Digital Object Indentifier 10.1109/TPAMI.2012.205

Local Evidence Aggregation for Regression Based Facial Point Detection

Brais Martinez, *Member, IEEE,* Michel F. Valstar, *Member, IEEE,* Xavier Binefa, *Member, IEEE,* Maja Pantic, *Fellow, IEEE*

***Abstract*—We propose a new algorithm to detect facial points in frontal and near-frontal face images. It combines a regression-based approach with a probabilistic graphical model-based face shape model, that restricts the search to anthropomorphically consistent regions. While most regression-based approaches perform a sequential approximation of the target location, our algorithm detects the target location by aggregating the estimates obtained from stochastically selected local appearance information into a single robust prediction. The underlying assumption is that by aggregating the different estimates, their errors will cancel out as long as the regressor inputs are uncorrelated. Once this new perspective is adopted, the problem is reformulated as how to optimally select the test locations over which the regressors are evaluated. We propose to extend the regression-based model to provide a quality measure of each prediction, and use the shape model to restrict and correct the sampling region. Our approach combines the low computational cost typical of regression-based approaches with the robustness of exhaustive-search approaches. The proposed algorithm was tested on over 7,500 images from 5 databases. Results showed significant improvement over the current state of the art.**

***Index Terms*—Facial point detection, object detection, probabilistic graphical networks, support vector regression.**

I. Introduction

FACE detection algorithms are very reliable nowadays [22], [40], and commonly used as the first step in face analysis systems. Given that it is desirable to know the shape of the face for achieving detailed analysis of the face, the next step in automatic face analysis is often aimed at uncovering the face shape. One way of achieving this is by locating a number of fiducial facial points, such as the corners of the mouth and eyes, and the tip of the nose. In challenging domains such as facial expression recognition, algorithms can capitalise on precise fiducial facial point localisation by extracting appearance features from locations relative to these points (e.g. [17], [35]) or by using the locations of the points directly (e.g. [34], [38]). Many important tasks, such as face recognition, gaze detection, head pose detection, sign language interpretation, gender detection, age estimation, lip reading, and facial point tracking applications, can benefit from precise facial point detection.

B. Martinez and M. Pantic are with the Department of Computing, Imperial College London, London, United Kingdom. E-mail: B.Martinez, M.Pantic@imperial.ac.uk. M.F. Valstar is with the School of Computer Science, University of Nottingham, Nottingham, UK, and was with Imperial College London during part of this work. E-mail: michel.valstar@nottingham.ac.uk. M. Pantic is also with the Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente, The Netherlands. X. Binefa is with the Information and Communication Technologies Department, Universitat Pompeu Fabra. E-mail: xavier.binefa@upf.edu.

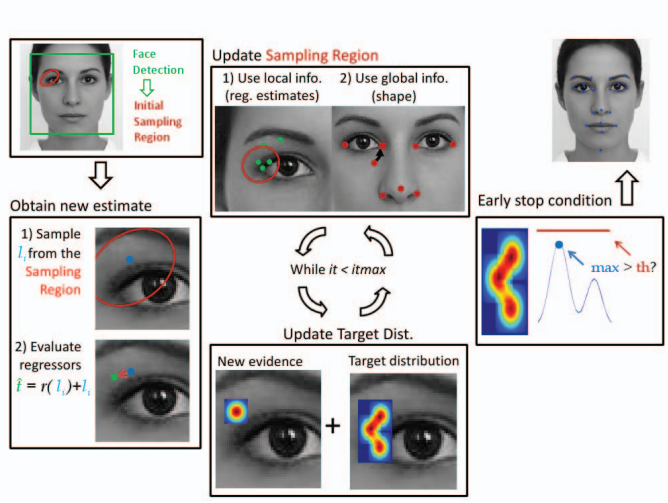

Fig. 1. Diagram showing the flow of the algorithm proposed. The input is an image where at least one face can be detected. A new test location is sampled according to a sampling distribution, and a prediction is obtained from it. The obtained prediction is added to the target distribution, which summarises the target location knowledge so far. An early stop criterion is evaluated over the updated target distribution and, if not accepted, a shape correction procedure is applied, and the sampling regions redefined.

The definition of the set of points to be detected depends on the desired application. The algorithm we present in this article is regression-based and can detect an arbitrary set of facial points, the only requirement being that the target points have a distinctive local texture. Without loss of generality, in this work we defined the location of these points to be the minimal set required to detect any facial expression [29], [38]. In total, a set of 20 points are defined as our target locations, shown in Fig. 2.

Two different sources of information are typically used to detect facial points: face appearance (i.e. texture), and shape information. The latter aims to explicitly model the spatial relations between the locations of facial points. Although some methods make no use of the shape information, it is common to combine the two sources of information. This can be done by using a face shape model to either restrict the search region (e.g. [14]), which is the approach we follow, or by correcting the estimates obtained during the local search (e.g. [6]).

We can further distinguish between different approaches by the methods they employ to search for the location of the facial points. Some approaches use exhaustive search (e.g. [9], [41]) through raster scan search in a region of

interest determined by e.g. a face detector. These methods typically use a classifier that indicates whether an image patch contains the target facial point or not. Instead, estimation-based approaches try to directly estimate the target location at each iteration/evaluation (e.g. [6], [39]). Such approaches typically employ regression techniques that, when evaluated over a patch, estimate the relative position of the target facial point with respect to the patch. Methods employing exhaustive search are inherently slower as they have to analyse every possible solution. Furthermore, since every location in a region of interest is analysed, multiple target candidates are usually identified. Hence these methods have the added complexity of deciding the final output. Instead, regression-based methods typically yield a single output, and are computationally more efficient. For these reasons we follow an estimation-based approach.

Estimation-based approaches commonly perform an iterative sequential refinement of the estimate, where the previous target estimate becomes the test location at the next iteration. This includes gradient-based, Newton-like and Taylor-based methods [1], [5]. Another approach is to obtain the estimates from a regressor evaluation. Regressors are more general class-specific models, and the explicit construction of a loss function is not required [6]. This approach has been followed in the facial point detector presented in [39], where regressor estimates are combined with a probabilistic graph-based shape model. The shape model is evaluated after each iteration, checks the shape of the predicted point locations and corrects them in case they do not form a consistent face shape. Overall, sequential estimation approaches present a series of drawbacks:

1) Sequential algorithms are very sensitive to the starting point. If it is not accurate enough, regressors will not yield good estimates and the algorithm may drift.
2) Sequential approaches suffer in case of wrong estimates, even when happening for a single iteration e.g. due to partial occlusion. That is to say, they are sensitive to any errors in the estimation process.
3) Regression methods are not inherently capable of providing an early stop condition based on the quality of the estimate obtained. Instead, the stop criterion must be defined based on a lack of change between successive iterations or based on a pre-set maximum number of iterations.
4) Sequential approaches may converge even when still far from the target[1].
5) If textures far from the target location are included into the training set, the regressor will make fewer mistakes. However, this comes at a cost of the precision of the estimates, especially when evaluating test locations close to the target location. The reciprocal also holds. This is a precision-generality trade off. Since sequential approaches are not robust to poor estimates, regressors with high generality are required, reducing in turn the precision of the method.

[1]Regression-based sequential approaches do not utilise a likelihood or a cost function. Therefore, even when the method converges far from the true target location, it cannot be called a local minima. However, a parallel can be drawn easily: the regressor might produce no change on an estimate when tested at a non-target location, or produce a cyclic path.

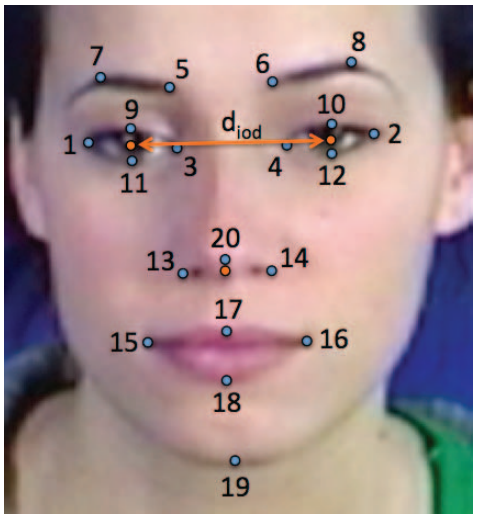

Fig. 2. The twenty facial points detected in this work (blue) and the centre points for the eyes and nose (orange). The inter-ocular distance d_{IOD} is the Euclidean distance between the centre eye points.

The facial point detection approach presented in this work aims to overcome these drawbacks. An overview of our method is shown in Figure 1. The method starts by detecting the face and initialising, for every point, a region from which to sample test locations. After sampling a test location, a set of regressors generates a prediction for the target location as well as a likelihood estimate of this prediction. Based on the predictions, the sampling region is refined and the target location distribution is updated. After each iteration, a Markov Random Field (MRF) based shape model enforces consistency between the sampling regions of facial points to ensure that new test locations are sampled from regions forming a valid face shape. The algorithm stops when a preset confidence value is obtained or a maximum number of iterations has been reached. A diagram showing this flow can be seen in Fig. 1.

The utilised shape model is based on that proposed in [39]. This shape model uses pairs of line segments, where a line segment is defined as the line connecting two facial points. For each pair of line segments, the shape model encodes their relative angles and their relative lengths. A Markov Network is used to combine the information from all pairs of line segments into a face shape model. This representation makes the shape model robust to scale changes and in-plane rotations.

We rely on regressors to translate the local appearance information to a prediction of the displacement vector relating the sampled test location to the target point location. As local appearance descriptors we use Local Binary Patterns (LBP), together with a correlation based feature selection (CFS) to reduce the dimensionality of the appearance descriptors. We trained Support Vector Regressors (SVRs) to predict the horizontal and vertical components of the displacement vector, and the Euclidian distance to the target, which is used to determine the quality of a prediction.

The method applies regressors trained to be precise rather than general. Some possible test locations will not be rep-

resented on the training set. However, the target location is obtained by aggregating the estimates obtained in all previous iterations. Hence, a single erroneous estimate is not likely to affect the detection outcome whenever several correct estimates are obtained. In turn, the method can capitalise on precise estimates very efficiently. This is how we address the precision-generality trade-off issue. Since the method relies on the accumulated evidence provided by all the estimates obtained up to iteration i, we refer to this algorithm as Local Evidence Aggregated Regression, or LEAR for short.

The remainder of this work is organised as follows. In section II, we position our work relative to existing facial point detection techniques. In section III, we describe the facial point localisation strategy. The novel sampling strategy and the facial shape model are explained in sections IV and V, respectively. A comprehensive evaluation of the performance of our method, including comparison with the state of the art methods, is provided in section VII. We close with concluding remarks in section VIII.

II. Related Work

Regression-based approaches: Relatively few regression-based approaches to facial point detection have been proposed. Cristinacce & Cootes [6] presented a sequential regression-based approach to facial point detection. It combines a GentleBoost regressor with an Active Shape Model (ASM) used to correct the estimates obtained. Another sequential regression-based approach was presented in [39], where a SVR was combined with a probabilistic MRF-based shape model.

In contrast to these methods, the work in [19] proposes an adaptation of the Implicit Shape Model [20] to the problem of facial point detection. A grid of regularly spaced point locations is defined within the detected face region. During training, textures local to each point are clustered, and a set of discrete target location estimates is assigned to each of the clusters found. For testing, the local textures are simply assigned to one of the clusters. The final output results from minimising the MSE to all of the estimates. However, it is not clear how this work would adapt to expressive faces, since most textures analysed are uncorrelated with expressions.

The work proposed in [12] uses an extension of the sequential regression-based approach proposed at [10] in the context of general object pose estimation. In particular, a different regressor is evaluated at each iteration, so more general regressors are applied in the first stages, while more specific regressors are applied in the later stages. The sequence of regressors is defined entirely during training. On it, regressor i is chosen to minimise the average error over the set of training examples. Then, regressor i is applied to all the training examples, and its estimates are taken as the training examples for regressor $i+1$. This is a solution to the problem of generality vs. precision alternative to ours. Through the use of shape regressors [43], the authors achieve multi-pose facial point detection. However, this approach assumes a worst-case scenario, since the examples defining the pace at which the regressors become more specific are those yielding the worst estimates. Furthermore, the regressors lack precision. In consequence, the number of iterations required is daunting: in [12] the algorithm runs for 800 iterations.

Shape models: One of the most common approaches to modelling the face shape is the use of an ASM (e.g. [6]). A face shape representation is obtained for each of the training examples as the concatenation of the facial point coordinates. Then PCA is used to obtain a linear manifold of possible face shapes. Given a set of points expected to correspond to the target facial point locations, shape-consistent estimates are obtained from projecting these points onto the manifold. Therefore, even if a single point is incorrect the whole set of points will be transformed through the projection. In comparison, the face shape model used in this paper can pinpoint which of the facial points is inconsistent with the others, and produce corrected point configurations that preserve well-estimated points.

Other works model the face shape using line segments between facial points. For example, Cosar and Cetin [4] use the fact that the ratio of distances between co-linear points is fixed under affine transformations to restrict the inter-frame change in facial point tracking. Liang et al. [21] use a condensation algorithm modified with spatial constraints. It considers the segments forming the contours delimiting facial components, and a shape model is used to constrain consecutive segments to have coincident limits (closing the contour), and to keep a valid angle between them. In [36] a MRF models the spatial relations between points, and a hierarchy between what is called global and local shapes (i.e., component-level shape and global face shape) is defined to alleviate complexity of the optimal shape search.

The shape model used here follows the same idea of using line segments between points as the basic descriptors of the face shape. However, we go one step further. We learn the relations between any two line segments connecting two pairs of facial points.

Other facial point detection techniques: One of the most popular approaches to facial point detection consists of combining a face shape model based on ASMs with a local texture-based search. The local search produces candidates for the facial point locations, which are corrected into a valid face shape using the face shape model. Examples include [6], that presents a classification-based and a regression-based variants for the local texture-based search, and [25], that use a refinement and tweaking of the ASM to further improve their performance. In [7] the Constrained Local Models are introduced, where a joint face shape and local face texture model is used. When a new estimate of the face shape is attained, the joint shape-texture model proposes adapted local templates for the local search in the next iteration.

Examples of classification-based sliding window approaches to facial point detection are given by [33], where very descriptive classifiers are trained using a Multiple Kernel Learning SVM and multi-scale LBP features. In [2] a SVM classifier using SIFT descriptors is trained, while the strength of that method comes from the constraints over the detection constellations, modelled non-parametrically using a very large set of annotations. In [9] a Subclass Discriminant Analysis classifier is trained to learn local facial point texture and the texture of its

surroundings, and the method includes the intermediate steps of face and facial component detection. Finally, [8] constructs a statistical model that combines Gabor feature responses and a prior shape model, and detects the facial feature location as its MAP. In section VII we compare our approach against these methods where possible.

III. Facial Point Detection through Local Evidence Aggregation

A key aspect of our algorithm is to consider that each image patch evaluated by the regressors adds some evidence regarding the target location, instead of just considering the last estimate in the sequence of estimates and discarding the rest of them. The detection arises from the aggregation of different evidences. To make this process effective, we need a strategy to select the test locations from which local evidences are gathered. To this end, we also define a strategy to assess the quality of the estimate provided by the regressors.

A. Obtaining Target Estimates from Local Descriptors

Our approach relies on the ability to obtain an estimate of the target location from local image information. More precisely, given a test location in an image, a feature representation is extracted from a patch centred at it. This feature vector is used to approximately infer the true target location. That is, we perform regression to predict the horizontal and vertical components Δx and Δy of the displacement vector pointing from the test location to the true target location. In this article we use Support Vector Regression (SVR) as the regression algorithm, although other regression algorithms are also applicable. An SVR takes a feature vector as input, and yields a 1-dimensional real-valued prediction as the output. Therefore, to estimate the displacement vector (a 2-dimensional vector), two regressors have to be trained, for every point. In our case, this results in $20 \cdot 2 = 40$ regressors used to obtain target estimates from local appearance descriptors.

Formally, given an input grey valued image I, the true target location T, and a test location l, the regression problem is to estimate the vector $v(l) = l - T$. Regressors r_x and r_y account for the estimation of Δx and the Δy respectively, where $v(l) = (\Delta x, \Delta y)$. Both regressors share the appearance description x_l of a patch centred at the test location l as their input, denoted as:

$$x_l = f_a(I, l) \tag{1}$$

and in the following variable I will be omitted for simplicity. The output of the regressor is an estimate of v defined as:

$$\hat{v}(l) = (r_x(x_l), r_y(x_l)) \tag{2}$$

and an estimation of the true target location can be computed as $\hat{t} = l + \hat{v}$.

To avoid confusion, we use $\hat{t}$ to denote the estimates as obtained from the evaluation of the regressor, and $\hat{T}$ to denote the target location estimate as provided by the full algorithm. The feature representation and the details of the SVR used to obtain $\hat{v}$ are described in detail in section VI.

B. Assessing the Quality of Local Evidences

One of the major drawbacks of SVR is that it does not provide a probability/likelihood of the estimate obtained. Some works have already tackled this problem. For instance [42] uses a regression method with probabilistic output as Relevance Vector Machine to obtain the predictions. Similarly, [30], [43] evaluate a classifier with probabilistic output at the estimated target location. In our case the estimated target location is also analysed. However, we use instead an SVR trained to estimate the distance between an image location and the true target location. More precisely, we compute the local appearance descriptor at the estimated target location $\hat{t}$ and evaluate the distance regressor:

$$\hat{d} = r_d(x_{\hat{t}}) \approx \|v(\hat{t})\| \tag{3}$$

where $x_{\hat{t}} = f_a\left(\hat{t}\right)$.

Finally, a likelihood based on the estimated distance can be computed as:

$$f_{lik}\left(\hat{t}\right) = e^{-\hat{d}/\sigma^2_{lik}} \tag{4}$$

where the variance σ^2_{lik} is a fixed parameter (see section VII-C for details on how its value is set).

C. Accumulating local information

Each iteration involves the evaluation of one test location per point. The observations over multiple iterations are combined, so the point detector builds upon multiple local evidences gathered. The estimate of the location of the facial point i at iteration k is then a function of the up-to-date local evidences $\{\hat{t}^i_j\}_{j=1:k}$. This information is combined and summarised through an unnormalised pdf, denoted as S_{acc}, which accumulates the obtained evidences as an unnormalised mixture of Gaussian distributions. Each local evidence adds a component to it with predefined covariance as:

$$S^i_k(x) = \sum_{j=1}^{k} N(x; \hat{t}^i_j, \Sigma_{ev}) \tag{5}$$

where Σ_{ev} can be set depending on the regressor average error or empirically determined (see Sec. VII-C). Based on this definition, our method estimates the target location at step k as:

$$\hat{T}^i_k = \arg\max_x S^i_k(x) \tag{6}$$

and, using an acceptance threshold θ_{acc}, we can also provide a measure of confidence on the prediction $\hat{T}^i_k$ as:

$$p(\hat{T}^i_k) = \max\left(S^i_k\right)/\theta_{acc} \tag{7}$$

Note that by taking the sum of the evidences in Eq. (5), it is unlikely that a single wrong estimate will have an impact on the peak location (i.e., will not affect the detection). This is in contrast to a multiplicative relation, where the effect of a wrong estimate can be dramatic.

One important consequence of this definition is that we have a criterion to stop the iterative process early, that is, as soon as the point has reliably been found according to Eq. 7.

IV. Evidence-driven sampling

The selection of each new test location l_k^i is performed through a random sampling strategy. Due to the definition of the facial point estimates as the combination of different sources of local image information, it is important to use the most meaningful and uncorrelated information possible. In our case, meaningful locations are those that yield a high estimate likelihood given by Eq. 4, and the benefit of using samples with high likelihood is obvious. Using maximally uncorrelated image information should help prevent shared biases on the local evidences. The use of correlated image information will result in correlated outputs, i.e., correlated predictions. Therefore, the accumulation of local evidences would not arise as a consequence of meaningful image patterns, but instead from the use of clustered test locations that might give poor but consistent estimates. Using maximally uncorrelated image information is the best way to increase the probability that coincident estimates are meaningful, and that erroneous estimates will not group together.

In this section we detail the sampling strategy used to select the test locations. The initial sampling region is defined by the face detection. Then, an evidence-driven sampling strategy is applied, so that the sampling region is updated after each iteration. This strategy is aimed at enforcing two aspects. On the one hand, the evidence-driven sampling strategy uses previous local evidences to focus sampling on a region close enough to the target location to guarantee that the sampled test locations are represented in the training set with high likelihood. On the other hand, we perform what we call non-repetitive sampling within this region, which enforces low correlation between the test locations sampled.

A. Initialisation: Prior Distribution

The point detection process starts with face detection using the Viola&Jones face detection algorithm [40]. This process provides a normalised face box F_{fd} containing the face (in our case normalised to 200x280 pixels). The face location and scale can already provide some information regarding the location of facial points. In order to model this information, a normal distribution is fitted as:

$$p(T^i|F_{fd}) = N\left(T^i; \mu_0^i, \Sigma_0^i\right) \tag{8}$$

where the superindex indicates the facial point. The process to obtain the mean μ_0^i and covariance Σ_0^i is as follows. Firstly, a set of examples where a ground truth location for each point is manually annotated. Subsequently, F_{fd} is obtained for every example image, and the manually annotated points are transformed into the face-box normalised space. This results in a set of scale-and-translation normalised examples. We fit a Gaussian distribution for each point i, which we call the prior distribution for point i.

For each test image I, F_{fd} is computed, and the sampling region of point i is initialised depending on the distribution in Eq. 8 as:

$$R_s^{0,i} = \{x : D_M^{0,i}(x) < a\} \tag{9}$$

where $D_M^{0,i}$ is the Mahalanobis distance defined by the prior distribution, and a can be selected so that the true target location would be included within the initial sampling region for any of the training examples, provided that the same initialisation procedure is followed. It is possible otherwise to optimise it during training.

B. Evidence-driven Sampling Region

To evaluate the SVRs at locations close to the true target location is fundamental. This is due to two reasons. First of all, for a given SVR, the prediction improves when the test location is closer the true target location. Nearby test locations are more likely to yield good estimates. Secondly, it is possible to train more precise regressors rather than general ones, so that the average regressor error is lower. We start the search based on the prior distribution, but keeping this as the sampling region becomes suboptimal as more information about the possible target location is obtained (see Sec. VII-D).

Consequently, in order to provide a more effective sampling region, all the acquired knowledge up to the current iteration is used. The initial sampling region, defined by Eq. (9), is kept fixed for a certain amount of iterations. After that, the sampling region for iteration k is defined as:

$$R_s^{i,k} = \{x : D_M^{i,k}(x) < 2\} \tag{10}$$

where the Mahalanobis distance $D_M^{i,k}$ is defined using mean and covariance:

$$\mu_k^i = \frac{1}{C}\sum_{j=1}^{k} f_{lik}\left(\hat{t}_j^i\right)\hat{t}_j^i \tag{11}$$

$$\Sigma_k^i = \max\left(\frac{1}{C}\sum_{j=1}^{k} f_{lik}\left(\hat{t}_j^i\right)\hat{d}_j^i, R_{min}\right)^2 \cdot I_2 \tag{12}$$

where C is a normalisation factor, and I_2 is the diagonal identity matrix. Since the aim is to use different local information, allowing the region to shrink arbitrarily would be counterproductive. Hence we introduce R_{min} as a term to limit the sampling region shrinkage. This value can be set depending on the training radius, so that the algorithm uses a sampling region large enough to adequately represent the training set.

An example of this strategy is shown in figure 3. It shows how increasingly accurate knowledge of the target location drives the sampling region to more informative areas.

C. Non-repetitive Sampling

The sampling probability is not defined as uniform over the sampling region. Sampling from the sampling region with uniform probability would very likely lead to correlated inputs, and regions producing poor estimates might be sampled repeatedly. The sampling distribution is defined so as to guarantee that regions providing poor information are less likely to be sampled, and that already sampled regions are less likely to be sampled from again. We call this strategy non-repetitive sampling. Thus, the evidence-driven sampling region increases the chance of sampling close to the target,

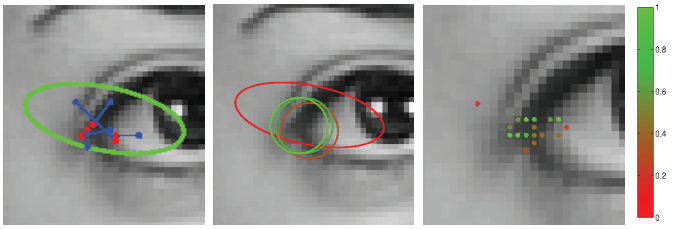

Fig. 3. Left: initial sampling region and first test locations (blue)/estimates (red) (ordering of the samples is not important at this stage). Middle: sampling region evolution (from red to green). Right: estimates for all iterations and their likelihood (green=1, red=0)

while the non-repetitive sampling accounts for both avoiding poor test locations within the sampling region, and removing bias in the inputs as much as possible.

Previously sampled locations are used to modify the sampling probability. Therefore, the sampling distribution depends on the sampling region R_s^i, the previous test locations $\mathbf{l}_k^i$, and their estimated likelihood $\hat{\mathbf{d}}_k^i$. The bold typeface denotes the set of variables up to iteration k, i.e., $\mathbf{l}_k^i = \{l_j^i\}_{j=1:k}$, and $\hat{\mathbf{d}}_k^i = \{\hat{d}_j^i\}_{j=1:k}$. We construct a function that summarises the sampled locations so far and the estimated quality of their predictions as:

$$F_r^i\left(x; \mathbf{l}_k^i, \hat{\mathbf{d}}_k^i, \rho\right) = \sum_{j=1}^{k} \exp\left(-\frac{1}{2}(x - l_j^i)^t \widetilde{\Sigma}_k^{-1}(x - l_j^i)\right) \tag{13}$$

where the covariance term is defined as $\widetilde{\Sigma}_k = \frac{1}{\rho}\hat{d}_k^i I_2$, and $\rho \in (0, \inf)$. This way, test locations with low likelihoods have a larger covariance. The factor ρ controls how much sparsity is enforced, and it is dynamically adjusted.

The sampling probability for location x is defined as:

$$F_s^i(x) = \frac{1}{C}\max\left(\mathbf{1}_{R_s^i} - F_r^i\left(x; \mathbf{l}_k^i, \hat{\mathbf{d}}_k^i, \rho\right), 0\right) \tag{14}$$

where the parameters of F_s^i have been omitted for simplicity, $\mathbf{1}_{R_s^i)}$ is a function with value 1 for any point in R_s^i and 0 otherwise, and C normalised the expression so it integrates to 1.

With this definition, locations around already sampled ones will be less likely to be sampled again. Also, the estimated likelihood defines how large the region of reduced sampling probability is. The sampling probability can be interpreted as sampling from a uniform distribution, and then rejecting and sampling again with a probability given by the normalised version of F_r^i, each of the components of the mixture being centred at a test location, and their covariances being defined by the corresponding likelihood and the sparsity factor ρ.

In practice, the factor ρ can be initialised to 1. After some iterations, it will not be possible to sample from R_s^i anymore. At this stage, the value of ρ is increased by one, thereby lowering the sparsity requirement. We use integer values for ρ purely for computational reasons, and it would be possible to change the value of ρ at each iteration for example using the integral of F_s^i instead.

The point where sampling using $\rho = 1$ is not possible anymore is used in our algorithm to start the evidence-driven sampling region process. This typically happens between iteration 4 and 6, which ensures enough exploration over the initial sampling region. Figure 3 shows in its left image an example of the samples drafted using the initial sampling region.

The aggregation to the target distribution of an evidence in the form of a Gaussian with fixed covariance Σ_{ev} (Eq. 5) relates to this sampling strategy. Since the estimate likelihood is used to modify the sampling probability, it already gives more importance to test locations with higher likelihoods since they are more likely to be sampled again. This introduces a bias in the potential estimates that we corrected by adding evidences with fixed covariances.

An example of the sampling procedure during one full run of the point detection algorithm is shown in Figure 3 for the right-outer eye corner.

V. Probabilistic Graphical Spatial Relation Modelling

Without a shape model, problems such as the presence of partial occlusions, caused by e.g. hair or glares on the glasses, are difficult to overcome. By using a face shape model it is also possible to identify erroneous estimates and provide valid alternatives. The spatial relations are modelled as proposed in [39]. However, instead of correcting the target estimates after each iteration, the face shape model is used to constrain the centre of the search regions. Therefore, the input to the shape model is the set μ_k^i (see eq. (11)). Throughout this section, we will omit the iteration subindex for clarity.

A spatial relation between two points is defined as the line segment between their locations, expressed in polar coordinates. For example, if $\mu^j - \mu^i$ is expressed in polar coordinates as $(\alpha_{i,j}, \rho_{i,j})$, then the spatial relation between locations i and j can be computed as:

$$r_{i,j} = (\alpha_{i,j}, \rho_{i,j}) \tag{15}$$

The construction of the spatial relations is depicted in Fig. 5. Although a subset of spatial relations can be considered, in this work the whole set of spatial relations are considered.

In order to obtain a constellation with consistent spatial relations, we build a probabilistic model to encode their interactions. This is done using an MRF (see [3]) with binary states $s_{i,j}$, where $s_{i,j} = 1$ indicates that the relation $r_{i,j}$ is a valid shape (i.e., it is consistent within the trained models), and 0 indicates otherwise.

Therefore, each node of the network is associated to a spatial relation. We define pairwise relations between nodes, so the probability of the network is decomposed as:

$$p(\{s_{i,j}\}) = \frac{1}{Z}\prod \varphi_{i,j,k,l}(s_{i,j}, s_{k,l}) \prod \psi_{i,j}(s_{i,j}) \tag{16}$$

where Z is a normalisation factor, $\varphi_{i,j,k,l}$ is a function that depends on a training stage, and encodes the compatibility of $s_{i,j}$ and $s_{k,l}$ (which depends on the values of $r_{i,j}$ and $r_{k,l}$), and $\psi_{i,j}$ is the observation potential or, in other words, how

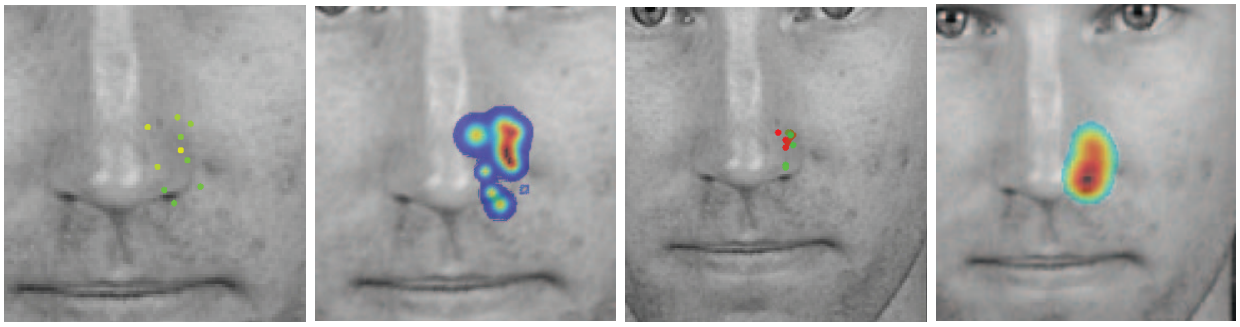

Fig. 4. From left to right: (i) sampled locations, ordered from yellow to green. Note that the effect of the non-repetitive sampling is reflected on the sparseness of the test locations shown here; (ii) final sampling rejection probability; (iii) estimated point locations, the colour indicating its likelihood (red for low, green for high); (iv) target distribution, from which the MAP is the final target detection. N.B. the nose is particularly long, so the search evolves from a location above the nostril towards the true target location.

likely it is that $s_{i,j} = 0/1$ before considering the other nodes. The function φ is defined as follows:

$$\varphi_{i,j,k,l}\left(s_{i,j}=1, s_{k,l}=1\right) = f_\alpha(\alpha_{i,j} - \alpha_{k,l}) \cdot f_\rho(\frac{\rho_{i,j}}{\rho_{k,l}}) \quad (17)$$

where $f = f_\alpha \cdot f_\rho$ are models constructed during a training stage. More precisely, a set of manually annotated images are used for training. The values of $\alpha_{i,j} - \alpha_{k,l}$ and $\frac{\rho_{i,j}}{\rho_{k,l}}$ are computed for each of the examples, and their mean and standard deviation are finally obtained (μ_α, σ_α, μ_ρ and σ_ρ, respectively). During the testing stage, the Mahalanobis distance with respect to these distributions is computed, obtaining d_α and d_ρ. Then $f_{\alpha/\rho}(d_{\alpha/\rho}) = S(2-d_{\alpha/\rho})$, where S is a Sigmoid function. The Sigmoid function can be seen intuitively as a smoothed step function and, in this way, the step is centred at twice the Mahalanobis distance of the Gaussian fitted to the training values. Finally, the other 3 values of $\varphi_{i,j,k,l}$ are equal and defined so that the values of φ sum up to 1.

We opted for this modelling instead of the more standard Gaussian distribution to prevent biasing in favour of more common configurations. That is to say, only those configurations that are not anthropomorphically possible should be penalised. Using a Gaussian distribution introduces a higher bias towards the mean and, in consequence, penalises those configurations that are feasible but less common. For example, with our model, configurations at twice the standard deviation from the mean are penalised around 3 times less than using a Gaussian distribution, while more extreme values get a harsher penalisation than with a Gaussian distribution. Also note that since the measures over the spatial relations are done in terms of their relative angle and ratio of lengths, these measures are invariant to isotropic scale changes and in-plane rotations.

The observation potential ψ encodes the confidence on a spatial relation beforehand. It is defined as:

$$\psi(s_{i,j}) = (1 - \mathrm{p}(r_{i,j}), \mathrm{p}(r_{i,j})) \quad (18)$$

where $\mathrm{p}(r_{i,j}) = \mathrm{p}(\mu^i) \cdot \mathrm{p}(\mu^j)$, and $\mathrm{p}(\mu^j)$ is defined in Eq. (7). This is as opposed to [39], where the observation model was non-informative (i.e. all spatial relations were equally likely *a priori*).

Finally, to test a facial point configuration, the joint MRF is maximised using a Belief Propagation (BP) algorithm [3], [31]. BP consists of an iterative maximisation of the posterior distribution (Eq. (16)). To this end, the algorithm updates iteratively each node using *messages* from the other nodes, which are basically their marginal distribution with respect to the other variables. The output is a *belief* for the state of each node, so a probability $p(s_{i,j} = 0/1)$ is obtained. In our case $p(s_{i,j} = 1)$ is the probability of spatial relation $r_{i,j}$ being correct.

These probabilities refer to the spatial relations, and have to be translated into a confidence over the input points. This can be done as:

$$p(\mu^i) = \frac{1}{N} \sum_{j \neq i} p(s_{i,j} = 1) \quad (19)$$

where N is the number of relations that include point i.

In order to provide an alternative facial point configuration when necessary, i.e., when $p(\mu^j)$ is low, it is possible to predict μ^j from each of the other facial points as $\mu^i + \hat{r}_{i,j}$, where:

$$\hat{r}_{i,j} = \sum p(s_{k,l}) \arg\max_{r_{i,j}} f(r_{i,j}, r_{k,l}) \quad (20)$$

Therefore, each spatial relation casts a prediction on $r_{i,j}$, and its final estimated value, $\hat{r}_{i,j}$, is the weighted average. Finally, the point can be predicted combining all those predictions as:

$$\hat{\mu}^j = p(\mu^j)\mu^j + \sum_{i \neq j} p(\mu^i)\left(\mu^i + \hat{r}_{i,j}\right) \quad (21)$$

One drawback of this approach is how the complexity scales with the number of points. The number of spatial relations for N points is $N \cdot (N-1)/2$, and the number of edges likewise depends on the number of spatial relations. In our case, we opt to divide the point search into successive stages. Following [39] we first detect 7 stable points (points 1-4, 13, 14, and 20), which are those that do not move significantly due to facial expression, using a shape model restricted to these 7 points. We use the estimated positions of such stable points to define the centres of the eyes and the nose (see Fig. 2), and to register the face based on these centre points. The points belonging to

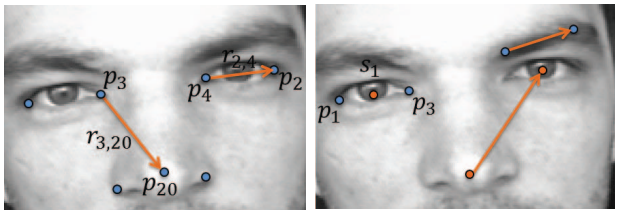

Fig. 5. Left: construction of spatial relations and the 7 stable points. E.g. points p_2 and p_4 produce spatial relation $r_{2,4}$. Right: The shape model for detecting the right eyebrow are the 3 component centres (in orange) and the 2 eyebrow points. Points p_1 and p_3 detected earlier, are used to compute the centre of the left eye.

different facial components are then detected separately (i.e. each eye, each brow, mouth, and chin). The shape model in these cases includes the detected points plus the estimated centres of the eyes and nose. This strategy limits the maximum number of points used in a shape model to 7.

Algorithm V.1: LEAR($priors$)

$F_s^i \leftarrow U_{R_s^{0,i}}$
for $k \leftarrow 1$ *to* $maxIts$
do
- **for** $i \in \mathcal{I}$
  - **do**
    - $l_k^i \sim F_s$
    - $x_l = f_a(I, l_k^i)$
    - $\hat{v} = (r_x(x_l), r_y(x_l))$
    - $\hat{t}_k^i = l_k^i + \hat{v}$
    - $\hat{d}_k^i = r_d(x_{\hat{t}_k^i})$
    - $f_{lik}(\hat{t}_k^i) = exp\left(-\hat{d}_k^i / \sigma_{lik}^2\right)$
    - $S_k^i = UpdateEvidenceDist(S_{k-1}^i, \hat{t}_k^i)(eq.5)$
    - $\hat{T}_k^i, p(\hat{T}_k^i) = updatePrediction(S_k^i)(eq.6)$
    - $F_r^i = UpdateRejectDist(F_r^i, l_k^i, f_{lik})(eq.13)$
    - **if** $updateSampleRegion\ i$
      - **do** $\mu^i = updateMean(f_{lik}, \hat{t}_k^i)(eq.11)$
      - $\Sigma^i = updateCov(f_{lik}, \hat{d}_k^i)(eq.12)$
- **if** $shape == bad$
  - **do** $\mu^i = correctShape(M_{shape}, \mu^i, p(\hat{T}_k^i))(eq.21)$
- **for** $i \in \mathcal{I}$
  - **do**
    - $F_s^i = updateSampleDist(\mu^i, \Sigma^i, F_r^i)(eq.14)$
    - **if** $p(\hat{T}_k^i) > \theta_{acc}$
      - **do** $\mathcal{I} = \mathcal{I} \setminus i$

Now that all the components of the LEAR algorithm have been explained, we provide a summary in algorithm V.1

VI. Appearance Model

The process of estimating the target location from a local patch relies on a feature descriptor to represent the patch, a feature selection procedure, and the SVR. We refer to these elements as the appearance model. In this section we describe the different options considered for the appearance model.

Feature descriptors: We experimented with three popular dense local appearance descriptors: Haar filters, Local Binary Pattern (LBP) histograms, and Local Phase Quantisation (LPQ) histograms.

Haar filters have been used for many object detection and face analysis problems, e.g. for the detection of faces in the popular Viola & Jones face detector [40], or in [39] for facial point detection, from which we adopt the descriptor's parameter values.

Local Binary Patterns were introduced by Ojala et al. [26], and proved to be a powerful means of texture description. By thresholding a 3×3 neighbourhood of each pixel with respect to the value of the central pixel, the operator labels each pixel. We use a lower-dimensional version that only retains 'strong' edges, called uniform local binary patterns [27].

The frequency of occurrence of all possible patterns over a region is recorded by a histogram. This removes information about what pattern occurred where. To retain some of the shape information, the patches were divided into a 3 by 3 grid of sub-patches. The LBP feature histograms are extracted from each sub-patch separately and concatenated into a single feature vector, resulting in a $3 \times 3 \times 59 = 531$ dimensional descriptor.

Local Phase Quantisation was originally proposed by Ojansivu and Heikkila as a texture descriptor that is robust to image blurring [28]. The descriptor uses local phase information extracted using a short-term Fourier transform (STFT) computed over a rectangular M-by-M neighbourhood at each pixel position x of an image patch. Again, histograms are calculated from a 3 by 3 grid of sub-patches, resulting in a 2,304 dimensional feature vector.

Regression The relation between the feature descriptor and the target location is learnt by SVR. SVRs are capable of dealing with nonlinear problems and have a reportedly high generalisation capability with few training data. We employ Epsilon-SVRs with a histogram intersection kernel. We thus need to optimise for the slack variable C and the error-insensitive margin ϵ. Parameter optimisation is performed through exhaustive multi-scale grid-search in a separate subject independent cross-validation loop during training, i.e. independently from the test data. We follow the same procedure to train r_x, r_y and r_{dist}.

Feature selection SVR performance decreases when the dimensionality of the feature vector is too large. Our training set consists of some 800 images, so we are indeed in danger of over-fitting to the training set. One way to overcome this problem is to reduce the number of features used for training using feature selection algorithms.

We considered using AdaBoost regression [11] for feature selection, where multi-ridge regression was used as the weak regressor. We observed that the performance of AdaBoost was not stable for all facial points, often selecting as few as two features. We therefore experimented with another basic feature selection technique called Correlation-based Feature Selection (CFS, [16]). CFS iteratively adds features to the set of selected features that correlate strongly with the label, but weakly with the set of features that have been selected so far.

VII. Evaluation

In this section we present four sets of experiments. We first optimise the parameters used, including the feature representation, training example selection parameters, and internal parameters. We also experiment with the impact of

TABLE I
OVERVIEW OF EXPERIMENTS AND THE DATA USED IN EACH FOR TRAINING AND TESTING.

Exp	Goal	Test data
1	Param. optimisation	MMI (122), FERET (259), XM2VTS (92)
2	Benchmarking	BioID (1300)
3	Expressions	MMI (122), SEMAINE (2380)
4	Pose & Illumination	Multi-PIE (3360)

different algorithm components in the performance, and the variance between different runs. Secondly, we perform a set of database-independent experiments that include a comparison of our algorithm with other state-of-the-art methods. Thirdly, we evaluate its performance on expressive data, both posed and spontaneous. Finally, we test the robustness of our system with respect to variations in head pose and illumination conditions. Table I gives an overview of the different experiments, and what data the algorithm was tested on. All systems were trained using the same set of core databases.

A. Databases

Each of the experiments conducted in this section requires different database characteristics. Below we list the databases used and describe their particularities.

The *MMI Facial Expression Database* [37] contains posed activations of Action Units (AUs), as defined in the Facial Action Coding System (FACS) [13], and is AU labelled per frame. It has high image quality, little variation in head pose and frontal homogeneous illumination.

The *FERET* [32] and the *XM2VTS* [24] databases contain images of high quality, with frontal illumination, and with little or no expression. The ethnicity, gender, and age is varied. Many images display significant glare (see Fig. 9).

The *BioID database* [18] has been recorded using a low-cost web-cam placed in an office scenario. It has natural illumination conditions and head pose variation. The quality of the images is worse than XM2VTS and FERET, and expressions and speech are frequently displayed. As a drawback, subjects are mostly Caucasian. This database is traditionally used for comparison against state-of-the-art methods in the facial point detection topic (see Fig. 11).

The *SEMAINE database* [23] contains sequences with spontaneous expressions in dyadic interactions. There are frequent speech-related poses, and face movements include in- and out-of-plane rotations. There is frontal illumination and the image quality is high.

The *Multi-PIE Database* [15] contains images taken in a controlled setting, where the subjects pose different facial expressions. Images are captured from different angles, and 20 directional illumination conditions.

The only pre-condition for an image to be included in the experiments was that the face was correctly detected by a Viola & Jones face detector. For the BioID this resulted in 1520 images, while from the MultiPIE database 3360 images were used, containing 34 different subjects coming from the third partition (we considered the facial expressions displayed in this partition to be of more interest). In this case we averaged the face detection over the 20 different illuminations. By doing so we avoid including the sensitivity of the face detector used to illumination conditions in the evaluation of our method, and guarantee that all images are used. The SEMAINE database consists of a number of partitions; we use here the solid-SAL partition, which is the most heavily annotated. We selected 20 frames from each of the 119 videos, resulting in 2380 images. The selected frames were equidistant in time, and always included the first and last frame of the video. The selection of images from the MMI, XM2VTS, and FERET databases is described below.

B. Experimental Methodology

We created what we call the core dataset using a combination of 244 images taken from the MMI Facial Expression database, 518 from the FERET database and 184 from the XM2VTS database, totalling 946 images of 622 different subjects. All facial points were manually annotated.

In experiment 1, the core dataset was used to find the optimal parameters for our system (see section VII-C) and to assess the variability of the detections over multiple runs. The results for these tests were obtained by splitting the core set into two subject-independent partitions, one for training of the algorithm and the other to test it. In experiments 2-4 we trained the regressors on the entire core dataset. Please note that experiments 2 and 4 are thus entirely database independent, and that experiment 3 is partially database independent.

Throughout the experimental sections, we use the inter-ocular distance (IOD)-normalised error, defined as:

$$e_i = \frac{||p_d^i - p_m^i||_2}{d_{IOD}} \tag{22}$$

where p_m^i is the manually annotated location of point i, p_d^i is the automatically detected facial point location, and d_{IOD} is the inter-ocular distance, defined as the distance between the eye centres (see Fig. 2). In this way, the error measure does not depend on the size of the image analysed.

Since there is a stochastic component in our algorithm, the performance should not only include the mean error but also its variance. However, we include an experiment that evaluates the variance of the detection error over 10 runs of the algorithm on the same set of images. Since the experiment showed very little variance for different runs, we spare the inclusion of the variance of the error in the remaining results.

We used the BioID database (see Fig. 11) for comparison against other methods. This is a procedure followed by most facial point detection works. The BioID database includes manual annotation for 20 facial points. However, our point definition differs slightly from that provided with the BioID database, most notably for the outer brow point and nostril points. We thus had to partially re-annotate the BioID database. To guarantee reproducibility of the results, we provide our annotations on the group website http://ibug.doc.ic.ac.uk/.

C. Experiment 1: Parameter Optimisation

A number of parameters influence the performance of our method, which can be grouped as follows. On the *appearance*

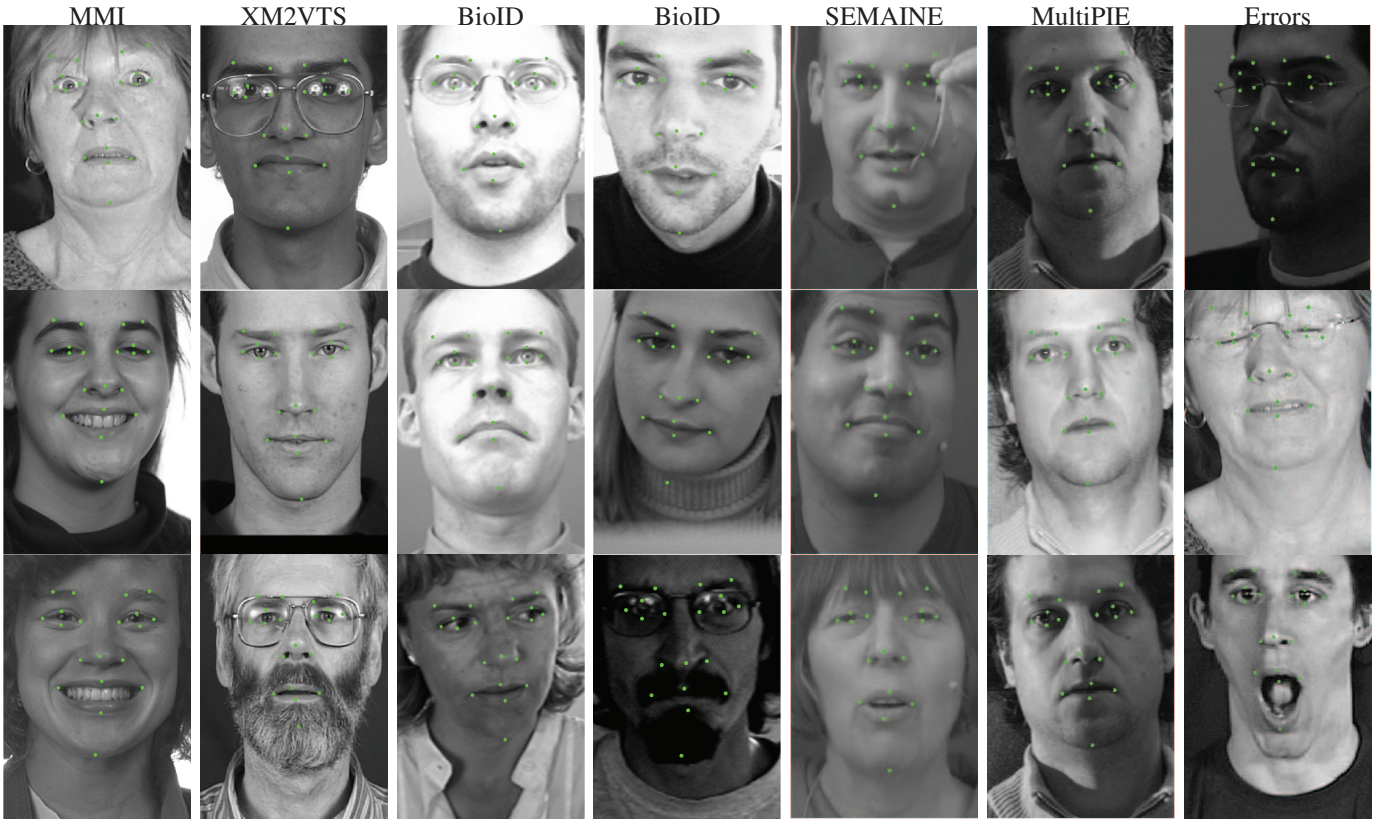

Fig. 6. Results of the point detector on images in the different databases used, and prototypical errors (out of plane rotation, facial hair, and extreme facial expressions). Each column displays examples of the database indicated on its top. The examples of the MultiPIE database show the detection for the first subject of session 3 (first subject of the portion considered here) for frontal and lateral illumination conditions. The errors are 0.067, 0.046 and 0.052 respectively.

modelling side we have to decide on the type of dense appearance descriptor used (we tested Haar features, LBP, or LPQ descriptors), and the feature selection technique applied (AdaBoost or CFS). The *selection of regressor training instances* is controlled by two parameters: the radius of the circular area around a target point from which training elements are sampled (sampling radius), and the number of training elements sampled from that area per image (sampling density). For the *internal parameters* of the detection algorithm, we need to find the initial search area (a in eq. (9)), the minimum radius of the sampling region (R_{min} in eq. (12)), the covariance of the Gaussian which is added to the target distribution for each tested location (Σ_{ev} in eq. 5), the acceptance threshold (θ_{acc} in eq. (7)), and a parameter for transforming the predicted distance to a likelihood (σ^2_{lik} in eq. 4).

Appearance modelling: The appearance descriptor and feature selection technique are selected to minimise the Mean Square Error (MSE) of the predictions, measured in pixels in IOD-normalised image coordinates. They can therefore be optimised independently from the internal parameters of the full detection algorithm. To this end, we evaluate the regressors over all points at a distance to the facial point location smaller than the sampling radius. The test was carried out with the sampling radius set to 10 pixels, and the sampling density for training set to 20 examples per point per image for all tested combination.

Figure 7 shows the results for the three different appearance descriptors and two feature selection techniques we tested. The figure clearly shows that a combination of LBP with CFS performs best.

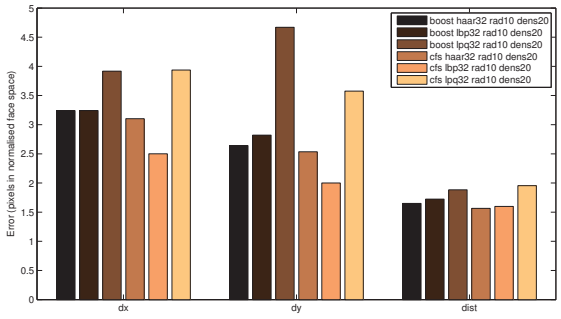

Fig. 7. Evaluation results for different appearance descriptors and feature selection techniques.

Selection of training instances: For simplicity's sake, we assumed that the sampling density and sampling radius are independent of the appearance model parameters, using the best performing LBP features and CFS feature selection combination. The selection of the ideal sampling density can be based only on the performance of the regressors trained. Figure 8 shows the results for the prediction of Δx, Δy, and $dist$ for different values of the sampling density. It shows that

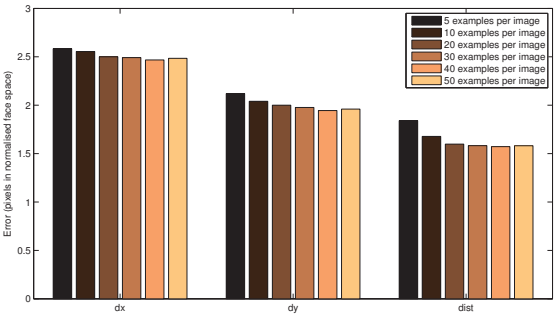

Fig. 8. Average appearance model error for varying numbers of training examples used per image.

TABLE II
POINT-WISE POINT AVERAGE DETECTION ERROR EVALUATED ON THE CORE SET USING THE OPTIMAL PARAMETERS FOUND (IN PERCENTAGE OF THE IOD).

Point	Mean error	Std Error	Point	Mean error	Std error
1	0.033	0.023	11	0.027	0.018
2	0.036	0.034	12	0.031	0.023
3	0.032	0.019	13	0.036	0.030
4	0.031	0.022	14	0.036	0.032
5	0.056	0.022	15	0.041	0.049
6	0.061	0.042	16	0.036	0.040
7	0.064	0.046	17	0.037	0.041
8	0.063	0.052	18	0.061	0.093
9	0.028	0.021	19	0.077	0.098
10	0.029	0.027	20	0.035	0.036

saturation is reached when 40 examples per image are used.

To select the best sampling radius, however, we need to evaluate the performance of the full detection algorithm. Training a regressor with smaller radius will always yield lower MSE within the training area, understood as test locations at a lower distance than the training radius, since the learning task is simpler. In consequence, the adequate balance between accuracy and generality has to be decided in terms of the general algorithm performance.

Since this evaluation depends on the detection algorithm parameters, which have not yet been optimised, we set the parameters to a reasonable default value and optimise the radius using them. Doing so we found that the optimal radius for determining Δx and Δy radius was 10 pixels in IOD-normalised coordinates, while for the distance regressor it was 20. From the distance regressor we need a roughly good estimate for all possible locations, because large errors can erroneously focus the search in a wrong area. Thus, a wider radius of the region to include training instances from is better. In turn, this makes the algorithm more robust to wrong estimates for the Δx and Δy regressors, as they will be assigned lower likelihood by the distance regressor.

The results obtained over the core dataset using the optimal parameters is shown in Table II.

Internal parameters: Some internal parameters of the detection algorithm have to be specified, some of them through a performance evaluation. The value of R_{min} at equation (12) can be set to depend on the training radius used. Sampling in a region larger than the training radius would lead to potential test locations we did not train for. In practise, we use $2/3$ of the training radius, since the centre of the sampling region will seldom be exactly over the true target location. The value of Σ_{ev} can be set in terms of the expected error of the regressor when tested at locations trained for. That is, $\Sigma_{ev} = cov\left(\|\hat{t}(x_l)\| : \|l - p_m^i\| \leq s_r\right) \cdot I_2$, where s_r is the sampling radius and p_m^i is the annotated target location. This yields a value of 14 (we restrict Σ_{ev} to be isotropic). We evaluated the influence of the performance for different values of these parameters in a grid search that also included the parameters θ_{acc}, σ_{lik}^2 and a, but we found that the performance is not sensitive to the values of Σ_{ev} and R_{min}.

We used the same grid search to optimise the parameters θ_{acc}, σ_{lik}^2 and a. The experimental results show a very limited impact of the parameters, except for the initial search region a, for which we found the optimal value to be 2. Both the acceptance thresholds θ_{acc} and σ_{lik}^2 yielded almost no variation in the results. We decided to set θ_{acc} to 0.2 to obtain an earlier stop of the process. σ_{lik}^2 was set to 15, but the performance does not vary for values between 10 and 25.

TABLE III
AVERAGE, MEDIAN, AND STANDARD DEVIATION OF THE ERROR FOR THE FULL ALGORITHM AND THE ALGORITHM WITH SOME COMPONENT DEACTIVATED. THE RESULTS WERE OBTAINED USING 2040 IMAGES OF THE MULTIPIE DATABASE.

Alg. Var.	Mean error	Std Error	Median error
Full	0.052	0.015	0.049
No shape model	0.065	0.023	0.060
No ev.-driv. sampling	0.059	0.018	0.056
No non-repet. sampling	0.059	0.018	0.056

D. Experiment 1: Importance of the algorithm components

In order to assess the importance of each of the algorithm components we performed an experiment using a set of images from the MultiPIE database. In total, 2040 frontal images have been used. We computed the performance of the full algorithm, a version where the shape correction was deactivated, another where the evidence-driven sampling was deactivated, and a final one without non-repetitive sampling. The results are shown in table III in terms of average error, its standard deviation, and the median error.

E. Experiment 1: Detection stability

The detection process relies heavily on a stochastic sampling strategy. It is therefore necessary to analyse the variance of the results when the experiments are repeated in exact conditions. To this aim, we have run the full test on the core test set (403 images) 10 times.

For each image and for each point, we can consider the set of error vectors $e_k = p_m^k - p_d^k$, where $k = 1:10$ is an index for the 10 runs, and the point coordinates are in the normalised facebox space. Over these values, we can compute the standard deviation on the x and y component. Averaging these values for all the images and all the points we obtain a value of 0.92 and 0.74 respectively. This means that the average deviation of a point detection over 10 runs is less than 1 pixel in either direction. When considering the standard deviation per point,

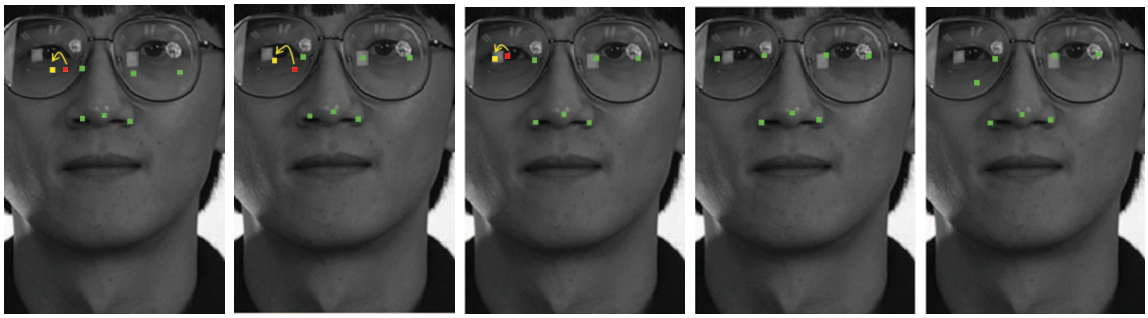

Fig. 9. The first 3 images show the sampling region centres before (red) and after (yellow) shape correction. Green points were not corrected. The two last images show the stable point detection obtained with and without using the shape model.

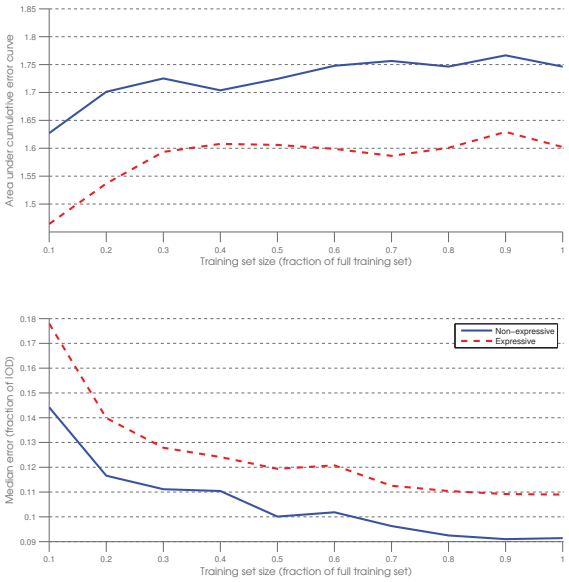

Fig. 10. Effect of the training size for expressive and non-expressive faces. Results on the left show the area under the cumulative error curve, results on the right the median error, measured in terms of the inter-ocular distance.

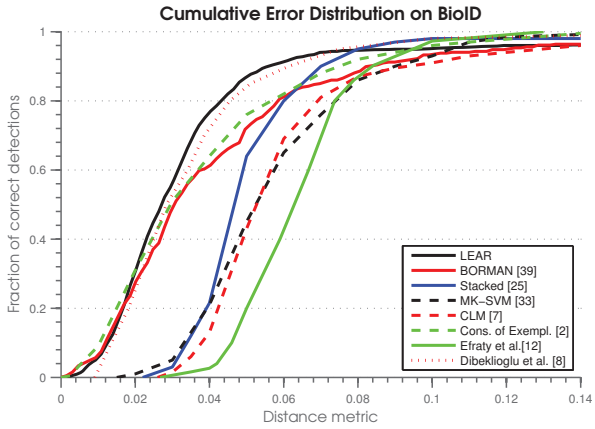

Fig. 11. Performance on data extracted from the BioID database. Left shows the cumulative error distribution of correct detections respect to the acceptance threshold (see text) of a number of recent point detection methods.

the chin shows the maximum of all points with a value of 1.49. This is due to the poor local image texture of this point. We also considered the variation of the error per image, measured as the IOD-normalised error average over the 20 points. In this case, the standard deviation is $3.3{\cdot}10^{-4}$. When considering the stable and the unstable points separately, the typical deviation is kept in the same levels, $3.6{\cdot}10^{-4}$ and $4.6{\cdot}10^{-4}$, respectively. It is worth remarking that while the point detection algorithm contains an important stochastic step inside, the effect of this random component is negligible in the final results. This indicates that the solution is almost entirely determined by the image properties, and that the search strategy is effective. This desirable property is typical for exhaustive search approaches, and traditionally one of their main advantages over estimation-based approaches. We have now shown that the same property can hold for an estimation based approach.

F. Experiment 2: Comparison with State of the Art

To compare our work with the current state of the art, we evaluated our method on the BioID database. This publicly available database is one of the benchmark databases for facial point detection methods. Methods recently evaluated on this dataset are a statistical coarse-to-fine Gabor wavelet method [8], the Consensus of Exemplars method [2], the multiple kernel Support Vector Machines (MK-SVM) method [33], two Active Shape Model methods ([7], [25], denoted as CLM and Stacked Model respectively), a sub-shape based regression method [12], and a regression-based sequential approach [39]. All systems shown in Fig. 11 are database independent, including ours, except for the work of Efraty et al. [12], which used 104 images of BioID to train their system.

The results of this test are shown in Fig. 11. It shows the cumulative error distribution of the m_{e17} error measure. The measure m_{e17} is defined in [7] as the mean error over all internal points, that is, all points that lie on facial features instead of the edge of the face. In our case, that would mean all points except for the point on the chin. However, neither the CLM nor the Stacked Model approaches detect the eyelids. So, to allow a fair comparison, we have excluded the four points on the eyelids as well when calculating m_{e17}.

A point concerning the validity of the performed comparison

TABLE IV
ROBUSTNESS TO EXPRESSION. PI IS THE SET OF POINTS THAT CHANGE POSITION AND/OR APPEARANCE DURING EXPRESSION.

AU	PI	Nr Images	Error on PI	Error all points
1	5-8	20	0.10	0.052
2	5-8	24	0.097	0.050
4	5-8	20	0.083	0.048
5	1-4, 9-12	15	0.033	0.049
6	1-4, 9-12	8	0.028	0.042
7	9-12	7	0.033	0.047
9	1-14, 17	7	0.055	0.054
12	15-18	13	0.076	0.058
15	15-18	3	0.047	0.042
17	18, 19	8	0.046	0.047
18	15-18	8	0.051	0.045
20	15-18	10	0.041	0.037
22	15-18	4	0.076	0.051
24	15-18	2	0.050	0.045
25	15-18	44	0.088	0.058
26	15-19	13	0.041	0.040
27	15-19	8	0.230	0.090
45	1-4, 9-12	14	0.052	0.066

should be noted here: a completely objective comparison with other methods is not possible due to the difference in training sets. However, this is beyond our control, and we can assume that the authors have tried their best to gather as good a training set as possible.

G. Experiment 3: Robustness to facial expressions

We tested how robust our approach is under facial expression changes. The MMI database is used to show performance per action unit (AU) on posed facial expressions (database-dependent experiment), while the SEMAINE database is used to show performance under spontaneous facial expressions (database independent). Furthermore, the partition of the SEMAINE database used contains human-human dyadic interaction.

Posed facial expressions: In this database, facial expressions are described in terms of Action Units (AUs), as defined in the Facial Action Coding System (FACS) [13]. Table IV gives the error per AU. The table shows the number of images that contained that AU, the average error over all points that change their position and appearance during the AU involved, as well as the error over all points for that image.

Spontaneous facial expressions: All the datasets used in the tests above are to a large extend posed in terms of the facial behaviour shown. To test how well our system would perform in a real-world scenario of human-human, human-machine, or human-robot interaction, we evaluated our system on the SEMAINE database.

Figure 12 shows the performance of LEAR on this dataset. As this is the first time facial point localisation is applied to the SEMAINE database, we slightly modify the performance measure. We used the error measure me_{19}, which is defined as a modification of the me_{17} where the eyelids are included, and the pupils are excluded. Figure 12 shows the me_{19} cumulative error distribution, while the right-hand side shows the median error per facial point, in terms of the inter-ocular distance.

Amount of training data required: We were interested to see how much training data is needed for expressive images versus non-expressive images. To test this, we trained the point detector using a variable fraction of the training data. We then determined the performance on images of the core test set derived from the MMI database (expressive images) and from the FERET/XM2VTS databases (non-expressive images) separately. Figure 10 shows the results of this test, measured in terms of the area under the cumulative error curve and the median error. The area under the curve was obtained by taking the integral of the cumulative error for errors between 0 and 1 $\times$ IOD. It indicates that approximately twice as much training data is needed to detect points in expressive images with the same accuracy as points in non-expressive images.

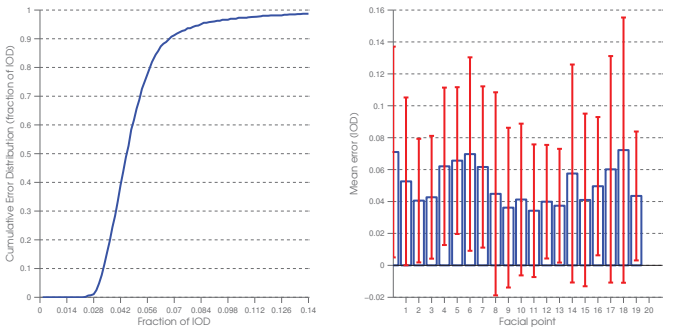

Fig. 12. Cumulative error distribution (left) and median error per point (right) for the SEMAINE dataset.

H. Experiment 4: Robustness to pose and illumination

We used the MultiPIE database to test the performance of our algorithm with respect to variations in head pose and illumination conditions. Despite our algorithm being trained for near-frontal images with homogeneous frontal illumination, we wanted to test its robustness against these conditions. The MultiPIE database offers a discrete set of poses in combination with different illumination conditions and facial expressions.

Head pose: Our algorithm can cope at some extent with out-of-plane rotations. Two components play a major role in the degradation of the performance with respect to head pose. The shape model becomes too restrictive, and the textures are increasingly different from the ones used for training the regressors, thus degrading their performance. The tests were carried over the third partition of the MultiPIE database, since we considered the expressions displayed on it the most interesting ones. We restricted the tests to the examples at 0, 15 and 30 degrees (we assume the performance is symmetric, so it would perform equally well for -30 as for 30 degrees). We tested all illuminations and expressions within these parameters for a total of 2040 images. The results are shown in Fig. 13. It is clear from the graph that the algorithm performs well for 15 degrees, but degrades significantly for 30 degrees.

Illumination: The MultiPIE database includes 20 different illumination conditions for each of the faces (the images are actually taken within a 0.7 sec. time lapse using synchronised flashes). Fig. 13 shows the results obtained for a test conducted over 34 subjects and 3 different facial expressions per subject. Some of the illumination conditions are lateral, while index 7

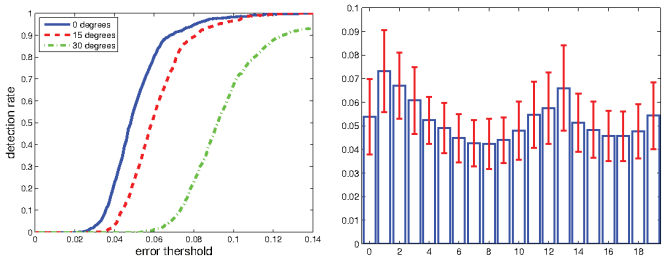

Fig. 13. Experiments over the MultiPIE database. Left: cumulative error distribution for different head poses. Right: average error and its standard deviation (y-axis) for 20 different illumination conditions (x-axis, indexes preserve the database conventions)

corresponds to frontal illumination. Illumination condition indices 0 and 19 correspond to no illumination, and illumination conditions with indices 1 and 13, the worst performing ones, have a 90 degrees angle with respect to the frontal face pose. It is important to note that no illumination normalisation has been performed, either to the face as a whole or to each of the analysed patches individually.

VIII. Conclusions

We presented a novel facial point detector algorithm that uses an estimation-based approach that employs Local Evidence Aggregated Regression (LEAR). Extensive experimental results on images taken from 6 databases, picturing over 700 different persons in over 7000 images show that LEAR outperforms the current state of the art in terms of accuracy. We have also shown that the point detector performs well in the presence of facial expressions, although a performance drop of 10-30% is still observed. While the robustness to facial expressions is currently wholly due to learning the various appearances of facial points caused by expressions and by including faces with expressions in the shape model, we intend to explicitly address the detection of point in the presence of facial expressions in our future work. With regards to addressing arbitrary head-pose, the algorithm presented in its current form can only process near-frontal views. In particular the experiments show robustness for a range of at least 30 degrees of head pose variation. We believe it can be extended to deal with arbitrary head pose given a modestly accurate pose detector.

References

[1] S. Avidan. Support vector tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(8):1064–1072, 2004.
[2] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
[4] S. Coşar and M. Çetin. A graphical model based solution to the facial feature point tracking problem. *Image Vision Computing*, 29(5):335–350, 2011.
[5] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE European Conf. Computer Vision*, 2:484–498, 1998.
[6] D. Cristinacce and T. Cootes. Boosted regression active shape models. *Proc. British Machine Vision Conference*, pages 880–889, 2007.
[7] D. Cristinacce and T. Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41:3054 – 3067, 2008.
[8] H. Dibeklioglu, A. Salah, and T. Gevers. A statistical method for 2-d facial landmarking. *Image Processing, IEEE Transactions on*, 21(2):844 –858, 2012.
[9] L. Ding and A. M. Martinez. Features versus context: An approach for precise and detailed detection and delineation of faces and facial features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:2022–2038, 2010.
[10] P. Dollar, P. Welinder, and P. Perona. Cascaded pose regression. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1078–1085, 2010.
[11] H. Drucker. Improving regressors using boosting techniques. *Int'l workshop on Machine Learning*, pages 107–115, 1997.
[12] B. Efraty, C. Huang, S. Shah, and I. Kakadiaris. Facial landmark detection in uncontrolled conditions. In *Proc. Int'l Joint Conf. Biometrics*, 2011.
[13] P. Ekman, W. V. Friesen, and J. C. Hager. *FACS Manual*. A Human Face, May 2002.
[14] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.
[15] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807 – 813, 2010.
[16] M. Hall. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, The University of Waikato, 1999.
[17] Y. Hu, Z. Zeng, L. Yin, X. Wei, X. Zhou, and T. S. Huang. Multi-view facial expression recognition. In *Proc. IEEE Int'l conf. on Automatic Face and Gesture Recognition*, pages 1–6, 2008.
[18] O. Jesorsky, K. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. *Lecture Notes in Computer Science*, pages 90–95, 2001.
[19] T. Kozakaya, T. Shibata, M. Yuasa, and O. Yamaguchi. Facial feature localization using weighted vector concentration approach. *Image and Vision Computing*, 28(5):772–780, 2010.
[20] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, 2008.
[21] L. Liang, F. Wen, Y.-Q. Xu, X. Tang, and H.-Y. Shum. Accurate face alignment using shape constrained markov network. In *Computer Vision Pattern Recognition*, volume 1, pages 1313–1319, 2006.
[22] S. Liwicki, S. Zafeiriou, and M. Pantic. Fast and robust appearance-based tracking. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition (FG'11)*, pages 507 –513, 2011.
[23] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schröder. The semaine database: annotated multimodal records of emotionally coloured conversations between a person and a limited agent. *Transactions on Affective Computing*, 2011. in print.
[24] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSbd: The extended M2VTS database. In *Conf. Audio and Video-base Biometric Personal Verification*. Springer Verlag, 1999.
[25] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. *Proc. IEEE European Conference on Computer Vision*, pages 504–513, 2008.
[26] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51—59, 1996.
[27] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.
[28] V. Ojansivu and J. Heikkilä. Blur insensitive texture classification using local phase quantization. *ICISP*, pages 1–8, 2008.
[29] M. Pantic and L. Rothkrantz. Expert system for automatic analysis of facial expressions. *Image and Vision Computing Journal*, 18(11):881–905, 2000.
[30] I. Patras and E. R. Hancock. Coupled prediction classification for robust visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1553–1567, 2010.
[31] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.
[32] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.
[33] V. Rapp, T. Senechal, K. Bailly, and L. Prevost. Multiple kernel learning svm and statistical validation for facial landmark detection. In *Automatic Face and Gesture Recognition*, pages 265–271, 2011.
[34] O. Rudovic, I. Patras, and M. Pantic. Coupled gaussian process regression for pose-invariant facial expression recognition. In *Proceedings of European Conf. Computer Vision (ECCV'10)*, pages 350–363, 2010.
[35] T. Senechal, V. Rapp, H. Salam, R. Seguier, K. Bailly, and L. Prevost.

Combining AAM coefficients with LGBP histograms in the multi-kernel SVM framework to detect facial action units. *Proc. IEEE Int'l. Conf. Automatic Face and Gesture Recognition*, pages 1–6, 2011.

[36] 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 *British Machine Vision Association*, 2009.

[37] M. Valstar and M. Pantic. Induced disgust, happiness and surprise: an addition to the MMI facial expression database. *Proc. 3rd Intern. Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect*, page 65, 2010.

[38] M. Valstar and M. Pantic. Fully automatic recognition of the temporal phases of facial actions. *Systems, Man, and Cybernetics, Part B*, 2011. in print.

[39] M. F. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *Conf. on Computer Vision and Pattern Recognition*, pages 2729–2736, 2010.

[40] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2002.

[41] D. Vukadinovic and M. Pantic. Fully automatic facial feature point detection using Gabor feature based boosted classifiers. *IEEE Int'l Conf. Systems, Man and Cybernetics*, 2:1692– 1698, 2005.

[42] O. M. C. Williams, A. Blake, and R. Cipolla. Sparse bayesian learning for efficient visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1292–1304, 2005.

[43] S. K. Zhou and D. Comaniciu. Shape regression machine. In *Information Processing in Medical Imaging*, pages 13–25, 2007.

Brais Martinez (M'10) received his BC degree in Mathematics from the Universidad de Santiago de Compostela in 2003 and a MD and PhD in Computer Science from the Universitat Autonoma de Barcelona in 2006 and 2010 respectively, under the supervision of Xavier Binefa. He is currently affiliated as a research associate with the intelligent Behaviour Understanding group (iBUG) of Maja Pantic at Imperial College London.

Michel F. Valstar (M'09) is a Lecturer in the Mixed Reality Lab at the University of Nottingham. He received his masters degree in Electrical Engineering at Delft University of Technology in 2005 and his PhD in computer science with the intelligent Behaviour Understanding Group (iBUG) at Imperial College London in 2008. Currently he is working in the fields of computer vision and pattern recognition, where his main interest is in automatic recognition of human behaviour. In 2011 he was the main organiser of the first facial expression recognition challenge, FERA 2011. In 2007 he won the BCS British Machine Intelligence Prize for part of his PhD work. He has published technical papers at authoritative conferences including CVPR, ICCV and SMC-B and his work has received popular press coverage in New Scientist and on BBC Radio. He is also a reviewer for many journals in the field, including Transactions on Pattern Analysis and Machine Intelligence, Transactions on Affective Computing, Systems, Man and Cybernetics-B and the Image and Vision Computing journal.

Xavier Binefa received BSc degrees in Mathematics from the Barcelona University and in Computer Engineering from the Universitat Autonoma de Barcelona. He received his PhD degree in Computer Vision from the Universitat Autonoma de Barcelona in 1996, where he was an Associate Professor in the Computer Science Department until 2009, when he was contracted by the Universitat Pompeu Fabra as an Associate Professor at the Information and Communication Technologies Department, where he leads the Cognitive Media Technologies research group. He currently hold the position of Head of the Department, to which he was appointed in 2010.

Maja Pantic (M'98, SM'06, F'12) is Professor in Affective and Behavioural Computing at Imperial College London, Department of Computing, UK, and at the University of Twente, Department of Computer Science, the Netherlands. She received various awards for her work on automatic analysis of human behaviour including the European Research Council Starting Grant Fellowship 2008 and the Roger Needham Award 2011. She currently serves as the Editor in Chief of Image and Vision Computing Journal and as an Associate Editor for both the IEEE Transactions on Systems, Man, and Cybernetics Part B and the IEEE Transactions on Pattern Analysis and Machine Intelligence. She is an IEEE Fellow.