Constrained Local Neural Fields for robust facial landmark detection in the wild

Tadas Baltrušaitis Peter Robinson
University of Cambridge Computer Laboratory
15 JJ Thomson Avenue
tb346@cl.cam.ac.uk pr10@cl.cam.ac.uk

Louis-Philippe Morency
USC Institute for Creative Technologies
12015 Waterfront Drive
morency@ict.usc.edu

Abstract

Facial feature detection algorithms have seen great progress over the recent years. However, they still struggle in poor lighting conditions and in the presence of extreme pose or occlusions. We present the Constrained Local Neural Field model for facial landmark detection. Our model includes two main novelties. First, we introduce a probabilistic patch expert (landmark detector) that can learn non-linear and spatial relationships between the input pixels and the probability of a landmark being aligned. Secondly, our model is optimised using a novel Non-uniform Regularised Landmark Mean-Shift optimisation technique, which takes into account the reliabilities of each patch expert. We demonstrate the benefit of our approach on a number of publicly available datasets over other state-of-the-art approaches when performing landmark detection in unseen lighting conditions and in the wild.

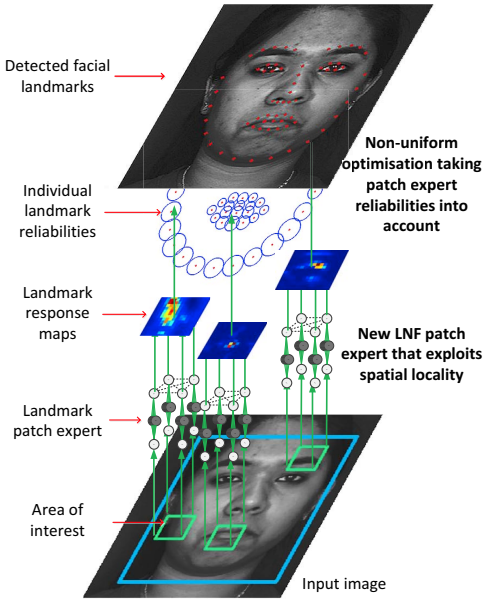


Figure 1: Overview of our CLNF model. We use our Local Neural Field patch expert to calculate more reliable response maps. Optimisation over the patch responses is performed using our Non-Uniform Regularised Mean-Shift method that takes the reliability of each patch expert into account leading to more accurate fitting. Note that only 3 patch experts are displayed for clarity.

1. Introduction

Facial expression is a rich source of information which provides an important communication channel for human interaction. Humans use them to reveal intent, display affection, and express emotion [13]. Automated tracking and analysis of such visual cues would greatly benefit human computer interaction [13]. A crucial initial step in many affect sensing, face recognition, and human behaviour understanding systems is the detection of certain facial feature points such as eyebrows, corners of eyes, and lips. This is an interesting and still an unsolved problem, especially for faces *in the wild* — exhibiting variability in pose, lighting, facial expression, age, gender, race, accessories, make-up, occlusions, background, focus, and resolution.

There have been many attempts, with varying success, at tackling the problem of accurate and person independent facial landmark detection. One of the most promising is the Constrained Local Model (CLM) proposed by Cristinacce and Cootes [5], and various extensions that followed [1, 3, 15, 20]. CLM methods, however, they still struggle in poor lighting conditions, in the presence of occlusion, and when detecting landmarks in unseen datasets.

In this paper, we present the Constrained Local Neural Field (CLNF), a novel instance of CLM that deals with the issues of feature detection in complex scenes. First of all, our CLNF model incorporates a novel Local Neural Field (LNF) patch expert, which allows us to both capture more complex information and exploit spatial relationships between pixels. We also propose Non-Uniform Regularised Landmark Mean-Shift (NU-RLMS), a novel CLM

fitting method which trusts reliable patch experts more. An overview of our model can be seen in Figure 1.

We demonstrate the benefit of our CLNF model by outperforming state of the art approaches when detecting facial landmarks across illumination and *in the wild*. We compare our approach to CLM [15], tree based method [22], DRMF [1], AOM [17], and supervised descent [21] methods. Our approach shows improvement over all of these approaches for across database and illumination generalisation on a number of publicly available datasets.

2. Related work

Facial feature detection refers to the location of certain facial landmarks in an image. For example, detecting the nose tip, corners of the eyes, and outline of the lips. There have been a number of approaches proposed to solve this problem. This section provides a brief summary of recent landmark detection methods followed by a detailed description of the CLM algorithm.

2.1. Facial landmark detection

Zhu *et al*. [22] have demonstrated the efficiency of tree-structured models for face detection, head pose estimation, and landmark localisation. They demonstrated promising results on a number of benchmarks.

Tzimiropoulos presented the Active Orientation Model (AOM) – a generative model of facial shape and appearance [17]. It is a similar to Active Appearance Model (AAM) [10], but has a different statistical model of appearance and a robust algorithm for model fitting and parameter estimation. AOM generalizes better to unseen faces and variations than AAM.

Discriminative Response Map Fitting (DRMF) presented by Asthana *et al*. [1] extends the canonical Constrained Local Model [5]. DRMF uses dimensionality reduction to produce a simpler response map representation. Parameter update is computed from the simplified response maps using regression. Furthermore, the authors demonstrate the benefits of computing the response maps from Histograms of Oriented Gradients [6].

Xiong and De la Torre proposed using the Supervised Descent Method (SDM) for non-linear least squares problems and applied it to face alignment [21]. During training, the SDM learns a sequence of optimal descent directions. During testing, SDM minimizes the non-linear least squares objective using the learned descent directions without the need to compute the Jacobian and/or Hessian.

2.2. Constrained Local Model

Our approach uses the Constrained Local Model (CLM) framework, hence it is described in detail here. There are three main parts to a CLM: a point distribution model (PDM), patch experts, and the fitting approach used. PDM models the location of facial feature points in the image using non-rigid shape and rigid global transformation parameters. The appearance of local patches around landmarks of interest is modelled using patch experts. The fitting strategies employed in CLMs are varied, a popular example is the Regularised Landmark Mean Shift (RLMS) [15]. Once the model is trained on labelled examples, a fitting approach is used to estimate the rigid and non-rigid parameters $\mathbf{p}$, which fit the underlying image best:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}}[\mathcal{R}(\mathbf{p}) + \sum_{i=1}^{n} \mathcal{D}_i(\mathbf{x}_i; \mathcal{I})]. \quad (1)$$

Here $\mathcal{R}$ represents the regularisation term that penalises overly complex or unlikely shapes, and $\mathcal{D}$ represents the amount of misalignment the i^{th} landmark is experiencing at $\mathbf{x}_i$ location in the image $\mathcal{I}$. The value of $\mathbf{x}_i = [x_i, y_i, z_i]^T$ (the location of the i$^{\text{th}}$ feature) is controlled by the parameters $\mathbf{p}$ through the PDM:

$$\mathbf{x}_i = s \cdot R_{2\text{D}} \cdot (\overline{\mathbf{x}}_i + \mathbf{\Phi}_i \mathbf{q}) + \mathbf{t}, \quad (2)$$

where $\overline{\mathbf{x}}_i = [\overline{x}_i, \overline{y}_i, \overline{z}_i]^T$ is the mean value of the i$^{\text{th}}$ feature, $\mathbf{\Phi}_i$ is a $3 \times m$ principal component matrix, and $\mathbf{q}$ is an m dimensional vector of parameters controlling the non-rigid shape. The rigid shape parameters can be parametrised using 6 scalars: a scaling term s, a translation $\mathbf{t} = [t_x, t_y]^T$, and orientation $\mathbf{w} = [w_x, w_y, w_z]^T$. Rotation parameters $\mathbf{w}$ control the rotation matrix $R_{2\text{D}}$ (the first two rows of 3×3 rotation matrix R), and are in axis-angle form, due to ease of linearising it. The whole shape can be described by $\mathbf{p} = [s, \mathbf{t}, \mathbf{w}, \mathbf{q}]$

2.2.1 Patch Experts

Patch experts (also called local detectors), are a very important part of the Constrained Local Model. They evaluate the probability of a landmark being aligned at a particular pixel location. The response from the i^{th} patch expert $\pi_{\mathbf{x}_i}$ at the image location $\mathbf{x}_i$ based on the surrounding support region is defined as:

$$\pi_{\mathbf{x}_i} = \mathcal{C}_i(\mathbf{x}_i; \mathcal{I}) \quad (3)$$

Here $\mathcal{C}_i$ is the output of a regressor for the i^{th} feature. The misalignment can then be modelled using a regressor that gives values from 0 (no alignment) to 1 (perfect alignment).

There have been a number of different methods proposed as patch experts: various SVR models and logistic regressors, or even simple template matching techniques. The most popular expert by far is the linear Support Vector Regressor in combination with a logistic regressor [3, 15, 20]. Linear SVRs are used, because of their computational simplicity, and potential for efficient implementation on images using convolution [20]. Some sample response maps from patch experts can be seen in Figure 2.

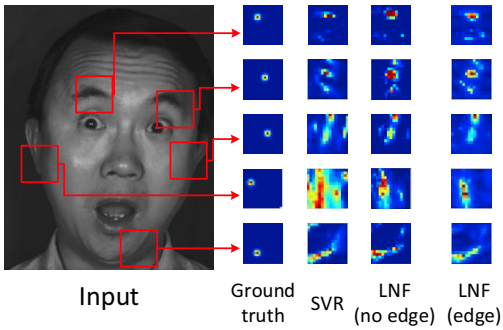


Figure 2: Sample response maps from patch experts of four features (redder is higher probability). The ideal response is shown in ground truth column. SVR refers to the standard patch expert used by CLM approaches. We show two instances of our CLNF model: one with spatial features, (g_k and l_k) and one without. Note how the edge features lead to fewer peaks and a smoother response. Furthermore, note the noisiness of the SVR response.

2.2.2 Regularised Landmark Mean Shift

CLM is a local approach and relies on an initial parameter estimate (often from a face detector). If we have an initial estimate $\mathbf{p}_0$, we want to find a parameter update $\Delta\mathbf{p}$ to get closer to a solution $\mathbf{p}^* = \mathbf{p}_0 + \Delta\mathbf{p}$ (where $\mathbf{p}^*$ is the optimal solution). Hence the iterative fitting objective is as follows:

$$\arg\min_{\Delta\mathbf{p}}[\mathcal{R}(\mathbf{p}_0 + \Delta\mathbf{p}) + \sum_{i=1}^{n} \mathcal{D}_i(\mathbf{x}_i; \mathcal{I})] \tag{4}$$

This can be solved using various methods, the most common of which is the Regularised Landmark Mean Shift [15] which finds a least squares solution to the following [1]:

$$\arg\min_{\mathbf{\Delta p}}(||\mathbf{p}_0 + \Delta\mathbf{p}||^2_{\Lambda^{-1}} + ||J\Delta\mathbf{p}_0 - \mathbf{v}||^2), \tag{5}$$

where J is the Jacobian of the landmark locations with respect to the parameter vector $\mathbf{p}$ evaluated at $\mathbf{p}$, Λ^{-1} is the matrix describing the prior on the parameter $\mathbf{p}$. A Gaussian distribution prior $p(\mathbf{p}) \propto \mathcal{N}(\mathbf{q}; \mathbf{0}, \mathbf{\Lambda})$ is used for non-rigid shape and uniform distribution for rigid shape parameters. Lastly, $\mathbf{v} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]^T$ is the mean-shift vector over the patch responses that approximate the response map using a Gaussian Kernel Density Estimator:

$$\mathbf{v}_i = \sum_{\mathbf{y}_i \in \Psi_i} \frac{\pi_{\mathbf{y}_i} \mathcal{N}(\mathbf{x}_i^c; \mathbf{y}_i, \rho\mathbf{I})}{\sum_{\mathbf{z}_i \in \Psi_i} \pi_{\mathbf{z}_i} \mathcal{N}(\mathbf{x}_i^c; \mathbf{z}_i, \rho\mathbf{I})} - \mathbf{x}_i^c. \tag{6}$$

Mean-shift vector computation depends on the current estimate of the feature $\mathbf{x}_i^c$ and the empirically determined ρ parameter.

[1] $||\cdot||_W$ refers to a weighted l2 norm

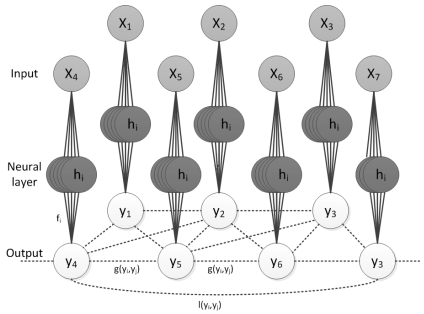


Figure 3: Overview of our patch expert graphical model. Solid lines represent vertex features (f_k, dashed lines represent edge features g_k or l_k). The input vector $\mathbf{x}_i$ is connected to the relevant output scalar y_i through the vertex features that combine the neural layer (Θ) and the vertex weights $\boldsymbol{\alpha}$. The outputs are further connected with edge features g_k (similarity) or l_k (sparsity)

The update rule can be derived using Tikhonov regularised Gauss-Newton method (with regularisation term r):

$$\Delta\mathbf{p} = -(J^T J + r\Lambda^{-1})^{-1}(r\Lambda^{-1}\mathbf{p} - J^T\mathbf{v}) \tag{7}$$

The mean-shifts are calculated and the update is computed iteratively until convergence is met.

3. CLNF

This section presents the Constrained Local Neural Field (CLNF) landmark detection model. It includes a novel Local Neural Field patch expert which learns the non-linearities and spatial relationships between pixel values and the probability of landmark alignment. CLNF also uses a novel Non-uniform Regularised Landmark Mean Shift fitting technique which takes into consideration patch reliabilities. An overview of our technique can be seen in Figure 1.

3.1. Local Neural Field patch expert

Our Local Neural Field (LNF) patch expert, shown in Figure 3, brings the non-linearity of Conditional Neural Fields [11] together with the flexibility and continuous output of Continuous Conditional Random Fields [12]. The proposed patch expert can capture relationships between pixels (neighbouring and longer distance) by learning both similarity and long distance sparsity constraints. LNF also includes a neural network layer that can capture complex non-linear relationships between pixel values and the output responses. It is a continuous output model, with simple and efficient inference.

We identified two types of spatial relationships we want a patch expert to capture. First of all, spatial similarity, that is

pixels nearby should have similar alignment probabilities. Secondly, in the whole area the patch expert is evaluated, only one peak should be present. We want to enforce some sparsity in the response.

We visually show the advantages of modelling spatial dependencies and input non-linearities in Figure 2, that shows patch responses maps from SVR patch experts [20], our LNF patch expert without spatial constraints, and our full LNF patch expert with similarity and sparsity constraints. Note how our patch expert response has fewer peaks and is smoother than the one without edge features, and both of them are more accurate than the SVR patch expert. These spatial constraints are designed to improve the patch response convexity, leading to more accurate fitting.

3.1.1 Model definition

LNF is an undirected graphical model that can model the conditional probability of a continuous valued vector $\mathbf{y}$ (the probability that a patch is aligned) depending on continuous $\mathbf{x}$ (the pixel intensity values in the support region). A graphical illustration of our model can be seen in Figure 3.

In our discussion we will use the following notation: $\boldsymbol{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is a set of observed input variables, $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ is a set of output variables that we wish to predict, $\mathbf{x}_i \in \mathcal{R}^m$ is a vectorised pixel intensities in patch expert support region (*e.g.* $m = 121$ for an 11×11 support region), $y_i \in \mathcal{R}$, and n is the area of interest where we evaluate our patch expert.

Our model for a particular set of observations is a conditional probability distribution with the probability density:

$$P(\mathbf{y}|\mathbf{X}) = \frac{\exp(\Psi)}{\int_{-\infty}^{\infty} \exp(\Psi) d\mathbf{y}} \tag{8}$$

Above $\int_{-\infty}^{\infty} \exp(\Psi) d\mathbf{y}$ is the normalisation (partition) function which makes the probability distribution a valid one (by making it sum to 1). The following section describes the potential function used by our LNF patch expert.

3.1.2 Potential functions

Our potential function is defined as:

$$\begin{array}{rcl} \Psi & = & \sum_i \sum_{k=1}^{K1} \alpha_k f_k(y_i, \mathbf{X}, \boldsymbol{\theta}_k) + \\ & & \sum_{i,j} \sum_{k=1}^{K2} \beta_k g_k(y_i, y_j) + \\ & & \sum_{i,j} \sum_{k=1}^{K3} \gamma_k l_k(y_i, y_j) \end{array}, \tag{9}$$

where model parameters $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots \alpha_{K1}\}$, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{K1}\}$, and $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots \beta_{K2}\}$, $\boldsymbol{\gamma} = \{\gamma_1, \gamma_2, \ldots \gamma_{K3}\}$ are learned and used for inference during testing. We define three types of potentials in our model, vertex features f_k, and edge features g_k, and l_k:

$$f_k(y_i, \boldsymbol{X}, \boldsymbol{\theta}_k) = -(y_i - h(\boldsymbol{\theta}_k, \boldsymbol{x}_i))^2, \tag{10}$$

$$h(\boldsymbol{\theta}, \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}, \tag{11}$$

$$g_k(y_i, y_j) = -\frac{1}{2} S_{i,j}^{(g_k)} (y_i - y_j)^2, \tag{12}$$

$$l_k(y_i, y_j) = -\frac{1}{2} S_{i,j}^{(l_k)} (y_i + y_j)^2. \tag{13}$$

Vertex features f_k represent the mapping from the input $\mathbf{x}_i$ to output y_i through a one layer neural network and $\boldsymbol{\theta}_k$ is the weight vector for a particular neuron k. $\boldsymbol{\Theta}$ can be thought of as a set of convolution kernels that are applied to an area of interest. The corresponding α_k for vertex feature f_k represents the reliability of the k^{th} neuron (convolution kernel).

Edge features g_k represent the similarities between observations y_i and y_j. In our LNF patch expert g_k enforces smoothness on connected nodes. This is also affected by the neighbourhood measure $S^{(g_k)}$, which allows us to control where the smoothness is to be enforced. For our patch expert we define $S^{(g_1)}$ to return 1 (otherwise return 0) only when the two nodes i and j are direct (horizontal/vertical) neighbours in a grid. We also define $S^{(g_2)}$ to return 1 (otherwise 0) when i and j are diagonal neighbours in a grid.

Edge features l_k represent the sparsity constraint between observations y_i and y_j. For example the model is penalised if both y_i and y_j are high, but is not penalised if both of them are zero. This has a slightly unwanted consequence of penalising just y_i or y_j being high, but the penalty for both of them being high is much bigger. This is controlled by the neighbourhood measure $S^{(l_k)}$ that allows us to define regions where sparsity should be enforced. We empirically defined the neighbourhood region $S^{(l)}$ to return 1 only when two nodes i and j are between 4 and 6 edges apart (where edges are counted from the grid layout of our LNF patch expert).

3.1.3 Learning and Inference

In this section we describe how to estimate the parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Theta}\}$. It is important to note that all of the parameters are optimised jointly.

We are given training data $\{\mathbf{x}^{(q)}, \mathbf{y}^{(q)}\}_{q=1}^{M}$ of M patches, where each $\mathbf{x}^{(q)} = \{\mathbf{x}_1^{(q)}, \mathbf{x}_2^{(q)}, \ldots, \mathbf{x}_n^{(q)}\}$ is a sequence of inputs (pixel values in the area of interest) and each $\mathbf{y}^{(q)} = \{y_1^{(q)}, y_2^{(q)}, \ldots, y_n^{(q)}\}$ is a sequence of real valued outputs (expected response maps).

In learning we want to pick the $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\Theta}$ values that maximise the conditional log-likelihood of LNF on the training sequences:

$$L(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Theta}) = \sum_{q=1}^{M} \log P(\mathbf{y}^{(q)}|\mathbf{x}^{(q)}) \tag{14}$$

$$(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}, \bar{\boldsymbol{\gamma}}, \bar{\boldsymbol{\Theta}}) = \arg\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Theta}} (L(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Theta})) \tag{15}$$

It helps with the derivation of the partial derivatives of Equation 14 and with explanation of inference to convert the Equation 8 into multivariate Gaussian form (see Baltrušaitis *et al*. [2] for a similar derivation):

$$P(\mathbf{y}|\mathbf{X}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu})), \tag{16}$$

$$\Sigma^{-1} = 2(A+B+C) \tag{17}$$

The diagonal matrix A represents the contribution of $\boldsymbol{\alpha}$ terms (vertex features) to the covariance matrix, and the symmetric B and C represent the contribution of the $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ terms (edge features).

$$A_{i,j} = \begin{cases} \sum_{k=1}^{K1} \alpha_k, & i = j \\ 0, & i \neq j \end{cases} \tag{18}$$

$$B_{i,j} = \begin{cases} (\sum_{k=1}^{K2} \beta_k \sum_{r=1}^{n} S_{i,r}^{(g_k)}) - (\sum_{k=1}^{K2} \beta_k S_{i,j}^{(g_k)}), & i = j \\ -\sum_{k=1}^{K2} \beta_k S_{i,j}^{(g_k)}, & i \neq j \end{cases} \tag{19}$$

$$C_{i,j} = \begin{cases} (\sum_{k=1}^{K2} \gamma_k \sum_{r=1}^{n} S_{i,r}^{(l_k)}) + (\sum_{k=1}^{K2} \gamma_k S_{i,j}^{(l_k)}), & i = j \\ \sum_{k=1}^{K2} \gamma_k S_{i,j}^{(l_k)}, & i \neq j \end{cases} \tag{20}$$

It is useful to define a vector $\boldsymbol{d}$, that describes the linear terms in the distribution, and $\boldsymbol{\mu}$ which is the mean value of the Gaussian form of the CCNF distribution:

$$\mathbf{d} = 2\alpha^T h(\boldsymbol{\Theta}\mathbf{X}). \tag{21}$$

$$\boldsymbol{\mu} = \Sigma \boldsymbol{d}. \tag{22}$$

Above $\mathbf{X}$ is a matrix where the i^{th} column is $\mathbf{x}_i$, $\boldsymbol{\Theta}$ is the concatenated neural network weights and $h(M)$, is an element-wise application of sigmoid (activation function) on each element of M, and thus $h(\boldsymbol{\Theta}\mathbf{X})$ represents the response of each of the gates (neural layers) at each $\mathbf{x}_i$.

Intuitively $\boldsymbol{d}$ is the contribution from the the vertex features. These are the terms that contribute directly from input features $\mathbf{x}$ towards $\mathbf{y}$. Σ on the other hand, controls the influence of the edge features on the output. Finally, $\boldsymbol{\mu}$ is the expected value of the distribution, hence it is the value of $\mathbf{y}$ that maximises $P(\mathbf{y}|\boldsymbol{x})$.

In order to guarantee that our partition function is integrable, we constrain $\alpha_k > 0$ and $\beta_k > 0, \gamma_k > 0$ [12], while $\boldsymbol{\Theta}$ is unconstrained.

The log-likelihood can be maximised using constrained BFGS. We use the standard Matlab implementation of the algorithm. In order to make the optimisation more accurate and faster we used the partial derivatives of the $\log P(\mathbf{y}|\mathbf{X})$.

To train our LNF patch expert, we need to define the output variables y_i. Given an image with a true landmark at $\mathbf{z} = (u, v)^T$ we can model the probability of it being aligned at $\mathbf{z}_i$ as $y_i = \mathcal{N}(\mathbf{z_i}; \mathbf{z}, \sigma)$ (we experimentally found that best results are achieved with $\sigma = 1$). We can then sample the image at various locations to get training samples. An example of such *synthetic* response maps can be seen in Figure 2, in the ground truth column.

3.2. Non-uniform RLMS

A problem facing CLM fitting is that each of the patch experts is equally trusted, but this should clearly not be the case. This can be seen in Figures 1 and 2, where the response maps of certain features are noisier. RLMS does not take this into consideration. To tackle this issue, we propose minimising the following objective function:

$$\arg\min_{\boldsymbol{\Delta}\mathbf{p}} (||\mathbf{p} + \Delta\mathbf{p}||^2_{\Lambda^{-1}} + ||J\Delta\mathbf{p} - \mathbf{v}||^2_W. \tag{23}$$

The diagonal weight matrix W allows for weighting of mean-shift vectors. Non-linear least squares with Tikhonov Regularisation leads to the following update rule:

$$\Delta\mathbf{p} = -(J^T W J + r\Lambda^{-1})(r\Lambda^{-1}\mathbf{p} - J^T W \mathbf{v}). \tag{24}$$

Note that, if we use a non-informative identity $W = I$, the above collapses to the regular RLMS update rule.

To construct W, we compute the correlation scores of each patch expert on the holdout fold of training data. This leads to $W = w \cdot \text{diag}(c_1; \ldots; c_n; c_1; \ldots c_n)$, where c_i is the correlation coefficient of the i^{th} patch expert on the holdout test fold and w is determined experimentally. The i^{th} and $i + n^{\text{th}}$ elements on the diagonal represent the confidence of the i^{th} patch expert. Patch expert reliability matrix W is computed separately for each scale and view. This is a simple but effective way to estimate the error expected from a particular patch. Example reliabilities are displayed in Figure 4a.

4. Experiments

We conducted a number of experiments to validate the benefits of the proposed CLNF model. First, an experiment was performed to confirm the benefit of both the LNF patch experts and the NU-RLMS approach to model fitting. The second experiment explored how well the CLNF model generalises to unseen illumination. The final set of experiments evaluated how well our approach generalises out of database and *in the wild*.

4.1. Baselines

As the first baseline we used the CLM model proposed by Saragih *et al*. [15]. It uses SVR patch experts and RLMS

fitting for landmark detection. We extended it to a multi-scale formulation for a fairer comparison, hence in the experiments it is called **CLM+**. The exact same training data and initialisation was used for CLM+ and CLNF.

Another baseline used, was the **tree based** face and landmark detector, proposed by Zhu and Ramanan [22]. It has shown good performance at locating the face and the landmark features on a number of datasets. Two differently trained models were used: trained on Multi-PIE [7] data by the authors (called p99) [22], trained on *in the wild* data by Asthana *et al*. (called p204) [1].

Active Orientation Model (**AOM**) is a generative model of facial shape and appearance [17]. We used the trained model (on close to frontal Multi-PIE) and the landmark detection code provided by the authors. In the experiments it was initialised using same face detection as CLNF.

As an additional baseline, we used the Discriminative Response Map Fitting (**DRMF**) model of CLM [1]. We used the code and the model provided by the authors. It was trained using LFPW [4] and Multi-PIE datasets. The model was initialised using a tree based face detector (p204), as that lead to the best results.

As a final baseline, the Supervised Descent Method (**SDM**) was used [21]. This approach is trained on the Multi-PIE and LFW [8] datasets. It relies on face detection from a Viola-Jones face detector, therefore the image was cropped around the desired face for a fairer comparison.

The above baselines were trained to detect 49, 66, or 68 feature points, making exact comparisons difficult. However, they all share 49 feature points that they detect – the feature points in Figure 4a without the face outline. The accuracy on these points was reported in our experiments.

4.2. CLNF

We performed an experiment to see how our novel patch expert and fitting approach affect landmark detection accuracy on the same dataset. This experiment evaluated the effect of using an LNF patch expert instead of an SVR one, and NU-RLMS fitting instead of RLMS.

The experiment was performed on the CMU Multi-PIE dataset [7]. For this experiment we used 3557 frontal and close to frontal images (at poses $051, 050, 140$ corresponding to $-15, 0, 15$ degrees yaw) with all six expressions and at frontal illumination. A quarter of subjects were used for training (890 images) and the rest for testing (2667).

Each of the images was sampled in 21 locations (20 away from the landmark and 1 near to it) in window sizes of 19×19 pixels - this led to 21×89 samples per image. It was made sure that the same person never appeared in both training and testing. Nine sets of patch experts were trained in total: at three orientations — $-20, 0, 20$ degrees yaw; and three scales – 17px, 23px and 30px of interocular distance. The PDM trained by Saragih *et al*. [15] was used.

For fitting we used a multi-scale approach where we first fit on the smallest scale moving to largest. We use $\{15 \times 15, 21 \times 21, 21 \times 21\}$ areas of interest for each scale, and 11×11 patch expert support regions. Other parameters used are: $\rho = 1.5, r = 25, w = 7$.

For fitting on images, the rigid shape parameters were estimated from an off-the-shelf Viola-Jones [19] face detector and non-rigid parameters were set to $\mathbf{0}$. If a face was not detected in an image the rigid parameters were initialised away from the perfect value by the amount expected from the face detector.

Cumulative error curves of using LNF and NU-RLMS can be seen in Figure 4b. The benefit of both our patch expert (LNF) and the fitting technique (NU-RLMS) can be clearly seen, their combination (CLNF) leads to best results. Interestingly, the effect of NU-RLMS is greater when using LNF, this is possibly due to the weights associated with each patch expert being more accurate for LNF.

4.3. Unseen illumination

We conducted an experiment to validate the CLNF ability to generalise to unseen illuminations, which is a very difficult task for facial landmark detectors. The experiment was performed on the left, right and poorly illuminated faces from the Multi-PIE dataset (8001 images).

We used the same models as in the previous experiment – trained on frontal illumination Multi-PIE. The fitting approach was identical to the one in the previous section. We did not want the landmark detection to be affected by face detection accuracy so the detection was initialised using parameters from frontally illuminated images.

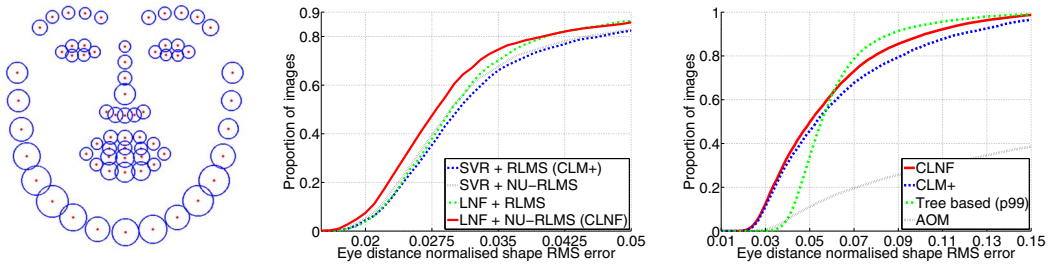
The approaches compared were all trained on frontal illumination Multi-PIE. We do not include other baselines as they were exposed to more difficult illuminations, making the comparison unfair. The results of this experiment can be seen in Figure 4c. The lower error rates of CLNF when compared to other models can be seen. This highlights the benefit of our new patch expert and fitting approach.

4.4. In the wild

The final set of experiments conducted, evaluated how our approach generalises on unseen and completely unconstrained *in the wild* datasets.

For training we used the subsets of two *in the wild* datasets: Labelled Face Parts in the Wild (LFPW) [4] and Helen [9] datasets. Both of them contain unconstrained images of faces in indoor and outdoor environments. In total 1176 training images were used for training the LNF patch experts. As before, nine patch expert sets were trained. Labels from the Helen and LFPW datasets were used to learn the PDM, using non-rigid structure from motion [16].

For fitting we used a multi-scale approach, with $(15 \times 15, 21 \times 21, 21 \times 21)$ areas of interest for each scale, and



(a) Patch reliabilities (b) Fitting on Multi-PIE (c) Fitting across illumination

Figure 4: a) The reliabilities of CLNF patch experts, smaller circles represent more reliability (less variance). b) Fitting on the Multi-PIE dataset, observe the performance boost from both the LNF patch expert and the NU-RLMS fitting method. Their combination (CLNF) leads to the lowest error, since NU-RLMS performs even better on more reliable response maps. c) Fitting on the unseen illumination Multi-PIE subset. Note the good generalisability of CLNF.

11×11 patch expert support regions. Other parameters used are: $\rho = 2.0, r = 25, w = 5$.

In order to evaluate the ability of CLNF to generalise on unseen datasets we evaluated our approach on the datasets labelled for the 300 Faces in-the-Wild Challenge [14]. For testing we used three datasets: Annotated Faces in the Wild [22], IBUG [18] and 300 Faces in-the-Wild Challenge (300-W) [18] datasets. The IBUG, AFW, and 300-W datasets contains 135, 337, and 600 images respectively.

To initialise model fitting, we used the bounding boxes provided by the 300 Faces in-the-Wild Challenge [18] that were initialised using the tree based face detector [22]. In order to deal with pose variation the model was initialised at 5 orientations – $(0, 0, 0), (0, \pm 30, 0), (0, 0, \pm 30)$ degrees of roll, pitch and yaw. The final model with the lowest alignment error (Equation 4) was chosen as the correct one. This makes the approach five times slower, but more robust.

Note that we were unable to use bounding box initialisations for the SDM method, and they were unnecessary for tree based methods.

The results of this experiment can be seen in Figure 5. Our approach can be seen outperforming all of the other baselines tested. This confirms the benefits of CLNF and its ability to generalise well to unseen data. Some examples of landmark detections can be seen in Figure 6. CLNF gap between CLNF and CLM+ is much greater on in the wild images than on the constrained ones. Furthermore, note the huge discrepancy between CLNF and the AAM baseline provided by the authors [10]. This illustrates the greater generalisability of our proposed model over other approaches.

5. Conclusions

We have presented a Constrained Local Neural Field model for facial landmark detection and tracking. The two main novelties of our approach show an improvement in landmark detection accuracy over state-of-art approaches. Our LNF patch expert exploits spatial relationships between patch response values, and learns non-linear relationships between pixel values and patch responses. Our Non-uniform Regularised Landmark Mean-Shift optimisation technique, allows us to take into account the reliabilities of each patch expert leading to better accuracy. We have demonstrated the benefit of our approach on a number of publicly available datasets.

CLNF is also a fast approach: a Matlab implementation can process 2 images per second on *in the wild* data, and 10 images per second on Multi-PIE data, on a 3.5GHz dual core Intel i7 machine. Lastly, all of the code and testing scripts to recreate the results will be made publicly available.

Acknowledgements

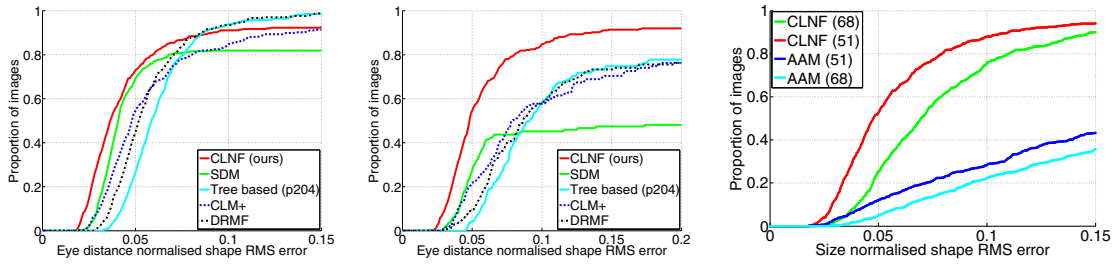
We acknowledge funding support from Thales Research and Technology (UK) and from the European Community's Seventh Framework Programme (FP7/2007- 2013) under grant agreement 289021 (ASC-Inclusion). This material is partially supported by the U.S. Army Research, Development, and Engineering Command (RDECOM). The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *CVPR*, 2013.

[2] T. Baltrušaitis, N. Banda, and P. Robinson. Dimensional affect recognition using continuous conditional random fields. In *FG*, 2013.

[3] T. Baltrušaitis, P. Robinson, and L.-P. Morency. 3D Constrained Local Model for Rigid and Non-Rigid Facial Tracking. In *CVPR*, 2012.



(a) AFW (b) IBUG (c) 300W

Figure 5: Fitting on the wild datasets using our CLNF approach and other state-of-the-art methods. All of the methods have been trained on in the wild data from different than test datasets. The benefit of CLNF in generalising on unseen data is clear. Furthermore, CLNF remains consistently accurate across all three test sets. a) Testing on the AFW dataset. b) Testing on the IBUG dataset. c) Testing on the challenge dataset [18], error rates using just internal and all of the points are displayed.

Figure 6: Examples of landmark detections in the wild using CLNF. Top row is from IBUG dataset, bottom row from AFW. Notice the generalisability across pose, illumination, and expression. The model can also deal with occlusions.

[4] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011.

[5] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *BMVC*, 2006.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[7] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *IVC*, 28(5):807 – 813, 2010.

[8] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. 2007.

[9] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*, 2012.

[10] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[11] J. Peng, L. Bo, and J. Xu. Conditional neural fields. *NIPS*, 2009.

[12] T. Qin, T.-Y. Liu, X.-D. Zhang, D.-S. Wang, and H. Li. Global ranking using continuous conditional random fields. In *NIPS*, 2008.

[13] P. Robinson and R. el Kaliouby. Computation of emotions in man and machines. *Philosophical Transactions B*, 364(1535):3441–3447, 2009.

[14] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *Workshop on Analysis and Modeling of Faces and Gestures*, 2013.

[15] J. Saragih, S. Lucey, and J. Cohn. Deformable Model Fitting by Regularized Landmark Mean-Shift. *IJCV*, 2011.

[16] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: estimating shape and motion with hierarchical priors. *TPAMI*, 30(5):878–892, May 2008.

[17] G. Tzimiropoulos, J. Alabort-i Medina, S. Zafeiriou, and M. Pantic. Generic Active Appearance Models Revisited. In *ACCV*, pages 650–663, 2012.

[18] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge, 2013.

[19] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[20] Y. Wang, S. Lucey, and J. Cohn. Enforcing convexity for improved alignment with constrained local models. In *CVPR*, 2008.

[21] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013.

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