Regression based automatic face annotation for deformable model building

Akshay Asthana a,*, Simon Lucey b, Roland Goecke c,d,**

a Research School of Engineering, College of Engineering and Computer Science, Australian National University, Canberra, Australia
b Information Communication Technology (ICT) Centre, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Sydney, Australia
c Vision and Sensing Group, Faculty of Information Sciences and Engineering, University of Canberra, Australia
d Research School of Computer Science, College of Engineering and Computer Science, Australian National University, Canberra, Australia

A B S T R A C T

A major drawback of statistical models of non-rigid, deformable objects, such as the active appearance model (AAM), is the required pseudo-dense annotation of landmark points for every training image. We propose a regression-based approach for automatic annotation of face images at arbitrary pose and expression, and for deformable model building using only the annotated frontal images. We pose the problem of learning the pattern of manual annotation as a data-driven regression problem and explore several regression strategies to effectively predict the spatial arrangement of the landmark points for unseen face images, with arbitrary expression, at arbitrary poses. We show that the proposed fully sparse non-linear regression approach outperforms other regression strategies by effectively modelling the changes in the shape of the face under varying pose and is capable of capturing the subtleties of different facial expressions at the same time, thus, ensuring the high quality of the generated synthetic images. We show the generalisability of the proposed approach by automatically annotating the face images from four different databases and verifying the results by comparing them with a ground truth obtained from manual annotations.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Over the last 20 years, modelling non-rigid, deformable visual objects through statistical approaches has been a very active area of research. Methods in this family of approaches include, for example, the active shape model (ASM) [10], active appearance model (AAM) [14] and 3D morphable model (3DMM) [6]. These approaches have been shown to perform very well for tasks that require the fitting of a deformable model to a visual object in an image or the tracking of a visual object in an image sequence. They provide highly accurate results, while (at least some of the) methods maintaining computational efficiency [3,33], through a combination of a compact parametric representation with an efficient alignment method.

However, a drawback of these approaches is that they require labelled training images, which typically have to be annotated in a tedious and error prone manual process. The model training process is, thus, largely a manually driven process, rather than an automated process. In general, statistical learning approaches require a reasonable amount of training data, for example, in the form of face images if the task is face tracking. Taking the example of AAMs, they typically require in the order of 10–30 images per subject for person-specific models, depending on the head poses and facial expressions expected. While that may not sound much initially, it must be kept in mind that annotating each of these images involves labelling 50–70 facial landmark points by hand, making annotation of a large database tedious and time consuming, especially when annotating images at arbitrary pose and expression.

Developing an automatic annotation method would allow for an automatic model building process. However, we face a chicken-and-egg problem in that we could automatically annotate unseen images of a visual object, if we had a good model for fitting/tracking, but to build such a good model, we require annotated images. While various approaches for automatic annotation and model building can be found in the literature [4,8,9,18,20,31,32,39], these suffer from problems such as requiring a large number of parameters for the warping function resulting in slow optimisation, requiring a sufficient number of salient features in the visual object to build good appearance models, ignoring the global image structure, or not taking into account the sequential nature of image sequences where available.

1.1. The problem

Generative parametric models, such as active appearance model (AAM) [14], have been well studied in the literature.
An extensive study of state-of-the-art AAMs [16] shows that the performance of person-specific AAMs is substantially better than that of Generic AAMs, in that person-specific AAMs are not only easier to train but also have a more robust fitting as compared to the Generic AAMs. The important conclusions, from [16], that we use to motivate our proposed approach are:

1. The simultaneous inverse compositional (SIC) method [2] is better suited for the task of fitting an AAM to an unseen face image.
2. Person-specific AAMs that use the SIC fitting method are quite robust to pose, expression and illumination variation.
3. Generic AAM fitting on the face of an unseen person, having minor pose and expression variation, using SIC is problematic with a convergence rate of nearly 60% and, hence, it is very challenging to train a Generic AAM to perform fitting on the face of an unseen person having major expression and pose variations.

We now turn our attention to the problem of automatic annotation of unseen face images, having arbitrary expression and pose variations. Referring back to the chicken-and-egg problem (Section 1) one possible solution is to collect a large amount of data to train a Generic AAM and use it for automatic annotation. However, as discussed above, fitting a Generic AAM to the face of an unseen person having unknown pose and expression variations can be problematic and might result in a low convergence rate.

One simple way to approach the problem of automatic face annotation, given just a single annotated frontal face image or a small set of annotated frontal face images, is to generate a set of synthetic images at any arbitrary pose from the given frontal image(s) and to use these to train a deformable model (2D or 3D), e.g., an AAM, to perform fitting on unseen non-frontal images and, hence, annotate them automatically. The advantage here is that the AAMs trained on synthetically generated images are person-specific in nature and can use existing AAM algorithms to perform automatic annotation with consistent accuracy.

Broadly speaking, there are two kinds of approaches to generate synthetic images at any arbitrary pose from a given frontal face image: (1) geometry-based approaches, (2) regression-based approaches.

Geometric approaches, such as non-rigid structure from motion (NRSFM) [37], have been well studied in the literature. Though NRSFM generates reasonably good synthetic images and has the ability to work in real-time, there are few issues with NRSFM approaches for solving the problem of automatic face annotation. Typically, they require annotated images at more than one pose to recover an accurate 3D non-rigid shape. This requirement runs counter to our main aim of using frontal images only. Some extensions of NRSFM [7,38] are capable of recovering the 3D non-rigid shape from a frontal image sequence having little or no out-of-plane motion. However, the availability of an image sequence is an assumption we do not wish to make.

We aim to solve the problem of automatic face annotation given a single annotated frontal image or a small set of annotated frontal images, for example, a set of 4–5 images exhibiting various facial expressions in no particular order and taken at any random point in time. Another main issue, also stated in [38], is the ability of NRSFM approaches to handle noisy data as occur in real world problems. It should be noted here that both training and test face images are manually annotated, which is an error prone process and consistently annotating certain facial parts, such as the chin line, is not possible due to a lack of distinct features. When only frontal images are used, the dataset can contain significant depth ambiguity. Since the manual annotation is typically done in 2D, small displacements in the 2D image plane (due to mis-annotations) can correspond to a large displacement in 3D (for example, face outline or nose outline). As a result, we deal with data potentially having significant noise stemming from mis-annotations.

Hence, the problem of generating non-frontal images from a single annotated frontal image via the NRSFM approach is ill-posed.

1.2. Our approach

In this paper, we propose a regression-based automatic face annotation and model building approach that only requires annotated frontal images, thus drastically simplifying the deformable model building process. We pose the problem of learning the pattern of manual annotation between frontal face images, having arbitrary expressions, and corresponding face images at different poses in a data-driven regression framework. Using this learnt regressor, synthetic images of unseen frontal faces at any arbitrary pose, for which the regressor was trained, are generated by predicting the spatial arrangement of the landmark points and warping the texture from the frontal image. These synthetic images are then used to train a 2D (or 3D) deformable models that can be used to perform fitting on unseen non-frontal images and, hence, annotate them automatically (Fig. 1).

The proposed regression-based approach is dictated by the nature of the automatic face annotation problem. In order to train a 2D (e.g. [14]) or a 3D (e.g. [25]) deformable model, our approach does not require annotated face images at all possible poses. Instead, it generates a set of synthetic images at fixed non-frontal poses from a given single frontal image, for example at 22.5° left, 45° left, 22.5° right, 45° right, 30° up and 30° down. Assuming that these synthetic images are of sufficiently high quality, the approach is able to train a (2D or 3D) deformable model to perform fitting on unseen non-frontal images and, hence, annotate them automatically. Given the relevant training data, generating a set of synthetic images at fixed non-frontal poses is a simpler problem than generating a synthetic image at any arbitrary pose (say via NRSFM) directly from a given single frontal image and
The goal is to learn the regression function $R$ between the set of annotated frontal face images (Pose 0) and corresponding non-frontal face images at Pose $P$, for example, 45° left as shown here.

Fig. 2. The goal is to learn the regression function $R$ between the set of annotated frontal face images (Pose 0) and corresponding non-frontal face images at Pose $P$.

1.3. Contributions

The contributions of this paper can be summarised as follows:

- We propose a simple and efficient regression-based approach to learn the pattern of manual annotation and generate high quality synthetic images at desired poses.
- We look beyond the standard linear regression framework and explore several regression strategies, i.e., a combination of linear or non-linear regression methods and dense or sparse representations, to find the most effective way of modelling the changes in the structure of the face under varying pose and capturing the subtleties of different facial expressions at the same time.
- We empirically test the quality of generated synthetic images and discuss why it is advantageous to use the proposed fully sparse non-linear regression approach to model this problem.
- We show that the generated synthetic images can be effectively used to train both 2D and 3D deformable models. We test the accuracy of these synthetically generated deformable models by automatically annotating unseen images from the CMU PIE database [35] and comparing them to the performance of the person specific 2D and 3D AAMs trained on the manually annotated data (i.e., ground truth data) and the Generic AAM approach. The results show that the performance of the synthetically generated deformable models is comparable to the performance of the person specific 2D and 3D AAMs trained on the ground truth data and drastically better than the Generic AAM approach.
- We show the generalisability of the proposed framework by automatically annotating images from the CMU Multi-PIE [17], Face Pointing [15] and FERET [28] databases, and verifying the results by comparing them with a ground truth obtained from manual annotations.

1.4. Structure of the paper

The rest of this paper is structured as follows. Section 2 discusses related work. Section 3 begins by motivating the proposed approach in a data-driven regression framework, followed by discussing various regression strategies that are explored in this paper. Section 4 describes the methodology to generate the synthetic images at various poses using just a single annotated frontal image via the proposed approach in detail. The quality of synthetic images generated by the proposed approach is empirically evaluated in Section 5. In Section 6, we show that the generated synthetic images can be used efficiently to train both 2D and 3D deformable models. In Section 7, we show that these deformable models, trained on the synthetic images, can be effectively used to automatically annotate unseen face images, having arbitrary pose and expression variations. We also analyse and verify the annotation results by comparing them with the ground truth obtained from manual annotations. Finally, in Section 8, we evaluate the generalisability of the proposed approach by automatically annotating images from four different unseen databases.

2. Related work

The issue of automatically annotating face images and building AAMs has received considerable attention in the research literature in recent years. More generally, the issue is one of the finding pseudo-dense correspondences across images of the same object class. Approaches can be broadly categorised into either image or feature based approaches.

In image-based methods, dense image correspondences are found through a non-linear warping function that minimises some error measure between the pixel intensities. In [20], images are modelled as ‘bags’ of pixel values, enabling the computation of correspondences simultaneously for all images. In [27], a probabilistic approach for modelling the correspondence between the features that were generated by stacking the shape and texture was explored. In [9], a groupwise registration using a set of non-linear diffeomorphic warps is proposed, providing dense correspondences between all images, avoiding the need for manual annotation of training images. In [4], the problem of automatic annotation and model building is recast as an energy-minimising image coding problem. Image-based methods have the advantage that the global image structure is taken into account, thus better mimicking the AAM for which the correspondences are used later. The main disadvantage is that the warping function will generally need to be parameterized using a large number of parameters (as a set of landmark points), which results in a very large optimisation problem that is slow to optimise and prone to terminating in local minima.

In feature-based methods, correspondences are found between salient image features through examining the local structure of the features. In [18], a sparse polygonal representation of one shape’s boundary is matched onto a second shape’s boundary via a greedy optimisation of a cost function. In [8], the point matching algorithm uses an iterative joint clustering and matching strategy, which reduces the computational complexity while maintaining accuracy. In [31,39], image sequences are used to automatically build models, with only the first frame need to be annotated manually, thus exploiting the fact that it is easier to track correspondences in image sequences than finding them between two arbitrary images. In [32], this line of work is extended to take the scene geometry into account through the epipolar constraints in stereo images. The advantage of feature-based methods is that the feature comparisons and calculations are comparatively cheap. Their disadvantages are that they require a sufficient number of salient features in the object and that the global image structure is ignored, as the feature comparisons generally only consider local image structure, which can lead to suboptimal fitting results.
3. Motivation

In this section, we propose the problem of learning the pattern of manual annotation between the frontal and corresponding non-frontal face image in a regression framework.

Given \(M\) annotated frontal face images and their corresponding non-frontal face images (say at pose \(P\)) with \(n\) landmark points each, the goal is to find the regression function \(\mathcal{R}\) that minimises (Fig. 2):

\[
\sum_{m=1}^{M} |x_m - \mathcal{R}(x_m)|,
\]

where \(x\) and \(x'\) are the shape vectors for frontal and non-frontal images, respectively:

\[
x = [x_1, y_1 \ldots x_n, y_n]^T, \quad x' = [x'_1, y'_1 \ldots x'_n, y'_n]^T.
\]

Therefore, the objective is to find a regression function \(\mathcal{R}\) that most accurately models the changes in the shape of the face under varying pose and expression.

Considering the shape of the face as a whole, the function \(\mathcal{R} : x \rightarrow x'\) is non-linear in \(x\). Hence, in this paper:

- First, we pose Eq. (1) in a linear regression framework and explore how well \(\mathcal{R}\) can be approximated as a linear function;
- Second, we seek a simpler and more efficient way to solve Eq. (1) in a non-linear regression framework.

Considering the individual landmark points, the function \(\mathcal{R} : x \rightarrow x'\) is also trying to learn the correspondence between landmark points at pose \(0\) and pose \(P\). Hence, we take into account the relationship between the location and local movements of the individual landmark points. This is motivated by the plethora of work done for pose-invariant face recognition via “patch-based” approaches (e.g. [1,2,23,29]), proposed initially in [23], where a face image is divided into a number of small sub-image patches.

This methodology is based on the hypothesis that: (1) in a holistic approach, it is difficult to take into account the local changes in appearance due to pose differences, because the appearance in different parts of the face changes in a different manner due to the underlying 3D structure of the face and illumination; (2) this problem, discussed above in (1), can be effectively solved by dividing the face into several independent sub-image patches to take into account the local appearance changes as a function of pose more accurately. This was further extended in [1], where the approach not only models how a face patch varies in appearance, but also how it deforms spatially as the pose changes. In [29], patches around marked landmark points were considered independently for the face recognition task. These patch-based approaches have shown strong improvements over the holistic approaches in the ability to handle pose variation for face recognition and have become a common practice.

Motivated by this, we use a similar hypothesis for learning the changes in the shape of the face induced by pose variation. In a dense representation framework, we assume complete dependence between the location and local movements of the landmark points to solve for \(\mathcal{R}\). However, since the face has an underlying complex 3D structure and landmark points, representing different facial features, lie at different depths, sometimes even for the same facial feature (such as the nose), it is difficult to learn how the shape changes due to pose variation because different facial features tend to change the shape in a different manner.

Therefore, in this paper, we solve Eq. (1) in a sparse representation framework to model the relationship between the location and local movements of landmark points automatically by using the training data at hand. We explore how well the local changes in the shape, induced by pose variation, can be modelled for unseen faces with arbitrary expressions.

- Contrary to the assumption of complete dependence between landmark points in the dense representation framework, we solve Eq. (1) in a fully sparse representation framework, where we assume total independence between the landmark points. The benefit of this approach is that it greatly simplifies Eq. (1), thus making it easy to pose the problem in a non-linear regression framework. In order to evaluate whether the assumption of total independence is too restrictive in nature, we solve Eq. (1) in both linear and non-linear regression frameworks and investigate which of the two frameworks is flexible enough to model \(\mathcal{R}\) efficiently.

Therefore, in the following subsections, we discuss different regression strategies to find \(\mathcal{R}\), which solve Eq. (1) by:

- Assuming \(\mathcal{R}\) to be either a linear or non-linear function of \(x\).
- Solving for \(\mathcal{R}\) by posing the problem in a dense or sparse representation framework.

3.1. Dense linear regression (DLR) approach

In the dense-linear regression approach, we pose Eq. (1) in a dense representation framework and assume \(\mathcal{R}\) to be a linear function of \(x\):

\[
\mathcal{R}_L = W_L x + v_L,
\]

where \(W_L\) is an unknown regression/ transformation matrix and \(v_L\) is the noise. Hence, Eq. (1) becomes a standard linear least squares problem [26], which we solve for \(W_L\) by:

\[
\min_j (\|P_j - W_L G_j 0 + \lambda_L \|_{F}^2),
\]

where \(\|g\|_F = \sqrt{\sum_{i} g_{i}^2}\) is the L2 norm of \(a\) and \(\lambda_L > 0\) is a regularisation factor used to avoid over-fitting:

\[
\mathcal{G}_j = [x_1 \ldots x_M] \quad \text{and} \quad \mathcal{P}_j = [x'_1 \ldots x'_M].
\]

Solving Eq. (4) for \(W_L\):

\[
W_L = P_j (G_j G_j^T + \lambda_L I)^{-1}.
\]

Hence, the computed transformation matrix \(W_L\) is a dense matrix of size \(2n \times 2n\).

3.2. Sparse linear regression (SLR) approach

Assuming \(\mathcal{R}_L\) to be a linear function of \(x\) (Eq. (3)), the transformation matrix \(W_L\) is computed by solving a \(L_1\)-regularised least squares problem (Eq. (5)). This gives a very dense solution for \(W_L\) (Eq. (5)), i.e. typically all the elements of matrix \(W_L\) are non-zero [22]. The goal here is to obtain a sparse solution for \(W\), i.e. very few elements of matrix \(W\) should be non-zero.

In [13], it has been shown that if there exists an optimal sparse solution, it can be efficiently computed by convex optimisation. Hence, we recast Eq. (1) as a standard \(L_1\)-regularised least squares problem, which can be reformulated as a convex quadratic program and then solved efficiently to give a sparse solution \(W_S\). With the sparse transformation matrix \(W_S \in \mathbb{R}^{2n \times 2n}\):

\[
G_S = \begin{bmatrix}
    x_1^T \\
    \vdots \\
    x_M^T
\end{bmatrix} \quad \text{and} \quad P_S = \begin{bmatrix}
    x'_1 \\
    \vdots \\
    x'_M
\end{bmatrix},
\]

where both \(G_S\) and \(P_S \in \mathbb{R}^{M \times 2n}\).
Consider the linear model:

\[ \mathbf{p}^i_l = \mathbf{G}_l \mathbf{w}^i_l + \mathbf{v}^i_l, \quad i = 1, \ldots, 2n, \]

where vectors \( \mathbf{p}^i_l \) and \( \mathbf{w}^i_l \) are the \( i \)th columns of matrices \( \mathbf{P}_l \) and \( \mathbf{W}_l \), respectively, and \( \mathbf{v}^i_l \) is the noise. We can determine \( \mathbf{w}^i_l \) by solving a \( L_1 \)-regularised least squares problem:

\[
\min_{\mathbf{w}^i_l} \left\| \mathbf{G}_l \mathbf{w}^i_l - \mathbf{p}^i_l \right\|^2 + \lambda_S \left\| \mathbf{w}^i_l \right\|_1, \tag{7}
\]

where \( \left\| \mathbf{a} \right\|_1 = \sum_{j=1}^{2n} |a_{ij}| \) is the \( L_1 \)-norm of \( \mathbf{a} \) and \( \lambda_S \) is a regularisation factor. This \( L_1 \)-regularised least squares problem (Eq. (7)) can be reformulated to a convex quadratic program [22] with linear inequality constraints:

\[
\min_{\mathbf{w}^i_l} \left\| \mathbf{G}_l \mathbf{w}^i_l - \mathbf{p}^i_l \right\|^2 + \lambda_S \sum_{j=1}^{2n} u_j \\
\text{subject to } -u_j \leq w_{ij} \leq u_j, \quad j = 1, \ldots, 2n, \tag{8}
\]

where \( u \in \mathbb{R}^{2n} \).

For the experiments presented in this paper, we solve this convex quadratic program (Eq. (8)) by a specialised interior-point method [22] that uses the preconditioned conjugate gradients algorithm to compute the search direction. This results in a sparse solution for \( \mathbf{w}^i_l \). Hence, the transformation matrix \( \mathbf{W}_l = [\mathbf{w}^1_l, \mathbf{w}^2_l, \ldots, \mathbf{w}^{2n}_l] \) is a sparse square matrix of size \( 2n \times 2n \).

3.3. Fully sparse linear regression (FSLR) approach

Assuming \( \mathcal{R} \) to be a linear function of \( \mathbf{x} \) (Eq. (3)), the goal here is to obtain a fully sparse solution for \( \mathbf{W} \), i.e. all non-diagonal elements of matrix \( \mathbf{W} \) should be zero. This approach treats the individual landmark points as independent of each other.

Let \( \mathbf{p}^i_l \) and \( \mathbf{g}^i_l \) be the \( i \)th row of matrix \( \mathbf{P}_l \) and \( \mathbf{G}_l \), respectively, where:

\[ \mathbf{G}_l = [\mathbf{x}_1, \ldots, \mathbf{x}_M] \quad \text{and} \quad \mathbf{P}_l = [\mathbf{x}'_1, \ldots, \mathbf{x}'_M]. \]

Here, we recast Eq. (1) in a linear regression framework to give a fully sparse solution \( \mathbf{W}_l \):

\[ \mathbf{W}_l = \text{diag}(\mathbf{w}^1_l, \mathbf{w}^2_l, \ldots, \mathbf{w}^{2n}_l). \tag{9} \]

Hence, we solve for individual \( \mathbf{w}^i_l \), where \( i = 1, \ldots, 2n \), as a standard \( L_2 \)-regularised least squares problem by:

\[
\min_{\mathbf{w}^i_l} \left\| \mathbf{w}^i_l \right\|^2 + \lambda_{RL} \left\| \mathbf{w}^i_l \right\|_2, \tag{10}
\]

where \( \lambda_{RL} > 0 \) is a regularisation factor used to avoid over-fitting, which gives \( \mathbf{w}^i_l \) as:

\[ \mathbf{w}^i_l = \mathbf{p}^i_l \mathbf{G}_l^T (\mathbf{G}_l \mathbf{G}_l^T + \lambda_{RL} I)^{-1}. \tag{11} \]

3.4. Dense non-linear regression (DNLR) approach

In the dense non-linear regression approach, the goal is to compute \( \mathcal{R} \) such that it is a non-linear function of \( \mathbf{x} \), thus, allowing more accurate predictions, better handling of noisy data and improving overall generalisability of the model.

To this end, we first simplify the task at hand and pose Eq. (1) in a multivariate input regression framework. Let:

\[ \mathbf{G}_N = [\mathbf{x}_1, \ldots, \mathbf{x}_M] \quad \text{and} \quad \mathbf{P}_N = [\mathbf{x}'_1, \ldots, \mathbf{x}'_M]. \]

where \( \mathbf{G}_N, \mathbf{P}_N \in \mathbb{R}^{2n \times M} \). \( \mathbf{g}^i_N \) is the \( i \)th column of matrix \( \mathbf{G}_N \) and \( p^i_N \) is the \( (i,j) \)th element of matrix \( \mathbf{P}_N \). Let the training set:

\[ \mathcal{T}_N = [(\mathbf{g}^i_N, p^i_N)]_{i,j=1}^{M}, \quad i = 1, \ldots, 2n, \tag{12} \]

where \( \mathbf{g}^i_N \in \mathcal{X} \) (the set of multivariate inputs) and \( p^i_N \in \mathcal{Y} \) (the set of outputs/targets). The goal is to learn a non-linear function \( \mathcal{R}_N : \mathcal{X} \rightarrow \mathcal{Y} \), where \( i = 1, \ldots, 2n \).

We use Gaussian process regression (GPR) [30] (see Appendix A for details), which implements automatic relevance determination (ARD) [40] to compute \( \mathcal{R}_N \). Hence, the result is a collection of regressors \( \mathbf{W}_N = [\mathcal{R}_N^1, \mathcal{R}_N^2, \ldots, \mathcal{R}_N^{2n}] \) where \( \mathcal{R}_N^i \) represents the regression model learnt over a particular training set \( \mathcal{T}_N^i \) via GPR.

3.5. Fully sparse non-linear regression (FSNLR) approach

Similar to the FSLR approach (Section 3.3), the goal is to obtain a fully sparse solution for \( \mathbf{W} \). However, here, we assume \( \mathcal{R} \) to be a non-linear function of \( \mathbf{x} \).

To this end, we first simplify the task at hand and pose Eq. (1) in a sparse framework. Let \( \mathbf{g}^i_N \) and \( p^i_N \) be the \( (i,j) \)th elements of matrices \( \mathbf{G}_N \) and \( \mathbf{P}_N \), respectively, where:

\[ \mathbf{G}_N = [\mathbf{x}_1, \ldots, \mathbf{x}_M] \quad \text{and} \quad \mathbf{P}_N = [\mathbf{x}'_1, \ldots, \mathbf{x}'_M]. \]

Let the training set:

\[ \mathcal{T}_N = [(\mathbf{g}^i_N, p^i_N)]_{i,j=1}^{M}, \quad i = 1, \ldots, 2n, \tag{13} \]

Fig. 3. (a) Facial feature clusters: clusters are represented by differently coloured landmark points. (b) Extracting normalisation and point vectors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
where \( g^{FN} \in \mathcal{X} \) (the set of inputs) and \( p^{FN} \in \mathcal{Y} \) (the set of outputs/targets). The goal is to learn a non-linear function \( R_i^{FN} : \mathcal{X} \rightarrow \mathcal{Y} \), where \( i = 1, \ldots, 2n \).

We again use GPR [30] to compute \( R_i^{FN} \) and the result is a collection of regressors \( \mathbf{W}_m^{FN} \):
\[
\mathbf{W}_m^{FN} = \text{diag}(\mathbf{R}_1^{FN}, \mathbf{R}_2^{FN}, \ldots, \mathbf{R}_{2n}^{FN}),
\]
where \( \mathbf{R}_i^{FN} \) represents the regression model learnt over a particular training set \( T_i^{FN} \) via GPR.

4. Methodology for automatic face annotation and model building

Fig. 1 shows a schematic of the proposed method, which only requires annotated frontal face images as input for automatic annotation and model building, thus involving minimal manual annotation. The relevant feature vector extraction procedure is described in Section 4.1. Section 4.2 gives a detailed account of the synthetic image generator that generates the synthetic face images at different poses by the various regression strategies discussed in Section 3.

4.1. Extraction of feature vectors

Given the annotation for a frontal (or non-frontal) image, we extract two vectors: a normalisation vector and a point vector (Fig. 3).

- **Normalisation vector** \( \mathbf{N} \) is a 1D vector containing information about the distances used to normalise the feature vectors with respect to the varying shape and size of faces of different people in the dataset. The horizontal normalisation distance \( N_h \) is the horizontal distance between the outer eye corners. The vertical normalisation distance \( N_v \) is the vertical distance between the outer eye corners and the nose tip (which also acts as the reference point in the normalised frame):
\[
\mathbf{N} = [N_h, N_v]^T.
\]

- **Point vector** \( \mathbf{P} \) is a 1D vector containing information about the location of each of the \( n \) landmark points in the normalised frame:
\[
\mathbf{P} = [px_1, py_1, \ldots, px_n, py_n]^T.
\]

4.2. Synthetic image generator

**Algorithm 1** (Learn the pattern of manual annotation for different poses).

**Require:** \( m \) frontal and corresponding non-frontal annotated images.

1. Extract \( \mathbf{N} \) from frontal and \( \mathbf{N}' \) from non-frontal image
\[
\mathbf{N} = [N_h, N_v]^T, \quad \mathbf{N}' = [N_h', N_v']^T
\]
2. Extract \( \mathbf{P} \) from frontal and \( \mathbf{P}' \) from non-frontal image
\[
\mathbf{P} = [px_1, py_1, \ldots, px_n, py_n]^T, \quad \mathbf{P}' = [px_1', py_1', \ldots, px_n', py_n']^T
\]
3. We have \( m \times 2 \) pairs of normalisation and point vectors, with each pair representing a frontal image and its corresponding non-frontal image
\[
T_N = \{ (\mathbf{N}_i, \mathbf{N}'_i) | i = 1, 2, \ldots, m \}, \quad T_P = \{ (\mathbf{P}_i, \mathbf{P}'_i) | i = 1, 2, \ldots, m \}
\]
4. Use regression (Section 3) to learn the transformation matrices (or a collection of regressors) \( \mathbf{W}_m^{N} \) and \( \mathbf{W}_m^{P} \) for predicting the normalisation and point vectors, respectively.

In the training phase, the synthetic image generator uses a regression based approach (Section 3) to learn the correspondence between the landmark points of frontal view images and non-frontal view images exhibiting arbitrary facial expressions. Once this learner has been trained, it can be used to predict the spatial arrangement of the landmark points for any other (unseen) face at arbitrary pose and expression. The training procedure is explained in Algorithm 1.

In the testing phase, given an annotated frontal face image, a synthetic image can be generated by predicting the new landmark locations via transformation matrices learnt in the training phase and warping the texture from the frontal image via **piecewise affine warping** (PAW). PAW, commonly used in AAM methods [14,24], is a spatial transformation function for which the reference frame is divided into a set of non-overlapping regions such that all locations within each region are warped using the same affine transformation. These regions are generally defined by some type of triangulation of a point set, such as the Delaunay triangulation [12], defined in the reference frame. The result is that locations in the reference frame are warped to locations in the destination frame with the same barycentric coordinates, with respect to its encompassing triangle. The step-by-step procedure is shown in Algorithm 2. As an example, Fig. 4 shows
the steps for generating a synthetic image at pose 45° left from a sample frontal image via the fully sparse non-linear regression approach (Section 3.5).

**Algorithm 2 (Generating a synthetic image).**

**Require:** Annotated frontal face image $l$.
1. Extract $\mathbf{N}$ and $\mathbf{P}$ from $l$ (Section 4.1).
2. Use $\mathbf{W}_N$ and $\mathbf{W}_P$ (Algorithm 1) to predict $\mathbf{N}$ and $\mathbf{P}$.
3. $\mathbf{P}$ contains the new location of each of the $n$ landmark points w.r.t. the origin in the normalised frame. Arrange the landmark points in the normalised frame.
4. Generated landmark points in the normalised frame are denormalised using $\mathbf{N}$.
5. for $i = 1$ to $n$ do
   6. $px_{\text{real}} = px_i \cdot (N_{x_i})$ where $\cdot$ denotes multiplication
   7. $py_{\text{real}} = py_i \cdot (N_{y_i})$
   8. Warp the texture from $l$ to the new locations using PAW.

5. Evaluating the quality of generated synthetic images

For experimental validation of the proposed framework, we start by comparing the quality of synthetic images generated by various regression strategies discussed in Section 3.

We first conducted experiments on the CMU PIE database [35]. Overall, 770 images across 50 subjects (35 male and 15 female), were manually annotated with 69 landmark points each. There were 100 images from each of the Cameras C27 (frontal pose), C09 (30° down), C07 (30° up), C29 (22.5° left), C11 (45° left), C37 (45° right) and 170 images from Camera C05 (22.5° right) exhibiting various facial expressions. Six different learners were trained by each of the five regression strategies discussed in Section 3, i.e. one for each predicting the new landmark locations for poses 22.5° left, 45° left, 22.5° right, 45° right, 30° up and 30° down using any of the regression strategies. Now, given an annotated frontal image of an unseen subject with arbitrary facial expression, the synthetic images can be generated using any of the regression strategies.

A leave-one-person-out cross-validation scheme was adopted throughout the experiments, i.e. data from 49 subjects was used for training and data from the remaining 1 subject for testing. For example, let subject 1 be the test case. The data from subject 2 to subject 50 was used for training the regressors. Hence for the regressors, subject 1 is completely unseen and the frontal image of test subject 1 is only used as the input in the test phase.

**Fig. 5** shows the synthetic images for sample subjects generated from each of the regression strategies, i.e. one example of one good synthesis, one bad synthesis and one complex expression synthesis for each regression strategy. It should be noted here that for the CMU PIE database, the subjects were asked to provide a neutral expression, to smile, to blink and to talk[35]. However, this video is only provided for cameras C27 (frontal), C22 (3/4 profile—not used for experiments in this paper) and C05. Thus, the learners for poses 30° up, 30° down, 22.5° left, 45° left and 45° right were trained only on images exhibiting neutral and smiling expressions, as they lacked the training data for the other expressions and arbitrary lip movements (talking).

To evaluate the quality of synthetic images generated by each of the regression strategies, the reconstruction error was computed as the RMSE measure between the location of landmark points in the synthetic image and the location of the manually annotated landmark points (ground truth) in the original image. However, a comparison between the quality of synthetic images generated by the different regression strategies is always subjective, mainly because the question “How good is this synthetic image?” has an aesthetic element to it.

The arrangement of landmark points is equally as important as the individual location of each landmark point. Consider a scenario (Fig. 6), in which the synthesis by the fully sparse linear regression approach of certain features (for example, upper half of the face) is accurate according to the error measure, but it is unable to synthesise the mouth and lower jaw region well. On the other hand, using the fully sparse non-linear regression approach, we get a consistently good synthesis where each feature is generated well, but the overall reconstruction error is still more than that of FSLR. This is also due to the fact that landmark points along the edges (i.e. the boundaries) of certain facial features (nose and outer boundary of the face, in particular) lack distinct features and are difficult to annotate consistently. Thus, numerical analysis of these points can be error prone. Hence, the aesthetic element of synthetic images, when judging its quality, cannot be ignored, but nonetheless the numerical analysis based on the reconstruction error criterion can be used to infer the trend effectively and conclude which regression strategy can best model the changes in the shape of the face under varying pose.

**Fig. 7** shows the reconstruction error for the entire CMU PIE dataset obtained from each of the regression strategies and these results have been summarised in **Table 1**. From these results, we can clearly infer that fully sparse non-linear regression outperforms all other regression approaches followed by sparse linear regression with dense non-linear regression, dense linear regression and fully sparse linear regression performing worse.

For the dense linear regression (DRL) approach, the issue is the complexity of Eq. (1). Considering the variation in shape of the face across different people and various expressions, the function $\mathcal{R} : \mathbf{x} \rightarrow \mathbf{x}$ is clearly non-linear in $\mathbf{x}$. In this approach, we try to model it in a linear framework, which is too restrictive and unable to model this complex mapping. This is evident from the error distribution graph of DLR (Fig. 7) for all poses.

Considering that the linear framework in the dense formulation is too restrictive, we model $\mathcal{R}$ as a non-linear function of $\mathbf{x}$ in the dense non-linear regression (DNLR) approach. As can be seen from the error distribution graph of DNLR (Fig. 7), there is no consistent improvement in the synthetic image generation accuracy over DLR. The non-linear framework can provides better flexibility to model $\mathcal{R}$ and can potentially provide more accurate predictions than their linear counterpart by virtue of their learnt model being more complex than the linear model, in general. But this invariably requires a more complicated training procedure and in practice, with limited training data at hand, the linear models can be expected to extrapolate (i.e. predict unseen data) better than their non-linear counterpart owing to their simpler predictive domain [19]. Therefore, we observe only a marginal performance improvement in this case. Notice that the mean reconstruction error of 3.8712 pixels obtained via DNLR is slightly better than the mean reconstruction error of 3.9202 pixels obtained via DLR.

To relax the assumption of total dependence between all the landmark points, the sparse representation framework is used in the sparse linear regression (SLR) approach, which drastically improves the image synthesis accuracy throughout the test dataset (Fig. 7). In SLR, the $L_1$-regularised least square problem (Eq. (7)) will always have a solution (which might not be unique). It can be efficiently solved as a convex optimisation program, which gives a sparse solution for $\mathcal{R}$ [13], i.e. the regression coefficients for irrelevant input features are set to 0. This reduces the model complexity and,
hence, avoids over-fitting. Moreover, in SLR, we use a specialised interior-point method, proposed in [22], to solve the convex quadratic program (Eq. (8)), which has been shown capable of solving complex sparse problems efficiently and accurately.

Contrary to the assumption of total dependence between all the landmark points in the dense representation, the fully sparse representation assumes total independence between all the landmark points and offers some interesting properties. From Fig. 7, we can clearly see that the fully sparse linear regression image synthesis accuracy is consistently the lowest except for the poses 30° up and down, which are relatively simpler synthesis problems as compared to the 22.5° and 45° left and right poses, where out-of-plane rotation causes more drastic changes in the shape of the face and occlusion. FSLR is able to capture certain subtleties of complex expressions better than its non-fully sparse linear counterparts (DLR and SLR). Notice the synthesis of the eye region for the complex expression (third row) in Fig. 5 is more accurate via FSLR than via DLR and SLR. This indicates that although the assumption of total independence between all the landmark points has a desirable ability to capture the subtle variations produced by a complex expression, the linear regression framework is not flexible enough and overall quality of synthetic images degrades as the pose variation increases.

On the other hand, the fully sparse non-linear regression approach stands out and outperforms all other regression approaches (Fig. 7). The fully sparse representation helps in simplifying the problem, avoiding over-fitting. When coupled with the flexibility of a non-linear regression framework to model \( R \), FSNLR proves to be the best approach for the task of learning the pattern of manual annotation across poses. Moreover, GPR [30] used in FSNLR offers a principled way of model fitting by maximising the log marginal likelihood, \( \log p(y|x) \):

\[
-\frac{1}{2} y^T (K + \sigma_n^2 I)^{-1} y - \frac{1}{2} \log |K + \sigma_n^2 I| - \frac{n}{2} \log 2 \pi,
\]

(17)

with respect to the hyperparameters, \( \Theta = \{l, \sigma_l^2, \sigma_n^2\} \) (see Appendix A). This model fitting procedure, which respects Occam's razor principle (choosing the simplest model, which best explains the
observed data), allows us to have different hyperparameters within $\mathbb{R}$ and, hence, leads to an efficient training procedure. Apart from this, by assuming an independent and normally distributed noise terms $\varepsilon_i \sim \mathcal{N}(0, \sigma^2_i)$ while training, GPR has a natural ability to deal with noisy data, i.e. manual annotations in this case. Refer to the supplementary videos,\footnote{Available at http://users.rise.anu.edu.au/~aasthana/PR/AAsthana_SuppVideos.zip.} to view the entire talking sequences generated at different poses via FSNLR from the frontal talking sequences for sample speakers. The results clearly show the ability of FSNLR to effectively model the changes in the shape of the face under varying pose and at the same time capturing the subtleties of different unseen facial expressions, thus, ensuring a high quality of the generated synthetic images.

6. Deformable model building via synthetic images

In this section, we demonstrate the utility of the generated synthetic images by building both 2D and 3D deformable models. We start by generating the synthetic images from frontal images via the various regression approaches discussed in Section 3. Since we have shown that the fully sparse non-linear regression approach outperforms all other methods (Section 5) and consistently generates high quality synthetic images, we will use these images in the following.

Given the annotated frontal images, we generate six synthetic images from each of the frontal images (Fig. 8(a)), so that we have sufficient synthetic data to train the deformable models. These synthetic images are then used to generate a 2D deformable model, e.g. active appearance model (AAM) [14], which is expected to cover the maximum range spanned by the input images, which in our experiments are the frontal and synthetic images spanning from 45° left to 45° right and 30° up to 30° down.

To generate a 3D deformable model from the synthetic images, we apply a non-rigid structure-from-motion approach [38] on the synthetic images. An example of a 3D deformable model generated from the synthetic images is shown in Fig. 8(b). Note that in this approach, we are essentially trying to learn a 3D structure (the shape) of the face from a set of 2D images, therefore, the model is bound to have some limitations [34], especially when trying to learn the shape at extreme poses. Despite its limitations, the generated 3D deformable model is capable of giving a comparable tracking/fitting performance to its 2D counterpart while being extremely fast (see Section 7).

7. Automatic annotation of unseen images

The goal here is to fit the 2D or 3D deformable model, trained on the synthetic images generated via FSNLR, to unseen images having any arbitrary pose and expression and, hence, to annotate them automatically. The problem here is that the generated synthetic images have no information about the area outside the convex hull of its generated shape. This makes the deformable model trained on these images highly susceptible to a changing background, poor initialisation and ill-defined borders [36]. In this section, we will discuss both 2D and 3D deformable model fitting methods used for annotating unseen images and will show that the above problems can be tackled efficiently. To start off, we use a face detector\footnote{For the experiments presented in this paper, we use faceAPI (Seeing Machines) to detect the location of the face. However, we wish to highlight that any suitably accurate face detector will suffice and that our approach is not dependent on a particular face detector.} to initialise both 2D and 3D deformable model fitting methods. This helps in avoiding the problem of poor initialisation to some extent.

7.1. 2D deformable model fitting method

Once we have trained a 2D deformable model, e.g. AAM, on the synthetic images generated via FSNLR, we use the simultaneous inverse compositional (SIC) generative fitting method [2] to perform fitting on unseen images. SIC is a very powerful generative fitting method, in which the update model is generated directly from background-free components (i.e. the mean appearance and their modes of variation) and, hence, has no specialisation to any particular background. However, SIC is highly prone to terminate in a local minima when the model initialisation is slightly too far from the optimum, with a large proportion of the image under the current warp estimate consisting of background pixels. To deal with this problem of poor initialisation, we use an iterative-fitting scheme, in which we initialise the model at several different positions and choose the fitting result that gives the minimum residual texture error. This scheme also helps in solving the problem of ill-defined borders. The iterative-fitting scheme is explained in Algorithm 3.

Algorithm 3 (Iterative-fitting scheme).

Notations:
- $E_i(x, y)$ – Residual texture error obtained by AAM fitting when model is initialised at location $(x, y)$
- $s_{x,y}$ – Shape vector obtained by AAM fitting when model is initialised at location $(x, y)$
- $\delta_x \times \delta_y$ – Size of the initialisation search window.

Input: Face Image, $I$, to be annotated.

1. Initialise the fitting procedure at the centre, $(x, y)$, of the bounding box obtained by the face detector.
2. for $i = (-\delta_x/2)$ to $(+\delta_x/2) do$
3. for $j = (-\delta_y/2)$ to $(+\delta_y/2)$ do
4. Compute residual texture error $E_i(x + i, y + j)$
5. if $E_i(x + i, y + j) < \text{MIN}(E_i)$ then
6. $\text{MIN}(E_i) = E_i(x + i, y + j)$
7. $s = s_{x+i,y+j}$

Output: Shape vector, $s$, representing the landmarks for $I$. 
It should be noted here that while fitting the model trained on synthetic images, getting a perfect fit (i.e. residual texture error \( \approx 0 \)) is highly improbable (if not impossible). Realistically, we are looking for the best set of parameters that gives the smallest residual texture error. The nature of the problem requires retaining most of the shape and texture variation while training the AAM. Therefore, for experiments presented in this paper, we retained 99% modes of shape and texture variations while training the AAM. This results in a single AAM being capable of performing accurate fitting across poses and expressions. However, doing so might slow down the convergence speed of SIC and might terminate in a local minimum when fitting images with extreme pose variations. One solution, commonly used by other, to this problem is to use a view-based AAM [11] building approach, which reduces the search space for a single AAM by dividing it among several AAMs. This can help in reducing the amount of shape and texture variation to be handled by a single AAM, thus, reducing the convergence time and also reducing the probability of the fitting procedure to converge to a local minima for extreme poses. This approach can be highly flexible and can assist in improving the fitting performance.

7.2. 3D deformable model fitting method

Once we have trained a 3D deformable model on the synthetic images generated via FSNLR (Section 6), we use the 2D+3D

---

Fig. 7. Reconstruction error for CMU PIE database achieved by methods discussed in Section 3. X in the graph represents a bad reconstruction with reconstruction error > 10. (a) Pose 30 down. (b) Pose 30 up. (c) Pose 22.5 left. (d) Pose 45 left. (e) Pose 22.5 right. (f) Pose 45 right.
Table 1
Mean reconstruction error for CMU PIE database achieved by methods discussed in Section 3.

<table>
<thead>
<tr>
<th>Reconstruction methods</th>
<th>Mean reconstruction error (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense linear regression (DLR)</td>
<td>3.9202</td>
</tr>
<tr>
<td>Dense non-linear regression (DNLR)</td>
<td>3.8712</td>
</tr>
<tr>
<td>Sparse linear regression (SLR)</td>
<td>3.4031</td>
</tr>
<tr>
<td>Fully sparse linear regression (FSNLR)</td>
<td>4.0073</td>
</tr>
<tr>
<td>Fully sparse non-linear regression (FSNLR)</td>
<td>2.8921</td>
</tr>
</tbody>
</table>

Fig. 8. Utility of generated synthetic images in generating high quality deformable models. (a) Generated synthetic images, (b) 3D deformable model.

inverse compositional method (2D + 3D IC) [25] to perform fitting on unseen images. In [25], it has been shown that 2D + 3D IC is highly robust and converges in fewer iterations than comparable 2D model fitting methods. It is also less prone to converge in a local minimum than 2D fitting methods, especially SIC, by virtue of its more natural and efficient 3D shape parametrisation whereas 2D models suffer over-parametrisation resulting in physically unrealisable model instances. Therefore, after initialisation, it is quite straightforward and convenient to use the 2D + 3D IC method for fitting unseen images using a 3D deformable model and annotating them automatically.

7.3. Analysing automatic annotation results

To evaluate the automatic annotation capabilities of the 2D and 3D deformable models trained on synthetic images generated via FSNLR, we start by automatically re-annotating the entire dataset from the CMU PIE database containing 670 images at various poses and expressions, with a face cropped area of 140 × 150 pixels (see Section 5).

We compare the automatic annotation performance of the 2D AAM trained in a person-specific manner on the synthetic images (Section 7.1), referred as synthetic 2D model fitting method, and the 3D AAM trained in a person-specific manner on the synthetic images (Section 7.2), referred as synthetic 3D model fitting method, to that of:

- Ground truth 2D model fitting method: This method uses the SIC fitting method [2] and the iterative-fitting scheme (Algorithm 3). The AAM is trained in a person-specific manner using the manually labelled (ground truth) data of the test subject. This method has been shown to perform exceedingly well under all conditions [16].
- Ground truth 3D model fitting method: This method uses the 2D + 3D IC fitting method [25]. The AAM is trained in a person-specific manner using the manually labelled (ground truth) data of the test subject. Although, not as accurate as SIC, 2D + 3D IC has been shown capable of fitting a 3D model accurately. A direct comparison between SIC and 2D + 3D IC can be found in [34] where the SIC has been shown to outperform 2D + 3D IC.
- Generic AAM fitting method: This method uses the SIC fitting method [16] and the iterative-fitting scheme (Algorithm 3). The Generic AAM is trained on the entire data used to train the regressors that generates the synthetic images for a subject (Section 5) and the frontal images of that subject. For example, let the subject 1 be the test case. For subject 1, the Generic AAM is trained using the entire data from subject 2 to subject 50, that includes frontal and non-frontal images. In addition, the training data of Generic AAM also included the frontal image of subject 1. We include this frontal image of subject 1 to maintain consistency in the experiments because our proposed method requires this frontal annotated image. Therefore, we believe it makes sense to add this to the Generic AAM as well. This generic AAM was then used to fit non-frontal unseen images of subject 1.

We have not made any comparison to the Generic AAM fitting method in 3D because, first, the direct application of SIC in 3D is not possible [25] and second, 2D + 3D IC is an extension of project-out 2D AAM fitting method in 3D which is an approximation of SIC in 2D. This project-out approximation of SIC is not suited for the task of Generic AAM building [16] and therefore, a 3D variant of the project-out method i.e. 2D + 3D IC cannot be used for Generic 3D model fitting.

To measure the accuracy of automatic annotations obtained from all the above methods, the annotation error was computed as RMSE measure between the manual and automatic annotations for every image. It should be noted here that consistently manually annotating the outer contour of the face is highly error prone due to its inconsistent structure and the lack of distinct features. Thus, numerical analysis of these points can be highly error prone. Therefore, we computed the annotation error by excluding the 13 landmark points that represent the outer contour for each face in our 69-point mark up. The cases where the fitting procedure failed to converge or the annotation error obtained is greater than 5 pixels are considered failure to annotate (FTA) cases. Fig. 9 shows the annotation errors for the entire CMU PIE dataset obtained from various methods described above in this section and have been summarised in Table 2. See Fig. 13 for sample annotation results for CMU PIE Db achieved by the synthetic 2D model fitting method.

From the results, we can clearly see that the Ground truth 2D model fitting method and the Ground truth 3D model fitting method show excellent performance with no convergence failure and an annotation error below 3 pixels with very few exceptions. This is hardly surprising. In fact, as mentioned in Section 1.1, this result is one of the main motivating factors and the goal is to generate synthetic images, which can be used to train AAMs that can show comparable results. On the other hand, the Generic AAM fitting method consistently fails to converge and results in large number of FTA cases. Moreover, even after excluding the FTA cases, the mean annotation error obtained from the Generic AAM fitting method is the worst (Table 2). Clearly, it cannot be used to perform automatic annotation accurately.
The synthetic 2D model fitting method and the synthetic 3D model fitting method show encouraging results. The performance enhancement by using the AAMs trained on synthetic images, generated via FSNLR, instead of the Generic AAM Fitting method is clearly evident across poses. More importantly, the annotation error obtained by the synthetic 2D model fitting method remains below 3 pixels with few exceptions and is comparable to the performance of the ground truth 2D and 3D model fitting methods. A slight performance drop can be seen by using the synthetic 3D model fitting method but is still more reliable than the Generic AAM fitting method. The number of FTA cases for the synthetic 3D model fitting method was 15/100 at pose 22.5° left and 4/100 at pose 45° left. This was due to considerable variation in illumination present in those images.

It should be noted here that the reason for the synthetic 3D model fitting method not performing as well as its 2D counterpart is mainly because of the different AAM fitting method used. The synthetic 2D model fitting method uses SIC, which is one of the most powerful generative fitting methods, whereas the synthetic 3D model fitting method used 2D+3D IC, which is a 3D variant of the project-out 2D AAM fitting method [25]. The project-out 2D AAM fitting method is an approximation of SIC that has been shown incapable of handling large texture variations [16]. Therefore, 2D+3D IC is more prone to fail when there is a variation in texture. Moreover, SIC is inherently more accurate than 2D+3D IC [34], which makes the synthetic 2D model fitting method show better annotation accuracy than the synthetic 3D model fitting method. Hence, we cannot directly conclude that the synthetic images, generated via FSNLR, are better suited for training a 2D deformable model than a 3D one.

To further test the robustness of the proposed framework for random expressions, we automatically annotated 600 images (10 subjects with 60 images each) containing the entire 2 s of the talking sequence captured from the camera C05, i.e. at pose 22.5°.
right. Fig. 10 shows the annotation error distribution for the talking sequence obtained from both the synthetic 2D and 3D model fitting methods. Note that ground truth (manual) annotations for these sequences were not available. Therefore, we treat the tracking results obtained by an AAM trained on the original images on these sequences as the ground truth and compare the automatic annotation results against these in Fig. 10.

However, it should be noted that better 2D fitting accuracy comes at the cost of a loss in the fitting speed. Our experiments were conducted on a computer with 2.0 GHz CPU and 2 GB RAM with C++ implementations. The 2D deformable model fitting methods that uses SIC and the iterative fitting scheme takes 10–20 s on an average (depending on how far the initialisation is from the optimum parameters) to annotate a single image. The performance of the 3D deformable model fitting methods that uses 2D+3D IC is real-time, offering fast and accurate annotation results. On the other hand, the Generic AAM approach is the slowest and computationally most expensive because the dimensionality of the shape and texture component of the Generic AAM is much higher than the dimensionality of the 2D and 3D deformable models trained on the synthetic images that are person specific in nature.

In this section, we have shown that both 2D and 3D models trained on the synthetic images generated via FSNLR can be effectively used to annotate unseen images of a face at any random pose and expression. The synthetic 2D model fitting method gives highly accurate annotations by virtue of SIC fitting method but is slow. In contrast, synthetic 3D model fitting methods is real-time but not as accurate as its 2D counterpart because of the short-comings of the 2D+3D IC fitting method used in it.

8. Automatic annotation of unseen databases

To evaluate the generalisability of the proposed framework, we tested the regression model trained entirely on the images of the CMU PIE database. Fig. 11 shows the generated synthetic images from unseen databases via FSNLR trained entirely on CMU PIE dataset. (a) Multi-PIE database. (b) Face pointing database. (c) Feret database 'b' series. (d) Color Feret database.

Fig. 10. Annotation error distribution for talking sequence from Camera C05 (22.5° right) of CMU PIE database.

Fig. 12. Annotation error distribution for three unseen databases, i.e. Multi-PIE, Color FERET and face pointing database.
from the CMU PIE database via FSRLR, for automatically annotating images, with arbitrary poses and expressions, from four different unseen databases: Color FERET database [28], Gray FERET database ‘b’ series [28], Face Pointing database [15] and the challenging CMU Multi-PIE database [17]. We use synthetic 2D model fitting methods for annotating these unseen databases.

Our Color FERET test set consisted of 150 random images across 50 subjects with poses varying from 45° left to 45° right. Our Gray FERET ‘b’ series test set consist of 300 images across 50 subjects with poses 40°, 25°, 15° left and right (i.e. subsets bc, bd, be, bf, bg and bh). Our face pointing test set consisted of 100 randomly chosen images across 15 subjects with poses varying from 45° left to 45° right and 30° up to 30° down. Our Multi-PIE test set consisted of 300 images across 50 subjects with poses 08.0 (45° left), 13.0 (30 left), 14.0 (15° left), 05.0 (15° right) 04.1 (30° right) and 19.0 (45° Right).

Fig. 11 shows some synthetic images generated for sample subjects from the four unseen databases. Fig. 12 shows the annotation error distribution obtained via synthetic 2D model fitting methods from each of the test sets. It should be noted here that we successfully annotated all 300 images of the Gray FERET ‘b’ series test set, but due to the ground truth for these images not being available, we are unable to show the error distribution for it in Fig. 12. See Fig. 13 for sample annotation results achieved across all databases by the synthetic 2D model fitting method.

From Fig. 12, we can clearly see that the proposed framework exhibits excellent generalisability by successfully annotating images from all four unseen databases with consistent accuracy. For the Color FERET and Face Pointing databases, the annotation error, for the majority of images, varied from 1–3 pixels with no FTA cases. For the challenging Multi-PIE database, the annotation error, for the majority of images, varied from 1.5–4 pixels.

9. Conclusion

A regression-based framework for automatic face annotation and deformable model building (both 2D and 3D) has been proposed. We have investigated several regression strategies to generate synthetic images at any random pose with any arbitrary expression from annotated frontal images and have shown that the fully sparse non-linear regression (FSNLR) approach outperforms all other approaches.

FSNLRLR is able to generate high quality synthetic images consistently and is able to capture the subtleties of different facial expressions by virtue of its non-linear predictor and sparse framework. The synthetic images are shown capable of training both 2D and 3D deformable models that are used to perform fitting on unseen images and, hence, annotating them automatically. The framework exhibits excellent generalisability and is demonstrated by automatically annotating images from four different unseen databases.

Acknowledgement

The authors would like to thank Jason Saragih (CSIRO) for the use of the DeMoLib software, and Tom Gedeon (ANU) and Novi Quadrianto (NICTA/ANU) for their insightful suggestions and comments.

Appendix A. Gaussian process regression (GPR)

Given m observed data points \( \mathcal{D} = \{(x_i,y_i)\}_{i=1}^m \), where \( y_i \in \mathcal{Y} \) (the set of outputs/targets) and \( x_i \in \mathcal{X} \) (the set of inputs), the goal of learning is to infer a function \( f: \mathcal{X} \rightarrow \mathcal{Y} \). In a regression setting, \( y_i \in \mathbb{R} \).

In this appendix, we briefly review the Gaussian process regression (GPR) [30] techniques used for the experiments in this paper.

GPR has gained increased popularity in statistical machine learning as it offers a principled non-parametric Bayesian framework for inference, model fitting and model selection [5]. In this framework, we observe a noisy query point \( y_i = f(x_i) + \epsilon_i \) at input location \( x_i \) and the noise term is assumed to be independent and normally distributed, \( \epsilon_i \sim \mathcal{N}(0,\sigma_n^2) \). Placing a Gaussian prior over functions will lead to a Gaussian predictive distribution:

\[
y^*|x^*,\mathcal{D} \sim \mathcal{N}(\mu,\sigma^2),
\]

with

\[
\mu = K^T[K + \sigma_n^2 I]^{-1}y,
\]

\[
\sigma^2 = k(x^*,x^*) + \sigma_n^2 - k(x^*)^T[K + \sigma_n^2 I]^{-1}k(x^*),
\]

for a noisy query point \( x^* \). In these equations, we have \( K \in \mathbb{R}^{m \times m} \), \( K_{ij} = k(x_i,x_j) \) and \( k^* \in \mathbb{R}^{1 \times m}, k^*_i = k(x^*,x_i) \). Here, \( k \) denotes a covariance matrix.
function, which encodes our assumptions about the function we wish to learn.

In a single input regression framework, we employ a squared exponential covariance function of form:

$$k(x_p, x_q) = \sigma_f^2 \exp \left( -\frac{1}{2} \frac{(x_p - x_q)^2}{\ell^2} \right) + \sigma_n^2,$$

which has the characteristic length scale $\ell$, the signal variance $\sigma_f^2$ and the noise variance $\sigma_n^2$ as free/ hyperparameter parameters, $\Theta = (\ell, \sigma_f^2, \sigma_n^2)$. In contrast, for a multivariate input regression framework, we use a squared exponential covariance function of form:

$$k(x_p, x_q) = \sigma_f^2 \exp \left( -\frac{1}{2} \sum_{i=1}^{d} (x_{p_i} - x_{q_i})^2 \right) + \sigma_n^2 \delta_{pq},$$

with $\Theta = (\{M_2\}, \sigma_f^2, \sigma_n^2)$ that implements automatic relevance determination (ARD) [40], where $\{M_2\}$ denotes the parameters in the symmetric matrix $M_2 = \text{diag}(\ell^{-2})$. See [30] for details.

From this Gaussian predictive distribution, we are interested to make a point prediction $y_{\text{optimal}}$ by minimizing the expected loss or risk as:

$$y_{\text{optimal}}(x^*) = \arg\min_{y_{\text{guess}}} \int L(y^*, y_{\text{guess}}) p(y^* | x^*, \Theta) dy^*$$

where $L(\cdot, \cdot)$ is the loss function. For squared loss functions (or any symmetric loss function), the optimal point prediction at query point $x^*$ is:

$$y_{\text{optimal}}(x^*) = E[y^* | x^*, \Theta] = \mu.$$

Appendix B. Supplementary material

Supplementary data associated with this article can be found in the online version of 10.1016/j.patcog.2011.03.014.

References


apolis (MN), USA, June 2007.


Akshay Asthana is a PhD student at the Research School of Engineering, Australian National University, Canberra, Australia. He was awarded the Australian Leadership Award Scholarship (ALAS) by the Australian government’s overseas aid program (AusAID) in 2008. He graduated from the Jaypee Institute of Information Technology University (JIITU), Noida, India with a degree of Bachelor of Technology (B.Tech) in Computer Science and Engineering in 2007. His research interests are in pattern recognition, computer vision and human-computer interaction.

Simon Lucey is Senior Research Scientist and Research Project Leader at the CSIRO ICT Centre Computer Vision (CI2CV) laboratory, Brisbane, Australia. He is also an Adjunct Associate Professor at the University of Sydney. He was awarded an ARC Future Fellowship by the Australian Research Council in 2009. He received his Ph.D. in Computer Science from the Queensland University of Technology, Brisbane, Australia in 2002. His research interests are in pattern recognition, computer vision and machine learning with specific interests in their application to biometric learning and space-time pattern recognition.

Roland Goecke is Assistant Professor of Software Engineering and Head of the Vision and Sensing Group at the Faculty of Information Sciences and Engineering, University of Canberra. He is also an Adjunct Faculty member at the Information and Human Centred Computing Group, Research School of Computer Science, Australian National University, Canberra. He received his Masters degree in Computer Science from the University of Rostock, Germany, in 1998 and his Ph.D. in Computer Science from the Australian National University, Canberra, Australia, in 2004. His research interests are in affective computing, pattern recognition, computer vision, human-computer interaction and multimodal signal processing.