Approximate Structured Output Learning for Constrained Local Models with Application to Real-time Facial Feature Detection and Tracking on Low-power Devices

Shuai Zheng, Paul Sturgess and Philip H. S. Torr

***Abstract*— Given a face detection, facial feature detection involves localizing the facial landmarks such as eyes, nose, mouth. Within this paper we examine the learning of the appearance model in Constrained Local Models (CLM) technique. We have two contributions: firstly we examine an approximate method for doing structured learning, which jointly learns all the appearances of the landmarks. Even though this method has no guarantee of optimality we find it performs better than training the appearance models independently. This also allows for efficiently online learning of a particular instance of a face. Secondly we use a binary approximation of our learnt model that when combined with binary features, leads to efficient inference at runtime using bitwise AND operations.**

We quantify the generalization performance of our approximate SO-CLM, by training the model parameters on a single dataset, and testing on a total of five *unseen* benchmarks. The speed at runtime is demonstrated on the ipad2 platform. Our results clearly show that our proposed system runs in real-time, yet still performs at state-of-the-art levels of accuracy.

I. INTRODUCTION

Facial feature detection involves localizing the facial landmarks [1] (e.g. spots on the eyes, nose, mouth) in a cropped image containing one face, as shown in Fig. 1. Facial feature tracking involves the consistent prediction of these facial landmarks over a temporal sequence of images. These techniques play essential roles in many popular computer vision and graphics applications such as face recognition [2] and face animation [3]. Recently, most smart mobile devices are equipped with frontal and back cameras [4]. In order to bring these applications to mobile devices, facial feature detection and tracking are required to work more efficiently and accurately [5], [6]. In this paper, we combine the state-of-the-art approaches [7], [8] in facial feature detection and tracking in order to deliver a system that is accurate and runs in real-time on both PC and low-powered smart mobile devices.

In general facial feature detection can be posed as a structured global optimisation problem, where the objective is to fit the sparse set of face landmarks to a pre-trained model of both the face structure *and* appearance. This problem has too many parameters making it computationally expensive to solve in such a general form, so approximations are necessary.

One way to approximate the problem is to first split the parameter space into a set for shape and another for appearance, then perform two stage learning: first learning the global face structure model, and then learning the global face appearance model, given the structure. This type of approach is taken by Constrained Local Models (CLMs) [10], [11], [7], [9], where the shape and appearance models are learnt over all landmarks together using Principle Component Analysis [10]. However, perhaps due to larger variations in appearance than shape, the appearance model tends not to generalise as well the shape model (or it would need large amounts of training data to do so). This had led to a focus on improving this stage [11], [7], [9]. Particularity of relevance to us is [9] where they found that a decomposition of the landmarks appearance into a set of independent pieces, one for each landmark, leads to better generalisation when the parameters for each piece (landmark) are learnt with Support Vector Machines (SVM). The inference in these types of models is extremely efficient as cleverly the learnt face shape can also be decomposed, leading to a set of feasible locations for each landmark, thus only a set of localised inferences need to be performed at run time i.e. each independently learnt appearance model is evaluated within a local region, then model fitting is performed over these local responses. However, the strong assumption that the appearance of a face landmark is independent of all other landmarks is not supported well i.e. the appearance of one of your eyes is not independent of the other. This has lead to a resurgence in joint learning for facial feature detection [12].

Another way to simplify the problem, in order to make the joint learning feasible, is via a second order approximation of the shape. This is done by adding local graph (usually a tree) based constraints on the output space where each

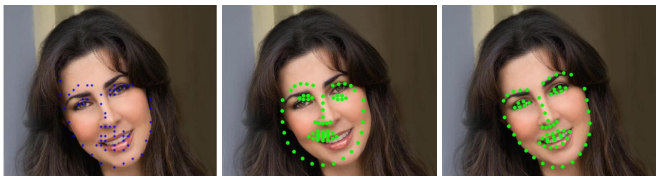


Fig. 1. **Facial feature detection results.** This figure shows facial feature detection results. The right image shows the results obtained with our jointly learnt CLM appearance model with BRIEF descriptors. The middle image shows the results obtained by CLM with independently learnt appearance models, with BRIEF descriptors [9]. The left is the result is obtained by the state-of-the-art graph-based approach [8]. See §V for further qualitative results.

The authors are with Oxford Brookes Vision Group, Oxford Brookes University, Oxford, U.K. szhengcvpr@gmail.com {paul.sturgess,philiptorr}@brookes.ac.uk

landmark is represented as a vertex and connections between then as edges. The joint learning of such models has been formulated as a Conditional Random Field (CRF) based [13] and a Structured Output Support Vector Machines (SO-SVM) [14]. Whilst these approaches are accurate due to the joint learning, additional computation is required to perform inference over the graph, making them too heavy to run on in real-time on modern low-power devices such as the popular ipad tablet. This is especially true as the number of nodes (landmarks) that are to be detected increases beyond the tens.

In this paper we aim to maintain both the efficiency of CLM inference and the good generalisation properties of the SO-SVM approaches. SO-SVM requires exact optimization at the inference stage, CLM has only a local optimization. However in general, the optimization is very near a global optimum. Hence, we propose using CLM to approximate the optimal inference in SO-SVM.

As well as giving a boost in accuracy by exploiting SVM training without assuming independence of landmark appearance, we also give a boost in run-time inference speed. This is achieved by using a binary approximation of our learnt joint appearance model, that when combined with a binary representation [15] (other possibilities could be [16], [17]) for the face features can be calculated via fast dot-product calculations in the form of bitwise AND operators. Now that we have an efficient facial feature detector, we also need a method for tracking these features through time. For tracking of our facial feature detector through time we use the Tracking-Learning-Detection (TLD) scheme similar to [18]. We do this for two reasons. Firstly it is an highly efficient method that can meet our real-time requirements when combined with our feature detector. Secondly, it is dynamic, which means that we can actually increase accuracy over deploying our pre-learned model i.e. the model is refined through time with-respect-to the targeted face.

We quantify the generalization performance of our SO-CLM, by training the model parameters on a single dataset, and testing on a total of five *unseen* benchmarks. The speed at runtime is demonstrated on the ipad2 platform. Our results clearly show that our proposed system runs in real-time, yet still performs at state-of-the-art levels of accuracy.

In summery our contributions are three fold:

- We train the popular CLM approach by introducing structured output optimisation to jointly learn the local appearance models.
- We employ model binary approximation scheme, which allows us to efficiently compute response maps by dot-product calculations when we use BRIEF as our low-level feature descriptor.
- We present a face landmark localization method on mobile device, which is integrated into a tracking-learning-detection (TLD) [18] system that allows us to robustly and efficiently detect the whole face and its feature landmarks through time.

II. Inference for Constrained Local Model

The Constrained Local Model (CLM) framework [19] consists of appearance and shape models. The CLM shape model includes mean shape, eigenvectors, and eigenvalues, which describe the variation of the possible spatial distribution of facial features. Facial features are represented by a consistently ordered vector of N landmarks $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, y_N]$; with each landmark being a single $2D$ coordinate $\mathbf{y}_i = \{u, v\}$, as illustrated in Fig. 1. The CLM appearance model for the ith facial feature is encoded by a linear classifier with weights $\mathbf{w}_i$. In the inference stage, as shown in Fig. 2, there are 3 steps summarized as follows:

1) We initialize the location of the facial features as follows: We locate a face region of interest (ROI) by applying face detector [20], we set the locations of the initial facial feature landmark on the face ROI by using the coordinates of the mean face in CLM shape model.
2) Extract feature descriptors in the local region that are around the coordinate of each facial feature landmark prediction. At each pixel in the local region of each facial feature landmark around we compute the dot product of the features with facial feature classifier $\mathbf{w}_i$, in order to generate the ith response score map.
3) Given the N response maps, one for each landmark, generated in stage 2, we find the set of facial landmarks constrained by the shape space that maximize the joint response. In order to do this we use the mean-shift algorithm implementation presented in [7].

Next we give more details of our implementation of CLM inference:

Generate a reference face shape: Our facial feature detection assumes that we have 4 coordinates of a bounding-box around a face. We locate the bounding-box, or face ROI, with the popular *Viola-Jones* detector [20]. Given this ROI we can crop the located face, rotate, and re-rescale it to a know reference frame from the CLM shape model. We then set each of the N landmark's coordinates to their mean value given by the CLM shape model.

Compute the response maps: For the ith estimated landmark in the face model $\mathbf{y}_i \in \mathbf{y}$ we define an ith local response image where each pixel, (u, v), takes a real response value $\psi_i^A : (u, v) \rightarrow \mathbb{R}$, which we calculate using an efficient dot product $\langle , \rangle$

$$\psi_i^A(u, v) = \langle \mathbf{w}_i, \mathbf{d}(u, v) \rangle \tag{1}$$

between the ith CLM appearance model $\mathbf{w}_i$ and an appearance feature $\mathbf{d}(u, v)$. For our appearance feature $\mathbf{d}$ we use the BRIEF [15] descriptor extracted from a 64×64 support region around each pixel. We calculate N local response images, one for each landmark $\mathbf{y}_i$. See §III for details of our appearance model.

Search best face landmarks: Given a set of response images, one for each landmark , and a CLM shape model

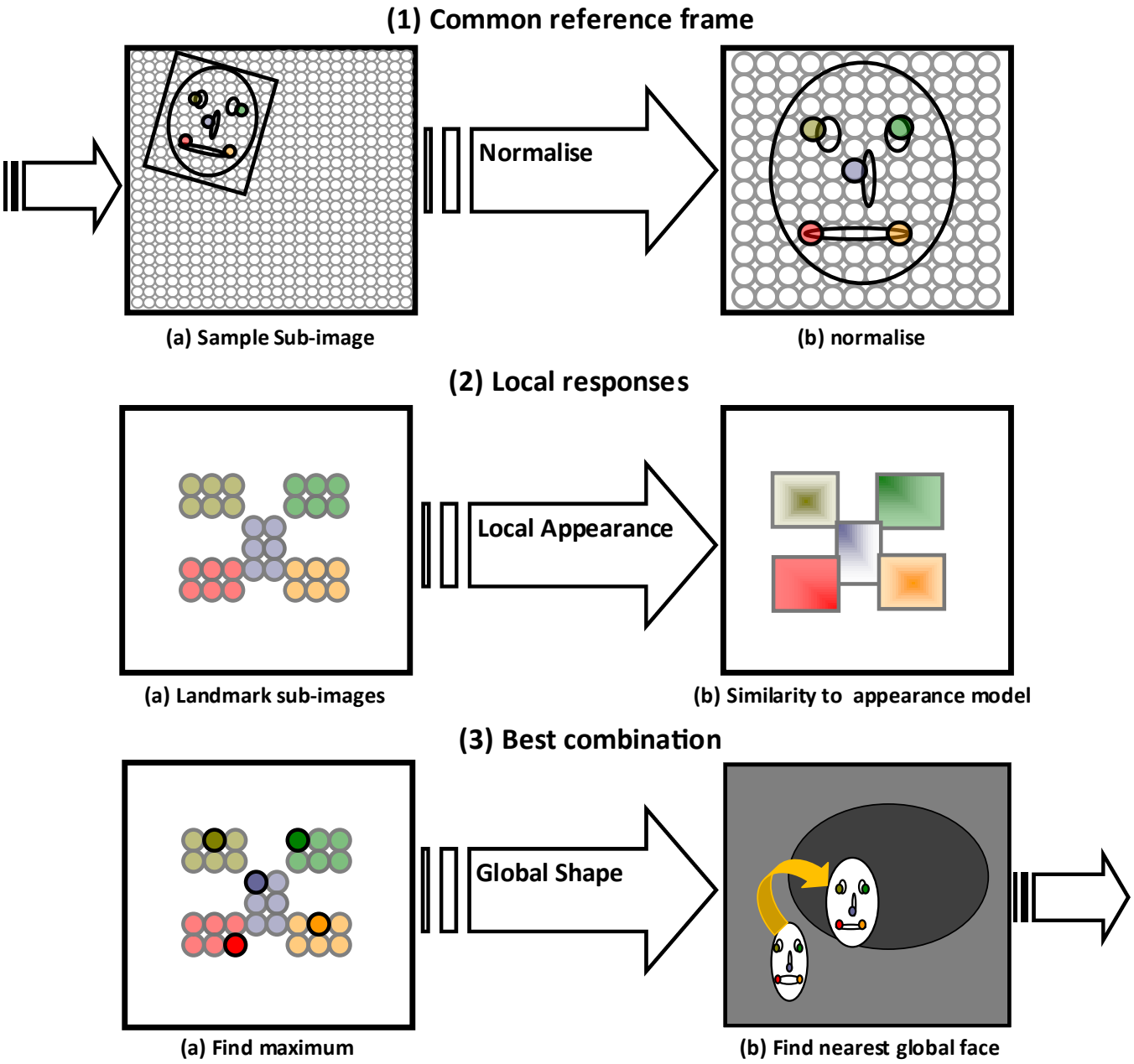


Fig. 2. **Constrained Local Model Framework** There are three steps in our proposed structured output learning for constrained local model (SO-CLM): A) Generate a reference face shape; B) Compute the *response map*; C) Search for the best face landmarks. The notations of the figure refer to §II.

(§III), the CLM joint cost is defined as

$$Q(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \bar{\mathbf{y}}\|^2 + \sum_{\mathbf{y}_i \in \mathbf{y}} \psi_i^A(\mathbf{y}_i) \cdot \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|^2, \quad (2)$$

where $\mathbf{y}_i$ is the current location of the i'th landmark in the image $\mathbf{x}$ and $\hat{\mathbf{y}}$ is the location of the predicted landmark from the shape model, and $\bar{\mathbf{y}}$ are the mean values of $\mathbf{y}$, further explained in §III-A. The ith landmark is restricted to only take those coordinates $\mathbf{y}_i \in \{\mathbf{u}, \mathbf{v}\}$ that lie within the ith response images. This joint cost is maximised using our own implementation of the mean-shift based approach of [9], that works on normalised response images [11].

III. Learning for Constrained Local Models

The CLM framework requires two models to be learnt. The global shape model, and the local appearance models. We use a point distribution model (PDM) to learn the shape, our main contribution pin this paper is to use an approximate SO-SVM to jointly the learn the appearance models.

A. Learning the Shape with PDM

We learn the CLM shape model using PDM approach [1], which allows us to represent $\mathbf{y}$ by a linear combination of

the mean face shape $\bar{\mathbf{y}}$, a basis of the shape variation $\mathbf{P}$, and non-rigid shape parameters $\mathbf{b}$,

$$\hat{\mathbf{y}} = \ S(\bar{\mathbf{y}} + \mathbf{P}\mathbf{b}; \theta) \tag{3}$$

where S is a linear function that may vary in form dependent on the global rigid transformation parameters θ that are included in the model (e.g. translation, rotation and scaling). $\mathbf{p} = \{\mathbf{b}, \theta\}$ is the parameters of PDM.

B. Structured Output Learning for CLM Appearance Model

Computer vision as a field has benefited greatly from progress in machine learning, and powerful statistical models which can be learned efficiently from large quantities of data now form the core of most modern vision techniques. The types of problems dealt with in computer vision often involve rich models with a large amount of structure. This has been particularly the case for pictorial structures in which discriminative learning has yielded an accurate and principled way of estimating parameters [21]. Here we explore how the same ideas might be applied to learning in the CLM approach. We shall see that because in CLM we can not guarantee a global optima, we can not exactly follow the SO-SVM approach, which requires exact optimisation in its inner loop. However, CLM approach allows us to do approximate SO-SVM still with good results.

Structured SVM provides a general framework for our task. In this setting, we have a prediction function $f : \mathcal{X} \rightarrow \mathcal{Y}$ from an input domain $\mathcal{X}$ to a structured output domain $\mathcal{Y}$.

$$f(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w}), \tag{4}$$

meaning that $\hat{\mathbf{y}}$ is the output which has the highest compatibility with the input $\mathbf{x}$. This prediction function is defined such that it makes use of an auxiliary function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ which can be seen as measuring the compatibility of an input-output pairs:

$$F(\mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \Phi^A(\mathbf{x}, \mathbf{y}) \rangle, \tag{5}$$

where $\Phi^A(\mathbf{x}, \mathbf{y})$ is a *joint feature mapping* of the input-output pairs. In SO-SVM the compatibility is leant, by learning the weights $\mathbf{w}$. Given a training set containing the known input-outputs pairs $(\mathbf{x}_1, \underline{\mathbf{y}}_1), ..., (\mathbf{x}_M, \underline{\mathbf{y}}_M)$, the optimisation program used to learn $\mathbf{w}$ is of the form of a generalised SVM:

$$\begin{aligned} &\min_{\mathbf{w}, \xi > 0} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \sum_{j=1}^{M} \xi_j, \\ &\quad s.t. \forall j, \forall \mathbf{y} \in \mathcal{Y} \setminus \underline{\mathbf{y}}_j : \\ &\qquad \langle \mathbf{w}, \Phi^A(\mathbf{x}_j, \underline{\mathbf{y}}_j) - \Phi^A(\mathbf{x}_j, \mathbf{y}) \rangle \geq \Delta(\underline{\mathbf{y}}_j, \mathbf{y}) - \xi_j, \end{aligned} \tag{6}$$

where λ is a free parameter used for regularisation.

Next we show how the learning for CLM can be approximately cast in a structured learning framework. The advantage of this is that it allows us to estimate the appearance models jointly in a discriminative framework. However typically structured learning requires an exact maximisation as the inner loop of the learning. In our case this is not available, but in general we find the inference part of CLM to be near the global optima so we use the CLM estimate as an approximation to the optima, whilst this has loses the theoretical guarantees of exact SO-SVM [22], interestingly our experiments show that it still works well in practice, as was also shown in [23].

For learning the CLM appearance model with SO-SVM, our training set consists of M images $\{\mathbf{x}_j\}_{j=1}^M$, each image $\mathbf{x}_j$ has an associated set of hand labelled landmarks $\underline{\mathbf{y}}_j$ with N true locations, $l = 1$. The joint appearance model is represented as a concatenation the N CLM appearance models $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N)$. In practice we initialise each of the N weights $\mathbf{w}_i$, one for each facial feature, with those of a N pre-learnt independent models trained with linear SVM.

In order to map each of the N appearance features, $\mathbf{d}_i$ of the CLM, to the joint feature space for all N landmarks we define the following joint appearance feature $\Phi^A(\mathbf{y}) = (\mathbf{d}(\mathbf{y}_1), \mathbf{d}(\mathbf{y}_2), \cdots, \mathbf{d}(\mathbf{y}_N))$, where $\mathbf{d}(\mathbf{y}_i)$ is the CLM appearance feature extracted at the location of the i'th facial landmark.

The joint loss $\Delta(\mathbf{y}, \underline{\mathbf{y}}) \rightarrow \mathbb{R}$ of SO-SVM measures the accuracy a prediction. For our application we define this loss as a hamming distance between the predicted and the true landmark locations. i.e.

$$\Delta(\mathbf{y}, \underline{\mathbf{y}}) = \sum_{\mathbf{y}_i \in \mathbf{y}} \mathbb{I}(\|\underline{\mathbf{y}}_i - \mathbf{y}_i\|^2 > \tau) \tag{7}$$

where $\mathbb{I}$ is an indicator function that activates when $\mathbf{y}_i$ is within a tolerated distance τ of the i'th true landmark.

During learning, the model $\mathbf{w}$ has to be updated in order to meet the constraints of Eq 6. In order to perform this update we need to find M maximums, one for each input image, of a function very similar to the objective Eq 4. We do this by following the steps

- We first initialize the weights $\mathbf{w}_i$ by applying a set of independent linear SVM on the labelled facial landmarks
- For each input-output pair, and for K iterations, we do CLM inference as that in §II to get the nearest estimated facial landmarks $\hat{\mathbf{y}}_j^k$ that are negative samples i.e. $\hat{\mathbf{y}}_j^k \neq \underline{\mathbf{y}}$. We call these *hard negatives*.
- We update $\mathbf{w}$ by using the implementation of online SO-SVM from [23][1]. This requires the joint CLM appearance features extracted at the true input-output pair $\{\mathbf{x}_j, \underline{\mathbf{y}}_j\}$ as well as that of the input-output pairs of the *hard negatives* found in the previous step.

Note we are using CLM as a approximate for the exact optimisation that is needed in cutting plane style methods, however we shall see in the results section that this still provides an improvement in results over learning all the $\mathbf{w}$ independently.

C. Model Binary Approximation

One important goal of our paper is to develop an approach targeting low-powered mobile devices. Following [23], we

[1] www.samhare.net/research/keypoints/code

give an extra boost in speed by choosing to use a binary appearance feature, BRIEF, and a binary approximation of our learnt model parameters, as shown in Fig 3. This allows for fast bitwise AND operators to compute the response maps. We approximate each $\mathbf{w}_i$ with a set of binary basis vectors following [23]:

$$\mathbf{w}_i \approx \sum_{k=1}^{N_b} \mathbf{z}_k \mathbf{a}_k, \tag{8}$$

where $\mathbf{a}_k$ is a binary vector that approximated the weights. This approximation is updated each time $\mathbf{w}_i$ changes.

As shown in Alg. 1 [23], we can compute the dot-product $\langle \mathbf{w}_i, \phi(\mathbf{y}_i) \rangle$ using only bit-wise operations. In order to do so, each $\mathbf{a}_i$ is represented as a binary vector and its complement: $\mathbf{a}_k = \mathbf{a}_k^+ - \bar{\mathbf{a}}_k^+$, where $\mathbf{a}_k^+ \in \{0,1\}^D$. then the dot-product can be re-write:

$$\langle \mathbf{w}_i, \phi(\mathbf{y}_i) \rangle \approx \sum_{k=1}^{N_b} \beta_k (\langle \mathbf{a}_k^+, \phi(\mathbf{y}_i) \rangle - \langle \bar{\mathbf{a}}_k^+, \phi(\mathbf{y}_i) \rangle), \tag{9}$$

where, and each dot-product in computing response map can be conducted very efficiently using a bitwise AND followed by a bit-count. We can also use pre-computed bit-count of $\phi(\mathbf{y}_i)$ to achieve more efficiency in running-time speed, although this will sacrifice the algorithm memory efficiency. As shown in the equation that $\langle \mathbf{a}_k^+, \phi(\mathbf{y}_i) \rangle - \langle \bar{\mathbf{a}}_k^+, \phi(\mathbf{y}_i) \rangle = 2\langle \mathbf{a}_k^+, \phi(\mathbf{y}_i) \rangle - \|\phi(\mathbf{y}_i)\|$, with the approximation method shown in Alg. 1, Generating response map in our approach is only roughly N_b times more expensive than computing binary Hamming distance. In practice, we set $N_b = 3$. See § V for the details of the experimental results.

Algorithm 1 Model Binary Approximation [23]

Input: $\mathbf{w}_i$, N_b.
Output: $\{\mathbf{z}_k\}_{k=1}^{N_b}, \{\mathbf{a}_k\}_{k=1}^{N_b}$.
Initialize: $\mathbf{r} = \mathbf{w}_i$ Initialize residual.
for $k = 1$ to N_b do:
1) $\mathbf{a}_k = \text{sign}(\mathbf{r})$
2) $\mathbf{z}_k = \frac{\langle \mathbf{a}_k, \mathbf{r} \rangle}{\|\mathbf{a}_k\|^2}$ Project $\mathbf{r}$ onto $\mathbf{a}_k$
3) $\mathbf{r} \longleftarrow \mathbf{r} - \mathbf{z}_k \mathbf{a}_k$ update residual
end for

IV. SO-CLM TLD TRACKING

We follow the Tracking-Learning-Detection (TLD) [18] approach as it provides a elegant solution to the long-term tracking problem. The TLD overcomes problems associated with naive tracking strategies as illustrated in Fig. 3. Three components make up the solution, those being; face localisation with a face ROI detector; a base tracker; and a model update step.

The structured SVM as framework presented so far assumes that all the training data are available at the time of learning. This scenario is referred to as batch learning. A different scenario is online learning, in which the training data arrive sequentially. In this setting, the learner must be incrementally updated each time a new training example

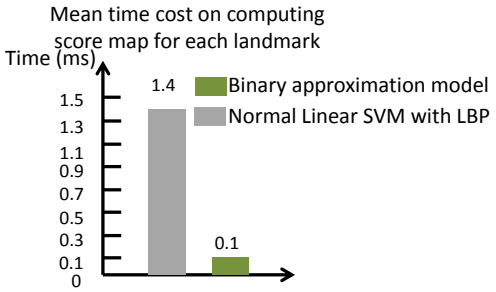


Fig. 3. **Timings with and without binary model approximation.** The bar chart clearly shows that the binary approximation of the appearance model significantly reduces the time it take to generate a response map for the CLM inference.

arrives. The current state of the learner is used in order to predict the label for this new example, which is then compared to the true label, and adjustments are made to the learner as appropriate. We use the online learning algorithm for SO-SVM presented in [23].

V. EXPERIMENTS

We conduct four sets of experiments: standard test, parameter test, generalization test, and mobile device test. The standard and generalisation test experiments are conducted on a PC. The software we used is Visual Studio 2010 and Windows 7 64-bit operation system, where the CPU is an Intel Xeon E5645 2.40GHz, and the Memory is 32.0GB. We used the same shape and appearance model for mobile device test. We implement our approach for *ios* device, and report the evaluation results of our approach on ipad2, which has 1 GHz CPU and 512MB DDR2 RAM.

Datasets and Annotations: Recent papers [8], [9] report state-of-the-art results on many challenging datasets by training their model on the CMU Multi-PIE dataset [24]. However, the facial landmark labels, that they used, on CMU Multi-PIE dataset are not publicly available. To achieve the results in this paper and make fair comparisons, we annotated 2245 frontal face images from the CMU Multi-PIE datasets. We evaluate the proposed approach on this database, and compared to the two state-of-the-art approaches [8], [9].

Many other face databases are annotated with facial features. The most widely used ones are XM2VTS [25], BioID [26] and Labelled Face Parts in the Wild (LFPW) [27] [28]. XM2VTS and BioID databases are acquired under controlled lighting conditions and contain only frontal faces. Different from XM2VTS and BioID, the LFPW database contains variations from real internet imaging conditions. In order to evaluate the generalization performance the proposed approach, we also tested our model on these databases.

To assess the tracking performance, we evaluate the accuracy performance of our model on the talking face video database[2]. We also report the speed on ipad2.

Evaluation Criteria: We follow the same evaluation criteria described in [10]. The 17 sample points we chosen

[2]www-prima.inrialpes.fr/FGnet/data/01TalkingFace/talking_face.html

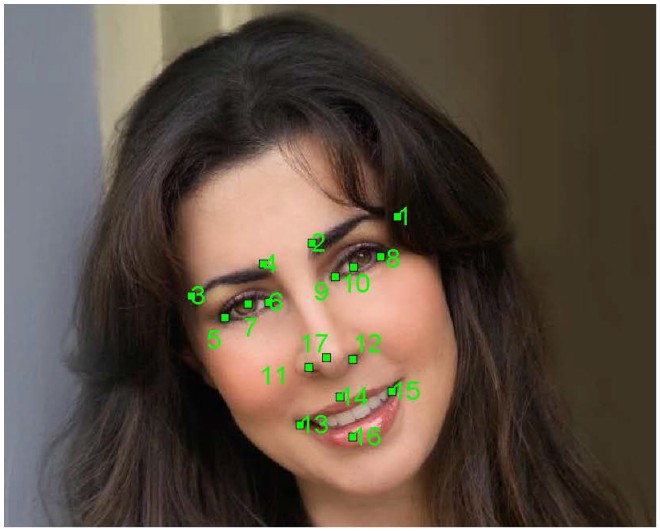


Fig. 4. **Chosen Landmark locations used for evaluation:** This figure shows the chosen evaluation sample points [10] described in §V. We compare the differences between the coordinates of these landmarks on the face and the landmark of estimated points from all the methods.

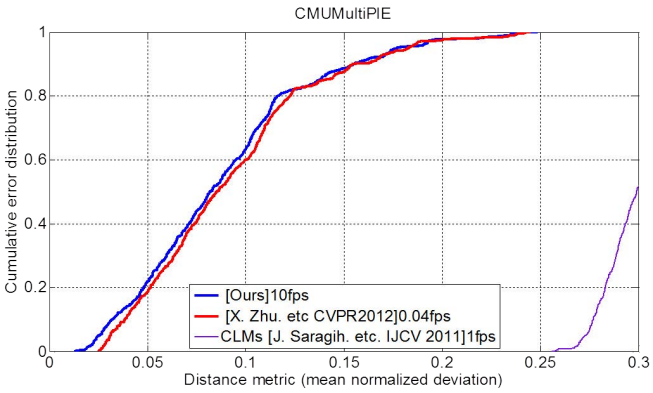


Fig. 5. **Standard Test:** This figure illustrates the results of different approaches on the 734 frontal face images from CMU Multi-PIE. We train our joint CLM appearance model on 1255 frontal face images and validate the parameters on another 466 frontal face images. For the tree-based structured approach, we used the model provided by [8]. Our evaluation criteria is the same as that in [10] and is discussed in §V. The time calculation for CLM and our approach does not include the initial face detection.

are shown in fig. 4. Suppose the ground truth (hand-labelled landmarks) are $\{\underline{\mathbf{y}}_i\}_{i=1}^{i=17}$, we find the most closet landmarks in the set of estimated landmarks $\{\hat{\mathbf{y}}_i\}_{i=1}^{i=17}$. We evaluate performance by computing the mean Euclidean distance for each estimated landmark. This is then normalized with respect to a reference distance d_r(defined as the Euclidean distance between eye centers in ground truth, which is called inter-ocular distance (IOD)) to make error invariant to scale:

$$m_e = \frac{1}{17}\sum_{i=1}^{17}\frac{\|\hat{\mathbf{y}}_i - \underline{\mathbf{y}}_i\|}{d_r}.$$

To evaluate the speed performance for different approaches, we report the frame per second (fps) speed which is computed based on the number of frames for 20 seconds. On the ipad2 evaluation, we report the fps based on the mean fps over 60 seconds.

A. Standard Test

From CMU Multi-PIE database, we randomly choose 1225 images as training images, 734 images as testing images, and 491 images as validation images.

We compare it with the following approaches: (1) Constrained local models (CLMs) [9]: This work is one of the response-fitting-based approaches, and produces the state-of-the-art results on CMU Multi-PIE. (2) Tree-structured SVMs part-based approach [8]: This work is the state-of-the-art approach in the graph-based approaches. The dimension of the shape subspace is 24. For the CLM approach we choose to compare with, we used the pre-trained tree-structured models [3] provided in [8] as they also trained on a subset of CMU-MultiPIE dataset because the ground truth coordinates of the training landmarks reported in [8], [9] are not public available. We used Viola face detector in OpenCV 2.4.2 to get the initial face ROI for our approach and CLMs. For comparison purpose, we used the pre-trained model for the tree-structured SVMs part-based approach, as the performance of this approach drops significantly when we train it on the same 1225 frontal face image data.

As illustrated in fig. 5, the results of our approach is comparable to the state-of-the-art. But ours performs faster than all of them. The proposed approach with BRIEF feature descriptor [15] performs best. The speed result is contributed by the two key components: 1) model binary approximation; 2) inexact search within shape priors. The model binary approximation part enables our approach to be working for the low-power devices. The CLMs [9] performs similar to the proposed approach due to that the two approaches obtained the shape priors following the same way.

B. Parameters Test

We test the accuracy and speed performance of the proposed approach with different parameter setting. We consider two cases here: 1) The accuracy performance of the proposed approach with different τ, we use τ as a ratio of inter-ocular distance (IOD); 2) The running-time performance of the proposed approach with different number of landmarks output. We still use the model trained on 1255 frontal face images from CMU Multi-PIE dataset, and test on the 734 frontal face images. Fig. 6 shows that the propose approach can be sped up by reducing the number of tracking feature points. Fig. 7 shows the RMES results for different IOD based our face feature detector.

C. Generalization Test

To evaluate the generalization of the proposed approach, we train a model on the CMU Multi-PIE and test on the XM2VTS, BioID, and LFPW datasets. As illustrated in fig. 8,

[3] www.ics.uci.edu/~xzhu/face/

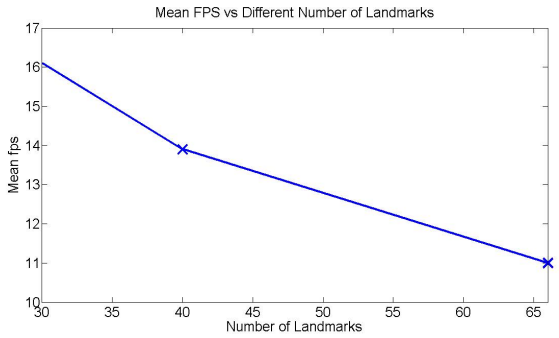


Fig. 6. **fps vs different number of landmarks.** This figure shows the mean frame per seconds (fps) results for different number of landmarks based on our face feature tracker.

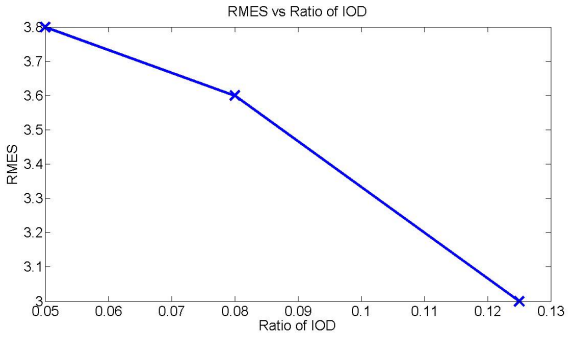


Fig. 7. **Mean RMES vs different inter-ocular distance.** This figure shows the root mean square error (RMES) results for different inter-ocular distance (IOD) based our face feature detector.

our approach performs better than other approaches on the BioID, XM2VTS, and LFPW datasets. This demonstrates that our joint training method improves the generalization over the other methods.

Fig. 8 shows the results of cumulative-error-distribution vs distance-metric-deviation of different approaches tested on XM2VTS, BioID, and LFPW datasets and trained on CMU Multi-PIE dataset. Most of the images in XM2VTS and BioID are frontal face images. The images of LFPW dataset contains some different poses. The proposed approach performs better than XM2VTS and BioID datasets in terms of accuracy and speed. In the LFPW dataset, the proposed approach performs close to the mixture-tree-based structured SVM [8], which is trained for multi-view data.

D. Mobile Device Test

For image sequences in video, we present the implementation details of the proposed facial feature detection and tracking approach on mobile device. The overall system is supported by OpenCV 2.4.2 and Eigen library[4] which are compiled by targeting system compilers.

As shown in Table I, we evaluate different facial feature tracking strategies. And we consider the fps speed performance on the ipad2 and the mean RMES performance on

[4]eigen.tuxfamily.org



Fig. 9. **ipad2 Demo.** This figure shows our SO-CLM running on an ipad2 low power device. We implement the proposed approach as a real-time facial feature tracker on an Apple IOS 5.1 system. The maximum frame per seconds of the propose facial feature tracker on ipad2 is 7fps. The size of the input image is 192×144. The number of the landmarks is 66.

the 5000 frames from Talking Face Video database. All the appearance and shape models of these methods are the same as those trained and evaluated on the standard test and generalization test. The first one is a facial feature detector on each frame, which performs accurately on the database but slow on the ipad2. The second one is facial feature detector on each 10 frames, which performs not very accurately on the database due to the poor detected face ROI. The third one is proposed facial feature tracker, which performs accurately and fast.

As illustrated in Fig. 9, we test an implementation of the proposed approach on popular ipad2. Since it is difficult to obtain the ground truth from the live video, we still use the trained model on 1250 frontal images from CMU Multi-PIE. The proposed approach can achieve maximum 7 fps when tracking 66 facial features. We found that the speed is not affected by the battery power percentage unless it has more than 10%.

VI. CONCLUSION

In this paper, we present an efficient approach for facial feature detection with a learning method based on approximate structured learning, which allows us to jointly learn the appearance parameters of the features. We found that the implementation of the proposed approach achieves real-time speed on ipad2. This result means that we could deploy dense (e.g. more than 17 landmarks) facial feature detection and tracking directly onto a low-power device. In this work we used the mean-shift based search of [9] to optimize the face in the inner loop of SO-SVM because it achieved state-of-the-art accuracy. We found that this approach suffers on un-normalised data, and that a normalisation step adds an extra time cost, that takes away some of the benefit of our binary approximation of $\mathbf{w}$. However as part of future work we plan to explore optimization of the facial features that exploits the fact we are using bit vectors.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the discussions and comments from Jonathan Warrell, Sam Hare, Sunando Sengupta, Vibhav Vineet, Lubor Ladicky, and FG13 reviewers. S. Zheng and P. Sturgess are supported by the EPSRC studentship. This work was funded by Engineering and Physical Sciences Research Council (EPSRC) project Scene Understanding using New Global Energy Models EP/I001107/1. This work also was supported in part by the IST Programme of the European

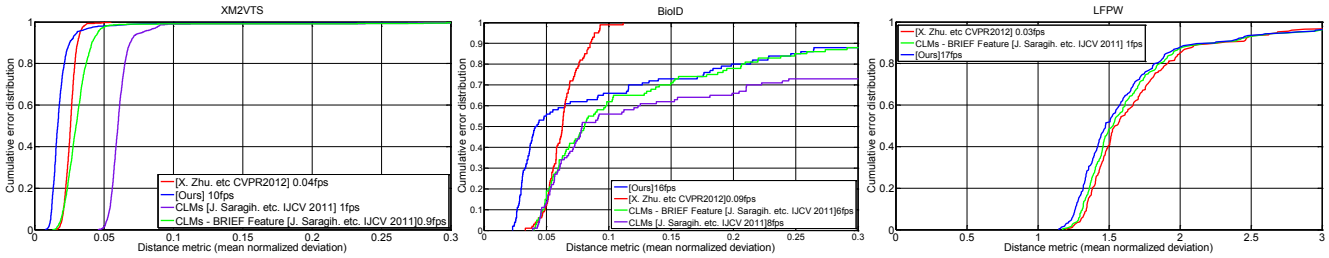


Fig. 8. **Generalisation Test** This figure illustrates the results of the generalization test. We train the models on CMU MultiPIE dataset, and test on the XM2VTS, BioID and LFPW datasets. We compare the proposed approach with other state-of-the-art graph-based and CLM approaches. We report the mean frame per second (fps) speed for different approaches on XM2VTS. We only report the results of graph-based approach and our approach on LFPW. The time calculation for CLM and our approach does not include the initial face detection.

TABLE I

Results from different facial feature tracking strategies. *This table shows results for different facial feature tracking strategies using our* SO-CLM*: a) with face detection on all frames; b) with face detection every* 10*th frame; c) with our Tracking Learning and Detection implementation, described in §IV. We report the frame per second (fps) performance on ipad2 and the Mean* RMES *performance on Talking Face Video.*

Methods	Detector on each frame	Detector on each 10 frames	Proposed tracker
fps on ipad2	0.5	3.0	7.0
Mean RMES	6.3	10.0	7.1

Community, under the PASCAL2 Network of Excellence, IST-2007-216886. P. H.S. Torr is in receipt of Royal Society Wolfson Research Merit Award.

REFERENCES

[1] T. F. Cootes and C. J. Taylor, "Active shape models - 'smart snakes'," in *BMVC*, 1992.

[2] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, "Static facial expressions in tough conditions: Data, evaluation protocol and benchmark," in *1st IEEE International Workshop on Benchmarking Facial Image Analysis Technologies BeFIT, ICCV2011*, 2011.

[3] I. Kemelmacher-Shlizerman and S. M. Seitz, "Face reconstruction in the wild," in *ICCV*, 2011.

[4] G. Hua, Y. Fu, M. Turk, M. Pollefeys, and Z. Zhang, "Introduction to the special issue on mobile vision," *International Journal of Computer Vision*, pp. 277–279, 2012.

[5] P. A. Tresadern, M. C. Ionita, and T. F. Cootes, "Real-time facial feature tracking on a mobile device," *International Journal of Computer Vision*, pp. 280–289, 2011.

[6] M. Davis, M. Smith, J. Canny, N. Good, S. King, and R. Janakiraman, "Towards context-aware face recognition," in *ACM Multimedia*, 2005.

[7] J. M. Saragih, S. Lucey, and J. F. Cohn, "Deformable model fitting with a mixture of local experts," in *ICCV*, 2009.

[8] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *CVPR*, 2012.

[9] J. M.Sargaih, S. Lucey, and J. F.Cohn, "Deformable model fitting by regularized landmark mean-shift," *International Journal of Computer Vision*, pp. 200–215, 2011.

[10] D. Cristinacce and T. F. Cootes, "Feature detection and tracking with constrained local models," in *BMVC*, 2006.

[11] Y. Wang, S. Lucey, and J. F. Cohn, "Enforcing convexity for improved alignment with constrained local models," in *CVPR*, 2008.

[12] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," in *Journal of Machine Learning Research*, 2005.

[13] P. A. Tresadern, H. Bhaskar, S. A. Adeshina, C. J. Taylor, and T. F. Cootes, "Combining local and global shape models for deformable object matching," in *BMVC*, 2009.

[14] M. Uricar, V. Franc, and V. Hlavac, "Detector of facial landmarks learned by the structured output svm," in *VISAPP*, 2012.

[15] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: binary robust independent elementary features," in *ECCV*, 2010.

[16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *ICCV*, 2011.

[17] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *CVPR*, 2012.

[18] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, pp. 1–14, 2011.

[19] D. Cristinacce and T. Cootes, "Automatic feature localisation with constrained local models," *Pattern Recognition*, pp. 3054–3067, 2008.

[20] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, 2001.

[21] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, vol. 61, pp. 55–79, 2005.

[22] L. Huang, S. Fayong, and Y. Guo, "Structured perceptron with inexact search," in *NAACL*, 2012.

[23] S. Hare, A. Saffari, and P. H. S. Torr, "Efficient online structured output learning for keypoint-based object tracking," in *CVPR*, 2012.

[24] R. Gross, I. Matthew, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, pp. 807–813, 2010.

[25] K. Messer, J. Kittler, M. Sadeghi, S. Marcel, C. Marcel, S. Bengio, F. Cardinaux, C. Sanderson, J. Czyz, L. Vandendorpe, S. Srisuk, M. Petrou, W. Kurutach, A. Kadyrov, R. Paredes, E. Kadyrov, B. Kepenekci, F. Tek, G. B. Akar, N. Mavity, and F. Deravi, "Face verification competition on the xm2vts database," in *Int. Conf. Audio and Video Based Biometric Person Authentication*, 2003.

[26] R. Frischholz and U. Dieckmann, "Bioid: A multimodal biometric identification system," *Computer*, vol. 33, pp. 64–68, 2000.

[27] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," in *CVPR*, 2011.

[28] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," in *CVPR*, 2012.