

Deep Energies for Estimating Three-Dimensional Facial Pose and Expression

Michael Bao^{1,2} Jane Wu¹

¹Stanford University

¹{mikebao, janehwu, yaodavid, rfedkiw}@stanford.edu

Xinwei Yao¹ Ronald Fedkiw^{1,2}

²Industrial Light & Magic

Abstract

While much progress has been made in capturing high-quality facial performances using motion capture markers and shape-from-shading, high-end systems typically also rely on rotoscope curves hand-drawn on the image. These curves are subjective and difficult to draw consistently; moreover, ad-hoc procedural methods are required for generating matching rotoscope curves on synthetic renders embedded in the optimization used to determine three-dimensional facial pose and expression. We propose an alternative approach whereby these curves and other keypoints are detected automatically on both the image and the synthetic renders using trained neural networks, eliminating artist subjectivity and the ad-hoc procedures meant to mimic it. More generally, we propose using machine learning networks to implicitly define deep energies which when minimized using classical optimization techniques lead to three-dimensional facial pose and expression estimation.

1. Introduction

For high-end face performance capture, either motion capture markers [4] or markerless techniques such as shape-from-shading [1] or optical flow [64] are typically used; however, these methods are generally unable to capture the intricacies of the performance, especially around the lips. To obtain high-end results, artists hand-draw rotoscope curves on the captured image; then, a variety of techniques are used to construct similar curves on the synthetic render of the estimated pose and to determine correspondences between the hand-drawn and synthetically generated curves. The simplest such approach would be to use a pre-defined contour on the three-dimensional face model, clip it for occlusions, and create correspondences in a length proportional way; although this provides some consistency to the curves generated on the synthetic render, it is quite difficult for an artist to emulate these curves. Thus, practical systems implement a number of ad-hoc methods in order to match the artist’s more subjective interpretation. The inability to embed the artist’s subjectivity into the optimiza-

tion loop and onto the synthetic render coupled with the artist’s inability to faithfully reproduce procedurally generated curves leaves a gaping chasm in the uncanny valley.

Although one might debate what works best in order to align a three-dimensional virtual model with an image, it is clearly the case that a consistent metric should be applied to evaluate whether the synthetic render and image are aligned. This motivates the employment of a machine learning algorithm to draw the rotoscope curves on both the captured image and the synthetic render, hoping that accurately representing the real world pose would lead to a negligible difference between the two curves. Although one might reasonably expect that the differences between real and synthetic camera behavior, albedo, lighting, etc. may lead to different rotoscope curves being generated by the deep learning algorithm, GAN-like approaches [22, 36] could be used to rectify such issues by training a network to draw the curves such that a discriminator cannot tell which curves were generated on real images versus synthetic renders.

Recent advancements in using deep neural networks to detect face landmarks (see e.g. [7]) make using machine learning to detect not only the lip curves but also other facial landmarks on both the captured image and the synthetic render an attractive option. A traditional optimization approach can then be used to minimize the difference between the outputs from the captured image and the synthetic render. This is feasible as long as one can backpropagate through the network to obtain the Jacobian of the output with respect to the synthetic render. Of course, this assumes that the synthetic render is fully differentiable with respect to the face pose parameters, and we use OpenDR [39] to satisfy this latter requirement; however, we note that any differentiable renderer can be used (e.g. [35]).

This approach is attractive as it replaces human subjectivity with a consistent, albeit sometimes consistently wrong, network to evaluate the semantic difference between two images. Furthermore, as networks improve their ability to detect facial features/descriptors on a wider variety of poses and lighting conditions, our approach will also benefit. More generally, we propose to use machine learning networks to embed various subjective, but consistent, “evalu-

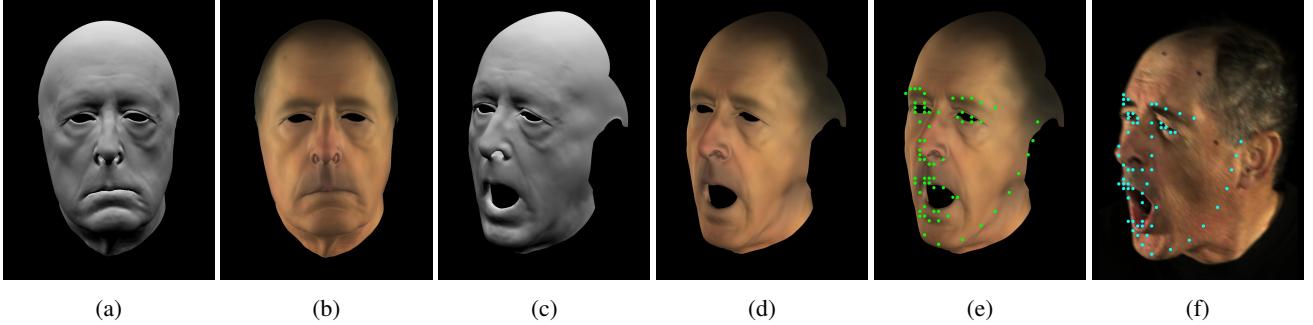


Figure 1. An overview of our approach: we use our three-dimensional face model of the actor (a) and estimate the albedo and spherical harmonics lighting in a neutral pose (b). Then, we can deform the face model into a variety of poses by changing the rigid parameters θ and t and blendshape parameters w (c), and generate synthetic images by rendering the face model in those poses (d). We feed that synthetic render through the network to produce a set of outputs (e), which are then compared to the outputs produced by the same network when feeding it the captured image (f).

ations” into classical optimization approaches that estimate facial pose and expression.

2. Related Work

Rotoscope Lip/Mouth Curves: Motion capture markers are commonly used to estimate facial pose and expression for a facial performance [13, 42, 56]. The corresponding points on the triangulated facial mesh are either selected by hand or found by projecting two-dimensional markers from the image to the three-dimensional mesh when the face is in a neutral pose [4]. Rotoscope curves are typically used to refine the results around the lips to produce higher fidelity [4, 14]. For view-independent curves such as the outer lip, the corresponding contour on the mesh can be pre-defined [4]; however, for view-dependent/silhouette curves such as the inner lip, the corresponding contour must be defined based on the current camera viewpoint and updated iteratively [4, 14].

Face Alignment: The earliest approaches to regression-based face alignment trained a cascade of regressors to detect face landmarks [9, 10, 18, 29, 70]. More recently, deep convolutional neural networks (CNNs) have been used for both 2D and 3D facial landmark detection from 2D images [28, 65]. These methods are generally classified into coordinate regression models [25, 28, 60, 66], where a direct mapping is learned between the image and the landmark coordinates, and heatmap regression models [7, 12, 67], where prediction heatmaps are learned for each landmark. Heatmap-based architectures are generally derived from stacked hourglass [7, 12, 24, 45] or convolutional pose machine [62] architectures used for human body pose estimation. Pixel coordinates can be obtained from the heatmaps by applying the argmax operation; however, [15, 58] use soft-argmax to achieve end-to-end differentiability. A more comprehensive overview of face alignment methods can be found in [26].

Optical Flow: Optical flow has had a long successful history started in part by [5]. Variational methods such as the Lucas-Kanade method [41] and the Brox method [6] are commonly used. Other correspondence finding strategies such as EpicFlow [49] and FlowFields [2] have also been used successfully. We focus instead on trained deep networks. End-to-end methods for learning optical flow using deep networks were first proposed by [17] and later refined in [23]. Other methods include DeepFlow [63], etc. use deep networks to detect correspondences. These methods are generally evaluated on the Middlebury dataset [3], the KITTI dataset [20], and the MPI Sintel dataset [8].

Faces and Networks: Neural networks have been used for various other face image analysis tasks such as gender determination [21] and face detection [50]. More recently, deep CNNs have been used to improve face detection results especially in uncontrolled environments and with more extreme poses [34, 68]. Additionally, CNNs have been employed for face segmentation [37, 46, 52], facial pose and reflectance acquisition [32, 55], and face recognition [53, 59].

Using deep networks such as VGG-16 [57] for losses has been shown to be effective for training other deep networks for tasks such as style transfer and super-resolution [27]. Such techniques have also been used for image generation [16] and face swapping [31]. Furthermore, deep networks have been used in energies for traditional optimization problems for style transfer [19], texture synthesis [54], and image generation [43, 61]. While [19, 54] use the L-BFGS [69] method to minimize the optimization problem, [43, 61] use gradient descent methods [51].

3. Overview

In this paper, we advocate for a general strategy that uses classical optimization where the energy to be minimized is based on metrics ascertained from deep learning neural net-

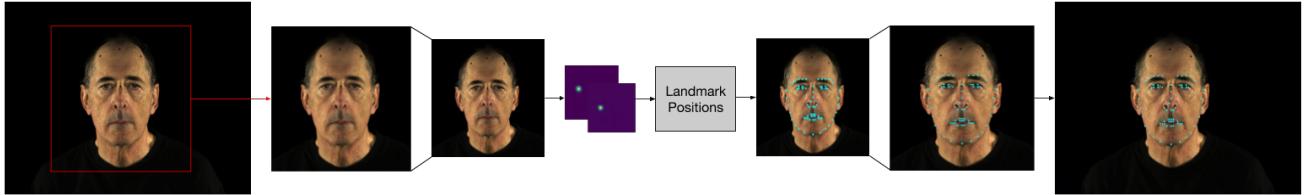


Figure 2. A visual overview of our approach applied to facial landmark detection. We pass the full resolution image through a facial detector and crop the face out of the image. This crop is then resized to pass through the neural network which outputs, in this case, heatmaps for every landmark. These heatmaps are processed using a soft-argmax operation to get facial landmark coordinates on the cropped and resized image. These positions are then transformed back onto the full resolution image before being used as part of the objective function. An identical process is performed for the synthetic render.

works. In particular, this removes the rotoscope artist and the ad-hoc contour drawing procedures meant to match the artist’s work, or vice versa, from the pipeline. This pins advancements in three-dimensional facial pose and expression estimation to those being made in machine learning, which are advancing at a fast pace. Generally, we take the following approach: First, we estimate an initial rigid alignment of the three-dimensional face model to the two-dimensional image using a facial alignment network. Then, we estimate an initial guess for the jaw and mouth expression using the same network. Finally, we temporally refine the results and insert/repair failed frames (if/when necessary) using an optical flow network.

We use a blendshape model hybridized with linear blend skinning for a six degree of freedom jaw joint [33]; let w denote the parameters that drive the face triangulated surface $x(w)$. The resulting surface has a rigid frame given by Euler angles θ , rotation matrix $R(\theta)$, and a translation t such that the final vertex positions are

$$x_R(\theta, t, w) = R(\theta)x(w) + t. \quad (1)$$

We note that other geometry such as the teeth can be trivially handled by Equation 1 as well. The geometry x_R is rendered using OpenDR [39] obtaining a rendered image $F(x_R)$. As a precomputation, we estimate the face’s albedo and nine coefficients for a spherical harmonics light [48] on a frame where the face is close to neutral; however, we note that a texture captured using a light stage [11] (for example) would potentially work just as well if not better. Then, our goal is to determine the parameters θ , t , and w that best match a given captured image F^* .

Both the pixels of captured image F^* and the pixels of the rendered image $F(x_R)$ are fed through the same deep network to get two sets of landmark positions $N(F^*)$ and $N(F)$. See Figure 1. We use the L2 norm of the difference between them

$$\|N(F^*) - N(F(x_R(\theta, t, w)))\|_2 \quad (2)$$

as the objective function to minimize via nonlinear least squares, which is solved using the Dogleg method [40] as

implemented by Chumpy [38]. This requires computing the Jacobian via the chain rule of the energy function; $\partial N / \partial F$, $\partial F / \partial x_R$, and $\partial x_R / \partial p$ where p is one of θ , t , and w all need to be evaluated. We use OpenDR to compute $\partial F / \partial x_R$, and Equation 1 yields $\partial x_R / \partial p$. $\partial N / \partial F$ is computed by back-propagating through the trained network using one’s deep learning library of choice; in this paper, we use PyTorch [47]. Note that for computational efficiency, we do not compute $\partial N / \partial F$ explicitly, but instead compute the Jacobian of Equation 2 with respect to the rendered image pixels output by F instead.

4. Rigid Alignment

We first solve for the initial estimate of the rigid alignment of the face, *i.e.* θ and t using the pre-trained 3D-FAN network [7]. Note that 3D-FAN, and most other facial alignment networks, requires taking in a cropped and resized image of the face as input; we denote these two operations as C and S respectively. The cropping function requires the bounding box output of a face detector D ; we use the CNN-based face detector implemented by Dlib [30]. The resize function resizes the crop to a resolution of 256×256 to feed into the network. The final image passed to the network is thus $S(C(F(x_R)), D(F(x_R))), D(F(x_R))$ where we note that both the crop and resize functions depend on the output of the face detector; however, aggressively assuming that $\partial D / \partial F = 0$ did not impede our ability to estimate a reasonable facial pose. Given D , C is merely a subset of the pixels of F so $\partial C / \partial p = \partial F / \partial p$ for all the pixels within the detected bounding box and $\partial C / \partial p = 0$ for all pixels outside. S resizes the crop using bilinear interpolation so $\partial S / \partial C$ can be computed using the size of the detected bounding box.

3D-FAN outputs a tensor of size $68 \times 64 \times 64$, *i.e.* each of the 68 landmarks has a 64×64 heatmap specifying the likelihood of a particular pixel containing that landmark. While one might difference the heatmaps directly, it is unlikely that this would sufficiently capture correspondences. Instead, we follow the approach of [15, 58] and apply a dif-

ferentiable soft-argmax function to the heatmaps obtaining pixel coordinates for each of the 68 landmarks. That is, given the marker position m_i computed using the argmax function on heatmap H_i , we use a 3×3 patch of pixels M_i around m_i to compute the soft-argmax position as

$$\hat{m}_i = \frac{\sum_{m \in M_i} m e^{\beta H_i(m)}}{\sum_{m \in M_i} e^{\beta H_i(m)}} \quad (3)$$

where $\beta = 50$ is set experimentally and $H_i(m)$ returns the heatmap value at a pixel coordinate m . We found that using a small patch around the argmax landmark positions gives better results than running the soft-argmax operation on the entire heatmap.

The soft-argmax function returns an image coordinate on the 64×64 image, and these image coordinates need to be remapped to the full resolution image to capture translation between the synthetic face render and the captured image. Thus, we apply inverse rescale S_m^{-1} and crop operations C_m^{-1} , i.e. $\tilde{m}_i = C_m^{-1}(S_m^{-1}(4\hat{m}_i, D), D)$. The multiplication by 4 rescales from the 64×64 heatmap to the original 256×256 . To summarize, we treat the 68 \tilde{m}_i as the output of N , and Equation 2 measures the L2 distance between the 68 \tilde{m}_i on the captured image and the corresponding \tilde{m}_i on the synthetic render. We once again assume $\partial D / \partial F = 0$. $\partial C_m^{-1} / \partial S_m^{-1}$ is the identity matrix, and $\partial S_m^{-1} / \partial \tilde{m}_i$ contains the scalar multipliers to resize the image from 256×256 to the original cropped size. We stress that this entire process is end-to-end differentiable. See Figure 2.

5. Expression Estimation

After the rigid alignment determines θ and t , we solve for an initial estimate of the mouth and jaw blendshape parameters (a subset of w). Generally, one would use hand-drawn rotoscope curves around the lips to accomplish this as discussed in Section 2; however, given the multitude of problems associated with this method as discussed in Section 1, we instead turn to deep networks to accomplish the same goal. We use 3D-FAN in the same manner as discussed in Section 4 to solve for a subset of the blendshape weights w keeping the rigid parameters θ and t fixed. It is sometimes beneficial or even preferred to also allow θ and t to be modified somewhat at this stage, although a prior energy term that penalizes these deviations from the values computed during the rigid alignment stage is often useful.

We note that the ideal solution would be to instead create new network architectures and train new models that are designed specifically for the purpose of detecting lip/mouth contours, especially since the 64×64 heatmaps generated by 3D-FAN are generally too low-resolution to detect fine mouth movements such as when the lips pucker. However, since our goal in this paper is to show how to leverage existing architectures and pre-trained networks especially so

one can benefit from the plethora of existing literature, for now, we bootstrap the mouth and jaw estimation using the existing facial landmark detection in 3D-FAN.

6. Optical Flow for Missing Frames

The face detector used in Section 4 can sometimes fail, e.g. on our test sequence, the Dlib’s HOG-based detector failed on 20 frames while Dlib’s CNN-based detector succeeded on all frames. We thus propose using optical flow networks to infer the rigid and blendshape parameters for failed frames by “flowing” these parameters from surrounding frames where the face detector succeeded. This is accomplished by assuming that the optical flow of the synthetic render from one frame to the next should be identical to the corresponding optical flow of the captured image. That is, given two synthetic renders F_1 and F_2 and two captured images F_1^* and F_2^* , we can compute two optical flow fields $N(F_1, F_2)$ and $N(F_1^*, F_2^*)$ using FlowNet2 [23]. We resize the synthetic renders and captured images to a resolution of 512×512 before feeding them through the optical flow network. Assuming that F_2^* is the image the face detector failed on, we solve for the parameters p_2 of F_2 starting with an initial guess p_1 , the parameters of F_1 , by minimizing the L2 difference between the flow field vectors $\|N(F_1^*, F_2^*) - N(F_1, F_2)\|_2$. $\partial N / \partial F_2$ can be computed by back-propagating through the network.

7. Temporal Refinement

Since we solve for the rigid alignment and expression for all captured images in parallel, adjacent frames may produce visually disjointed results either because of noisy facial landmarks detected by 3D-FAN or due to the non-linear optimization converging to different local minima. Thus, we also use optical flow to refine temporal inconsistencies between adjacent frames. We adopt a method that can be run in parallel. Given three sequentially captured images F_1^* , F_2^* , and F_3^* , we compute two optical flow fields $N(F_1^*, F_2^*)$ and $N(F_2^*, F_3^*)$. Similarly, we can compute $N(F_1, F_2)$ and $N(F_2, F_3)$. Then, we solve for the parameters p_2 of F_2 by minimizing the sum of two L2 norms $\|N(F_1^*, F_2^*) - N(F_1, F_2)\|_2$ and $\|N(F_2^*, F_3^*) - N(F_2, F_3)\|_2$. The details for computing the Jacobian follow that in Section 6. Optionally, one may also wish to add a prior penalizing the parameters p_2 from deviating too far from their initial value. Here, step k of smoothing to obtain a new set of parameters p_i^k uses the parameters from the last step $p_{i \pm 1}^{k-1}$; however, one could also use the updated parameter values $p_{i \pm 1}^k$ whenever available in a Gauss-Seidel style approach.

Alternatively, one could adopt a self-smoothing approach by ignoring the capture image’s optical flow and solving for the parameters p_2 that minimize $\|N(F_1, F_2) -$

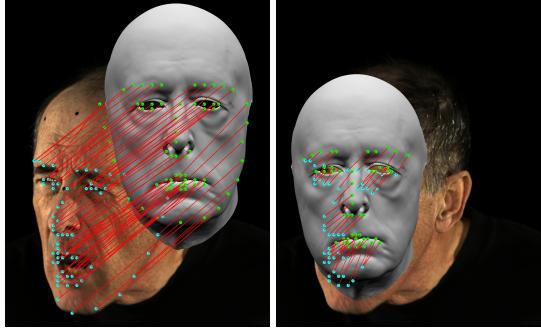


Figure 3. The green dots denote the landmarks detected on the synthetic render, the teal dots denote the landmarks detected on the captured image, and the red lines show correspondences. Left: The initial state where the face is front facing and centered (note the figure is cropped) in the image plane. Right: The initial translation stage in the rigid alignment step roughly aligns the synthetic render of the face to the face in the captured image. (Note that we display the synthetic render without the estimated albedo for clarity, but the network sees the version with the albedo as in Figures 1d and 1f, not 1c).

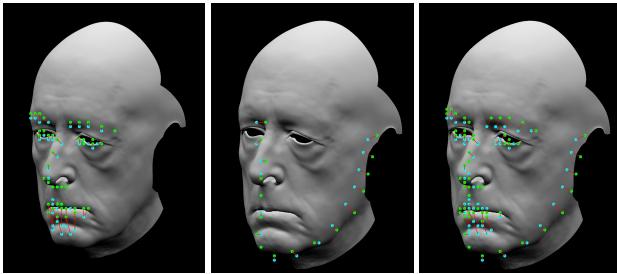


Figure 4. From left to right: the result after solving for θ and t using the non-jaw markers, using only the jaw markers, and using all the markers.

$N(F_2, F_3)\|_2$. Such an approach in effect minimizes the second derivative of the motion of the head in the image plane, causing any sudden motions to be smoothed out; however, since the energy function contains no knowledge of the data being targeted, it is possible for such a smoothing operation to cause the model to deviate from the captured image.

While we focus on exploring deep learning based techniques, more traditional smoothing/interpolation techniques can also be applied in place of or in addition to the proposed optical flow approaches. Such methods include: spline fitting the rigid parameters and blendshape weights, smoothing the detected landmarks/bounding boxes on the captured images as a preprocess, smoothing each frame’s parameters using the adjacent frame’s estimations, etc.

8. Results

We estimate the facial pose and expression on a moderately challenging performance captured by a single ARRI

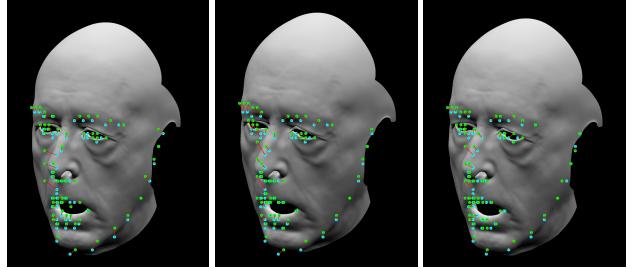


Figure 5. From left to right: the results after solving for only the jaw open blendshape, all jaw-related blendshapes, and all jaw and mouth related blendshapes.

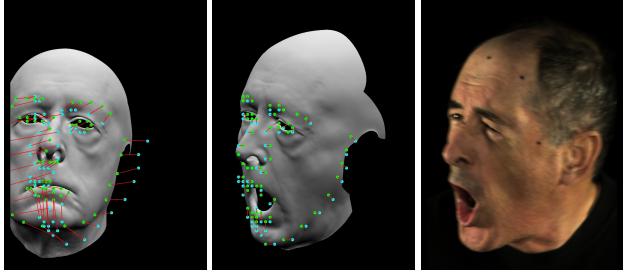


Figure 6. From left to right: the rigid alignment overfit to the mouth markers, the correction by expression estimation, and the target captured image. The rigid alignment overfits to the non-jaw contour markers during the initial translation and rotation steps; however, this is corrected during expression estimation when the jaw is allowed to open.

Alexa XT Studio running at 24 frames-per-second with an 180 degree shutter angle at ISO 800 where numerous captured images exhibit motion blur. These images are captured at a resolution of 2880×2160 , but we downsample them to 720×540 before feeding them through our pipeline. We assume that the camera intrinsics and extrinsics have been pre-calibrated, the captured images have been undistorted, and that the face model described in Equation 1 has already been created. Furthermore, we assume that the face’s rigid transform has been set such that the rendered face is initially visible and forward-facing in all the captured viewpoints.

8.1. Rigid Alignment

We estimate the rigid alignment (*i.e.* θ and t) of the face using 3D-FAN. We use an energy $E_1 = W(N(F) - N(F^*))$ where N are the image space coordinates of the facial landmarks as described in Section 4 and W is a per-landmark weighting matrix. Furthermore, we use an edge-preserving energy $E_2 = \sum_i (\tilde{m}_i^{F^*} - \tilde{m}_{i-1}^{F^*}) - (\tilde{m}_i^F - \tilde{m}_{i-1}^F)$ where $\tilde{m}_i^{F^*}$ are the landmark positions on the captured image and \tilde{m}_i^F are the landmark positions on the synthetic renders to ensure that the face does not erroneously grow/shrink in projected size as it moves towards the target landmarks, which may prevent the face detector from

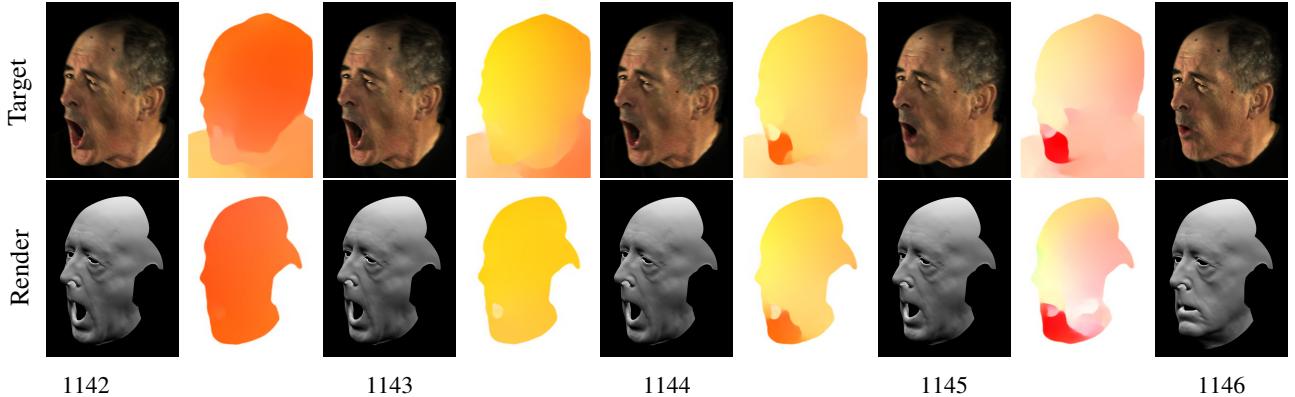


Figure 7. We use optical flow to infill frames where the face detector fails. Starting from frame 1142, the optimization moves the head to match the optical flow of the synthetic render to the optical flow of the captured image. After solving for frame 1145, we perform another round of optimization except that we start from frame 1146 instead; this way, each frame will capture optical flow information from both anchor frames. Using optical flow allows the mouth to stay open longer (*e.g.* in frame 1144) than what one would obtain using simple interpolation.

working.

First, we only solve for t using all the landmarks except for those around the jaw to bring the initial state of the face into the general area of the face on the captured image. See Figure 3. We prevent the optimization from overfitting to the landmarks by limiting the maximum number of iterations. Next, we solve for both θ and t in three steps: using the non-jaw markers, using only the jaw markers, and using all markers. We perform these steps in stages as we generally found the non-jaw markers to be more reliable and use them to guide the face model to the approximate location before trying to fit to all existing markers. See Figure 4.

8.2. Expression Estimation

We run a similar multi-stage process to estimate facial expression using the detected 3D-FAN landmarks. We use the same energy term E_1 as Section 8.1, but also introduce L2 regularization on the blendshape weights $E_2 = \lambda w$ with $\lambda = 1 \times 10^2$ set experimentally. In the first stage, we weight the landmarks around the mouth and lips more heavily and estimate only the jaw open parameter along with

the rigid alignment. The next stage estimates all available jaw-related blendshape parameters using the same set of landmarks. The final stage estimates all available jaw and mouth-related blendshapes as well as the rigid alignment using all available landmarks. See Figure 5. This process will also generally correct any overfitting introduced during the rigid alignment due to not being able to fully match the markers along the mouth. See Figure 6.

Our approach naturally depends on the robustness of 3D-FAN’s landmark detection on both the captured images and synthetic renders. As seen in Figure 8, the optimization will try to target the erroneous markers producing inaccurate θ , t , and w which overfit to the markers. Such frames should be considered a failure case and thus require using the optical flow approach described in Section 6 for infill. Alternatively, one could manually modify the multi-stage process for rigid alignment and expression estimation to remove the erroneous markers around the jaw; however, such an approach may then overfit to the potentially inaccurate mouth markers. We note that such concerns will gradually become less prominent as these networks improve.

8.3. Optical Flow Infill

Consider, for example, Figure 7 where frames 1142 and 1146 were solved for successfully and we wish to fill frames 1143, 1144, and 1145. We visualize the optical flow fields using the coloring scheme of [3]. We adopt our proposed approach from Section 6 whereby the parameters of frames 1143, 1144, and 1145 are first solved for sequentially starting from frame 1142. Then, the frames are solved again in reverse order starting from frame 1146. This back-and-forth process which can be repeated multiple times ensures that the infilled frames at the end of the sequence have not accumulated so much error that they no longer match the other known frame.

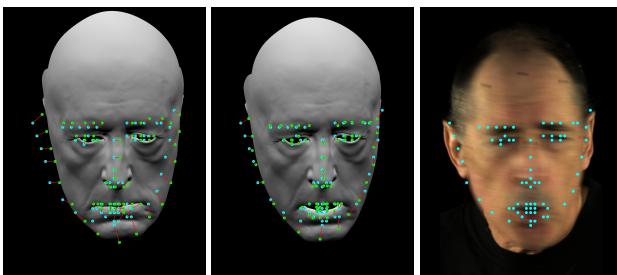


Figure 8. From left to right: the result after rigid alignment, after expression estimation, and the captured image. Erroneous markers such as those around the jaw cause the optimization to land in an inaccurate local minima.

Using optical flow information is preferable to using simple interpolation as it is able to more accurately capture any nonlinear motion in the captured images (*e.g.* the mouth staying open and then suddenly closing). We compare the results of our approach of using optical flow to using linear interpolation for t and w and spherical linear interpolation for θ in Figure 9.

8.4. Multi-Camera

Our approach can trivially be extended to multiple calibrated camera viewpoints as it only entails adding another duplicate set of energy terms to the nonlinear least squares objective function. We demonstrate the effectiveness of this approach by applying our approach from Sections 8.1 and 8.2 to the same performance captured using an identical ARRI Alexa XT Studio from another viewpoint. See Figure 10.

We also compare the rigid alignment estimated by our automatic method to the rigid alignment created by a skilled matchmove artist for the same performance. The manual rigid alignment was performed by tracking the painted black dots on the face along with other manually tracked facial features. In comparison, our rigid alignment was done using only the markers detected by 3D-FAN on both the captured images and the synthetic renders. See Figure 11. Our approach using only features detected by 3D-FAN produces visually comparable results. In Figure 13, we assume the manually done rigid alignment is the “ground truth” and quantitatively evaluate the rigid alignment computed by the monocular and stereo solves. Both the monocular and stereo solves are able to recover similar rotation parameters, and the stereo solve is able to much more accurately determine the rigid translation. We note, however, that it is unlikely that the manually done rigid alignment can be con-

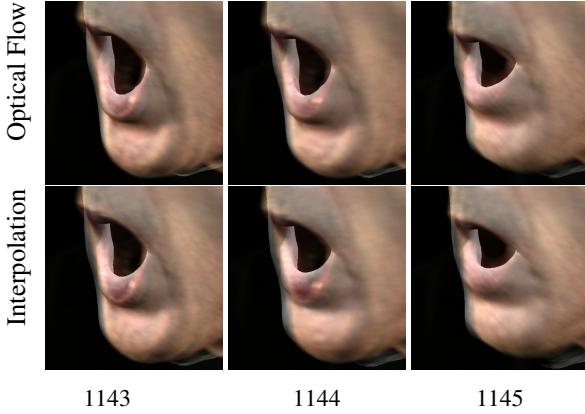


Figure 9. Using optical flow to interpolate missing frames produces results that better match the plate than simple linear interpolation. Notice how the lips and the face boundary match better in the optical flow results (particularly in frame 1144) than in the simple interpolation results.

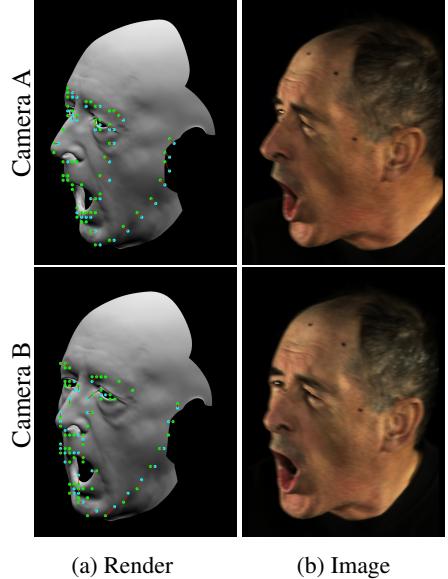


Figure 10. We trivially extend Sections 8.1 and 8.2 to handle landmarks from two cameras by simply combining the objective functions.

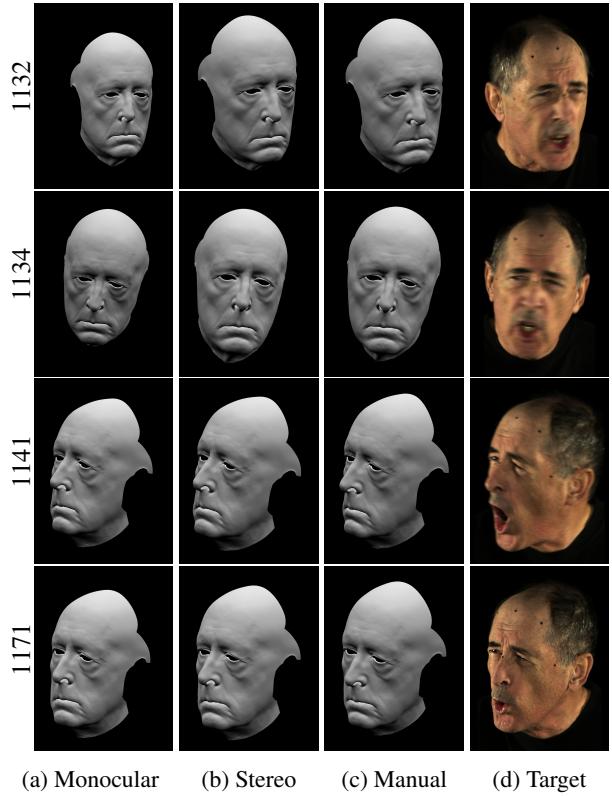


Figure 11. A comparison of the rigid alignment computed manually by a skilled artist versus the rigid alignment computed by our approach in both the monocular and stereo case.

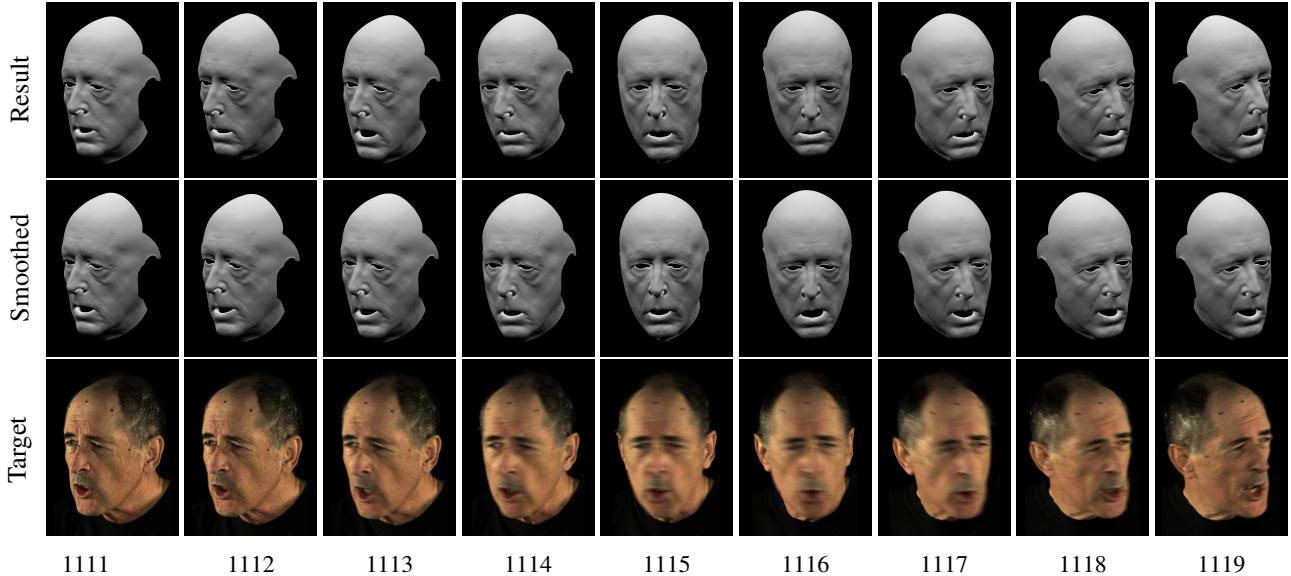


Figure 12. Smoothing the performance using the captured images’ optical flow fields between adjacent frames produces more temporally consistent results. Similar to the optical flow fill-in stage, any errors in the initial landmark estimation will propagate into the optical flow solve, *e.g.* frame 1116 where the jaw opens in an attempt to smooth the rigid of the face.

sidered “ground truth” as it more than likely contains errors as well.

8.5. Temporal Refinement

As seen in the supplementary video, the facial pose and expression estimations are generally temporally inconsistent. We adopt our proposed approach from Section 7. This attempts to mimic the captured temporal performance which not only helps to better match the synthetic render to the captured image but also introduces temporal consistency between renders. While this is theoretically susceptible to noise in the optical flow field, we did not find this to be a problem. See Figure 12. We explore additional methods of performing temporal refinement in the supplementary ma-

terial.

9. Conclusion and Future Work

We have proposed and demonstrated the efficacy of a fully automatic pipeline for estimating facial pose and expression using pre-trained deep networks as the objective functions in traditional nonlinear optimization. Such an approach is advantageous as it removes the subjectivity and inconsistency of the artist. Our approach heavily depends upon the robustness of the face detector and the facial alignment networks, and any failures in those cause the optimization to fail. Currently, we use optical flow to fix such problematic frames, and we leave exploring methods to automatically avoid problematic areas of the search space for future work. Furthermore, as the quality of these networks improve, our proposed approach would similarly benefit, leading to higher-fidelity results. While we have only explored using pre-trained facial alignment and optical flow networks, using other types of networks (*e.g.* face segmentation, face recognition, etc.) and using networks trained specifically on the vast repository of data from decades of visual effects work are exciting avenues for future work.

Acknowledgements

Research supported in part by ONR N00014-13-1-0346, ONR N00014-17-1-2174, ARL AHPCRC W911NF-07-0027, and generous gifts from Amazon and Toyota. In addition, we would like to thank both Reza and Behzad at ONR for supporting our efforts into computer vision and machine learning, as well as Cary Phillips and Industrial Light &

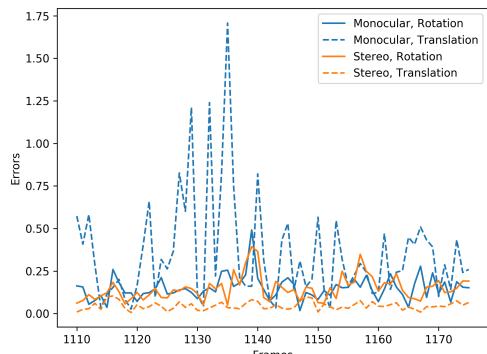


Figure 13. Assuming the manually done rigid alignment is the “ground truth,” we measure the errors for rigid parameters for the monocular and stereo case.

Magic for supporting our efforts into facial performance capture. M.B. was supported in part by The VMWare Fellowship in Honor of Ole Agesen. J.W. was supported in part by the Stanford School of Engineering Fellowship. We would also like to thank Paul Huston for his acting.

A. Temporal Smoothing Alternatives

Figure 14 (third row) shows the results obtained by matching the synthetic render’s optical flow to the captured image’s optical flow (denoted plate flow in Figure 14). Although this generally produces accurate results when looking at each frame in isolation, adjacent frames may still obtain visually disjoint results (see the accompanying video). Thus, we explore additional temporal smoothing methods.

We first explore temporally smoothing the parameters (θ , t , and w) by computing a weighted average over a three frame window centered at every frame. We weigh the current frame more heavily and use the method of [44] to average the rigid rotation parameters. While this approach produces temporally smooth parameters, it generally causes the synthetic render to no longer match the captured image. This inaccuracy is demonstrated in Figure 14 (top row, denoted as averaging) and is especially apparent around the nose (frames 1147 and 1148) and around the lower right cheek (frame 1150).

One could also carry out averaging using an optical flow network. This can be accomplished by finding the parameters p_2 that minimize the difference in optical flow fields between the current frame’s synthetic render and the adjacent frames’ synthetic renders, *i.e.* $\|N(F_1, F_2) - N(F_2, F_3)\|_2$. See Figure 14 (second row, designated self flow). This aims to minimize the second derivative of the motion of the head in the image plane; however, in practice, we found this method to have little effect on temporal noise while still causing the synthetic render to deviate from the captured image. These inaccuracies are most noticeable around the right cheek and lips.

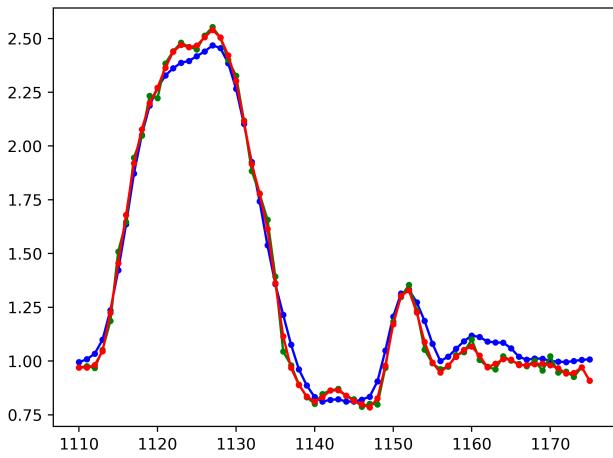
We found the most effective approach to temporal refinement to be a two step process: First, we use averaging to produce temporally consistent parameter values. Then, starting from those values, we use the optical flow approach to make the synthetic render flow better target that of the plate. See Figure 14 (bottom row, denoted hybrid). This hybrid approach produces temporally consistent results with synthetic renders that still match the captured image. Figure 15 shows the rigid parameters before and after using this hybrid approach, along with that obtained manually by a matchmove artist for reference. Assuming the manual rigid alignment is the “ground truth,” Figure 16 compares how far the rigid parameters are from their manually solved for values both before and after the hybrid smoothing approach. Figure 17 compares all the proposed smoothing methods on this same example.

A.1. Expression Reestimation

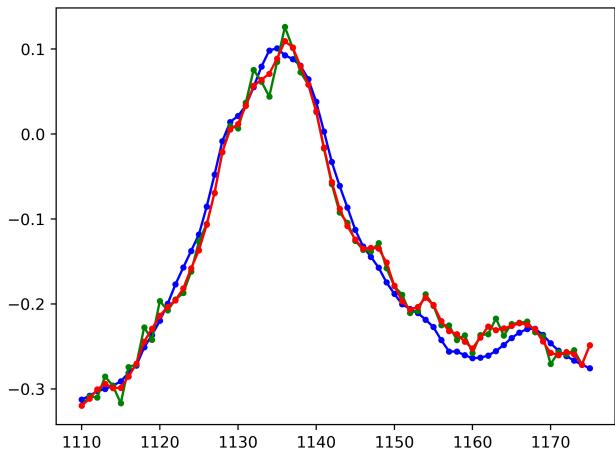
The expression estimation and temporal smoothing steps can be repeated multiple times until convergence to produce more accurate results. To demonstrate the potential of this approach, we reestimate the facial expression by solving for the mouth and jaw blendshape parameters (a subset of w) while keeping the rigid parameters fixed after temporal smoothing. As seen in Figure 18, the resulting facial expression is generally more accurate than the pre-temporal smoothing result. Furthermore, in the case where temporal smoothing dampens the performance, performing expression re-estimation will once again capture the desired expression (frame 1159).



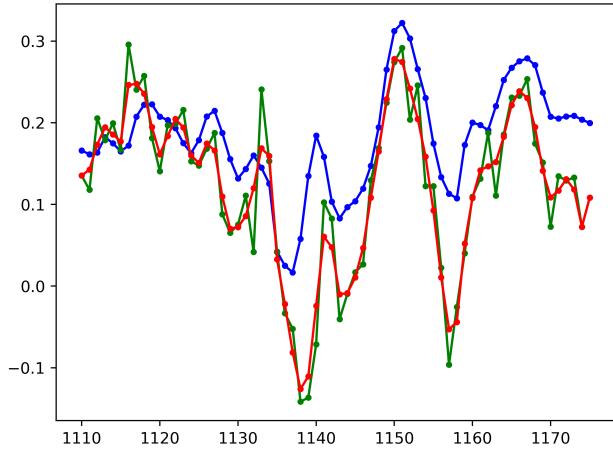
Figure 14. A comparison of the geometry produced by the various proposed temporal smoothing methods overlaid on the captured images. Notice how the techniques involving optical flow information from the captured images better track the nose and the contours of the face.



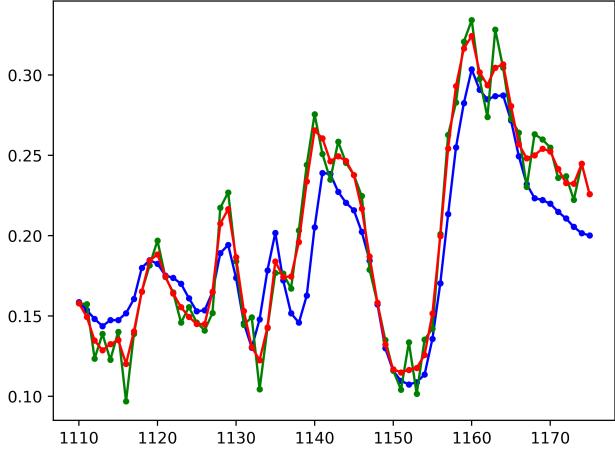
(a) Euler Angle X



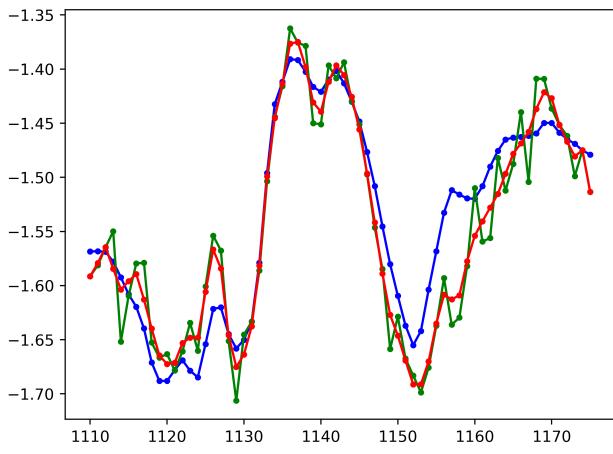
(d) Translation X



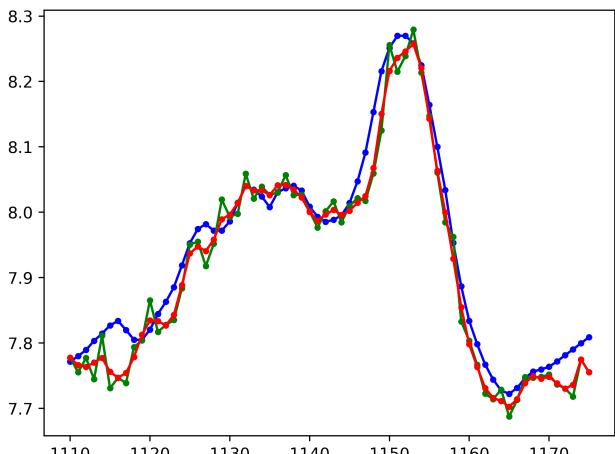
(b) Euler Angle Y



(e) Translation Y



(c) Euler Angle Z



(f) Translation Z

Figure 15. A comparison of the rigid parameters θ and t from a manual process (blue), from the initial expression estimation (green), and from the hybrid smoothing approach (red).

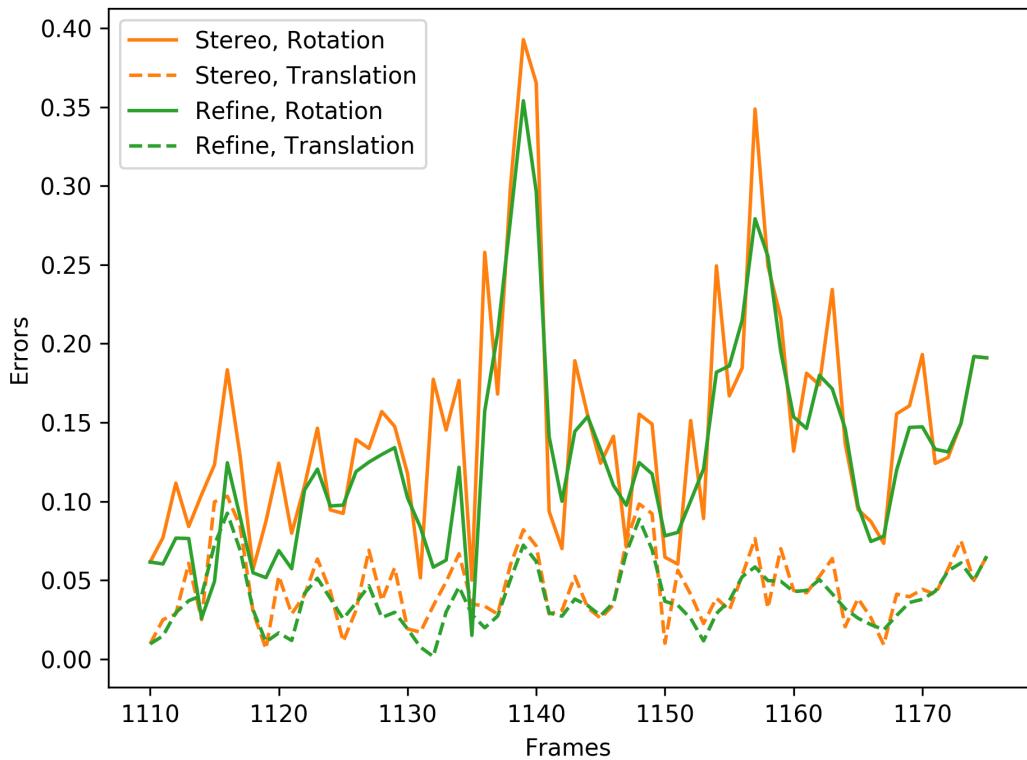
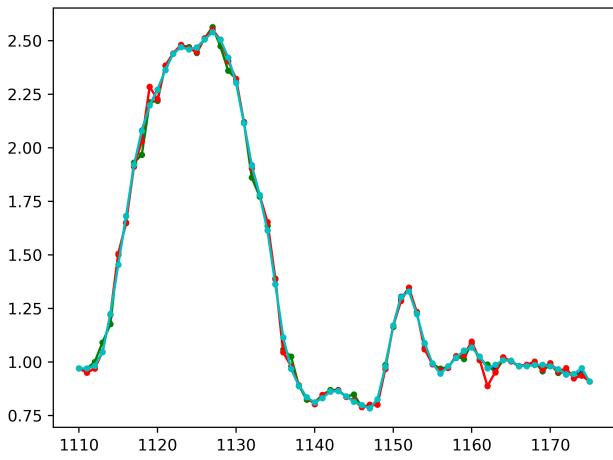
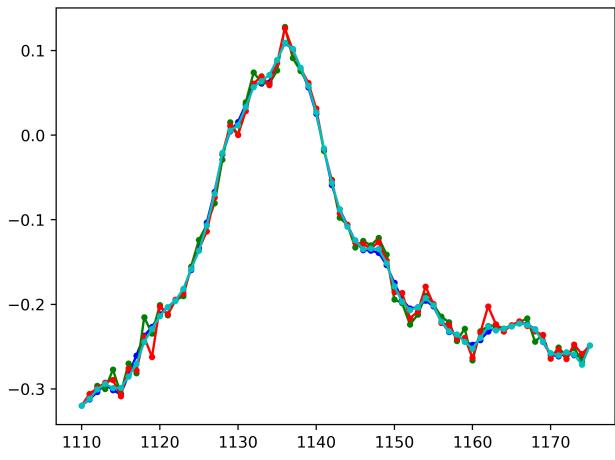


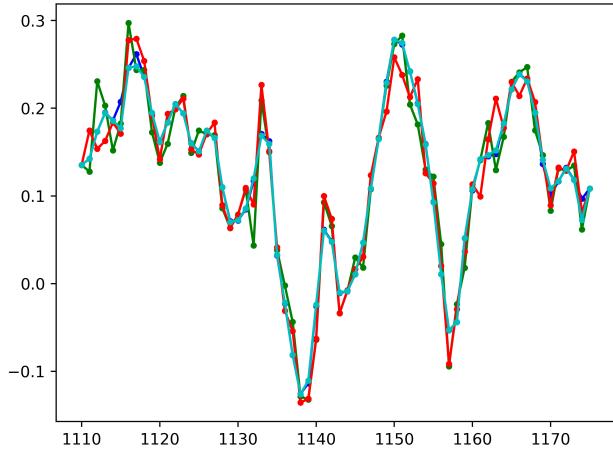
Figure 16. Assuming the manual rigid alignment is the “ground truth,” we compare the errors for rigid parameters for the stereo results (orange) and the temporally smoothed stereo results (green). Note that the temporal smoothing does not increase the errors already found in the stereo rigid alignment.



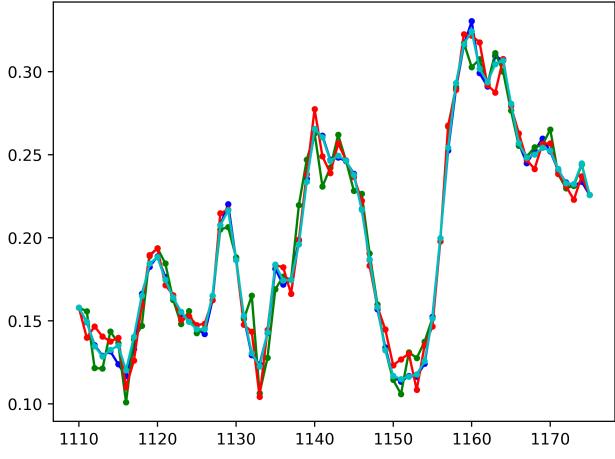
(a) Euler Angle X



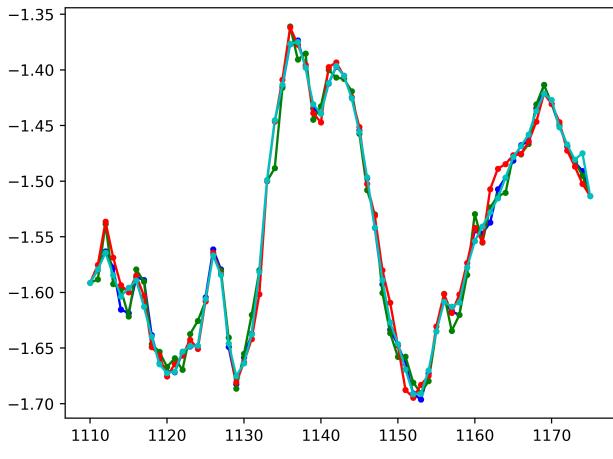
(d) Translation X



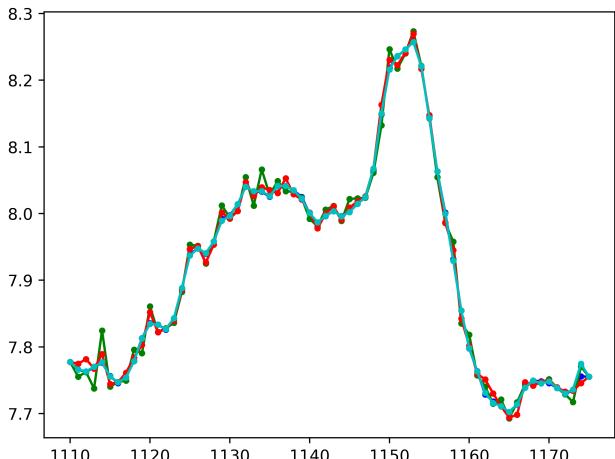
(b) Euler Angle Y



(e) Translation Y



(c) Euler Angle Z



(f) Translation Z

Figure 17. A comparison of the rigid parameters obtained by averaging (blue), self flow (green), plate flow (red), and the hybrid smoothing approach (cyan).

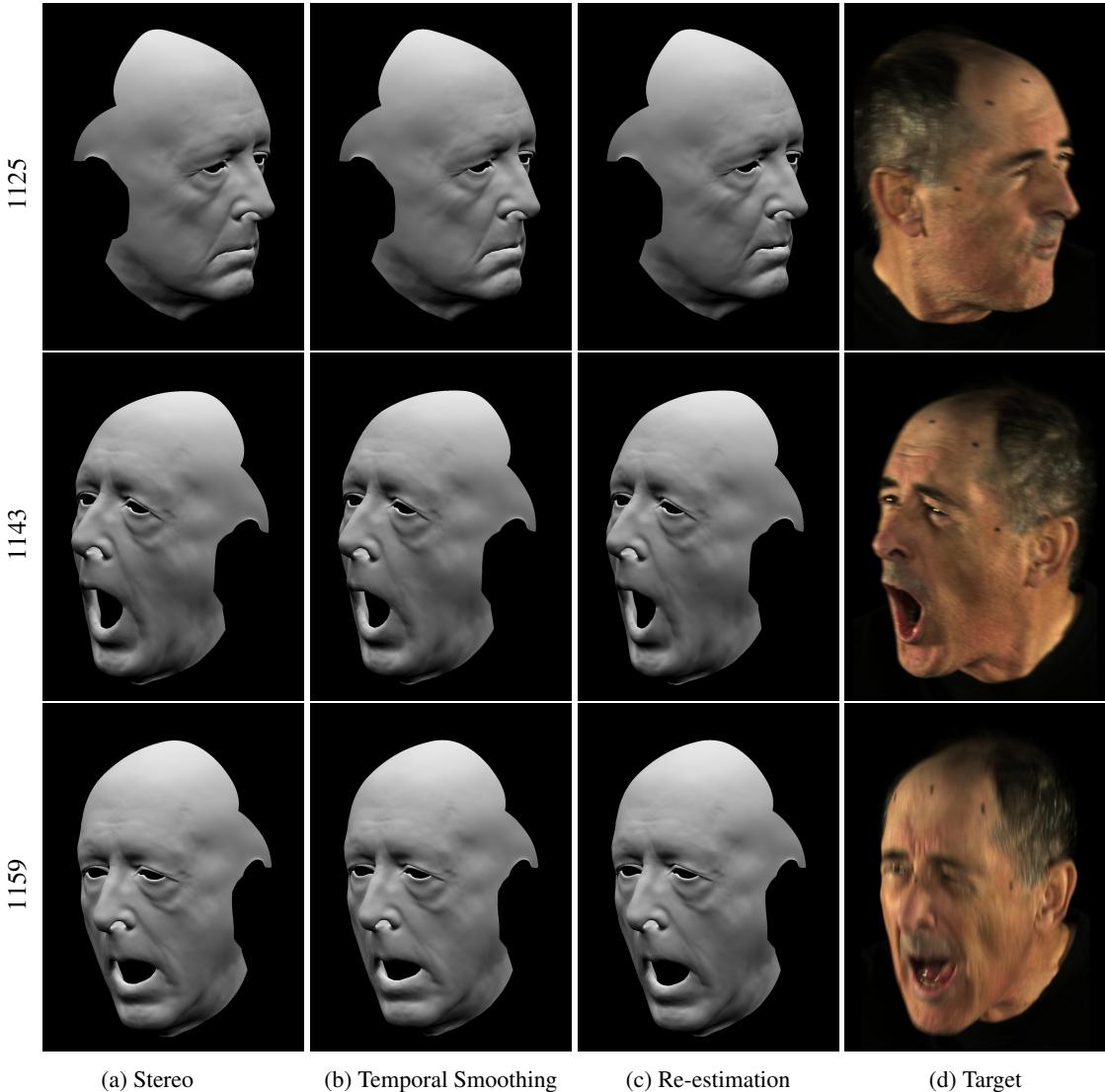


Figure 18. A comparison of the result of stereo expression estimation, stereo temporal smoothing, and stereo expression re-estimation after temporal smoothing.

References

- [1] O. Aldrian, W. A. Smith, et al. Inverse rendering of faces with a 3d morphable model. 2012.
- [2] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 4015–4023, 2015.
- [3] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.
- [4] K. S. Bhat, R. Goldenthal, Y. Ye, R. Mallet, and M. Koppenwas. High fidelity facial animation capture and retargeting with contours. In *Proceedings of the 12th ACM SIGGRAPH/eurographics symposium on computer animation*, pages 7–14. ACM, 2013.
- [5] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.
- [6] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004.
- [7] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, volume 1, page 4, 2017.
- [8] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, 2012.
- [9] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014.
- [10] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *European Conference on Computer Vision*, pages 109–122. Springer, 2014.
- [11] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 145–156. ACM Press/Addison-Wesley Publishing Co., 2000.
- [12] J. Deng, Y. Zhou, S. Cheng, and S. Zaferiou. Cascade multi-view hourglass model for robust 3d face alignment. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 399–403. IEEE, 2018.
- [13] Z. Deng, P.-Y. Chiang, P. Fox, and U. Neumann. Animating blendshape faces by cross-mapping motion capture data. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 43–48. ACM, 2006.
- [14] D. Dinev, T. Beeler, D. Bradley, M. Bächer, H. Xu, and L. Kavan. User-guided lip correction for facial performance capture. In *Computer Graphics Forum*, volume 37, pages 93–101. Wiley Online Library, 2018.
- [15] X. Dong, S.-I. Yu, X. Weng, S.-E. Wei, Y. Yang, and Y. Sheikh. Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 360–368, 2018.
- [16] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016.
- [17] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.
- [18] Z.-H. Feng, J. Kittler, W. Christmas, P. Huber, and X.-J. Wu. Dynamic attention-controlled cascaded shape regression exploiting training data augmentation and fuzzy-set sample weighting. *arXiv preprint arXiv:1611.05396*, 2016.
- [19] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.
- [20] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.
- [21] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski. Sexnet: A neural network identifies sex from human faces. In *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3*, NIPS-3, pages 572–577, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.
- [22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [23] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.
- [24] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1031–1039. IEEE, 2017.
- [25] L. A. Jeni, J. F. Cohn, and T. Kanade. Dense 3d face alignment from 2d video for real-time use. *Image and Vision Computing*, 58:13–24, 2017.
- [26] X. Jin and X. Tan. Face alignment in-the-wild: A survey. *Computer Vision and Image Understanding*, 162:1–22, 2017.
- [27] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.
- [28] A. Jourabloo and X. Liu. Pose-invariant face alignment via cnn-based dense 3d model fitting. *International Journal of Computer Vision*, 124(2):187–203, 2017.

- [29] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.
- [30] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.
- [31] I. Korshunova, W. Shi, J. Dambre, and L. Theis. Fast face-swap using convolutional neural networks. In *The IEEE International Conference on Computer Vision*, 2017.
- [32] S. Laine, T. Karras, T. Aila, A. Herva, S. Saito, R. Yu, H. Li, and J. Lehtinen. Production-level facial performance capture using deep convolutional neural networks. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, page 10. ACM, 2017.
- [33] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. H. Pighin, and Z. Deng. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)*, 1(8), 2014.
- [34] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [35] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018.
- [36] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative face completion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017.
- [37] S. Liu, J. Yang, C. Huang, and M.-H. Yang. Multi-objective convolutional learning for face labeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [38] M. Loper. Chumpy autodifferentiation library. <http://chumpy.org>, 2014.
- [39] M. M. Loper and M. J. Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014.
- [40] M. Lourakis and A. A. Argyros. Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1526–1531. IEEE, 2005.
- [41] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.
- [42] W.-C. Ma, A. Jones, J.-Y. Chiang, T. Hawkins, S. Frederiksen, P. Peers, M. Vukovic, M. Ouhyoung, and P. Debevec. Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Transactions on Graphics (TOG)*, 27(5):121, 2008.
- [43] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.
- [44] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman. Quaternion averaging. 2007.
- [45] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.
- [46] Y. Nirkin, I. Masi, A. T. Tuan, T. Hassner, and G. Medioni. On face segmentation, face swapping, and face perception. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)(FG)*, volume 00, pages 98–105, May 2018.
- [47] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.
- [48] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500. ACM, 2001.
- [49] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015.
- [50] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(1):23–38, Jan. 1998.
- [51] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [52] S. Saito, T. Li, and H. Li. Real-time facial segmentation and performance capture from rgb input. In *European Conference on Computer Vision*, pages 244–261. Springer, 2016.
- [53] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [54] O. Sendik and D. Cohen-Or. Deep correlations for texture synthesis. *ACM Transactions on Graphics (TOG)*, 36(5):161, 2017.
- [55] S. Sengupta, A. Kanazawa, C. D. Castillo, and D. W. Jacobs. Sfsnet: Learning shape, reflectance and illuminance of faces in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [56] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. In *Acm transactions on graphics (tog)*, volume 24, pages 417–425. ACM, 2005.
- [57] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [58] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. Integral human pose regression. In *The European Conference on Computer Vision (ECCV)*, September 2018.
- [59] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR ’14*, pages 1701–1708, Washington, DC, USA, 2014. IEEE Computer Society.
- [60] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

- [61] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 18, 2018.
- [62] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.
- [63] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1392, 2013.
- [64] C. Wu, D. Bradley, M. Gross, and T. Beeler. An anatomically-constrained local deformation model for monocular face capture. *ACM Transactions on Graphics (TOG)*, 35(4):115, 2016.
- [65] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2138, 2018.
- [66] J. Xing, Z. Niu, J. Huang, W. Hu, X. Zhou, and S. Yan. Towards robust and accurate multi-view and partially-occluded face alignment. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):987–1001, 2018.
- [67] A. Zadeh, Y. C. Lim, T. Baltrušaitis, and L.-P. Morency. Convolutional experts constrained local model for 3d facial landmark detection. In *ICCV Workshops*, pages 2519–2528, 2017.
- [68] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23:1499–1503, 2016.
- [69] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.
- [70] S. Zhu, C. Li, C.-C. Loy, and X. Tang. Unconstrained face alignment via cascaded compositional learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3409–3417, 2016.