Pose-Robust 3D Facial Landmark Estimation from a Single 2D Image

Brandon M. Smith
http://www.cs.wisc.edu/~bmsmith
Charles R. Dyer
http://www.cs.wisc.edu/~dyer

Department of Computer Sciences
University of Wisconsin-Madison
Madison, WI USA

Abstract

An algorithm is presented that estimates 3D facial landmark coordinates and occlusion state from a single 2D image. Unlike previous approaches, we divide the 3D cascaded shape regression problem into a set of *viewpoint domains*, which helps avoid problems in the optimization, such as local minima at test time, and averaging conflicting gradient directions in the domain maps during training. These problems are especially important to address in the 3D case, where a wider range of head poses is expected. Parametric shape models are used and are shown to have several desirable qualities compared to the recent trend of modeling shape nonparametrically. Results show quantitatively that our approach is significantly more accurate than recent work.

1 Introduction

Despite much research interest in facial landmark estimation in recent years, relatively little work has been done to handle the full range of head poses encountered in the real world (*e.g.*, beyond $\pm 45^\circ$ rotation). Large head pose variation is challenging for several reasons:

1. The 2D shapes of profile faces and frontal faces are significantly different;
2. Many landmarks become self-occluded on profile faces; and
3. Even when visible, landmark appearance changes significantly with head pose.

As a result, the large majority of face alignment algorithms are limited to near fronto-parallel faces, and break down on profile faces.

We propose an approach to face alignment that can handle 180 degrees of head rotation. The foundation of our approach is cascaded shape regression (CSR), which has emerged as the leading strategy (see, *e.g.*, [18, 23] and references therein). CSR methods are computationally efficient and the core idea is elegant. They are purely discriminative, which means they can capitalize on large and diverse training datasets to perform robust face alignment in the wild. To better handle a wide range of head poses, we extend the 2D CSR approach to 3D. That is, instead of fitting a 2D face model to single 2D images, we fit a 3D face model to single 2D images (3D-to-2D). Intuitively, as the range of head poses increases, the 3D geometry of the face becomes increasingly important in explaining its 2D image projection.

Pages 18.1-18.12
DOI: https://dx.doi.org/10.5244/C.30.18

Recent facial landmark estimation methods, including 3D-to-2D approaches [16, 25], employ *local* optimization algorithms at each cascade level, which can fail on face collections with large head pose variation. It is unlikely that a single cascade of generic domain maps (from input features to output landmark updates) will consistently find the true solution. We therefore partition the shape regression problem into a set of simpler *viewpoint domains*, and learn a separate cascade of regressors for each. Each viewpoint domain corresponds to an automatically-learned range of camera viewpoints/head poses.

Self-occlusion of landmarks is a significant problem for non-frontal faces. The first challenge is estimating the occlusion state of each landmark, which is difficult in general: occluding objects (*e.g.*, sunglasses, hair, scarves) can have arbitrary shape and appearance, and can be easily confused with image noise, shadows, *etc*. Provided that occlusion state can be estimated, the second challenge is to reduce the impact of spurious features associated with occluded landmarks. One strategy is to employ an ensemble of regressors that operate on non-occluded subregions of the face [4, 33]. Unfortunately, with N landmarks, there are $N!$ possible combinations of occluded/visible landmark states in general. The regions must either be small (*e.g.*, $1/9$-th of the face [4]), which leads to weak regressors, or many different overlapping regressors must be employed [33] to handle different occlusion combinations. For these reasons, we focus on the more tractable problem of detecting only self-occlusion of landmarks (*i.e.*, when one part of the face occludes another), which is well-correlated with head pose.

A recent trend (*e.g.*, [4, 6, 27, 29]) has been to model face shape nonparametrically, and directly update landmark coordinates. However, parametric point distribution models (PDMs) [11] have several desirable qualities:

1. There are fewer parameters to optimize, which results in a smaller (and faster to apply) set of regression coefficients.

2. They generalize well to unfamiliar faces.

3. All landmarks are optimized simultaneously.

In fact, we show empirically that there are no significant differences in accuracy between parametric and nonparametric shape models when used in otherwise identical systems.

CSR methods commonly use off-the-shelf feature mapping functions (*e.g.*, SIFT [20]) to produce features from the image. Instead, we *learn* the feature mapping functions. Specifically, we use regression random forests [3, 12] to induce local binary features that predict ideal PDM parameter updates. To summarize, we make the following contributions:

1. *Viewpoint domain cascaded shape regression:* We automatically partition the regression problem by camera viewpoint/head pose, which results in better performance.

2. *Faces with arbitrary head pose:* The proposed algorithm estimates 3D landmark coordinates and is robust to extreme head pose, including profile faces.

3. *Nonparametric vs. parametric:* We show empirically that there is virtually no difference in accuracy between parametric and nonparametric shape models used in otherwise identical systems.

4. *High accuracy on challenging faces:* The proposed approach produces results with favorable accuracy compared to the state-of-the-art on a wide range of head poses.

2 Related Work

Face alignment has a rich history in computer vision. We forgo an extensive overview here due to space limitations and focus on the most relevant work. Shape regression approaches have recently come to dominate the face alignment landscape (*e.g.*, [4, 6, 18, 19, 23, 26, 29]). Due to the complex relationships between image appearance and face shape, finding the true face shape in one step is difficult. The most popular strategy is to split the regression problem into multiple iterations, *i.e.*, a cascaded shape regressor (CSR).

Although conventional CSRs split the problem into stages, each regressor still attempts to fit the entire dataset. This is problematic for large pose variation because the objective function includes many conflicting gradient directions. Xiong and De la Torre [30] proposed the Global Supervised Descent Method (GSDM) to address this problem for 2D face alignment. GSDM splits the objective function into regions of similar gradient directions and constructs a separate CSR for each region. As presented, GSDM is limited to video because it uses the alignment result from one frame to select which CSR to use for the subsequent frame. GSDM reverts to conventional CSR (*e.g.*, SDM [29]) on single images. At test time our algorithm adaptively chooses which CSR to apply *based on the current test image alone*. Thus, we extend the theoretical benefits of GSDM to single image 3D-to-2D face alignment.

Several CSR-based methods *learn* feature mapping functions (*e.g.*, using ensembles of regression trees [18, 19, 23, 25]), which are task-specific and can be extremely efficient at test time [18, 23]. We use ensembles of regression trees in our approach to induce pixel-difference features correlated with our regression targets.

A recent trend among CSR methods, including those that learn feature mapping functions, is to model shape nonparametrically and update the landmark coordinates directly (*e.g.*, [6, 18, 23, 25]). Instead, we employ PCA-based shape models [11] to parameterize the 3D geometry of faces, and our system learns feature mapping functions that are optimized for these parametric regression targets. We show empirically in Section 4 that *there are no significant differences in accuracy between parametric and nonparametric shape models.*

Work that addresses significant head rotation includes view-based models [9, 35], nonlinear statistical models (*e.g.*, kernel methods [24] or mixture models [10, 17, 34]), and 3D shape models [2, 5, 7, 14, 16, 22, 25, 32]. Nonlinear statistical models tend to be too slow for real-time applications. View-based methods employ a separate model for each viewpoint mode. Traditionally, the modes are specified as part of the algorithm design (*e.g.*, every 15° yaw [35]), and problems can arise at midpoints between models. In contrast, we propose an automatic way to select viewpoint modes for training, and we train the models with overlapping subsets of examples so that midpoint cases are insensitive to model assignment. Most 3D algorithms focus on either near fronto-parallel faces [5, 7, 32], and/or employ local optimization techniques to minimize a single objective function across all head poses [2, 5, 7, 16, 22, 25], which is problematic for reasons described in [30]. Another approach is *dense* 3D face alignment and reconstruction (*e.g.*, [15] and its predecessors), although results suggest that test faces must still be near-frontal.

Researchers have recently considered the problem of locating landmarks on faces with significant head rotation within the CSR framework. Wu and Ji [27] proposed an approach for estimating 2D landmarks and their occlusion state; inference of landmark visibility with their model is nontrivial, and they used Monte Carlo approximation, which is relatively slow. Our work is most similar to Tulyakov and Sebe [25] and Jourabloo and Liu [16], who proposed approaches for estimating 3D landmark coordinates from single 2D images. However, they both employ single generic CSRs across all faces, which can get stuck in local minima

or stall in regions of conflicting gradient directions. This is especially problematic for diverse collections of frontal and non-frontal faces, which they target. We address this problem by partitioning the regression problem into *viewpoint domains*, which are each simpler and less prone to optimization problems. As a result, our approach produces results with favorable accuracy, especially on non-frontal faces.

3 Approach

We begin with an overview of conventional CSR, which motivates our approach. Let $\mathbf{s}_i$ be the ground truth set of landmarks for training example i, and $\hat{\mathbf{s}}_i$ be an estimate of $\mathbf{s}_i$. Conventional CSRs learn a sequence of $t = 1,\ldots,T$ descent maps $\{\mathbf{Q}^t\}_{t=1}^T$ that minimize the following:

$$\hat{\mathbf{Q}}^t = \underset{\mathbf{Q}^t}{\operatorname{argmin}} \sum_i \|\Delta\mathbf{s}_i^t - \mathbf{Q}^t\mathbf{d}^t(I_i,\hat{\mathbf{s}}_i^{t-1})\|_2^2 \qquad \Delta\mathbf{s}_i^t = \mathbf{s}_i - \hat{\mathbf{s}}_i^{t-1}, \tag{1}$$

where $\mathbf{d}(I,\mathbf{s})$ is a feature descriptor that captures the local appearance in image I relative to shape $\mathbf{s}$. At test time, starting with a mean face shape initialization $\hat{\mathbf{s}}^0 = \mu$, $\hat{\mathbf{s}}$ is updated over $t = 1,\ldots,T$ iterations:

$$\Delta\hat{\mathbf{s}}^t = \mathbf{Q}^t\mathbf{d}^t(I,\hat{\mathbf{s}}^{t-1}) \tag{2}$$

$$\hat{\mathbf{s}}^t = \hat{\mathbf{s}}^{t-1} + \Delta\hat{\mathbf{s}}^t. \tag{3}$$

A single sequence of $\{\mathbf{Q}^t\}_{t=1}^T$ can result in undesirable performance because the descent maps average conflicting gradient directions [30], which is increasingly problematic with more head pose variation. Xiong and De la Torre [30] proved that there exists a finite partition of the domain of Eq. (1) such that each part is a *domain of homogeneous descent*. However, they used only 2D training data. We extend their work to 3D landmark estimation, and show results on a wider range of head poses. Specifically, we partition Eq. (1) into $v = 1,\ldots,V$ *viewpoint domains* and learn a separate CSR for each one. Eq. (1) then becomes

$$\hat{\mathbf{Q}}^{t,v} = \underset{\mathbf{Q}^{t,v}}{\operatorname{argmin}} \sum_{i\in\Phi^v}^{N} \|\Delta\mathbf{s}_i^t - \mathbf{Q}^{t,v}\mathbf{d}^t(I,\hat{\mathbf{s}}_i^{t-1})\|_2^2 \qquad \Delta\mathbf{s}_i^t = \mathbf{s}_i - \hat{\mathbf{s}}_i^{t-1}, \tag{4}$$

where Φ^v is the subset of training instances that belong to viewpoint domain v.

Section 3.5 describes how $\{\Phi^v\}_{v=1}^V$ are defined during training, and how the best view domain is chosen at test time. First, we describe the mathematical relationship between 3D face landmarks and their 2D image projections. We then describe our parametric 3D face shape model in Section 3.2, followed by our method for estimating a 3D face shape model from a single 2D image in Sections 3.3 and 3.4.

3.1 From 3D World Coordinates to 2D Image Coordinates

Assume that we have a pair of corresponding face shapes: a 3D shape $\mathbf{s}^w = [\mathbf{x}_1^w,\cdots,\mathbf{x}_L^w]$, where $\mathbf{x}^w = [X,Y,Z]^\mathsf{T}$ is in world space, and its 2D projection, $\mathbf{s}^i = [\mathbf{x}_1^i,\cdots,\mathbf{x}_L^i]$, where $\mathbf{x}^i = [x,y]^\mathsf{T}$ in the image. We can relate the two via the pinhole camera model:

$$\mathbf{x}^h = \mathbf{K}(\mathbf{R}\mathbf{x}^w + \mathbf{t}), \tag{5}$$

where $\mathbf{x}^h = \lambda[\mathbf{x}^{i\mathsf{T}}, 1]^\mathsf{T}$ is a homogeneous coordinate (λ is the homogeneous scaling factor), $\mathbf{K}$ is the intrinsic camera matrix, $\mathbf{R}$ is a rotation matrix, and $\mathbf{t}$ is a translation vector. If we have an estimate $\hat{\mathbf{x}}^h$ for $\mathbf{x}^h$, and we know $\mathbf{K}$, $\mathbf{R}$, and $\mathbf{t}$, we can compute an estimate $\hat{\mathbf{x}}^w$ for $\mathbf{x}^w$ by $\hat{\mathbf{x}}^w = \mathbf{R}^\mathsf{T}(\mathbf{K}^{-1}\hat{\mathbf{x}}^h - \mathbf{t})$. On the other hand, if $\mathbf{x}^h$ and $\mathbf{x}^w$ are known, we can estimate the intrinsic and extrinsic camera parameters via camera calibration.[1] This is a chicken-and-egg problem. Fortunately, because faces exhibit a relatively consistent 3D structure, $\mathbf{x}^w$ can be approximated by the average 3D shape, μ^w, among a set of training faces for the purpose of camera calibration [25]. Thus, given $\hat{\mathbf{x}}^h$ and μ^w, we can compute estimates $\hat{\mathbf{K}}$, $\hat{\mathbf{R}}$, and $\hat{\mathbf{t}}$. We describe in the next section a parametric model for $\mathbf{x}^h$.

3.2 3D Point Distribution Model

PDMs [11] model a set of N shapes $\mathbf{S} = [\mathbf{s}_1(:), \cdots, \mathbf{s}_N(:)]$[2] using a linear combination of shape bases $\mathbf{B}$ plus the mean shape μ:

$$\hat{\mathbf{s}}(:) = \mu(:) + \mathbf{B}\mathbf{p}, \quad \mathbf{s} \approx \hat{\mathbf{s}}, \tag{6}$$

where $\mathbf{p}$ contains the reconstruction parameters; μ, $\mathbf{B}$, and $\mathbf{p}$ are computed from $\mathbf{S}$ via PCA. Prior to PCA, the x- and y-dimensions of the shapes in $\mathbf{S}$ are aligned using Procrustes Analysis to remove variations due to 2D rotation, translation, and scale. To incorporate these variations back into the model, we append four global geometric transformation bases to $\mathbf{B}$, as described in Section 4.2.1 of [21], and re-orthonormalize. We define $\mathbf{s} = [\mathbf{x}_1, \cdots, \mathbf{x}_L]$, where $\mathbf{x}_l = [x_l, y_l, \lambda_l]^\mathsf{T}$. Note that $\mathbf{x}$ is a hybrid between a homogeneous coordinate, $\mathbf{x}^h = [x^h, y^h, \lambda]^\mathsf{T}$, where $x^h = \lambda x$ and $y^h = \lambda y$, and an image coordinate, $\mathbf{x}^i = [x, y]^\mathsf{T}$. We note that Jourabloo and Liu [16] also employed a 3DPDM, but they parameterized 3D world coordinates (*e.g.*, $\mathbf{x}^w$) instead of hybrid coordinates $\mathbf{x}$, which requires two independent estimates (3D shape + camera parameters) just to estimate the 2D landmark coordinates in the image. We use the above hybrid scheme (originally proposed in [25]) so that x and y are directly related to the 2D image observation. Once estimated, $\mathbf{s}$ contains enough information to compute the 3D world coordinates of each landmark (Section 3.1).

3.3 3DPDM Cascaded Shape Regression

Each $\mathbf{Q}^{t,v}$ is computed offline by solving the following ridge regression problem:

$$\underset{\hat{\mathbf{Q}}^{t,v}}{\operatorname{argmin}} \sum_{i \in \Phi^v} \|\Delta\hat{\mathbf{p}}_i^t - \mathbf{Q}^{t,v}\mathbf{d}^{t,v}(I_i, \mathbf{s}^{t-1})\|_2^2 + \alpha\|\mathbf{Q}^{t,v}\|_2^2, \tag{7}$$

where $\Delta\hat{\mathbf{p}}_i^t$ is the ideal parameter update for face i. Ridge regression (*i.e.*, the second term) is necessary because $\mathbf{d}(I, \mathbf{s})$ is very high dimensional (see Section 3.4) and substantial overfitting would result without it [23]. At test time, the shape is updated for $t = 1, \ldots, T$:

$$\Delta\hat{\mathbf{p}}^t = \mathbf{Q}^{t,v}\mathbf{d}^{t,v}(I, \mathbf{s}^{t-1}) \tag{8}$$

$$\hat{\mathbf{p}}^t = \hat{\mathbf{p}}^{t-1} + \Delta\hat{\mathbf{p}}^t \tag{9}$$

$$\hat{\mathbf{s}}^t = \mu^v + \mathbf{B}^v\hat{\mathbf{p}}^t. \tag{10}$$

$\mathbf{Q}^{t,v}$ and $\mathbf{d}^{t,v}(I, \mathbf{s})$ are learned separately for each cascade level and view domain, and $\mathbf{B}^v$ and μ^v are learned independently for each viewpoint domain.

[1] For example, using OpenCV's `calibrateCamera` function.

[2] Here we use MATLAB notation to represent a vector version of $\mathbf{s} \in \mathbb{R}^{3\times L}$: $\mathbf{s}(:) \in \mathbb{R}^{3L\times 1}$.

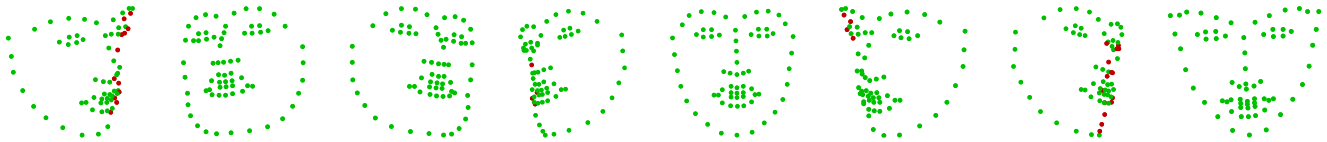


Figure 1: The modal viewpoints found for $V = 8$ viewpoint domains. The modal occlusion state of the landmarks is stored for each viewpoint domain (green is visible, red is occluded). Features are extracted around only visible landmarks.

3.4 Learning 3D-Invariant Feature Mapping Functions

$\mathbf{d}^{t,v}$ is composed of many local, independent feature mapping functions: $\mathbf{d} = [d_1, \cdots, d_M]$. We use an ensemble of regression trees [3] to learn each $\mathbf{d}$ (one tree for each d_m). Instead of using the 2D landmark coordinates as regression targets (*e.g.*, as in [23]), we use the 3DPDM parameter updates $\Delta\mathbf{p}$. Rather than allocate trees uniformly across all landmarks, *we allocate them to only the visible landmarks in each view domain*. For a view domain that covers leftward-looking faces, for example, the result is an ensemble of trees that uses only features from visible landmarks on the right side of the face.

To learn each tree split node, our algorithm first chooses one dimension of $\Delta\mathbf{p}$. The choice is random, but weighted proportional to the amount of variance that each dimension of $\Delta\mathbf{p}$ encodes. This emphasizes the most significant modes of variation in the 3DPDM. The training algorithm tests $F = 500$ random pixel-difference features, described below, and selects the one that gives rise to maximum variance reduction in the chosen dimension of $\Delta\mathbf{p}$. In other words, the algorithm chooses the feature that, when thresholded, splits the incoming training examples into two groups that have the most consistent values for the chosen dimension of $\Delta\mathbf{p}$. The result is a random forest with M trees, which outputs $\mathbf{d}$, a high-dimensional binary vector with M nonzero entries and length equal to the number of leaves in the forest. Once $\mathbf{Q}$ is found via Eq. (7), we can replace each leaf with the corresponding column of $\mathbf{Q}$ in order to shortcut the matrix-vector multiplication $\mathbf{Qd}$ at test time.

The pixel-difference features are defined as $I(\mathbf{u}_1) - I(\mathbf{u}_2)$, where $\mathbf{u} = \mathbf{x}_u + s\mathbf{R}_f\delta_u$, $\mathbf{x}_u$ is the nearest landmark coordinate in the image plane, and δ_u is selected during training and is defined in a scale- and rotation-normalized reference frame. In [23], s and $\mathbf{R}_f$ reflect the 2D scale and in-plane rotation of the face. However, to account for 3D head orientation, we define $\mathbf{R}_f$ as the upper-left 2×2 submatrix of $\hat{\mathbf{R}}$ in Eq. (5), which is re-computed at each t. This is the "3D transform indexing" method described in [25].

3.5 Assigning Faces to Viewpoint Domains

For each training instance i, we compute the head orientation vector $\vec{\mathbf{n}}_i = \mathbf{R}_i\vec{\mathbf{z}}$, where $\mathbf{R}_i$ is the known rotation and $\vec{\mathbf{z}} = [0,0,1]^\mathsf{T}$ (assuming the reference face, centered at the origin, is facing positive z). We then partition $\{\vec{\mathbf{n}}_i\}_{i=1}^N$ using the k-means++ algorithm [1]. Xiong and De la Torre [30] proposed an alternative scheme based on partitioning 2D shapes according to the two most dominant axes in PCA space. We found that the two most dominant axes in PCA space correspond roughly to yaw and tilt rotation in our experimental dataset. Thus, intuitively, directly partitioning the head orientation vectors, which encode yaw and tilt, should produce a similar partitioning, but in a more straightforward way. Figure 1 shows the modes of each viewpoint domain for $V = 8$.

At test time, we run all V models on the test face to produce V face shape estimates.

Although runtime scales linearly with V, the regression forest approach with pixel-difference features is extremely fast ($>$ 300 FPS on 68 2D landmarks in [23]), our 3DPDMs use fewer parameters than the nonparametric shape model approach, and the camera calibration step takes less than $1ms$ per cascade level. Thus, for $V < 10$, the algorithm runs in real time.

Our algorithm produces a single estimate by classifying the output from each viewpoint domain model, and choosing the result with highest classification score. Specifically, we train a random *classification* forest for each viewpoint domain v, which are similar to the random regression forests in Section 3.4, except the regression targets are replaced with a binary class variable c_i for each image i. Instead of minimizing the variance at each split node, we minimize the entropy of the classes that fall into each child node. To train each classification forest, we label all faces $i \in \Phi^v$ (*i.e.*, members of viewpoint domain v) with $c_i = 1$, and all faces $i \notin \Phi^v$ with $c_i = 0$. Note that the features used for the classification forest for viewpoint domain v are *shape-indexed* [6] relative to the final landmark estimates from model v. Chen *et al.* [8] showed that such a "post-classifier" produces more accurate results than the conventional approach, which indexes features relative to a bounding box. Each leaf node stores a classification score computed as $\sum_{i \in \Omega^l} c_i / |\Omega^l|$, where Ω^l is the subset of training examples that fall into leaf l. The final score is the average among all leaf nodes in the forest reached by instance i. In [8], they trained a linear SVM to classify the binary vector encoding the random forest output, $\mathbf{d}$. However, we found empirically that directly using the random forest leaves produces more accurate results.

For $V = 6$, we found that the above approach chooses the correct model for 82% of test faces, which is too low to ensure good results. We found that misclassifications almost always occured near the boundary between adjacent viewpoint domains. Therefore, to train the shape regressors, we expand each Φ^v outward to form $\Phi^v_{\mathbf{s}}$ such that $|\Phi^v_{\mathbf{s}}| = \frac{\beta N}{V}$, where $\beta > 1$. Larger (overlapping) view domains ensure that boundary faces can be handled by multiple models, and reduces over-fitting. We set $\beta = 2$ in our implementation, which results in correct assignment (true domain *or* an overlapping domain) for 99% of test faces. Intuitively, $\beta = 2$ means that the training set for viewpoint domain v is expanded to twice its original size, with 50% of the instances coming from neighboring classes.

3.6 Self-Occlusion Detection

Landmark visibility is computed automatically via $\hat{\mathbf{R}}$ in Eq. (5) and domain knowledge of each 3D landmark [16]. Specifically, we compute the surface normal $\vec{\mathbf{n}}_l$ at each 3D landmark l from the 3D mesh for each training example (after alignment to a reference), and produce an average (reference) normal $\vec{\mu}^{\mathbf{n}}_l$ for each landmark l. At test time we rotate $\vec{\mu}^{\mathbf{n}}_l$, $[\hat{n}_x, \hat{n}_y, \hat{n}_z]^\mathsf{T} = \hat{\mathbf{R}}\vec{\mu}^{\mathbf{n}}_l$. If the rotated normal is oriented away from the camera (*i.e.*, $\hat{n}_z < 0$), then we label the landmark as occluded.

4 Results and Discussion

Few datasets exist that include 2D images with 3D landmark annotations. Therefore, direct comparison with almost all face alignment algorithms is not possible. However, Tulyakov and Sebe [25] recently built an evaluation dataset for 3D-to-2D face alignment from the BU-4DFE database [31]. BU-4DFE includes sequences of 3D scans and color image maps of 101 diverse male and female subjects making six facial expressions (anger, disgust, fear, happy,

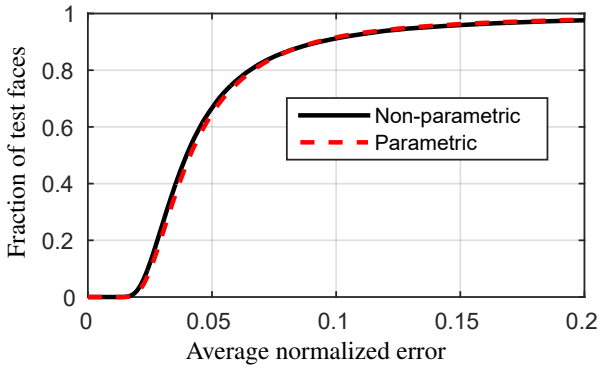
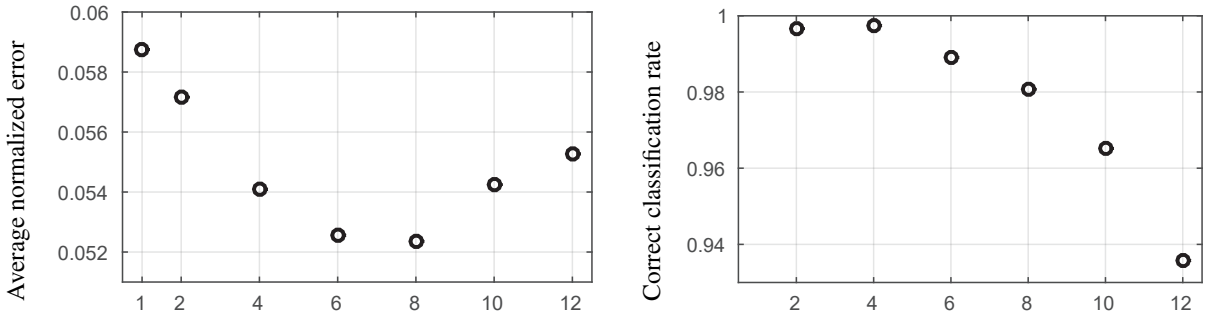


Figure 2: Comparison between two baseline algorithms, one with landmark coordinates used directly as regression targets (nonparametric), and the other with 3DPDM shape parameters used as regression targets (parametric). Observe that performance is almost identical.



(a) Number of viewpoint domains, V **(b)** Number of viewpoint domains, V

Figure 3: **(a)** Face alignment accuracy vs. number of viewpoint domains. Observe that $V = 8$ produces the most accurate results overall. However, $V = 6$ produces a good tradeoff between accuracy and efficiency (model size and runtime). **(b)** Viewpoint domain assignment accuracy vs. number of viewpoint domains. By 'correct' we mean the test face is correctly assigned to either its true viewpoint domain *or* an overlapping domain (Section 3.5).

sad, and surprised). Tulyakov and Sebe manually annotated 3000 images from BU-4DFE according to the 68-landmark MultiPIE markup [13]. They augmented the 2D landmarks with a depth value by mapping each image onto the corresponding 3D scan. We used their 3D annotations throughout our evaluation, and followed their procedure to generate an evaluation dataset as similar as possible to theirs. Specifically, we randomly generated 40 yaw and tilt angles within the range $[-50, 50]$ degrees for each 3D face. We then rotated each face according to the random angles, projected each rotated 3D face onto the 2D image plane, and superimposed each instance onto a background image chosen at random from the SUN database [28] (see Figure 4 for several examples). This generated a dataset of 120K images. Following [25], we performed six-fold cross validation throughout, where each fold contained a random subset of subjects from BU-4DFE (no subject belonged to more than one fold). Error values were computed by the average distance between 3D landmarks in $\mathbf{s}$ and $\hat{\mathbf{s}}$, normalized by the inter-ocular distance. In all cases, we set the number of cascade levels to $T = 5$, we constructed random forests with 600 trees at a maximum depth of 6 nodes, and we constructed 3DPDMs such that 99% of the shape variance was retained.

4.1 Parametric vs. Nonparametric Shape Models

Parametric shape models have several benefits over nonparametric ones, despite the recent trend in the literature: all landmarks are optimized simultaneously, and there are fewer parameters to optimize, which results in faster runtimes. Most important of all, we show in Figure 2 that *parametric models do not sacrifice accuracy, which suggests that they generalize just as well as nonparametric models.* To generate the two cumulative error distribution

Method	Average Normalized 3D Errors
Tulyakov and Sebe, Baseline indexing	0.0610
Tulyakov and Sebe, 3D Transform	0.0607
Tulyakov and Sebe, Basis Transform	0.0592
Our Baseline ($V = 1$), Nonparametric	0.0586
Our Baseline ($V = 1$), Parametric	0.0588
Ours, $V = 6$	0.0535
Ours, $V = 8$	**0.0524**

Table 1: Comparison of average normalized 3D landmark localization errors (average point-to-point error between estimated and ground truth landmarks divided by inter-ocular distance). The top three rows are copied from Tulyakov and Sebe [25].

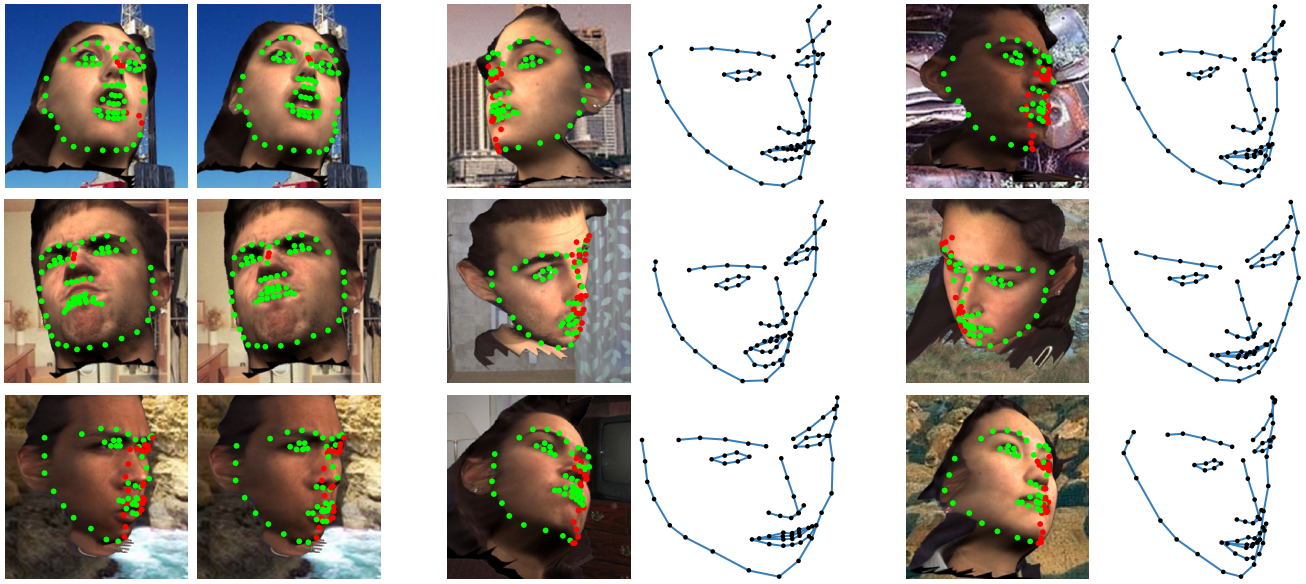


Figure 4: Qualitative results on faces from BU-4DFE [31]. The estimated visibility of each landmark is shown in green (visible) and red (occluded). Results in (a) are from our baseline algorithm, and results in (b) are from our algorithm with $V = 6$. Notice the improvements from (a) to (b). In (c) and (e), a selection of results with estimated 3D shapes in (d) and (f). Note that the 3D shapes were rotated to a common orientation for fair comparison.

(CED) curves in Figure 2, we trained two systems, one with landmark coordinates used directly as regression targets (nonparametric), and the other with 3DPDM shape parameters used as regression targets (parametric), as described Section 3.3. The systems are otherwise identical. We set $V = 1$. Therefore, these two systems serve as baseline algorithms.

4.2 Comparison with Tulyakov and Sebe [25]

Observe in Table 1 that our baseline algorithms are comparable in accuracy to [25]. However, our full system, with multiple viewpoint domains, is significantly more accurate.

4.3 Accuracy vs. Number of Viewpoint Domains

We trained and tested 7 different systems with a different number of viewpoint domains; $V = 1(\text{baseline}), 2, 4, \ldots, 12$. We set $\beta = 2$ in $|\Phi_{\mathbf{s}}^{v}| = \frac{\beta N}{V}$ (N is the number of images), which doubles the size of each viewpoint domain region, except for $V = 1$ ($\beta = 1$) and $V = 2$ ($\beta = 1.5$). Figure 3(a) shows the error drops significantly from $V = 1$ to $V = 6$, with small

improvement from $V = 6$ to $V = 8$. The error is worse for $V \geq 10$, in part because correct viewpoint domain assignment (Section 3.5) is more difficult as V increases, as shown in Figure 3(b). Larger V also results in slower runtime. $V = 6$ therefore offers a good tradeoff between accuracy and efficiency. We show qualitative results in Figure 4.

5 Conclusions

We proposed an algorithm for estimating 3D facial landmarks with self-occlusion detection from a single 2D image. We divide the 3D cascaded shape regression problem into a set of *viewpoint domains*, which helps avoid problems in the optimization, such as local minima and averaged conflicting gradient directions in the domain maps. These problems are especially important to address in the 3D case, where a wider range of head poses is allowed. We showed that localization error is reduced significantly from $V = 1$ to $V = 6$. For $V > 6$, each viewpoint domain is simpler and more tailored to viewpoint range, but the viewpoint domain assignment problem becomes more difficult and classification accuracy falls with larger V. Therefore, $V = 6$ is a good tradeoff between landmark accuracy, model size, and runtime. We also argued that parametric shape models have several desirable qualities and showed empirically that they perform just as well as modeling shape nonparametrically.

References

[1] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, 2007.

[2] T. Baltrusaitis, P. Robinson, and L. Morency. 3D constrained local model for rigid and non-rigid facial tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[3] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[4] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *IEEE International Conference on Computer Vision*, 2013.

[5] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3D shape regression for real-time facial animation. *ACM Transactions on Graphics*, 32(4):41:1–41:10, July 2013.

[6] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[7] X. Cao, Q. Hou, and K. Zhou. Displacing dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics*, 33(4), 2014.

[8] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *European Conference on Computer Vision*, 2014.

[9] T. Cootes, K. Walker, and C. Taylor. View-based active appearance models. In *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, 2015.

[10] T. F. Cootes and C. Taylor. A mixture model for representing shape variation. In *British Machine Vision Conference*, 1999.

[11] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[12] A. Criminisi and J. Shotton. Regression forests. In A. Criminisi and J. Shotton, editors, *Decision Forests for Computer Vision and Medical Image Analysis*, Advances in Computer Vision and Pattern Recognition, pages 47–58. 2013.

[13] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. *Image and Vision Computing*, 28(5):807–813, May 2010.

[14] L. Gu and T. Kanade. 3D alignment of face in a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[15] L. A. Jeni, J. F. Cohn, and T. Kanade. Dense 3D face alignment from 2d videos in real-time. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2015.

[16] A. Jourabloo and X. Liu. Pose-invariant 3d face alignment. In *IEEE International Conference on Computer Vision*, 2015.

[17] A. Kanaujia, Y. Huang, and D. N. Metaxas. Tracking facial features using mixture of point distribution models. In *Computer Vision, Graphics and Image Processing*, volume 4338 of *Lecture Notes in Computer Science*, pages 492–503. Springer Berlin Heidelberg, 2006.

[18] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[19] D. Lee, H. Park, and C. D. Yoo. Face alignment using cascade gaussian process regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[21] I. Matthews and S. Baker. Active Appearance Models revisited. *International Journal of Computer Vision*, 60(2):135 – 164, November 2004.

[22] I. Matthews, J. Xiao, and S. Baker. 2D vs. 3D deformable face models: Representational power, construction, and real-time fitting. *International Journal of Computer Vision*, 75(1):93–113, 2007.

[23] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[24] S. Romdhani, S. Gong, and A. Psarrou. A multi-view nonlinear active shape model using kernel PCA. In *British Machine Vision Conference*, 1999.

[25] S. Tulyakov and N. Sebe. Regressing a 3D face shape from a single image. In *IEEE International Conference on Computer Vision*, 2015.

[26] G. Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[27] Y. Wu and Q. Ji. Robust facial landmark detection under significant head poses and occlusion. In *IEEE International Conference on Computer Vision*, 2014.

[28] J. Xiao, J. Hays., K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[29] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[30] X. Xiong and F. De la Torre. Global supervised descent method. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[31] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

[32] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *IEEE International Conference on Computer Vision*, 2013.

[33] X. Yu, Z. Lin, J. Brandt, and D. N. Metaxas. Consensus of regression for occlusion-robust facial feature localization. In *European Conference on Computer Vision*, 2014.

[34] Y. Zhou, W. Zhang, X. Tang, and H. Shum. A bayesian mixture model for multi-view face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[35] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.