learning the linear mappings of the LP trackers.

In comparison to the AAM method, Table 1 shows that, with the exception of the smallest training set of 5 images, the remaining larger training sets resulted in LPs with tracking mean errors that are smaller than the AAM method. It should be noted that the specific number of training examples used for the AAM method is unknown. However, it was indicated in [9] that two sequences were used for training, totaling approximately 140 training examples. For completeness, a comparison is also made against the SubAdaBoost method [9] which has a higher average pixel error compared with both the LP and AAM methods. However, this method was person-independent.

We note that the standard deviation of the LPs is higher than the AAM method, given a fairly similar average error (Table 1). The reason for this is that the proposed method performs less well under significant occlusions (e.g. the feature of interest is completely occluded by the hand) of the face. This is due to the LPs having only local contextual information, which will be occluded as well. On the other hand, AAMs perform more accurately under occlusions due to their use of more global facial context. This is also aided by the large size of the face in the test images. As will be shown in the next section, when the quality of the face image is neither as good or is considerably smaller, AAMs fare less well.

In summary, we find that each method has different advantages. However, a particular advantage can have a negative impact on the performance of other aspects of the method. An example of this is the advantage that the SubAdaBoost method is person-independent. However, this comes at a cost of feature tracking accuracy. A more detailed listing of the advantages and disadvantages of popular facial feature tracking methods can be found in Table 2.

7.2 Youtube Sequences

We also attempt to compare the performance of the proposed tracking method to that of AAMs in other

Tracking Method	Pros	Cons
Selected Biased LP	Smaller training size compared to AAMs. More accurate than AAMs. Low complexity. Able to track features independently due to automatic LP selection.	Person dependent. Vulnerable to occlusions.
AAM	Small training size required. More accurate than person independent methods. Potentially robust to occlusions due to use of entire face shape and texture for tracking.	Person dependent. Single facial feature point tracking not possible. The entire face shape and texture required for effective feature tracking.
SubAdaBoost	Person Independent. Training performed once. Automatic selection of tracking context for more effective tracking. Single feature point tracking possible.	Not as accurate as person dependent methods. Large initial training set required. Vulnerable to occlusions
ASM	Person Independent. Training performed once.	Not as accurate as person dependent methods. Large initial training set required. Single feature point tracking not possible due to reliance on shape model.

TABLE 2: Table summarising the advantages and disadvantages of the tracking methods compared in Section 7.1.

challenging sequences. To this end, Youtube was used as a source of sequences, all of which contain significant amounts of image noise, motion blurring, pose and expression changes as well as occlusions. These factors make it almost impossible to ground-truth the location of facial features in such sequences. Consequently, the tests performed above are not possible.

However, some form of performance measure that is indicative of the robustness of a method in automatically tracking facial features is still needed. One important trait to achieving good automated tracking is the number of manual reinitialisations that need to be done whilst processing a sequence. A large number of reinitialisations will require greater user involvement, thus making the system less convenient to use, and consequently having a negative impact on the user productivity.

Whilst it is not possible to directly quantify the tracking performance using average tracking errors, an indirect comparison by means of divergence between the LP tracking results and that of AAM trackers is still possible. Where the divergence is large and there is a significant difference between the tracking results from both methods on a particular frame, there is a large chance that tracking failure has occurred for one of the methods. These frames are then visually inspected to determine whether one or both methods have failed in tracking the features accurately. Where there has been a catastrophic tracking failure, the respective tracker is reinitialised manually. We can then compare the tracking robustness of the proposed method to AAMs by identifying the total number of times reinitialisation was required for these test sequences. For both methods, the tracker was initialised at the position from the previous frame.

7.2.1 Experimental Setup

For our tracking experiments, 3 low-quality sequences obtained from Youtube were used. Example frames from each of the sequences can be seen in Figure 10. For each sequence, a multi-resolution selected LP flock is trained using a set of training images described more specifically below. For all sequences, we have found that 200 member LPs for each flock, with each LP having 30 support pixels was adequate for successful tracking. The training size across different sequences varies according to the amount of facial variations present in the scene. Typically, a training set that capture the extreme facial configurations is manually obtained.

The first test sequence is of a subject being interviewed. It requires the tracking method to cope with the following issues: pose changes, significant motion blur and compression artefacts, occlusions of the face and extreme facial expressions. The test sequence contains a total of 1216 frames, each having a resolution of 480 by 360 pixels. In this sequence, the face was typically of the resolution of 160 by 170 pixels. To train the trackers, 18 images were extracted from the sequence and labelled with the positions of facial features of interest. For the proposed LPs, 3 scales were used, at 40, 20 and 10 pixels respectively. A multi-resolution AAM model was trained using exactly the same images and position labels.

The second sequence consists of a subject giving a speech. This sequence tests the ability of the tracker to cope with tracking features on a face of small resolution in the presence of pose and expression changes. This sequence is comprised of a total of 943 frames, each again

with a resolution of 360 by 480 pixels. However, in these sequences, the face was typically of size 80 by 85 pixels, half that of the first sequence, and drastically lower than the resolution of faces in the convergence experiment dataset presented in Section 6. The trackers were trained with 24 labelled images. LPs of 4 scales were used, at 40, 20, 10 and 5 pixels respectively. The smaller scale was required due to the low resolution of the face. A multi-resolution AAM model was again trained using the exact same images and position labels.

The third sequence is extracted from a children's programme broadcast by the BBC with a subject performing sign language as an insert into the broadcast footage. The image was cropped to isolate the subject, however, the background around the face region was still retained. This sequence exhibits significant amounts of interlacing artefacts, occlusions of the face and expression changes. Additionally, the face is also small in resolution, typically at 72 by 92 pixels. In total, the sequence contains 514 frames, each of resolution 470 by 376 pixels. The trackers were trained with 18 labelled images. LPs of 3 scales were used, at 20, 10 and 5 pixels respectively. The smaller scale was again required due to the low resolution of the face. A multi-resolution AAM model was again trained using the exact same images and position labels.

7.2.2 Tracking Results

The tracking speed in unoptimised C++ is about 20fps on a standard single-core processor PC. In the first sequence, we found that the proposed LP method suffered an unrecoverable tracking error 3 times, despite the presence of occlusions, extreme facial expressions and pose changes. In comparison, the AAM irrecoverably failed 40 times. Example frames where the AAM failed to track in the first sequence can be seen in Figure 13a with the corresponding LP tracker on the same frame shown in 13b. Here, we note that the result of the proposed method is considerably more accurate than that of the AAMs.

For both the second and third sequences, there are inaccuracies in tracking, but the proposed method always recovered without re-initialisation. In contrast, the AAM method required a total of 12 and 9 reinitialisations for the second and third sequence respectively. Again, we show the example tracking results at frames where AAMs required reinitialisation in Figure 13c,d and Figure 13e,f for the second and third sequence respectively. As before, we notice that the results for the proposed method are visually more accurate than that of AAMs. However, it should be noted that while increasing the number of training examples would have little effect on the LP tracker approach it is likely that AAMs would benefit as discussed in Section 7.1. A full verification of the above claims can be seen in the video sequences provided in the supplementary material.

8 Conclusions

In this paper, we described the use of a hierarchical multi-resolution tracking framework that uses linear predictors. Accurate and robust tracking is made possible by firstly introducing a biased-LP. Additionally, the visual tracking context for a facial feature is automatically identified as surrounding visual support using a probabilistic selection method. This allowed us to track a range of difficult feature points containing either too much visual variations (e.g. inner lip points) or ambiguous feature points with too little visual information (e.g. points on the cheek). Finally, the addition of shape constraints provides a simple but effective method for further improving the tracking performance of the selected LP flocks by correcting the occasional tracking failure of a small number of points. Experimental results based on convergence tests quantitatively show that the proposed method is more robust than existing AAMs. Crucially, this was achieved with a minimal training set of 15 to 25 images. We also show that the method can successfully track facial feature points in sequences that range from SD to Youtube quality. This was also achieved in the presence of fast motion, occlusions due moving hands, simultaneous pose and expression changes, as well as in conditions of low image quality. We have also analysed the consistency of the selected visual support across different individuals. We have found that for certain visual features, there is consistent use of similar parts of the face for tracking the same facial feature point. In terms of future work, further investigations need to carried out on using other visual cues, for example, colour, edge strengths and orientations. The key to effectively using these different cues will lie integrating an information fusion process into the linear prediction framework proposed in this manuscript.

Acknowledgements

The work has been supported by the EPSRC project LILiR (EP/E027946/1) and the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no 231135 DictaSign. We would also like to acknowledge the authors of [28] and the following people: Yuxuan Lan, Barry Theobald and Richard Harvey from the University of East Anglia for providing us with an implementation of the AAM method and expertise in using it to perform the experiments described in Section 6.

References

[1] K. Atul, Y. Huang, and D. Metaxas. Tracking facial features using mixture of point distribution models. In *Proceedings of Indian Conference on Computer Vision Graphics and Image Processing*, pages 492–503, India, December 2006.

[2] 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.

[3] 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.

[4] M. Burl, T. Leung, and P. Perona. Face localization via shape statistics. In *Proceedings. 1st IEEE International Conference on Automatic Face and Gesture Recognition, 1995.*, pages 154–159, 1995.
[5] I. Castelli, M. Maggini, S. Melacci, and L. Sarti. Auto associative neural network based active shape models. In *Proceedings of 8th IEEE International Conference on Face and Gesture*, pages 1–6, 2008.
[6] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.
[7] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.
[8] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *Proceedings of British Machine Vision Conference*, pages 929–938, 2006.
[9] L. Ding and A. Martinez. Features versus context: An approach for precise and detailed detection and delineation of faces and facial featurse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.
[10] F. Dornaika and J. Ahlberg. Face model adaptation using robust matching and active appearance models. *Applications of Computer Vision, IEEE Workshop on*, 0:3, 2002.
[11] F. Dornaika and J. Ahlberg. Fitting 3d face models for tracking and active appearance model training. *Image and Vision Computing*, 24(9):1010 – 1024, 2006.
[12] M. Hamouz, J. Kittler, J.-K. Kamarainen, P. Paalanen, and H. Kalviainen. Affine-invariant face detection and localization using gmm-based feature detector and enhanced appearance model. In *Proceedings. 6th IEEE International Conference on Automatic Face and Gesture Recognition, 2004.*, pages 67–72, May 2004.
[13] J. Hoey. Tracking using flocks of features, with application to assisted handwashing. In *Proc. of British Machine Vision Conference*, pages 367–376, 2006.
[14] 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.
[15] 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.
[16] X. Liu. Discriminative face alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1941–1954, 2009.
[17] M. Lourakis. homest: A c/c++ library for robust, non-linear homography estimation. [web page] `http://www.ics.forth.gr/~lourakis/homest/`, Jul. 2006.
[18] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
[19] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(1):135 – 164, November 2004.
[20] 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.
[21] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. In *Proceedings of the 10th European Conference on Computer Vision*, pages 504–513, 2008.
[22] M. Nguyen and F. de la Torre. Local minima free parameterized appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
[23] E. Ong and R. Bowden. Robust lip-tracking using rigid flocks of selected linear predictors. In *Proc. 8th IEEE Conf. on Automatic Face and Gesture Recognition*, 2008.
[24] E. Ong, Y. Lan, B. Theobald, R. Harvey, and R. Bowden. Robust facial feature tracking using selected multi-resolution linear predictors. In *Proceedings of the 12th International Conference on Computer Vision*, 2009.
[25] I. Patras and E. Hancock. Regression tracking with data relevance determination. *IEEE Conference on Computer Vision and Pattern Recognition.*, pages 1–8, June 2007.
[26] F. Sukno, S. Ordas, C. Butakoff, S. Cruz, and A. Frangi. Active shape models with invariant optimal features: Application to facial analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1105–1117, 2007.
[27] J. Sung, T. Kanade, and D. Kim. Pose robust face tracking by combining active appearance models and cylinder head models. *International Journal on Computer Vision*, 80(2):260–274, 2008.
[28] B. Theobald, I. Matthews, and S. Baker. Evaluating error functions for robust active appearance models. In *Proceedings of the Seventh International Conference on Face and Gesture Recognition*, pages 149–154, 2006.
[29] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.
[30] 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.
[31] H. Wu, X. Liu, and G. Doretto. Face alignment via boosted ranking models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
[32] Z. Wu, P. Aleksic, and A. Katsaggelos. Lip tracking for mpeg-4 facial animation. In *Proc. of the 4th IEEE Conf. on Multimodal Interfaces*. IEEE Computer Society, 2002.
[33] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 535–542, June 2004.
[34] K. Yow and R. Cipolla. A probabilistic framework for perceptual grouping of features in human face detection. In *Proceedings. 2nd IEEE International Conference on Automatic Face and Gesture Recognition, 1996.*, pages 16–21, 1996.
[35] A. Yulle, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.
[36] L. Zhang, H. Ai, S. Xin, C. Huang, S. Tsukiji, and S. Lao. Robust face alignment based on local texture classifiers. In *IEEE International Conference on Image Processing*, volume 2, pages II–354–7, September 2005.
[37] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87 – 119, 1995.
[38] K. Zimmermann, J. Matas, and T. Svoboda. Tracking by an optimal sequence of linear predictors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):677–692, 2009.

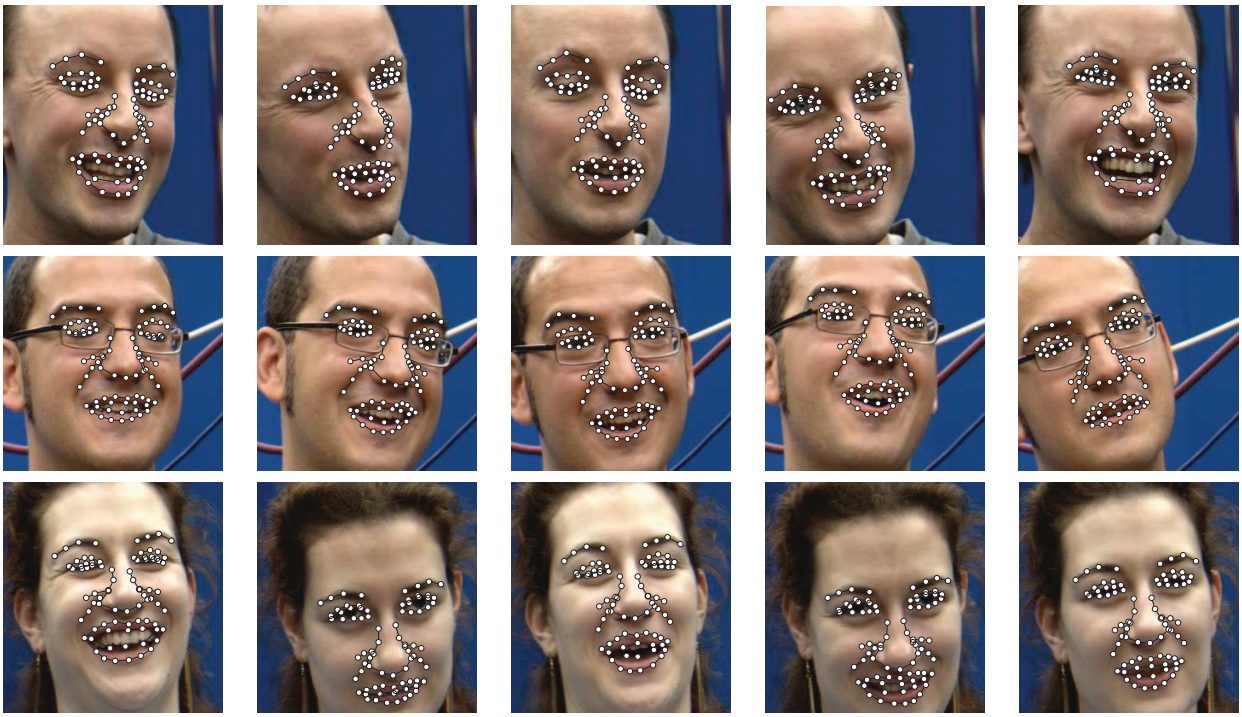


Fig. 11: Results of the convergence tests for displace facial features in SD camera sequences.

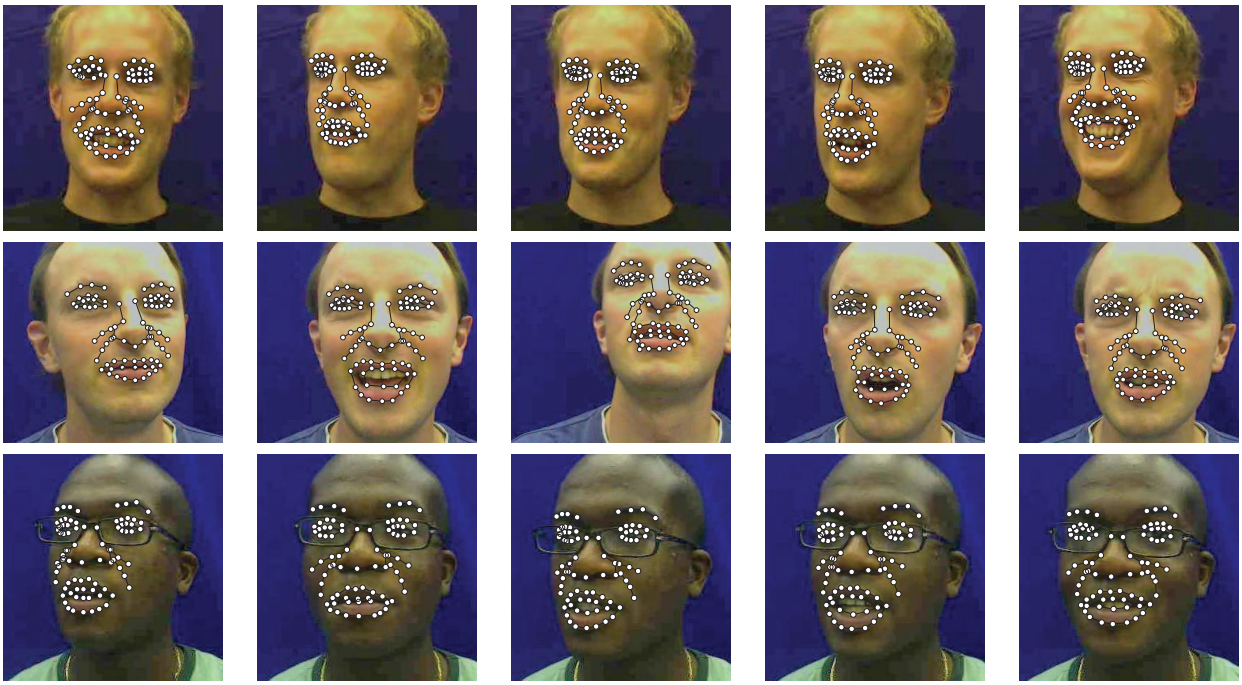
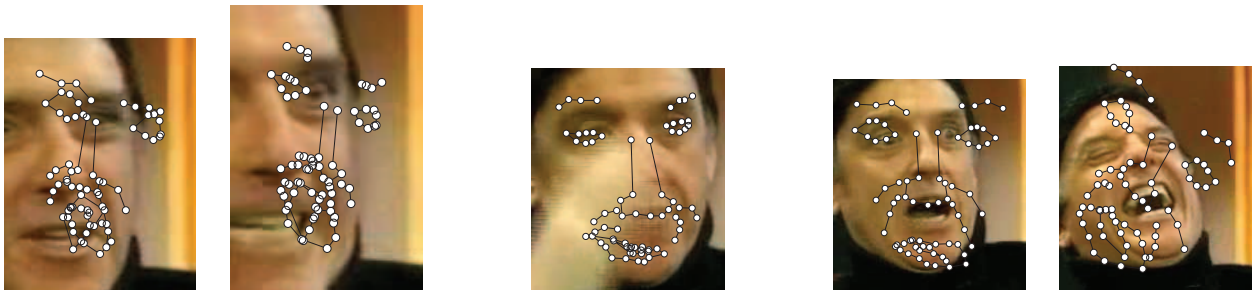
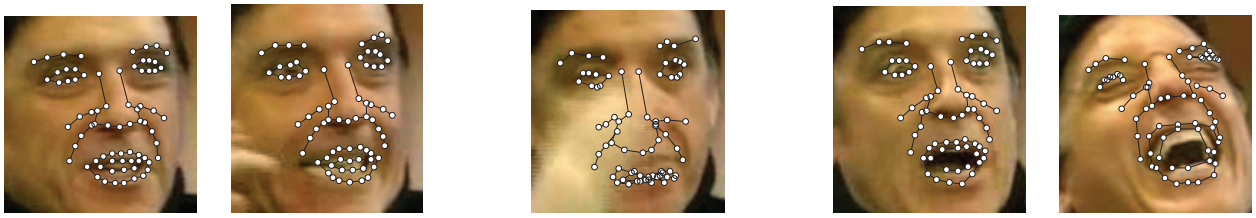


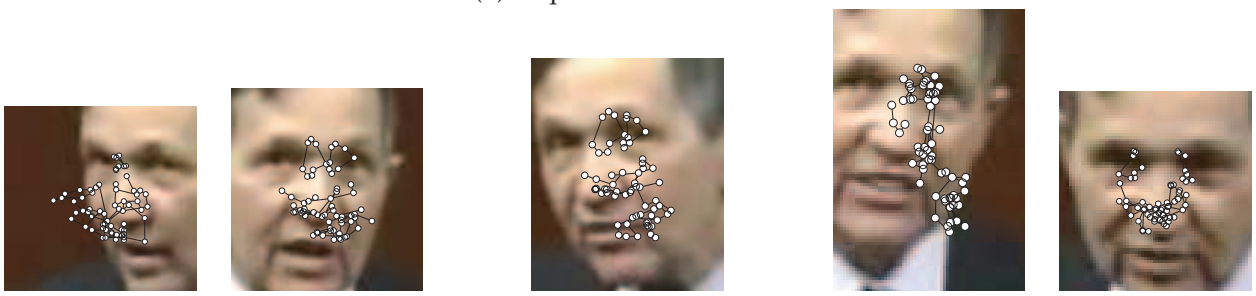
Fig. 12: Results of the convergence tests for displaced facial features in webcam sequences.



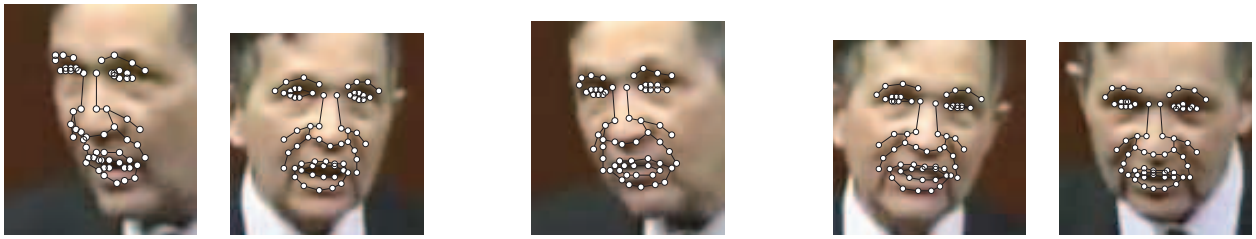
(a) Sequence 1 AAM Results



(b) Sequence 1 LP Results

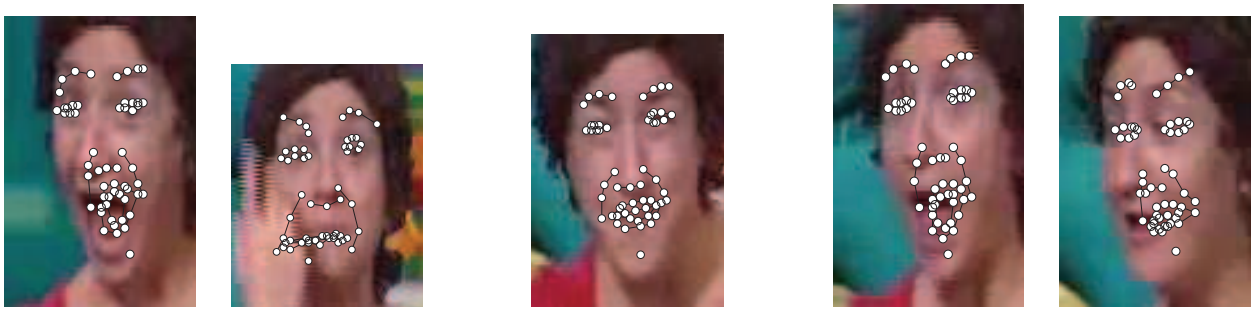


(c) Sequence 2 AAM Results



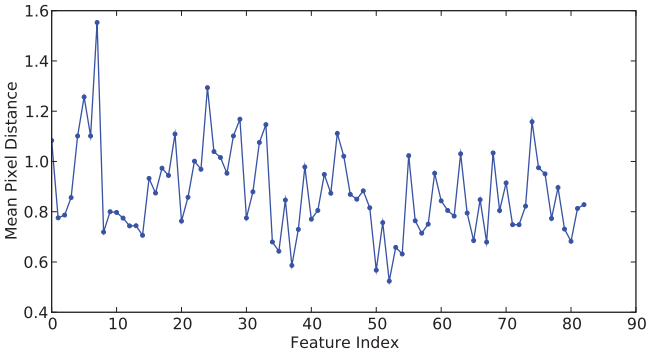
(d) Sequence 2 LP Results

(e) Sequence 3 AAM Results

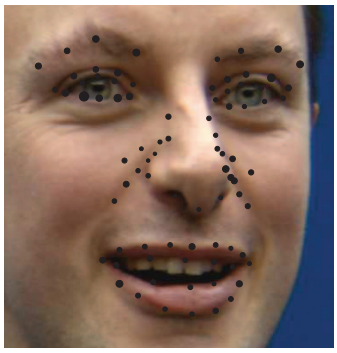


(f) Sequence 3 LP Results

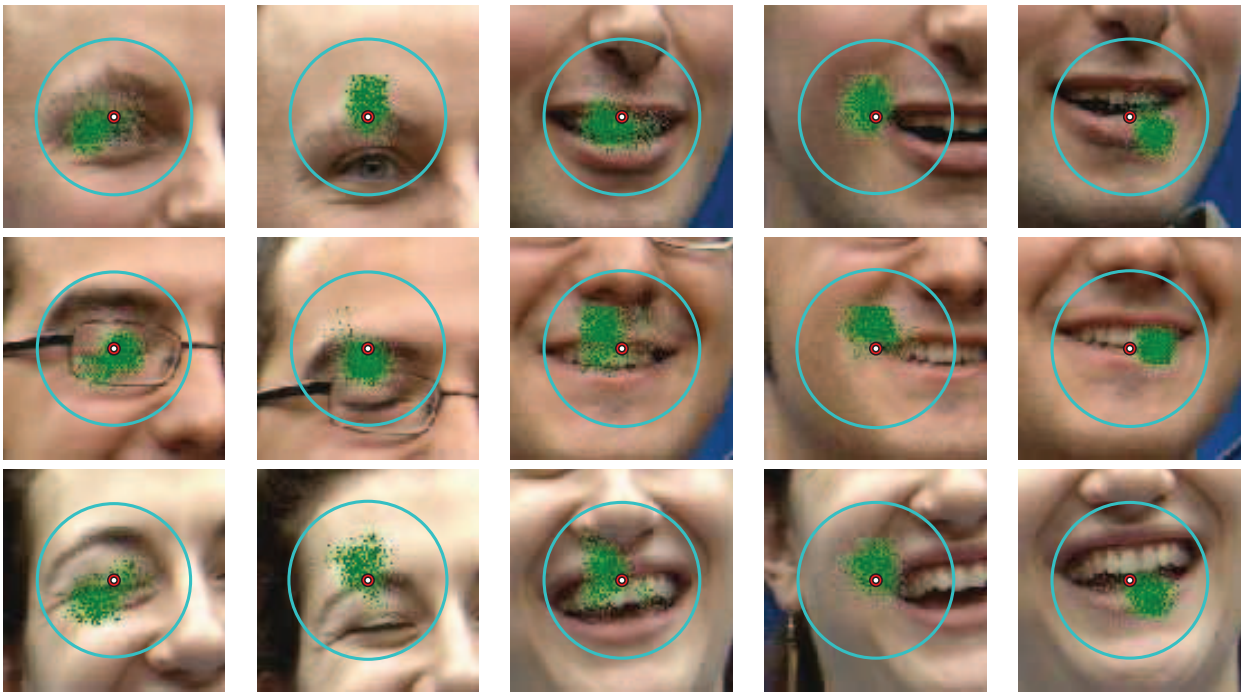
Fig. 13: Tracking results on the frames prior to AAMs being reinitialised when tracking the challenging sequences described in Section 7. For comparisons, we also show the results from the proposed LP method. Since the faces in the original images are quite small, to see the tracking results in greater detail, the results shown are cropped images that are 1/3 larger than the area of the tracked results.



(a) Average distance of support pixels for LP flocks across different subjects for all the facial features.

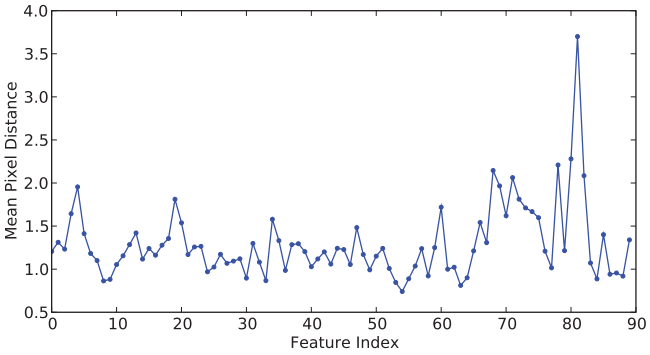


(b) Visualisation of the inter-subject average distances shown in 14a. The size of the black circles are proportional support pixel distances.

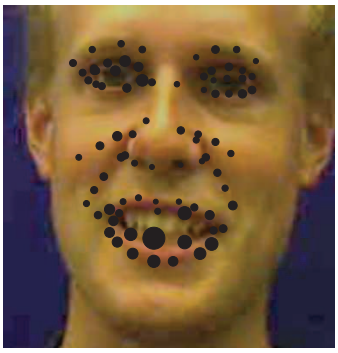


(c) SP Vis

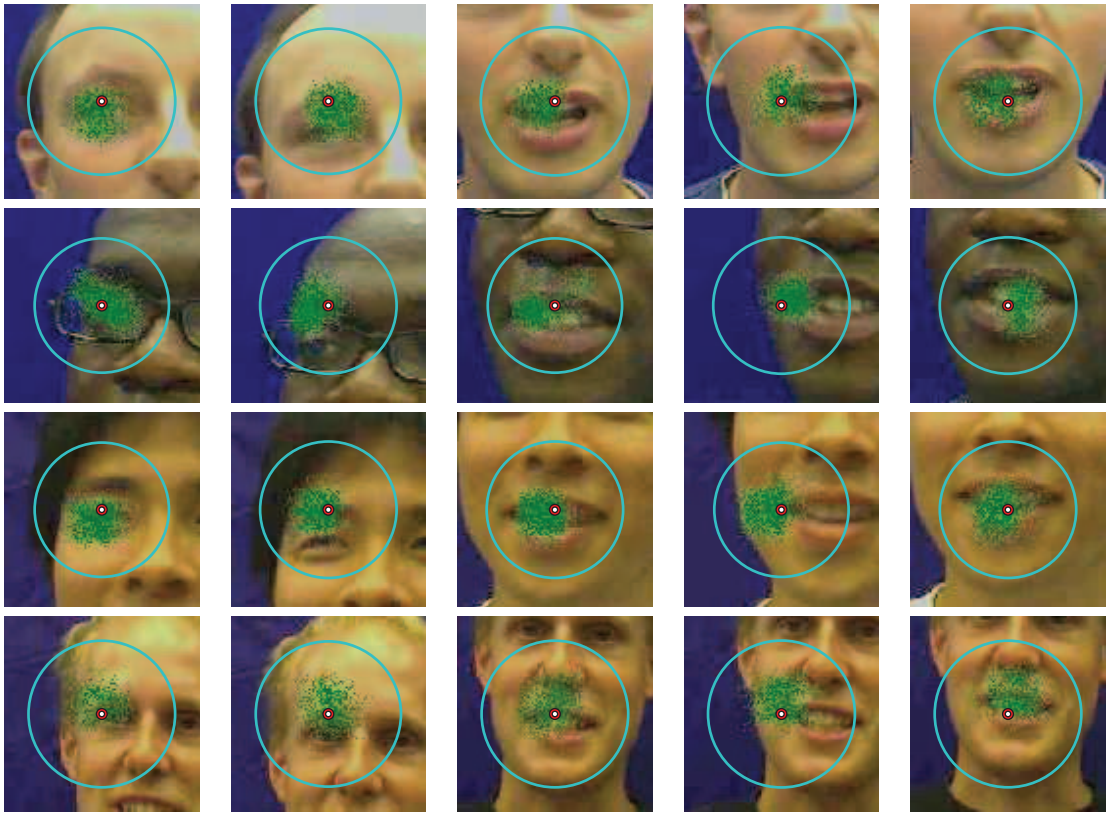
Fig. 14: Results of the similarities of the visual regions selected for tracking various facial features based on the support pixels of their corresponding SP flocks on SD data.



(a) Average distance of support pixels for LP flocks across different subjects for all the facial features.



(b) Visualisation of the average distances shown in 15a. The size of the black circles are proportional support pixel distances.



(c) Visualisation of the average distances shown in 14a

Fig. 15: Results of the similarities of the visual regions selected for tracking various facial features based on the support pixels of their corresponding SP flocks on webcam data.