Dense 3D Face Alignment from 2D Videos in Real-Time

László A. Jeni[1], Jeffrey F. Cohn[1,2], and Takeo Kanade[1]
[1] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[2] Department of Psychology, University of Pittsburgh, Pittsburgh, PA, USA

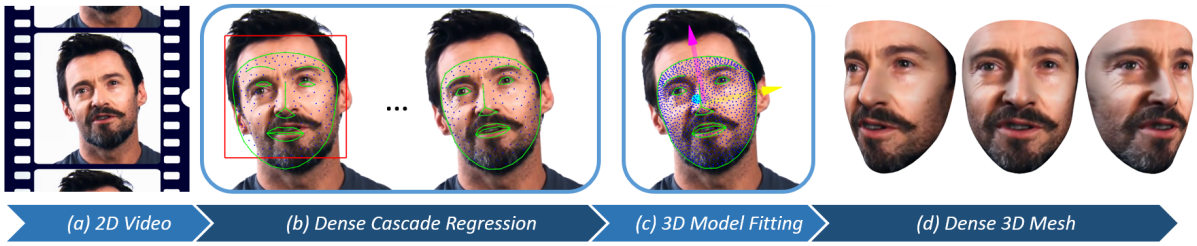

Fig. 1: From a 2D image of a person's face (a) a dense set of facial landmarks is estimated using a fast, consistent cascade regression framework (b), then a part-based 3D deformable model is applied (c) to reconstruct a dense 3D mesh of the face (d).

***Abstract*— To enable real-time, person-independent 3D registration from 2D video, we developed a 3D cascade regression approach in which facial landmarks remain invariant across pose over a range of approximately 60 degrees. From a single 2D image of a person's face, a dense 3D shape is registered in real time for each frame. The algorithm utilizes a fast cascade regression framework trained on high-resolution 3D face-scans of posed and spontaneous emotion expression. The algorithm first estimates the location of a dense set of markers and their visibility, then reconstructs face shapes by fitting a part-based 3D model. Because no assumptions are required about illumination or surface properties, the method can be applied to a wide range of imaging conditions that include 2D video and uncalibrated multi-view video. The method has been validated in a battery of experiments that evaluate its precision of 3D reconstruction and extension to multi-view reconstruction. Experimental findings strongly support the validity of real-time, 3D registration and reconstruction from 2D video. The software is available online at `http://zface.org`.**

I. INTRODUCTION

Face alignment is the problem of automatically locating detailed facial landmarks across different subjects, illuminations, and viewpoints. Previous methods can be divided into two broad categories. 2D-based methods locate a relatively small number of 2D fiducial points in real time while 3D-based methods fit a high-resolution 3D model offline at a much higher computational cost and usually require manual initialization. 2D-based approaches include Active Appearance Models [11], [28], Constrained Local Models [12], [31] and shape-regression-based methods [16], [10], [37], [6], [30]). These approaches train a set of 2D models, each of which is intended to cope with shape or appearance variation within a small range of viewpoints. In contrast, 3D-based methods [5], [15], [41], [18] accommodate wide range of views using a single 3D model. Recent 2D approaches enable person-independent initialization, which is not possible with 3D approaches. 3D approaches have advantage with respect to representational power and robustness to illumination and pose but are not feasible for generic fitting and real-time use.

Seminal work by Blanz et al. [5] on 3D morphable models minimized intensity difference between synthesized and source-video images. Dimitrijevic et al. [15] proposed a 3D morphable model similar to that of Blanz that discarded the texture component in order to reduce sensitivity to illumination. Zhang et al. [41] proposed an approach that deforms a 3D mesh model so that the 3D corner points reconstructed from a stereo pair lie on the surface of the model. Both [41] and [15] minimize shape differences instead of intensity differences, but rely on stereo correspondence. Single view face reconstruction methods [23], [20] produce a detailed 3D representation, but do not estimate the deformations over time. Recently, Suwajanakorn et al. [33] proposed a 3D flow based approach coupled with shape from shading to reconstruct a time-varying detailed 3D shape of a person's face from a video. Gu and Kanade [18] developed an approach for aligning a 3D deformable model to a single face image. The model consists of a set of sparse 3D points and the view-based patches associated with every point. These and other 3D-based methods require precise initialization, which typically involves manual labeling of the fiduciary landmark points. The gain with 3D-based approaches is their far greater representational power that is robust to illumination and viewpoint variation that would scuttle 2D-based approaches.

A key advantage of 2D-based approaches is their much lower computational cost and more recently the ability to forgo manual initialization. In the last few years in particular, 2D face alignment has reached a mature state with the emergence of discriminative shape regression methods [10], [6], [13], [16], [32], [36], [27], [37], [30], [39], [22], [2], [9]. These techniques predict a face shape in a cascade manner: They begin with an initial guess about shape and then progressively refine that guess by regressing a shape increment step-by-step from a feature space. The feature space can be either hand designed, such as SIFT features [37], or learned from data [10], [6], [30].

Accepted for the 11th IEEE International Conference on Automatic Face and Gesture Recognition, Ljubljana, Slovenia, May 2015. Please refer to IEEE Xplore for the final version.

Most previous work has emphasized 2D face tracking and registration. Relatively neglected is the application of cascade regression in dense 3D face alignment. Only recently did Cao et al. [9] propose a method for regressing facial landmarks from 2D video. Pose and facial expression are recovered by fitting a user-specific blendshape model to them. This method then was extended to a person-independent case [8], where the estimated 2D markers were used to adapt the camera matrix and user identity to better match facial expression. Because this approach uses both 2D and 3D annotations, a correction step is needed to resolve inconsistency in the landmark positions across different poses and self-occlusions.

Our approach exploits 3D cascade regression, where the facial landmarks are consistent across all poses. To avoid inconsistency in landmark positions encountered by Cao et al., the face is annotated completely in 3D by selecting a dense set of 3D points (shape). Binary feature descriptors (appearance) associated with a sparse subset of the landmarks are used to regress projections of 3D points. The method first estimates the location of a dense set of markers and their visibility, then reconstructs face shapes by fitting a part-based 3D model. The method was made possible in part by training on the BU-4DFE [38] and BP-4D-Spontaneous [40] datasets that contain over 300,000 high-resolution 3D face scans. Because the algorithm makes no assumptions about illumination or surface properties, it can be applied to a wide range of imaging conditions. The method was validated in a series of tests. We found that 3D registration from 2D video effectively handles previously unseen faces with a variety of poses and illuminations.

This paper advances two main novelties:

Dense cascade-regression-based face alignment
Previous work on cascade-regression-based face alignment was limited to a small number of fiducial landmarks. We achieve a dense alignment with a manageable model size. We show that this is achievable by using a relatively small number of sparse measurements and a compressed representation of landmark displacement-updates. Furthermore, the facial landmarks are always consistent across pose, eliminating the discrepancies between 2D and 3D annotations that have plagued previous approaches.

Real-time 3D part-based deformable model fitting
By using dense cascade regression, we fit a 3D, part-based deformable model to the markers. The algorithm iteratively refines the 3D shape and the 3D pose until convergence. We utilize measurements over multiple frames to refine the rigid 3D shape.

The paper is organized as follows: Section II details the dense 3D model building process and Section III describes the model fitting method in details. The efficiency of our novel solution method is illustrated by numerical experiments in Section IV. Conclusions are drawn in Section V.

Notations. Vectors ($\mathbf{a}$) and matrices ($\mathbf{A}$) are denoted by bold letters. An $\mathbf{u} \in \mathbb{R}^d$ vector's Euclidean norm is $\|\mathbf{u}\|_2 = \sqrt{\sum_{i=1}^{d} u_i^2}$. $\mathbf{B} = [\mathbf{A}_1; \ldots; \mathbf{A}_K] \in \mathbb{R}^{(d_1+\ldots+d_K)\times N}$ denotes the concatenation of matrices $\mathbf{A}_k \in \mathbb{R}^{d_k \times N}$.

II. DENSE FACE MODEL BUILDING

In this section we detail the components of the dense 3D face model building process.

A. Linear Face Models

We are interested in building a dense linear shape model. A shape model is defined by a 3D mesh and, in particular, by the 3D vertex locations of the mesh, called landmark points. Consider the 3D shape as the coordinates of 3D vertices that make up the mesh:

$$\mathbf{x} = [x_1; y_1; z_1; \ldots; x_M; y_M; z_M], \tag{1}$$

or, $\mathbf{x} = [\mathbf{x}_1; \ldots; \mathbf{x}_M]$, where $\mathbf{x}_i = [x_i; y_i; z_i]$. We have T samples: $\{\mathbf{x}(t)\}_{t=1}^{T}$.

We assume that – apart from scale, rotation, and translation – all samples $\{\mathbf{x}(t)\}_{t=1}^{T}$ can be approximated by means of a linear subspace.

The 3D point distribution model (PDM) describes non-rigid shape variations linearly and composes it with a global rigid transformation, placing the shape in the image frame:

$$\mathbf{x}_i = \mathbf{x}_i(\mathbf{p}) = s\mathbf{R}(\bar{\mathbf{x}}_i + \mathbf{\Phi}_i\mathbf{q}) + \mathbf{t} \quad (i = 1, \ldots, M), \tag{2}$$

where $\mathbf{x}_i(\mathbf{p})$ denotes the 3D location of the i^{th} landmark and $\mathbf{p} = \{s, \alpha, \beta, \gamma, \mathbf{q}, \mathbf{t}\}$ denotes the parameters of the model, which consist of a global scaling s, angles of rotation in three dimensions ($\mathbf{R} = \mathbf{R}_1(\alpha)\mathbf{R}_2(\beta)\mathbf{R}_3(\gamma)$), a translation $\mathbf{t}$ and non-rigid transformation $\mathbf{q}$. Here $\bar{\mathbf{x}}_i$ denotes the mean location of the i^{th} landmark (i.e. $\bar{\mathbf{x}}_i = [\bar{x}_i; \bar{y}_i; \bar{z}_i]$ and $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_1; \ldots; \bar{\mathbf{x}}_M]$). The d pieces of $3M$ dimensional basis vectors are denoted with $\mathbf{\Phi} = [\mathbf{\Phi}_1; \ldots; \mathbf{\Phi}_M] \in \mathbb{R}^{3M \times d}$.

Vector $\mathbf{q}$ represents the 3D distortion of the face in the $3M \times d$ dimensional linear subspace. To build this model we used high-resolution 3D face scans. We describe this in the next subsection.

B. Datasets

The algorithm was trained on two related 3D datasets. They were BU-4DFE [38] and BP4D-Spontaneous [40].

BU-4DFE consists of approximately 60,600 3D frame models from 101 subjects (56% female, 44% male). Subjects ranged in age from 18 years to 70 years and were ethnically and racially diverse (European-American, African-American, East-Asian, Middle-Eastern, Asian, Indian, and Hispanic Latino). Subjects were imaged individually using the Di3D (Dimensional Imaging [14]) dynamic face capturing system while posing six prototypic emotion expressions (anger, disgust, happiness, fear, sadness, and surprise). The Di3D system consisted of two stereo cameras and a texture video camera arranged vertically. Both 3D model and 2D texture videos were obtained for each prototypic expression and subject. Given the arrangement of the stereo cameras, frontal looking faces have the most complete 3D information and smallest amount of texture distortion.

Fig. 2: The 2D annotation of a profile-view image mapped on a frontal view face. Note, that certain landmarks (eyebrow, jawline) do not correspond to the same points on the two views because of the different head-poses and self-occlusions.

The 3D models of 3D video sequences have a resolution of approximately 35,000 vertices. BP-4D-Spontaneous dataset [40] consists of over 300,000 frame models from 41 subjects (56% female, 48.7% European-American, average age 20.2 years) of similarly diverse backgrounds to BU-4DFE. Subjects were imaged using the same Di3D system while responding to a varied series of 8 emotion inductions that elicited spontaneous expressions of amusement, surprise, fear, anxiety, embarrassment, pain, anger, and disgust. The 3D models range in resolution between 30,000 and 50,000 vertices. For each sequence, manual FACS coding [17] by highly experienced and reliable certified coders was obtained.

For training, we selected 3000 close-to-frontal frames from each dataset, (i.e., 6000 frames in total). In BU-4DFE, we sampled uniformly distributed frames from each sequence. In BP4D-Spontaneous, we sampled frames based on the available FACS (Facial Action Coding System [17]) annotation to include a wide range of expressions. Some 3D meshes in the two datasets are corrupted or noisy. During the selection we eliminated meshes that had large error.

C. 2D vs. 3D Annotation

Automatic face alignment requires a large number of training examples of annotated images. Annotation is usually done using 2D images, where the annotator selects the locations of fiducial points around permanent facial features (e.g., brows and eyes). For frontal faces, reliable annotation can be achieved. As face orientation varies from frontal, however, annotated points lose correspondence. Pose variation results in self-occlusion that confounds landmark annotation. For example, consider the landmarks on the eyebrow and jawline. With increasing rotation and associated self-occlusion, annotations no longer correspond to the same landmarks on profile and frontal view images. See Figure 2 for an illustration of this problem. This issue can be alleviated by using 3D face-scans and annotating the 3D meshes themselves, instead of the 2D images.

The 3D meshes were manually annotated with 77 landmarks, corresponding to facial fiducial points. This coarse set of markers had the same semantic meaning across subjects and expressions. Figure 3 shows the annotated and rotated meshes with the annotated markers and the corresponding depth maps. Since the annotation is 3D, we can identify the self-occluded markers from every pose.

The time-consuming annotation process can be accelerated by using the semi-automatic approach of Baltrusaitis et al. [4].

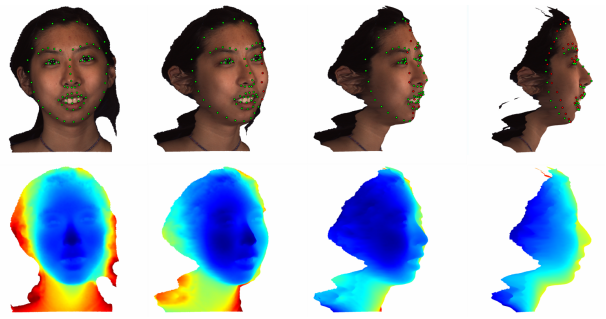

Fig. 3: The 77 point annotation of the 3D mesh (top-left image), and profile views (30-60-90 degrees of yaw rotation). The color of the markers indicate the visibility from the given viewpoint (green – visible, red – occluded). The bottom row shows the corresponding depth images.

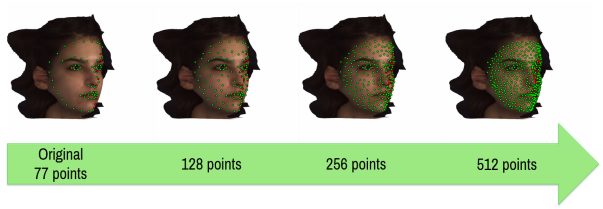

Fig. 4: Surface tessellation using the adaptive refinement scheme. The vertices are evenly distributed on the surface and they follow the original geometry.

D. Dense 3D Correspondence

While the coarse set of manually annotated markers has the same semantic meaning across subjects and expressions, to create a dense model that spans the data of multiple subjects requires establishing dense point-to-point correspondences among them [5], [1]. This means that the position of each vertex may vary in different samples, but its context label should remain the same. To establish dense correspondence, we used the Wave Kernel Signature (WKS) [3].

WKS is a novel shape feature descriptor. It is based on the Laplace–Beltrami operator [25] and carries a physical interpretation: it arises from studying the Schrödinger equation governing the dissipation of quantum mechanical particles on the geometric surface. The WKS allows for accurate feature matching (see [3] for more details).

The number of vertices and their locations vary across the 3D face scans. To establish a reference shape, we used ordinary Procrustes analysis [21] with the 77 3D markers and registered each mesh to the same frame.

We then calculated a dense mean shape by uniformly subsample the meshes down to 5000 vertices and calculated WKS descriptors for this reference shape as well.

We are interested in a model where we can easily control the level of detail of the 3D mesh. To build such a model, we employed a coarse-to-fine mesh refinement that resembles an adaptive $\sqrt{3}$-subdivision [24] scheme. We started from the reference shape and its triangulated 77-points mesh. Since this annotation corresponds to fiducial points and is not based

on uniform sampling or surface complexity, applying the original $\sqrt{3}$-subdivision would result in unnecessary details around these markers. Therefore, in every step we apply the subdivision only on the triangle with the largest surface area and project the centroid back to the dense mesh. This procedure results in a tessellation, where the vertices are evenly distributed on the surface, follow the original geometry and the level of detail can be easily managed. After we tessellated the reference mesh, we identify the corresponding vertices from every mesh by finding the closest WKS match. See Figure II-C for an illustration of the method. We stopped the process at 1024 vertices. In Section IV-B we give a more detailed explanation why we choose this level of detail. We used these 1024 vertices meshes to build our part-based linear model.

E. Part-based model building

In Eq. (2) one can assume that the prior of the parameters follow a normal distribution with mean $\mathbf{0}$ and variance $\mathbf{\Lambda}$ at a parameter vector $\mathbf{q}$: $p(\mathbf{p}) \propto N(\mathbf{q};\mathbf{0},\mathbf{\Lambda})$ and can use Principal Component Analysis (PCA) to determine the d pieces of $3M$ dimensional basis vectors ($\mathbf{\Phi} = [\mathbf{\Phi}_1;\ldots;\mathbf{\Phi}_M] \in \mathbb{R}^{3M\times d}$). This approach has been used successfully in a broad range of face alignment techniques, such as Active Appearance Models [28] or 3D Morphable Models [5]. This procedure would result in a holistic shape model with a high compression rate, but on the on the other hand, its components have a global reach and they lack of semantic meaning.

The deformations on the face can be categorized into two separate subsets: rigid (the shape of the face) and non-rigid (facial expressions) parts. We reformulate Eq. (2) to model these deformations separately:

$$\mathbf{x}_i = \mathbf{x}_i(\mathbf{p}) = s\mathbf{R}(\bar{\mathbf{x}}_i + \mathbf{\Theta}_i\mathbf{r} + \mathbf{\Psi}_i\mathbf{s}) + \mathbf{t} \quad (i = 1,\ldots,M), \quad (3)$$

where the d pieces of $3M$ dimensional basis vectors ($\mathbf{\Theta} = [\mathbf{\Theta}_1;\ldots;\mathbf{\Theta}_M] \in \mathbb{R}^{3M\times d}$) describes the rigid, and the the e pieces of $3M$ dimensional basis vectors ($\mathbf{\Psi} = [\mathbf{\Psi}_1;\ldots;\mathbf{\Psi}_M] \in \mathbb{R}^{3M\times e}$) describes the non-rigid deformations.

To build the rigid part, we selected neutral frames from each subjects and applied PCA to determine the basis vectors ($\mathbf{\Theta}$) and their mean ($\bar{\mathbf{x}}$). This provide us a holistic linear subspace, that describes the variation of the face shape only. Note that the neutral face is only required during the model building, it is not required for testing.

To build a linear subspace that describes the non-rigid deformations ($\mathbf{\Psi}$) we follow the method of Tena et al [34]. The goal is to build a model that composed of a collection of PCA part-models that are independently trained but share soft boundaries. This model generalizes to unseen data better than the traditional holistic approach. To create the part-based-models, we group vertices that are highly correlated and form compact regions, since these regions will be better compressed by PCA. To find a data-driven segmentation of the facial expressions, we used all the 6000 frames we selected from the BU-4DFE and BP-4D-Spontaneous datasets. From each mesh, we subtracted the person's own neutral face to remove all the personal variation from the data. Note, that if we would have used the global mean face for the subtraction ($\bar{\mathbf{x}}$) that would leave some of the rigid variation in the dataset.

Our data $\mathbf{D} \in \mathbb{R}^{6000\times 3072}$ consist of 6000 frames and 1024 3D vertices. We split $\mathbf{D}$ into three subsets $\mathbf{D}_x, \mathbf{D}_y, \mathbf{D}_z \in \mathbb{R}^{6000\times 1024}$ each containing the corresponding spatial coordinate of the vertices. To describe the measurement of the correlation between vertices, the normalized correlation matrices are computed from $\mathbf{D}_x, \mathbf{D}_y, \mathbf{D}_z$ and then averaged into a global correlation matrix $\mathbf{C}$. Vertices in the same region should also be close to each other on the face surface. Accordingly, we also compute the inter-vertex distance on the mesh as described in [34] for the isomap algorithm [35] to form a distance matrix $\mathbf{G}$ and normalized it to the [0,1] scale. Both matrices are added into an affinity matrix $\mathbf{A}$ and spectral clustering were performed on it using the method of Ng et al. [29].

In our experiment we obtained 12 compact clusters instead of 13 as reported in [34]. A possible reason for this is that we lowered the 11 forehead markers from the manual annotation before calculating the dense mesh, resulting in a missing separate forehead region. These markers were on the border of the hair region, which was not estimated correctly by the imaging hardware in the dataset.

III. MODEL FITTING

In this section we describe the dense cascade regression and the 3D model fitting process.

A. Training dataset

Automatic face alignment requires a large number of training examples of annotated images. We used our 6000 annotated meshes and from each mesh we generated 63 different views in 20 degrees yaw rotation and 15 degrees pitch rotation increments (9 yaw and 7 pitch rotations in total). Resulting in the total number of 378,000 frames. For each view we calculated the corresponding rotated 3D markers and their 2D projections with self-occlusion information. Since the 3D meshes do not contain backgrounds, we included randomly selected non-face backgrounds in the final 2D images to increase the variety. These generated and annotated images were used to train a dense, cascade regression based method, that we detail in the next subsection.

B. Dense Cascade Regression

In this section we describe the general framework of dense cascade regression for face alignment. We build on the work of Xiong and De la Torre [37]. Given an image $\mathbf{d} \in \mathbb{R}^{a\times 1}$ of a pixels, $\mathbf{d}(\mathbf{y}) \in \mathbb{R}^{b\times 1}$ indexes b markers in the image. Let $\mathbf{h}$ to be a feature extraction function (e.g. HOG, SIFT or binary features) and $\mathbf{h}(\mathbf{d}(\mathbf{y})) \in \mathbb{R}^{Fb\times 1}$ in the case of extracting features of length F. During training we will assume that the ground truth location of the b markers are known. We refer to them as $\mathbf{y}_\star$.

We used a face detector on the training images to provide an initial configuration of the markers ($\mathbf{y}_0$), which correspond

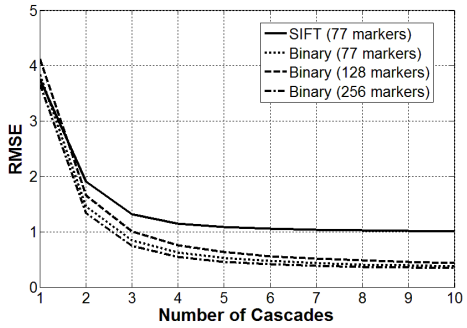

Fig. 5: Marker RMSE as a function of cascades. 1 RMSE unit correspond to 1 pixel error in all markers. The inter-ocular distance was normalized to 100 pixels.

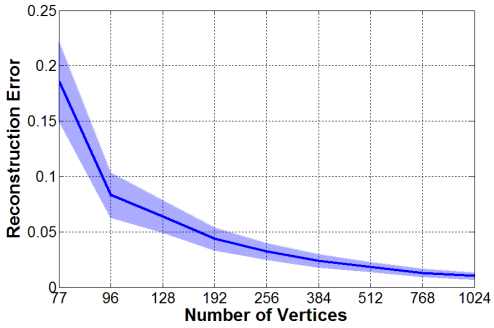

Fig. 6: The reconstruction error as a function of vertices. The solid line (shaded region) shows the mean error (standard deviation).

to the frontal projection of the 3D reference face built in Section II-D.

In this framework, face alignment can be framed as minimizing the following function over ($\Delta\mathbf{y}$):

$$f(\mathbf{y}_0+\Delta\mathbf{y}) = \|\mathbf{h}(\mathbf{d}(\mathbf{y}_0+\Delta\mathbf{y})) - \beta_\star\|_2^2 \tag{4}$$

where $\beta_\star = \mathbf{h}(\mathbf{d}(\mathbf{y}_\star))$ represents the feature values in the ground truth markers.

The feature extraction function ($\mathbf{h}$) can be highly non-linear and minimizing eq. (4) would require numerical approximations, which are computational expensive. Instead we learn a series of linear regressor matrices ($\mathbf{R}_i$), such that it produces a sequence of updates starting from $\mathbf{y}_0$ that converges to $\mathbf{y}_\star$ in the training data:

$$\Delta\mathbf{y}_i = \mathbf{R}_{i-1}\beta_{i-1} + \mathbf{b}_{i-1} \tag{5}$$

$$\mathbf{y}_i = \mathbf{y}_{i-1} + \Delta\mathbf{y}_i \rightarrow \mathbf{y}_\star \tag{6}$$

In our case, the annotation $\mathbf{y}$ consist of the projected 2D locations of the 3D markers and their corresponding visibility information:

$$\mathbf{y} = [x_1; y_1; v_1; \ldots; x_M; y_M; v_M], \tag{7}$$

where $v_i \in [0,1]$ indicates if the marker is visible ($v_i = 1$) or not ($v_i = 0$).

C. 3D Model Fitting

An iterative method was used [18] to register 3D model on the 2D landmarks. The algorithm iteratively refines the 3D shape and 3D pose until convergence, and estimates the rigid ($\mathbf{p} = \{s, \alpha, \beta, \gamma, \mathbf{t}\}$) and non-rigid transformations ($\mathbf{q}$ in the case of holistic model or $\mathbf{r}$ and $\mathbf{s}$ in the case of the part-based one).

To calculate of the non-rigid part the method requires the pseudo-inverse of the linear basis. In the holistic model $\mathbf{\Phi}$ was acquired by PCA and therefore its pseudo-inverse is its transpose. In the part-based model we have to re-calculate the pseudo-inverse of $[\mathbf{\Theta}, \mathbf{\Psi}]$.

Note that in both models, the fitting can be done even if we restrict the process to a limited set of visible landmarks ($M_{Obs} \leq M$).

In this case we just remove the rows corresponding to the occluded markers from the basis ($\mathbf{\Phi}$ or $[\mathbf{\Theta}, \mathbf{\Psi}]$) and re-calculate the pseudo-inverse for the fitting.

IV. EXPERIMENTS

We conducted a battery of experiments to evaluate the precision of 3D reconstruction and extensions to multi-view reconstruction.

A. Feature space for the cascade regression

In this experiment we evaluated SIFT [26] and localized binary features [7] for training the regression cascades. For each shape in the training set we generated 10 samples using the Monte-Carlo procedure by perturbing the model parameters of the ground truth shapes. SIFT and binary descriptors were computed on 32x32 local patches around the markers. In the case of binary features, we used a low-rank PCA [19] to learn a compressed representation of the patches. We kept 128 dimensions in the process for each marker.

We used a five-fold cross-validation to estimate the marker estimation precision, measured by the root mean squared error of marker displacements. We found that binary feature representation learned from the data outperformed hand crafted SIFT features (see Figure 5).

Binary features are a magnitude faster than SIFT, allowing more markers to be measured. We varied the number of observed vertices using binary features from 77 to 256. The effect in terms of RMSE is noticeable, but not significant. After 6-7 iterations, we observed a plateauing effect in the RMSE.

We also investigated the effect of different illumination conditions by varying the level of ambient light and adding directional light. The method was robust to these perturbations. Binary features, like SIFT features, were insensitive to photometric variation in the input images.

B. Optimal density of the model

In this experiment, we studied the reconstruction precision of the 3D model with different level of details. We identified a minimum set of vertices that are required in the model to reconstruct the 3D geometry of the face with high precision.

First we registered and tessellated the ground truth meshes according to Section II-D.

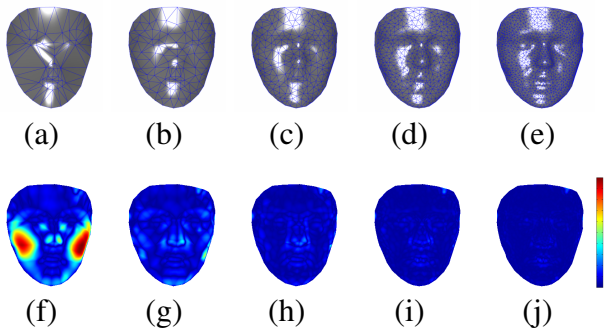

Fig. 7: Visualizing the reconstructed 3D meshes with different levels of detail. (a)-(e) Meshes consisting of 77, 128, 256, 512 and 1024 vertices, respectively. (f)-(j) The corresponding absolute depth map differences comparing with the ground truth mesh.

We rendered the reconstructed 3D shapes and their corresponding depth maps. Accurate depth maps of the ground truth meshes are also computed for comparison. The differences between the two depth maps are computed and they were summed up within the area bounded by the face outline. The final score was normalized to the area of the original face. This normalized score served as the measure for evaluating the reconstruction precision (Reconstruction Error). Since the tessellation is done in an adaptive manner, this provides an easy way to vary the number of vertices. We varied this number between 77 and 1024 on a logarithmic scale

The results are summarized in Figure 6. The original data consist of more than 30,000 vertices. The figure shows that we can precisely approximate them using around 1,000 vertices. We suspended refinement at $M = 1024$ vertices.

Figure 7 shows the different levels of detail and the corresponding absolute depth map differences comparing with the ground truth mesh.

C. Number of measurements and iterations for fitting

Two important questions are the number of vertices to measure and the number of iteration steps needed during model fitting (Section III-C). We expected that a much smaller subset of vertices would be sufficient for the fitting given the inherent 3D surface constraints.

To test this hypothesis, we kept the total number of vertices in the model fixed ($M = 1024$) and varied the number of vertices (M_{Obs}) from 77 to 1024 on a logarithmic scale. For selecting the observed vertices we used the same scheme as before: we used the first 77, 128, 256, etc. vertices from the refining process. This way we add more detail to the mesh and more constraint to the model fitting.

Another parameter is the number of iterations during the model fitting. We varied the number of iterations between 5 and 40 on a logarithmic scale. Figure 8 shows reconstruction error as a function of observed vertices and the number of iteration steps.

The figure shows that there is no further performance gain measuring more than 128 vertices. The size of the cascade regressor matrices and the fitting speed depends on

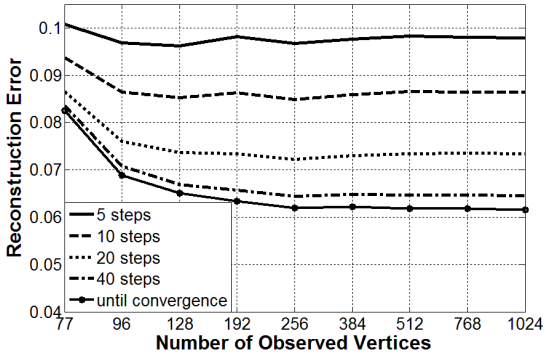

Fig. 8: The reconstruction error as a function of observed vertices M_{Obs} and the number of iteration steps using a single measurement.

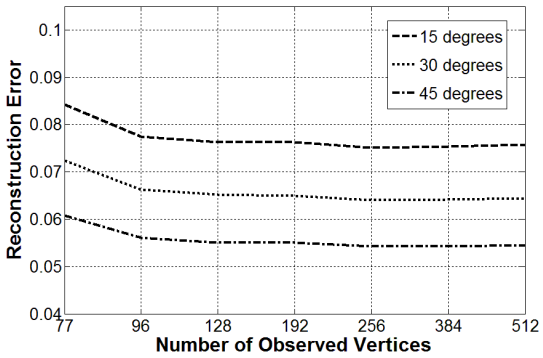

Fig. 9: The reconstruction error as a function of observed vertices M_{Obs} using two synchronized cameras that are separated by 15, 30 and 45 degrees of yaw rotations apart.

the number of observed vertices. As we seen, we can keep this number low without the loss of precision.

The number of iterations during the fitting had a significant effect on the reconstruction error. Increasing the number of iterations steps has no effect on the model size.

D. Multi-view measurements

Up until now, we used only a single frame to locate the landmarks and fit the 3D model. In this experiment, we investigated the performance gain when we have access to multiple measurements for each time-step.

Let us assume that we have a time-synchronized multi-camera setup that provides two frames at every time-step, but the exact camera locations and the camera calibration matrices are unknown.

We fixed the total number of vertices in the model ($M = 1024$) and varied the number of observed vertices (M_{Obs}) between 77 and 512 on a logarithmic scale. For selecting the observed vertices, we used the same scheme as before: we used the first 77, 128, 256, etc. vertices from the refining process. This way we add more detail to the mesh and more constraint to the model fitting.

Figure 9 shows the reconstruction error as a function of observed vertices M_{Obs} using two synchronized cameras that are separated by 15, 30 and 45 degrees of yaw rotations apart. The number of iterations was fixed in 10 steps.

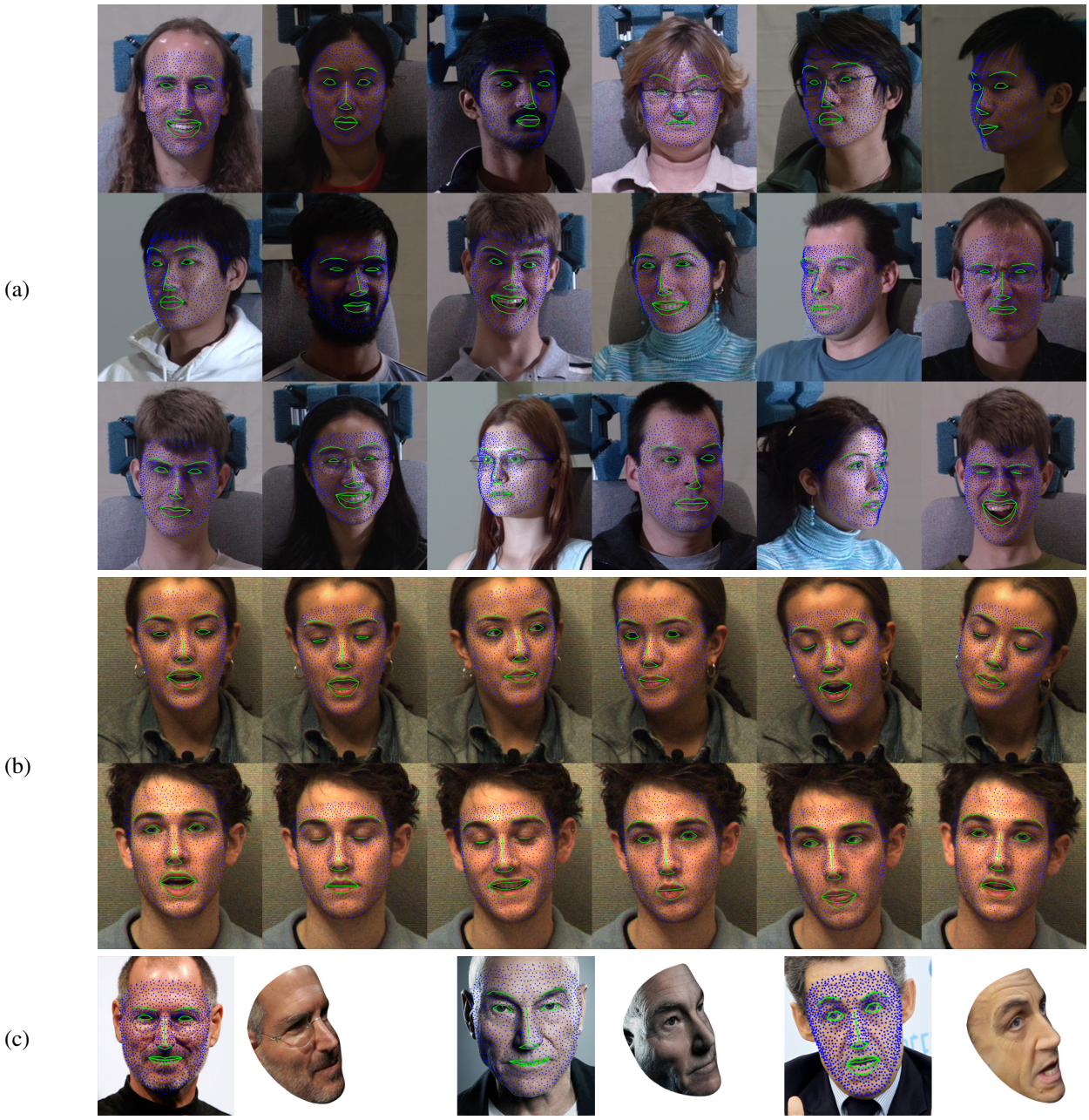

Fig. 10: Examples from (a) Multi-PIE with various illuminations and head poses, (b) RU-FACS tracking sequences and (c) celebrities with profile view renders using the high-resolution 3D shape. The contours of key facial parts are highlighted in green for display purpose.

The figure shows that larger viewpoint-angles yielded lower reconstruction error.

V. CONCLUSIONS AND FUTURE WORK

Real-time, dense 3D face alignment is a challenging problem for computer vision. In the last few years 2D face alignment has reached a mature state with the emergence of discriminative shape regression methods. On the other hand, relatively neglected is the application of cascade regression in dense 3D face alignment. To afford real-time, person-independent 3D registration from 2D video, we developed a 3D cascade regression approach in which facial landmarks remain invariant across pose over a range of approximately 60 degrees. From a single 2D image of a person's face, a dense 3D shape is registered in real time for each frame.

Our present method has two specific features.

Consistent 3D Annotation
: Landmark annotation in 2D is confounded by inconsistency in the landmark positions across different poses and self-occlusions. To avoid inconsistency in landmark positions, we annotate the meshes themselves completely in 3D by selecting a dense set of 3D points.

Sparse Measurements for Dense Registration
: By using only a sparse number of measurements,

the 3D reconstruction can be carried out with high precision. This results in real-time performance and a manageable model size. Our MATLAB implementation runs at 50 fps using a single core of an i7 processor.

We validated the method in a series of experiments that evaluate its precision of 3D reconstruction and extension to multi-view reconstruction. Experimental findings strongly support the validity of real-time, 3D registration and reconstruction from 2D video.

REFERENCES

[1] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, July 2003.

[2] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1859–1866, June 2014.

[3] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633, Nov 2011.

[4] T. Baltrusaitis, P. Robinson, and L. Morency. 3d constrained local model for rigid and non-rigid facial tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2610–2617. IEEE, 2012.

[5] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[6] X. Burgos-Artizzu, P. Perona, and P. Dollar. Robust face landmark estimation under occlusion. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1513–1520, Dec 2013.

[7] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. Brief: Computing a local binary descriptor very fast. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1281–1298, July 2012.

[8] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph.*, 33(4):43:1–43:10, July 2014.

[9] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3d shape regression for real-time facial animation. *ACM Trans. Graph.*, 32(4):41:1–41:10, July 2013.

[10] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2887–2894, June 2012.

[11] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681–685, Jun 2001.

[12] D. Cristinacce and T. Cootes. Automatic feature localisation with constrained local models. *Pattern Recogn.*, 41(10):3054–3067, Oct. 2008.

[13] M. Dantone, J. Gall, G. Fanelli, and L. V. Gool. Real-time facial feature detection using conditional regression forests. In *CVPR*, 2012.

[14] Dimensional Imaging Ltd. DI3D. `http://www.di3d.com`.

[15] M. Dimitrijevic, S. Ilic, and P. Fua. Accurate face models from uncalibrated and ill-lit video sequences. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–1034–II–1041 Vol.2, June 2004.

[16] P. Dollar, P. Welinder, and P. Perona. Cascaded pose regression. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1078–1085, June 2010.

[17] P. Ekman, W. Friesen, and J. Hager. *Facial Action Coding System (FACS): Manual*. Salt Lake City (USA): A Human Face, 2002.

[18] L. Gu and T. Kanade. 3d alignment of face in a single image. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1305–1312, June 2006.

[19] N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[20] T. Hassner. Viewing real-world faces in 3d. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3607–3614, Dec 2013.

[21] K. V. M. I. L. Dryden. *Statistical Shape Analysis*. John Wiley & Sons, 1998.

[22] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1867–1874, June 2014.

[23] I. Kemelmacher-Shlizerman and R. Basri. 3d face reconstruction from a single image using a single reference face shape. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(2):394–405, Feb 2011.

[24] L. Kobbelt. Sqrt(3)-subdivision. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 103–112, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[25] B. Levy. Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pages 13–13, June 2006.

[26] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.

[27] B. Martinez, M. Valstar, X. Binefa, and M. Pantic. Local evidence aggregation for regression-based facial point detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(5):1149–1163, May 2013.

[28] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[29] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 849–856. MIT Press, 2001.

[30] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1685–1692, June 2014.

[31] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

[32] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3476–3483, June 2013.

[33] S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. Seitz. Total moving face reconstruction. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision, ECCV 2014*, volume 8692 of *Lecture Notes in Computer Science*, pages 796–812. Springer International Publishing, 2014.

[34] J. R. Tena, F. De la Torre, and I. Matthews. Interactive region-based linear 3d face models. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 76:1–76:10, New York, NY, USA, 2011. ACM.

[35] J. B. Tenenbaum, V. De Silva, and J. C. Langfold. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[36] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2729–2736, June 2010.

[37] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 532–539, June 2013.

[38] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1–6, Sept 2008.

[39] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision, ECCV 2014*, volume 8690 of *Lecture Notes in Computer Science*, pages 1–16. Springer International Publishing, 2014.

[40] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692 – 706, 2014. Best of Automatic Face and Gesture Recognition 2013.

[41] Z. Zhang, Z. Liu, D. Adler, M. F. Cohen, E. Hanson, and Y. Shan. Robust and rapid generation of animated faces from video images: A model-based modeling approach. *Int. J. Comput. Vision*, 58(2):93–119, July 2004.