Automatic Acquisition of High-fidelity Facial Performances Using Monocular Videos

Fuhao Shi* Hsiang-Tao Wu† Xin Tong† Jinxiang Chai*

*Texas A&M University †Microsoft Research Asia

Figure 1: *Our system automatically captures high-fidelity facial performances using Internet videos: (left) input video data; (middle) the captured facial performances; (right) facial editing results: wrinkle removal and facial geometry editing.*

Abstract

This paper presents a facial performance capture system that automatically captures high-fidelity facial performances using uncontrolled monocular videos (*e.g.*, Internet videos). We start the process by detecting and tracking important facial features such as the nose tip and mouth corners across the entire sequence and then use the detected facial features along with multilinear facial models to reconstruct 3D head poses and large-scale facial deformation of the subject at each frame. We utilize per-pixel shading cues to add fine-scale surface details such as emerging or disappearing wrinkles and folds into large-scale facial deformation. At a final step, we iterate our reconstruction procedure on large-scale facial geometry and fine-scale facial details to further improve the accuracy of facial reconstruction. We have tested our system on monocular videos downloaded from the Internet, demonstrating its accuracy and robustness under a variety of uncontrolled lighting conditions and overcoming significant shape differences across individuals. We show our system advances the state of the art in facial performance capture by comparing against alternative methods.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: facial performance capture, face animation, facial modeling, facial detection and tracking, facial editing

Links: DL PDF WEB VIDEO

*{fuhaoshi, jchai}@cse.tamu.edu;{musclewu, xtong}@microsoft.com

1 Introduction

Facial animation is an essential component of many applications, such as movies, video games, and virtual environments. Thus far, one of the most popular and successful approaches for creating virtual faces often involves capturing facial performances of real people. Capturing high-fidelity facial performances remains challenging because it requires capturing spatial-temporal facial performances involving both large-scale facial deformation and fine-scale geometric detail.

An ideal solution to the problem of facial performance capture is to use a standard video camera to capture live performances in 3D. The minimal requirement of a single video camera is particularly appealing, as it offers the lowest cost, a simplified setup, and the potential use of legacy sources and uncontrolled videos (*e.g.*, Internet videos). Yet despite decades of research in computer graphics and a plethora of approaches, many existing video-based facial capture systems still suffer from two major limitations. Firstly, captured facial models are often extremely coarse and usually only contain sparse collections of 2D or 3D facial landmarks rather than detailed 3D shapes. Secondly, these results are often vulnerable to ambiguity caused by occlusions, the loss of depth information in the projection from 3D to 2D, and a lack of discernible features on most facial regions and therefore require a significant amount of manual intervention during the capturing process.

In this paper, we present an automatic technique for acquiring high-fidelity facial performances using monocular video sequences such as Internet videos (Figure 1). The key idea of our approach is to use both high-level facial features and per-pixel shading cues to reconstruct 3D head poses, large-scale deformations and fine-scale facial details in a spacetime optimization framework. We start the process by automatically detecting/tracking important facial features such as the nose tip and mouth corners across the entire sequence. The detected facial features are then used to reconstruct 3D head poses and large-scale deformations of a detailed face model at each frame. This step combines the power of non-rigid structure from motion, multilinear facial models, and keyframe based spacetime optimization for large-scale deformation reconstruction. Next, we utilize per-pixel shading cues to add fine-scale surface details such as emerging or disappearing wrinkles and folds into large-scale fa-

cial deformations. At a final step, We iterate our reconstruction procedure on large-scale facial geometry and fine-scale facial details to further improve the accuracy of facial reconstruction.

Our final system is robust and fully automatic, allowing for high-fidelity facial performance capture of large-scale deformation and fine-scale facial detail. We have tested our system on monocular videos downloaded from the Internet, demonstrating its accuracy and robustness under a variety of uncontrolled lighting conditions and overcoming significant shape differences across individuals. We show our system achieves the state-of-the-art results by comparing against alternative systems [Garrido et al. 2013].

Contributions. This paper makes the following contributions:

- First and foremost, an end-to-end facial performance capture system that automatically reconstructs 3D head poses, large-scale facial deformation and fine-scale facial detail using uncontrolled monocular videos.
- An automatic facial feature detection/tracking algorithm that accurately locates important facial features across the entire video sequence.
- A novel facial reconstruction technique that combines facial detection, non-rigid structure from motion, multilinear facial models, and keyframe based spacetime optimization to compute 3D poses and large-scale facial deformation from monocular video sequences. This step also requires estimating the unknown camera parameters across the entire sequence.
- An efficient facial modeling algorithm that infers fine-scale geometric details and unknown incident lighting and face albedo from the whole sequence of input images. Our algorithm builds on the state of the art in 3D face reconstruction from a single image [Kemelmacher-Shlizerman and Basri 2011]. However, we significantly extend the idea to reconstructing dynamic facial details using monocular videos.

2 Background

Our system automatically captures high-fidelity facial performances using monocular video sequences. Therefore, we focus our discussion on methods and systems developed for acquiring 3D facial performances.

One of the most successful approaches for facial capture is based on marker-based motion capture systems [Guenter et al. 1998], which robustly and accurately track a sparse set of markers attached to the face. Recent efforts in this area (*e.g.* [Bickel et al. 2007; Huang et al. 2011]) have been focused on complementing marker-based systems with other types of capturing devices such as video cameras and/or 3D scanners to improve the resolution and details of reconstructed facial geometry. Marker-based motion capture, however, is expensive and cumbersome for 3D facial performance capture.

Marker-less facial performance capture provides an appealing alternative because it is non-intrusive and does not impede the subject's ability to perform facial expressions. One solution to the problem of marker-less facial capture is the use of depth and/or color data obtained from structured light systems [Zhang et al. 2004; Ma et al. 2008; Li et al. 2009; Weise et al. 2009]. For example, Zhang and colleagues [2004] captured 3D facial geometry and texture over time and built the correspondences across all the facial geometries by deforming a generic face template to fit the acquired depth data using optical flow computed from image sequences. Ma et al. [2008] achieved high-resolution facial reconstructions by interleaving structured light with spherical gradient photometric stereo using the USC Light Stage. Recently, Li and his colleagues [2009] captured dynamic depth maps with their realtime structured light system and fitted a smooth template to the captured depth maps.

Reconstructing high-quality face models directly from multiview images offers another possibility for marker-less motion capture [Bradley et al. 2010; Beeler et al. 2010; Beeler et al. 2011; Valgaerts et al. 2012]. In particular, Bradley and his colleagues [2010] used multi-view stereo reconstruction techniques to obtain initial facial geometry, which was then used to capture 3D facial movement by tracking the geometry and texture over time. Beeler et al. [2010] presented an impressive multi-view stereo reconstruction system for capturing the 3D geometry of a face in a single shot and later extended it to acquiring dynamic facial expressions using multiple synchronized cameras [Beeler et al. 2011]. More recently, Valgaerts et al. [2012] combined image-based tracking with shading-based geometry refinement to reconstruct facial performances from stereo image sequences.

The minimal requirement of a single camera for facial performance capture is particularly appealing, as it offers the lowest cost and a simplified setup. However, the use of a single RGB camera for facial capture is often vulnerable to ambiguity caused by the loss of depth information in the projection from 3D to 2D and a lack of discernible features on most facial regions. One way to address the issue is to use person-specific facial prior models to reduce reconstruction ambiguity (*e.g.*, [Blanz et al. 2003; Vlasic et al. 2005]). However, fine face details such as wrinkles and large lines cannot be recovered with this approach. In addition, their tracking process is often performed in a sequential manner and therefore requires good initialization and manual correction for troublesome frames.

Recently, Cao and colleagues [2013a] proposed a 3D regression algorithm that utilized personalized blendshape models for automatic, realtime facial tracking/retargeting. Their approach, however, required an expensive offline training stage to construct person-specific blendshape models. In addition, they focused on tracking large-scale geometric deformation rather than authentic reconstruction of high-fidelity facial performances. Concurrently to this work, Suwajanakorn and colleagues [2014] propose a dense 3D flow algorithm coupled with shape-from-shading to reconstruct high-fidelity facial geometry from monocular videos. Though their method does not require person-specific scan/blendshapes, it uses a photo gallery of the subject's faces under different illuminations for reconstructing a person-specific average facial model, which could be unavailable for subjects in uncontrolled videos. Additionally, instead of using a person-specific average facial model for facial capture, we propose to use multilinear face models to reduce the reconstruction ambiguity of facial capture, thereby significantly improving the robustness of the system. Finally, their method assumes a known albedo map obtained from the subject's photo gallery and estimates lighting for every frame separately. In contrast, we use the whole sequence of input images to estimate a single lighting map and face albedo and therefore further improve the accuracy of facial reconstruction.

Among all the systems, our work is most closely related to Garrido et al. [2013], which captured detailed, dynamic 3D facial geometry using monocular video sequences. Briefly, they first created a personalized blendshape model for the captured actor by transferring the blendshapes of a generic model to a single static 3D face scan of the subject. They then tracked 2D image features across the entire sequence by combining sparse facial feature tracking and optical flow estimation. At a final step, they reconstructed fine-scale facial detail by estimating the unknown lighting and exploiting shading for shape refinement.

Our research shares a similar goal but there are important differences. They relied on manual specification of correspondences be-

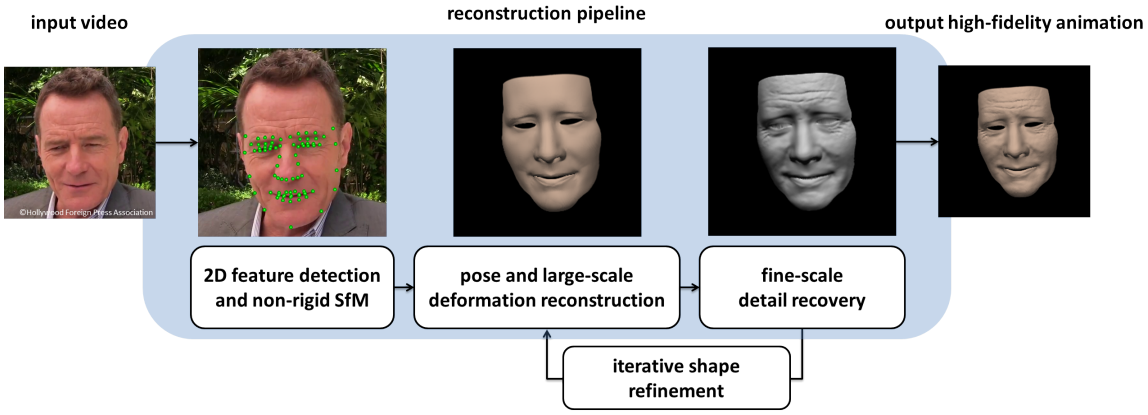


Figure 2: *Our high-fidelity facial performance reconstruction pipeline. We first detect and track a sparse set of 2D facial features and recover their 3D positions and unknown camera parameters using non-rigid SfM techniques. We then reconstruct 3D head poses and large-scale facial geometry using the estimated facial features and camera parameters. We assume that the lighting and albedo are constant across the entire sequence and extend shape-from-shading techniques to reconstruct fine-scale details throughout the whole sequence. Lastly, we iteratively refine large-scale facial deformations and fine-scale facial details to obtain the final result.*

tween a generic blendshape model and a static face scan of the subject to reconstruct a personalized blendshape model for the subject. In addition, their optical-flow based tracking process required manual intervention to improve the locations of 2D features in the first frame for texturing. In contrast, we automatically detect facial features across the entire sequence and use them along with multilinear facial models to simultaneously compute 3D head poses and large-scale deformations in a keyframe based optimization framework. Garrido et al. reported that typical manual intervention for a test video sequence required about 40 minutes, while our method is fully automatic. In addition, they required a static 3D face scan of the subject for transferring a generic blendshape model to the subject. Our system does not have such a limitation and therefore can be applied to capturing high-fidelity facial performances from uncontrolled monocular videos such as Internet videos. Finally, we have compared against their method and the comparison shows that our system produces more accurate results than theirs (Section 8.2).

Our work on fine-scale detail reconstruction builds on the success of modeling fine-scale facial geometry from a single image proposed by [Kemelmacher-Shlizerman and Basri 2011]. In particular, Kemelmacher-Shlizerman and Basri [2011] introduced a novel method for shape recovery of a face from a single image by using a reference 3D face model and a reference albedo. We present three novel extensions to their work. First, we extend their idea to shape recovery of a dynamic face from a monocular video sequence. We reduce the ambiguity of estimating lighting and albedo by utilizing the assumption that the lighting and albedo are constant across the entire sequence. This allows us to estimate the unknown lighting coefficients and albedo based on shading cues across the entire sequence. Second, we utilize results obtained from large-scale deformation reconstruction to initialize and guide the fine-scale geometry reconstruction process, thereby significantly improving the accuracy, robustness and speed of the process. Third, they simplified the reconstruction problem with linear approximations and directly reconstructed depth in a least-square fitting framework. In contrast, we propose a two-step optimization algorithm to sequentially compute normal map and depth. As shown in Section 8.2, our system produces more accurate results than their method.

The idea of using spherical harmonics approximation for fine-scale detail recovery is similar to previous methods proposed by Wu et al. [2011] and Valgaerts et al. [2012]. Our method, however, is different from theirs. First, we assume consistent lighting and albedo across the entire sequence while they estimate both maps for each frame. Additionally, we estimate a per-pixel albedo map while they assume it is piecewise uniform. Finally, we measure the differences between the synthesized and observed images based on RGB intensities rather than high-frequency components (image gradients) because we also want to use recovered surface details to refine large-scale facial deformation.

Our system is also relevant to recent successes in tracking 3D facial expression using a single RGBD camera such as Microsoft Kinect or time-of-flight (TOF) cameras [Weise et al. 2011; Bouaziz et al. 2013; Li et al. 2013]. Notably, Weise et al. [2011] used RGBD image data captured by a single *Kinect* and a facial 3D template, along with a set of predefined blendshape models, to track facial expression over time. Most recently, Bouaziz et al [2013] and Li et al. [2013] concurrently developed real-time monocular face trackers based on a run time shape correction strategy for combined depth and video data. All these systems are focused on modeling large-scale facial deformation rather than high-fidelity facial performances. In addition, they are based on depth and image data obtained by a calibrated RGBD camera rather than RGB images captured by an uncalibrated video camera. It is not clear how their approaches can be extended to capture high-fidelity facial performances from uncontrolled monocular videos.

3 Overview

Our system acquires high-fidelity facial performances from uncontrolled monocular video sequences. The problem is challenging because of complex facial movements at different scales and ambiguity caused by the loss of depth information in the projection from 3D to 2D and a lack of discernible features on most facial regions. Unknown camera parameters and lighting conditions further complicate the reconstruction problem. To this end, we decompose high-fidelity facial performances into three scales: high-level facial features, large-scale facial deformations and fine-scale facial details and reconstruct them from coarse to fine scales. We start with the coarse scale (high-level facial features). During the reconstruction, we utilize the results in coarse scales to initialize and guide the reconstruction in fine scales. At a final step, we iterate our reconstruction procedure on large-scale facial geometry and fine-scale facial detail to obtain the final output. The whole system consists of four

main components summarized as follows (Figure 2):

Facial feature detection and 3D reconstruction. We start the process by automatically detecting and tracking important facial features such as nose tip, eye and mouth corners in monocular video sequences. We apply non-rigid factorization techniques to recover 3D feature positions and unknown camera parameters, which are then used to initialize and guide the large-scale deformation reconstruction process.

Large-scale deformation reconstruction. Sparse facial features, however, do not provide detailed facial geometry. We introduce an efficient reconstruction process that utilizes the recovered 3D facial features and camera parameters, along with multlinear facial models, to model detailed facial geometry. We formulate it as a space-time optimization problem by simultaneously reconstructing head poses and large-scale facial geometry across the entire sequence. This, however, requires solving a challenging nonlinear optimization with a huge number of unknowns. We address the challenge by developing a keyframe based optimization algorithm.

Fine-scale detail recovery. Recovering dynamic geometric details such as wrinkles and folds is crucial for high-fidelity facial performance capture. We have developed an efficient shape-from-shading algorithm that infers fine-scale geometric detail, the unknown incident lighting and face albedo from the whole sequence of input images. Our method assumes both lighting and face albedo are constant throughout the whole sequence. Starting from large-scale deformation results, we simultaneously compute per-pixel normal map of each input image and unknown incident lighting, and face albedo by minimizing the inconsistency between the rendered and observed image sequences. We further reconstruct a per-pixel depth estimate from the reconstructed per-pixel normal map. Again, we use keyframe based optimization to facilitate reconstruction of fine-scale facial detail.

Iterative shape refinement. Large-scale facial geometry reconstructed from sparse facial features often does not closely fit actual facial geometry because of the lack of facial features in some facial regions such as cheeks. We address the issue by iteratively refining large-scale facial geometry using the reconstructed per-pixel normal map and then updating the per-pixel normal map with the refined large-scale deformation.

We describe these components in detail in the following sections.

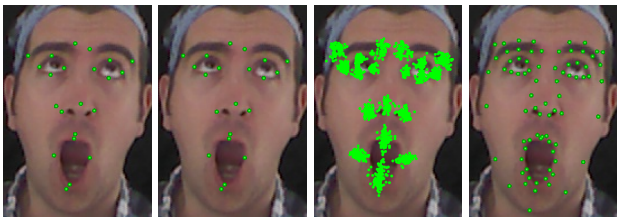
4 Feature Detection and Non-rigid SfM

Our first challenge is how to reconstruct 3D facial feature locations from uncalibrated monocular video sequences. This is achieved by detecting/tracking a sparse set of facial features across the entire sequence and performing non-rigid SfM on 2D tracking features.

4.1 Facial Feature Detection

We have developed an efficient facial feature detection/tracking algorithm that automatically locates important facial features across the entire video sequence. Our key idea is to combine the power of local detection, spatial priors for facial feature locations, Active Appearance Models (AAMs) [Matthews and Baker 2004] and temporal coherence for facial feature detection.

Briefly, we formulate local feature detection as a per-pixel classification problem and apply randomized forests [Amit and Geman 1997] to associate each pixel with a probability score of being a particular feature. The outputs of local feature detectors are often noisy and frequently corrupted by outliers due to classification errors. This motivates us to employ geometric hashing to robustly search the closest examples in a predefined database of labeled images and use a consensus of non-parametric global shape models to improve the outputs of local detectors. Furthermore, we develop an efficient facial registration method that integrates AAMs, local detection results, and facial priors into a Lucas-Kanade registration framework. Finally, we complement facial detection with temporal coherence to improve the robustness and accuracy of our facial detection and tracking process.



(a) (b) (c) (d)

Figure 3: *Robust detection of facial features: (a) candidate features after local detection and multi-mode extraction; (b) detected features after outlier removal; (c) closest examples of detected features via geometric hashing; (d) final output.*

4.1.1 Local Feature Detection

We introduce an efficient feature detection process which utilizes the local information of a pixel (*i.e.*, an input patch centered at a pixel) to detect a predefined set of facial features from single RGB images.

We formulate the feature detection process as a per-pixel classification problem. During training, we construct a set of $N = 21$ classes of keypoints. Each class corresponds a prominent facial feature such as the nose tip or the left corner of the mouth. Figure 3(b) shows all facial features considered by our local detection process. At runtime, given an input patch centered at a pixel $\mathbf{x}$, we want to decide the likelihood that a particular feature $c \in \{1,...,N\}$ is located at point $\mathbf{x}$ in the image.

We use randomized decision trees [Amit and Geman 1997; Lepetit and Fua 2006] to train a classifier for automatic labeling of pixels. For each randomized tree, similar binary tests are performed. The feature function calculates the difference of intensity values of a pair of pixels taken in the neighborhood of the classification pixel. Specifically, at a given pixel $\mathbf{x}$, the feature computes

$$f(x) = P(\mathbf{x}+\mathbf{u}) - P(\mathbf{x}+\mathbf{v}), \quad (1)$$

where $P(\mathbf{x})$ is the intensity value at pixel $\mathbf{x}$. The parameters $\mathbf{u}$ and $\mathbf{v}$ describe the offsets. We then infer feature locations from their probability maps by detecting peaks of important modes. Meanshift algorithm [Comaniciu and Meer 2002] is used to refine the location of each extracted mode. Figure 3(a) shows the result obtained from the local feature detection step.

4.1.2 KNN Search by Geometric Hashing

We now discuss how to utilize prior knowledge embedded in a training set of labeled facial images to remove misclassified features. Due to classification errors, feature candidates inevitably contain "outlier" features (*e.g.*, the "outlier" feature in the right side of the nose shown in Figure 3(a)). Similar to [Belhumeur et al. 2011], we robustly search closest examples in a training set of labeled images and use them to remove misclassified features. KNN search, however, requires computing the unknown similarity transformations

between the detection image and every training image. Instead of adopting a RANSAC-based sampling procedure [Belhumeur et al. 2011], we propose to use geometric hashing to find the closest examples. Geometric hashing [Lamdan and Wolfson 1988] has been successfully applied to 3D object detection and popular for its simplicity and efficiency. The use of geometric hashing for KNN search significantly improves the speed and accuracy of our search process. Figure 3(b) and 3(c) show the detected feature after outlier removal and the closest examples obtained from searching a training set of labeled facial images.

4.1.3 Facial Detection Refinement

We refine the feature detection results by complementing detection with facial alignment using Active Appearance Models (AAMs) [Matthews and Baker 2004]. Figure 3(d) show the improvement of feature locations as well as detection of non-salient facial features via the refinement step.

We formulate the refinement process in an optimization framework. The whole cost function consists of three terms:

$$E = w_1 E_{AAM} + w_2 E_{detection} + w_3 E_{prior}, \tag{2}$$

where the first term E_{AAM} is *Active Appearance Models* (AAMs) term, which measures the inconsistency between the input image and the AAM model instance (for details, refer to [Matthews and Baker 2004]). The second term is the *detection* term which penalizes the deviation of feature points from detected feature points from Section 4.1.2. The third term is the *prior* term which ensures the new feature points are consistent with 2D facial priors embedded in K closest examples. In this work, we fit a Gaussian prior based on K closest examples and obtain this term by applying the negative log to the Gaussian distribution. The local priors reduce bias towards the average face and avoid the problem of finding an appropriate structure for global priors, which would necessarily be high-dimensional and nonlinear.

We minimize the cost function by simultaneously optimizing the shape and appearance parameters of the AAM model instance, as well as the global similarity transformation for aligning the input image with the AAM model instance. We analytically derive Jacobian and Hessian of each objective term and optimize the function in Lucas-Kanade registration framework via iterative linear system solvers [Matthews and Baker 2004]. We initialize the shape parameter using the closest example of the input image. The appearance parameters are initialized by the average appearance image of AAM models. The optimization typically converges in 8 iterations because of very good initialization and local facial priors obtained from K closest examples. We set the weights of w_1, w_2 and w_3 to 2, 1, and 0.001 respectively. During the iterations, we gradually decrease the weight for the second term (w_2) from 1 to 0.0001 in order to ensure that the final feature locations can achieve a better accuracy via AAM fitting.

4.1.4 Incorporating Temporal Coherence

Single frame feature detection can automatically infer the locations of facial features from single RGB images but often with noisy and unstable tracking results. In addition, our detection refinement process builds upon AAMs and therefore might not generalize well to new subjects that are significantly different from training databases. This motivates us to utilize the temporal coherence to further improve the robustness and accuracy of our facial detection and tracking process. We utilize previously registered facial images to incrementally update the Active Appearance Models (AAMs) on the fly. During tracking, we maintain a buffer of registered facial images from previous frames and use them to incrementally update the mean and eigen basis of AAMs based on an incremental learning method proposed by Ross et al. [2008]. Note that we only push the registered frames whose corresponding AAM fitting errors are below a particular threshold (0.6) into the buffer. Once the buffer is full, we first check if the registered images in the buffer are sufficiently far from the subspace of the current AAMs. When the reconstruction residual is higher than a threshold (0.3), we update the AAMs and then reinitialize the whole buffer. In our experiment, we set the buffer size to 10.

Please refer to our supplementary material for evaluation of our facial detection component.

4.2 Non-rigid Structure from Motion

This step aims to reconstruct 3D feature positions and unknown camera parameters across the entire sequence, which requires solving a non-rigid structure from motion problem. Our solution is based on a prior-free non-rigid structure from motion method proposed by Dai et al. [2012]. We choose their method because it is purely convex, very easy to implement, and is guaranteed to converge to an optimal solution. By assuming a weak perspective camera model, the non-rigid structure from motion process reconstructs camera motion and non-rigid shapes through a SVD based factorization.

5 Reconstructing 3D Pose and Large-Scale Deformation

This section describes our idea on how to reconstruct large-scale facial geometry and 3D head poses from 2D feature locations obtained from Section 4. We formulate this in a spacetime optimization framework and simultaneously reconstruct large-scale deformations and 3D head poses across the entire sequence. Direct estimate of large-scale deformations and 3D head poses throughout the whole sequence is often time-consuming and memory intensive. This motivates us to develop a keyframe based optimization method to speed up the optimization process.

5.1 Representation and Formulation

We model large-scale facial deformation using multi-linear facial models [Vlasic et al. 2005; Cao et al. 2013a]. Specifically, we parameterize large-sale facial deformation using two low-dimensional vectors controlling "identity" and "expression" variation. As a result, we can represent large-scale facial geometry of the subject at any frame using

$$M = R(C_r \times_2 m_{id}^T \times_3 m_{exp}^T) + T, \tag{3}$$

where M represents large-scale facial geometry of an unknown subject. And R and T represent the global rotation and translation of the subject. C_r is the reduced core tensor, and m_{id} and m_{exp} are identity and expression parameters respectively. Our multi-linear model is constructed from FaceWarehouse [Cao et al. 2013b], which contains face meshes corresponding to 150 identities and 47 facial expressions. In our experiment, the numbers of dimensions for the identity and expression parameters are set to 50 and 25.

We assume that the camera projection is weak perspective. The relationship between a 2D facial feature $\mathbf{p}_k$ and its corresponding large scale facial geometry model can be described as follows:

$$\mathbf{p}_k = sR((C_r \times_2 m_{id}^T \times_3 m_{\text{exp}}^T)^{(k)}) + \mathbf{t}, \tag{4}$$

where s is the scalar for the weak perspective projection and $\mathbf{t}$ represents the translation components on the image space. Note that

t_z in T is dropped because of the weak perspective camera model assumption.

Our goal herein is to estimate a number of unknown parameters, including $\{R, \mathbf{t}, s, m_{id}, m_{\exp}\}_j$, across the entire sequence $j = 1, ..., N$. Since the identity is the same throughout the whole sequence, the parameters to be estimated in large-scale deformation reconstruction are $m_{id}, \{R, \mathbf{t}, s, m_{\exp}\}_j, j = 1, ..., N$.

We formulate large-scale deformation reconstruction in a spacetime optimization framework by estimating all the parameters simultaneously, resulting in the following objective function:

$$\underset{m_{id}, \{R, \mathbf{t}, s, m_{\exp}\}_{j=1,...,N}}{\arg\min} \; E_{feature} + w_1 E_{id} + w_2 E_{exp} + w_3 E^s_{exp} + w_4 E^s_{pose}, \tag{5}$$

where the first term is the $feature$ term that measures how well the reconstructed facial geometry matches the observed facial features across the entire sequence. The second and third terms are the *prior* terms used for regularizing the identity and expression parameters, which are formulated as multivariate Gaussians. The fourth and fifth terms are the *smoothness* terms that penalize sudden changes of expressions and poses over time. In all of our experiments, w_1, w_2, w_3 and w_4 are set to 0.05, 0.005, 0.05 and 50, respectively.

The $feature$ term utilizes both the locations of 2D facial features from facial feature detection and the 3D depth values of the reconstructed facial features obtained from non-rigid structure from motion, resulting in the following objective $E_{feature}$.

$$E_{feature} = E_{2d} + w_d E_{depth}, \tag{6}$$

where the first and second terms evaluate how well the reconstructed facial geometry matches the observed 2D facial features and the 3D depth values of the reconstructed facial features. The weight w_d is experimentally set to 0.01.

5.2 Optimization

The direct estimate of large-scale facial geometry described in Equation (5) is challenging because it requires solving a complex nonlinear optimization problem with a huge number of unknowns. We develop a keyframe based optimization algorithm to address this challenge. Briefly, we first automatically select a number of key frames to represent large-scale facial geometry across the entire sequence. The extracted key frames allow us to divide the whole sequence into multiple subsequences. We then simultaneously estimate all the unknown parameters associated with all the key frames. Lastly, for each subsequence, we keep the identity fixed and compute the unknowns across the entire subsequence using the parameters reconstructed by keyframe optimization.

Keyframe extraction. Given 3D feature locations obtained from non-rigid structure from motion, we aim to select a minimum set of frames in such a way that 3D feature locations across the entire sequence can be accurately interpolated by 3D feature locations defined at key frames. We perform principle component analysis (PCA) on 3D face shapes of the original sequence and obtain a PCA subspace for accurately representing 3D face shapes at any frame. We adopt a greedy strategy to find the optimal solution, initializing the key frame list with all frames in the original sequence and then incrementally decreasing it by one until the difference between the original PCA subspace and the new PCA subspace spanned by the remaining frames exceeds a user-specified threshold ε. The difference between the two PCA subspaces is measured by principal angle [Wedin 1983], and is experimentally set to 0.4. The details of our analysis algorithm are described in Algorithm 1.

Algorithm 1 Keyframe Extraction

Input: the N face shapes in a sequence $V = \{v_1, ..., v_N\}$,and a tolerance threshold ε
Output: indices of the minimum key frames K

```
set K = {1, ..., N} //initialized as the full set of video frames
while 1 do
    for i = 1, ..., |K| do
        evaluate the subspace angle SA(V_{K}-i, V)
    end for
    j = argmin_i SA(V_{K}-i, V)
    minError = SA(V_{k}-j, V)
    if minError < ε then
        K = {K} - j
    else
        break;
    end if
end while
return K
```

Keyframe reconstruction. Since the number of the key frames is usually small (11–16 in our experiment), we can estimate all the unknown parameters at key frames using the objective function described in Equation 5. This can be efficiently solved by coordinate-descent optimization techniques. Note that we initialize poses and camera parameters at key frames using the reconstruction results obtained from non-rigid structure from motion.

Keyframe interpolation. This step uses the reconstructed identity parameter, as well as the recovered parameters at key frames, to compute the unknown parameters at intermediate frames. This can be formulated as a keyframe interpolation problem. Specifically, given the reconstructed parameters at the starting and ending key frames, we estimate the parameters at inbetween frames in such a way that it minimizes the objective function described in Equation 5. During reconstruction, we assume the identity parameter is known and fixed. We initialize the camera parameters and pose parameters using the results obtained from non-rigid structure from motion. The expression parameters are initialized by tracking each subsequence in a sequential manner.

6 Fine-scale Detail Recovery

Dynamic facial details such as emerging or disappearing wrinkles are crucial for high-fidelity facial performance capture. This section describes our idea on using shading cues to add fine-scale surface details to large-scale deformation. Starting from large-scale deformation results, we compute per-pixel depth values associated with each input image, as well as unknown incident lighting and face albedo, by minimizing the inconsistency between the "hypothesized" and "observed" images.

6.1 Representation

We cast the fine-scale detail recovery problem as an image irradiance equation [Horn and Brooks 1989] with unknown lighting, albedo, and surface normals. We assume the face is a Lambertian surface. We further assume both lighting and face albedo are constant throughout the whole sequence. The reflected radiance equation for a Lambertian surface is formulated as follows:

$$I(x,y) = \rho R = \rho \int \max(\mathbf{l}(\omega_i \cdot \mathbf{n}(x,y)), 0) d\omega_i, \tag{7}$$

where $I(x,y)$ represents the color of the pixel located at (x,y), ρ is the albedo map, and the vector $\mathbf{l}$ indicates the lighting direction and intensity, the vector $\mathbf{n}$ is the normal at the pixel located at (x,y), and ω_i represents a subtend solid angle in 3D space. R is the irradiance of the surface.

Normal map representation. The normal for each pixel is represented by spherical coordinate (θ,ϕ), *i.e.*, $n_x = \sin\theta\cos\phi$, $n_y = \cos\theta\cos\phi$ and $n_z = \sin\phi$.

Lighting and albedo. We assume that the surface of the face is Lambertian with albedo $\rho(x,y)$. The albedo is represented as RGB values in texture space. We model the light reflected by a Lambertian surface (referred to as the reflectance function) using spherical harmonics [Basri and Jacobs 2003]

$$R(x,y) \approx \sum_{i=0}^{N} \sum_{j=-i}^{i} l_{ij}\alpha_i Y_{ij}(x,y), \tag{8}$$

where l_{ij} are the lighting coefficients of the harmonic expansion, α_i are the factors that depend only on the order i, and $Y_{ij}(x,y)$ are the surface spherical harmonic functions evaluated at the surface normal. In practice, α_i in the equation can often be omitted and the reflection function becomes

$$R(x,y) \approx \mathbf{l}^T Y(\mathbf{n}(x,y)), \tag{9}$$

with

$$\begin{array}{l} Y(\mathbf{n}(\mathrm{x},\mathrm{y})) = (1, n_x, n_y, n_z, n_x n_y, n_x n_z, n_y n_z, \\ n_x^2 - n_y^2, 3n_z^2 - 1)^T, \end{array} \tag{10}$$

where n_x, n_y, n_z are the components of the surface normals $\mathbf{n}$.

We now can synthesize an image based on the "hypothesized" lighting coefficients $\mathbf{l}$, albedo map $\rho(x,y)$ and a per-pixel normal estimate $\mathbf{n}(x,y)$:

$$I(x,y) = \mathbf{l}^T \rho(x,y) Y(\mathbf{n}(x,y)). \tag{11}$$

6.2 Objective Function

We adopt an analysis-by-synthesis strategy to reconstruct fine-scale facial geometry. Specifically, we reconstruct optimal normal maps $\mathbf{n}_i(x,y), i = 1,...,N$ representing the facial details, as well as unknown lighting coefficients $\mathbf{l}$ and albedo $\rho(x,y)$, so that the "hypothesized" image best matches the "observed" image. We formulate the problem as an optimization problem, resulting in the following objective function:

$$\underset{\rho,\mathbf{l},\{\mathbf{n}_i\}_{i=1,...,N}}{\arg\min} \quad w_1 E_{data} + w_2 E_{albedo} + w_3 E_{reg} + w_4 E_{integrability}. \tag{12}$$

In all of our experiments, the weights w_1, w_2, w_3 and w_4 are set to 1,10,15 and 50 respectively. Note that we assume the lighting and albedo are constant across the entire sequence. Therefore, we can combine shading cues throughout the whole sequence to model the unknown lighting coefficients and albedo.

The *data fitting* term, E_{data}, measures the inconsistency between the rendered and observed images at each frame:

$$E_{data} = \sum_{i=1}^{N} \left\| I_i - \mathbf{l}^T \rho Y(\mathbf{n}_i(x,y)) \right\|^2. \tag{13}$$

Similar to [Kemelmacher-Shlizerman and Basri 2011], we include the two regularization terms to reduce the reconstruction ambiguity for the albedo and normal maps, resulting in the second and third terms of the objective function:

$$\begin{array}{rcl} E_{albedo} & = & \left\| LoG(\rho) - LoG(\rho_{ref}) \right\|^2 \\ E_{reg} & = & \left\| LoG(\mathbf{n}_i) - LoG(\mathbf{n}_{i,ref}) \right\|^2, \end{array} \tag{14}$$

where E_{albedo} and E_{reg} are the prior terms that are used to constrain the reconstructed albedo and normal maps. The operator LoG represents the Laplacian of Gaussian filter. In our experiment, a texture map provided by FaceWarehouse [Cao et al. 2013b] is used as the reference albedo map ρ_{ref} and the reference normal map $\mathbf{n}_{i,ref}$ is initialized by normal maps of the reconstructed large-scale facial geometry at each frame.

The integrability term, $E_{integrability}$, is the integrability constraint described in [Horn and Brooks 1986] to ensure that the reconstructed normals can generate an integrable surface. Integrability is a fundamental mathematical property of smooth (C^2) surfaces. It restricts the independence of the surface normal, so that

$$\frac{\partial}{\partial y}\left(\frac{n_x}{n_z}\right) = \frac{\partial}{\partial x}\left(\frac{n_y}{n_z}\right). \tag{15}$$

In our implementation, this constraint is formulated as:

$$E_{integrability} = \left\| \frac{\partial}{\partial y}\left(\frac{n_x}{n_z}\right) - \frac{\partial}{\partial x}\left(\frac{n_y}{n_z}\right) \right\|^2. \tag{16}$$

Depth recovery. We now discuss how to estimate the depth from the reconstructed normal map. We represent depth information on image space. Given a pixel location (x,y), the depth value of its corresponding surface point is represented as $z(x,y)$. The surface normal $\mathbf{n}(x,y) = [n_x, n_y, n_z]^T$ can be obtained from the depth values as follows:

$$\mathbf{n}(x,y) = \frac{1}{\sqrt{p^2+q^2+1}}(p,q,-1)^T, \tag{17}$$

where $p(x,y) = \partial z/\partial x$ and $q(x,y) = \partial z/\partial y$.

We approximate $p(x,y)$ and $q(x,y)$ using forward differences by

$$\begin{array}{l} p(x,y) = z(x+1,y) - z(x,y) \\ q(x,y) = z(x,y+1) - z(x,y). \end{array} \tag{18}$$

Combining Equation (17) and (18), we obtain the following linear constraints:

$$\begin{array}{l} n_z z(x+1,y) - n_z z(x,y) = n_x \\ n_z z(x,y+1) - n_z z(x,y) = n_y. \end{array} \tag{19}$$

Once the normal map is estimated, we can compute the corresponding depth estimate by solving the following least-square fitting problem:

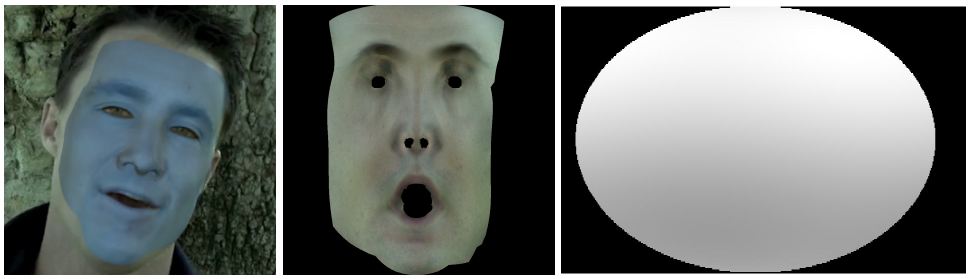
$$\underset{z_i}{\arg\min} E_{normal} + w_{d1} E_{depth1} + w_{d2} E_{depth2}. \tag{20}$$

The first term, E_{normal}, evaluates how well the reconstructed depth map matches the estimated normal map. We define the first term based on linear equations described in Equation (18).

The second term, E_{depth1}, is a Laplacian regularization term that preserves the geometric details in the reference mesh obtained from large-scale facial geometry reconstruction. We have

$$E_{depth1} \quad = \quad \left\| LoG(z_i) - LoG(z_{i,ref}) \right\|^2, \tag{21}$$

where the operator LoG represents the Laplacian of Gaussian filter.



(a) (b) (c)

Figure 4: *Reconstruction of detailed facial geometry, lighting and albedo: (a) the reconstructed facial geometry overlaid on the original image; (b) and (c) show the reconstructed albedo and lighting.*

The last term, E_{depth2}, prevents the estimated depth z_i from being away from the reference depth $z_{i,ref}$. We have

$$E_{depth2} = \left\| z_i - z_{i,ref} \right\|^2. \quad (22)$$

We introduce this term because the reference depth maps, which are initialized by results obtained by large-scale deformation reconstruction, are already close to actual facial geometry. This term is critical to estimating the absolute depth values of each pixel because neither the normal fitting term (E_{normal}) nor the Laplacian term E_{depth1} can constrain the absolute depth values. Besides, this term also provides the boundary constraints for the reconstructed depth maps. In our experiments, we set the weights of boundary pixels to a higher value ("10") in order to stabilize the boundary pixels.

6.3 Fine-scale Geometry Optimization

Similar to large-scale deformation reconstruction, we adopt keyframe based optimization for reconstructing fine-scale facial geometry. Specifically, we first estimate the lighting coefficients, albedo map, and depth maps at key frames by solving the optimization problem described Section 6.2. We then use the estimated lighting coefficients and albedo map to estimate the depth maps of the rest frames.

We use shading cues of all the key frames to estimate the unknown lighting coefficients $\mathbf{l}$ and albedo map ρ. We initialize the normal maps using results obtained from large-scale geometry reconstruction. The albedo map is initialized by the reference albedo map.

- Step 1: we estimate the spherical harmonic coefficients $\mathbf{l}$ by finding the coefficients that best fit the current albedo and the current normal maps at all the key frames. This requires solving a highly over-constrained linear least squares optimization with only nine or four unknowns (see the objective function defined in Equation (13)), which can be solved simply using least square techniques.
- Step 2: we update the albedo map $\rho(x,y)$ in texture space based on the estimated lighting coefficients $\mathbf{l}$ and the current normal map. This requires optimizing an objective function including the two terms (E_{data} and E_{albedo}). The objective function contains a linear set of equations, in which the first set determines the albedo values, and the second set smooths these values and can be optimized using linear system solvers.
- Step 3: we solve for normal maps by using the estimated lighting coefficients $\mathbf{l}$ and the updated albedo map $\rho(x,y)$. This requires solving nonlinear optimization described in Equation (12) except that we drop off the regularization term E_{albedo}. We analytically evaluate the Jacobian terms of the objective function and run a gradient-based optimization with the Levenberg-Marquardt algorithm [2009].
- Step 4: we solve for depth estimates at key frames by using the estimated normal maps. This again requires solving a least-square fitting problem described in Equation (20).

We repeat the procedure (Step 1, 2 and 3) iteratively, although in our experiments two iterations seem to suffice. Note that the number of degrees of freedom for lighting coefficients $\mathbf{l}$ can be either 4 or 9. We found similar results could be obtained by using the first order and second order harmonic approximations, while the first order approximation was more efficient. We thus use the first order approximation for light representation (*i.e.*, the length of $\mathbf{l}$ is 4).

Given the estimated lighting coefficients $\mathbf{l}$ and albedo ρ, the normal maps for the rest of the sequence are estimated by solving the objective function described in Equation (12), except that we drop off the prior term for albedo map (*i.e.*, E_{albedo}) and the optimization is done for each single frame. Similar to Step 4, we apply least square techniques to recover the depth maps from the reconstructed normal maps for the rest of the sequence. Figure 4 shows the reconstructed geometry, albedo and lighting for one frame of a test sequence.

7 Iterative Shape Refinement

Large-scale facial geometry reconstructed from high-level facial features often does not accurately match actual facial geometry because of the lack of facial features in some regions such as cheeks. We address the issue by iteratively refining large-scale facial geometry using the per-pixel normal maps obtained from fine-scale detail recovery process. In addition, we can further improve the accuracy of normal maps by using refined large-scale facial geometry. The two steps are repeated for a few times (3 in our experiment) to output the final reconstruction result. In the following, we focus discussion on the first step as the second step is the same as fine-scale detail recovery process described in Section 6.

To refine large-scale facial geometry using per-pixel normal maps, we include an extra term, *normal fitting* term, into the objective function described in Equation 5. The *normal fitting* term evaluates how well the normals of the refined large-scale geometry match the normal maps obtained from fine-scale detail recovery process. This allows us to refine large-scale facial geometry for the whole face region, especially the parts without salient features such as cheeks. In practice, fine-scale facial details such as wrinkles often dominate the normal fitting process because of large normal residuals. To address the issue, we filter the normal maps obtained from fine-scale detail recovery process by applying an exponential function to adjust the normal differences so that small differences are preserved

Figure 5: *Large-scale facial geometry refinement: (left) the input image; (middle) large-scale facial geometry before the refinement; (right) large-scale facial geometry after the refinement. Note that the nasolabial folds are lifted up with the constraint of fine-scale normal map.*

and large differences are marginalized. In our implementation, we solve large-scale geometry refinement using Levenberg Marquardt optimization [Lourakis 2009]. Note that we keep 3D head poses fixed during the optimization and only the identity and expression weights are refined. Figure 5 shows a side-by-side comparison between the original and refined large-scale facial geometry.

Displacement map. Because depth maps estimated from each frame are view-dependent, fine-scale facial details invisible to the camera are missing. The resulting depth maps are also difficult to be integrated with large-scale facial geometry for facial rendering and manipulation. To address the challenges, we bake fine-scale details (*i.e.*, the difference between the estimated depth map and large-scale facial geometry) into a displacement map. Briefly, we describe each depth pixel as 3D points in a global coordinate system and project them onto the texture space of large-scale face mesh. Note that the texture map and the texture coordinates of large-scale facial geometry are defined in advance by the artist. We generate the displacement map by computing 3D offsets between the depth points and their corresponding points on the large-scale facial geometry in the same texture space. We automatically fill in missing displacement values by using Poisson image editing technique [Pérez et al. 2003]. Therefore, we can render the reconstructed facial performances using large-scale face mesh and its displacement map using GPU accelerated displacement mapping techniques [Bunnell 2005].

8 Results and Applications

In this section, we first demonstrate the power and effectiveness of our system by capturing a wide range of high-fidelity facial performances using our proposed system (Section 8.1). We show our system achieves the state-of-the-art results on video-based facial capture by comparing against alternative systems (Section 8.2). We quantitatively evaluate the performance of our system on synthetic image data generated by high-fidelity 3D facial performance data captured by Huang and his colleagues [2011] (Section 8.3). Finally, we show novel applications of our reconstructed facial data in facial video editing (Section 8.4). Our results are best seen in the accompanying video.

8.1 Test on Real Data

We evaluate the performance of our system on video sequences of four different subjects with lengths ranging from 344 (11s) to 846 frames (28s). Three of videos (Greg, Ken and Bryan) are downloaded from the Internet. All of the test videos have a resolution of 640×360. Figure 6 shows some sample frames of our results.

8.2 Comparisons

We have evaluated the effectiveness of our system by comparing against alternative techniques.

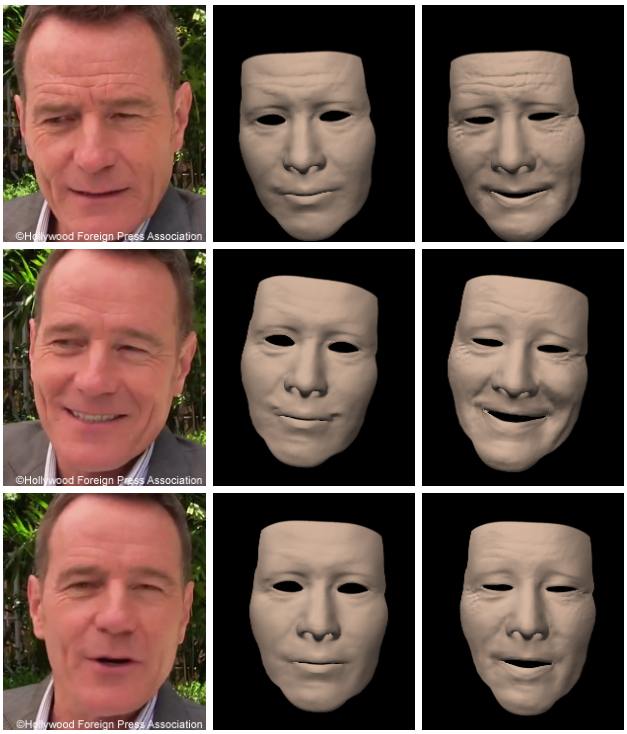


Figure 7: *Comparison against Kemelmacher-Shlizerman and Basri [2011] on real data. From left to right: input image, Kemelmacher-Shlizerman and Basri [2011], and our result.*

Comparison against Kemelmacher-Shlizerman and Basri [2011]. Our fine-scale detail reconstruction builds on the success of modeling fine-scale facial geometry from a single image [Kemelmacher-Shlizerman and Basri 2011]. Figure 7 shows a side-by-side comparison between our method and their method. The reference face template required by [Kemelmacher-Shlizerman and Basri 2011] is based on the neutral expression mesh model of the subject. The same reference albedo is used in both methods. As shown in Figure 7, our system produces more fine-scale facial details and obtain more accurate geometry than [Kemelmacher-Shlizerman and Basri 2011].

Comparison against Garrido et al. [2013]. We compare our method against state of the art in facial performance capture using monocular videos [Garrido et al. 2013]. The test video is directly downloaded from their website. The resolution is 900×600 and the total number of frames is 538. Since ground truth data is not available, we show a side-by-side visual comparison between the two results. As shown in Figure 8, our method captures richer details and produces better facial geometry than their method. Figure 8 also shows that our method is more robust to large pose variations and produces more accurate facial reconstruction results for extreme head poses.

8.3 Evaluation on Synthetic Data

We evaluate the importance of key components of our system based on synthetic data generated by high-fidelity facial data captured by Huang and colleagues [2011]. The whole test sequence consists of 300 frames with large variations of expressions and poses. We use ground truth head poses, facial geometry and texture to synthesize a sequence of color images. The resolution of image data is set to 640×480.

Figure 6: *High-fidelity facial performance capture: from left to right, we show the input image data, the reconstructed head poses and large-scale facial geometry, the reconstructed high-fidelity facial geometry overlaid on the original image data, and the reconstructed high-fidelity facial data. The subjects from top to bottom are Greg (©www.simplyshredded.com), Ken (©Ken Taylor), Bryan (©Hollywood Foreign Press Association) and Tiana respectively.*

Figure 8: *Comparisons against Garrido et al. [2013]. From left to right: input, their reconstruction results, and our results.*

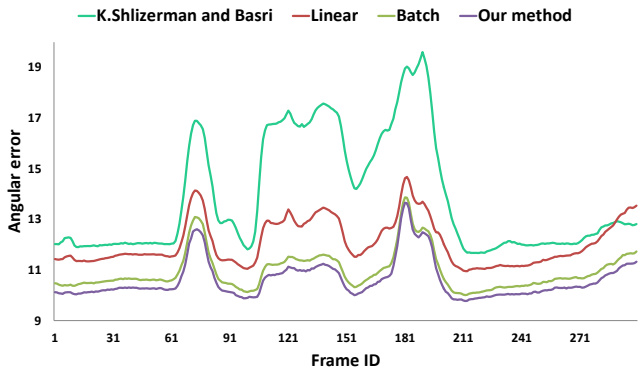


Figure 9: *Evaluation of key components of our system on synthetic image data generated by high-fidelity facial data captured by Huang and colleagues [2011].*

Figure 10: *Albedo editing: adding a beard (top) and large-scale facial geometry editing (bottom). At the bottom row, the left two images show the original and edited large-scale facial geometry. The right two images show the original and edited image data.*

The experiment is designed to show the importance of three key components of our facial reconstruction system: including "large-scale geometry and pose reconstruction" described in Section 5, "batch-based fine-scale geometry optimization" described in Section 6, and "iterative shape refinement" described in Section 7. We test four different methods on the synthetic data set. The four methods are noted as "Kemelmacher-Shlizerman and Basri", "Linear", "Batch" and "Our method", respectively. "Kemelmacher-Shlizerman and Basri" [2011] initializes the template mesh using the neutral expression model of the subject and applies the least square algorithm to solve lighting coefficients, albedo and facial geometry in each frame separately. Similar to "Kemelmacher-Shlizerman and Basri", "Linear" method solves facial geometry, lighting coefficients, and albedo in each frame separately. However, it improves "Kemelmacher-Shlizerman and Basri" by initializing the reference template model and poses using large-scale geometry and poses obtained from Section 5. "Batch" method improves "Linear" method by assuming constant lighting and albedo across the entire sequence and solving the unknown lighting coefficients, albedo and facial geometry based on shading cues throughout the whole sequence. "Our method" further improves the "Batch" method by running iterative shape refinement described in Section 7.

Figure 9 shows average reconstruction errors for all methods. We evaluate the reconstruction accuracy by computing the average normal direction discrepancy between ground truth normal maps and normal maps reconstructed from each method. As we can see, the fitting error decreases as we gradually improve the method.

8.4 Applications

With detailed facial geometry, albedo, and illumination recovered from monocular video sequences, our method allows user to easily edit the underlying geometry and albedo of the face.

Albedo editing. Albedo editing allows the user to edit the albedo map in texture space. Once the albedo map is edited, we can combine it with the reconstructed lighting coefficients and captured facial geometry to render a new sequence of images based on the rendering equation described in Equation (11). We further replace the original video data with the "synthesized" video data based on a user-specified mask. By solving Poisson equations with boundary constraints at every frame, we seamlessly blend the rendered image data with the original image data. Figure 10(top) shows two sample frames of albedo editing results, where we add a beard to the subject.

Large-scale facial geometry editing. We can edit the underlying large-scale facial geometry data at any frame and use the modified facial geometry to edit facial video data across the entire sequence. Figure 10(bottom) shows a video editing result based on large-scale face geometry editing. In this example, we modify the underlying facial geometry of the subject under the neutral expression. We then transfer large-scale facial deformation of the subject to the new subject (*i.e.*, the edited face) via the deformation transfer technique described by [Sumner and Popović 2004]. Specifically, for every pixel in the image, we find the corresponding point on the surface of the original facial geometry and project the 3D offset between the original and edited facial geometry onto the image space to obtain the corresponding image displacement. The per-pixel image displacement map is used to warp the original image into a new image (*i.e.*, the edited image). Note that the expressions of the original sub-

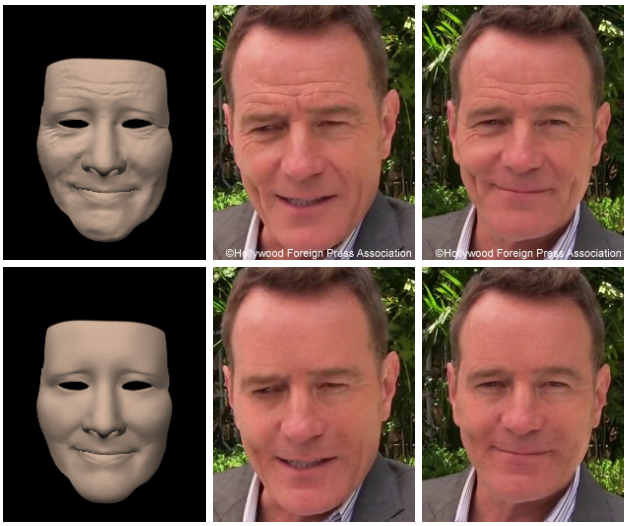


Figure 11: *Fine-scale geometric detail editing: wrinkle removal. The first column shows the original facial geometry with fine geometric details and the edited geometry after removing fine geometric details. The second and third columns show the video editing results.*

(a) (b) (c)

Figure 12: *Limitations: reconstruction artifacts caused by strong cast shadows, non-Lambertian reflectance and occlusions caused by glasses. (a) shows artifacts produced by strong cast shadows around the right eye, the nostril and the lip region; (b) shows artifacts due to non-Lambertian specular highlight on the forehead; (c) shows the reconstruction result under significant occlusions caused by sun glasses.*

ject are faithfully transferred to the new facial subject because our system can accurately reconstruct a space-time coherent 3D face geometry.

Fine-scale facial editing. In Figure 11, we modify fine-scale facial details reconstructed from the original video sequence to remove the wrinkles across the entire sequence. For this purpose, we first smooth the displacement map of each frame with a low-pass filter. We then use the edited facial geometry, as well as the reconstructed lighting coefficients and albedo, to render a new sequence of images. At a final step, we paste the "rendered" video data back to the original video data in the same way as albedo editing. Note that the shading details caused by fine-scale geometric details such as wrinkles are removed in the edited video sequence, while skin texture details and appearance variations corresponding to large-scale facial deformations are well preserved.

9 Conclusion and Limitations

In this paper, we have developed an end-to-end system that captures high-fidelity facial performances from monocular videos. The key idea of our method is to utilize automatically detected facial features and per-pixel shading cues, along with multilinear facial models, to infer 3D head poses, large-scale facial deformation and fine-scale facial detail across the entire sequence. Our system is appealing for facial capture because it is fully automatic, offers the lowest cost and a simplified setup, and can capture both large-scale deformation and fine-scale facial detail. We have tested our system on monocular videos downloaded from the Internet, demonstrating its accuracy and robustness under a variety of uncontrolled lighting conditions and overcoming significant shape differences across individuals. We have explored novel applications of captured facial performance data in facial video editing, including removing wrinkles, adding a beard, and modifying underlying facial geometry.

The current system has a few limitations. First, it ignores cast shadows, which can be created by the non-convex shapes on the face. Figure 12(a) illustrates such a concern. Fine-scale geometry is overfitted around the right eye, the nostril, and the lip region where strong cast shadows occur. Additionally, the current system assumes the face has Lambertian reflectance. Human faces, however, are not exactly Lambertian since specularities can be observed in certain face regions. For example, the specular highlight on forehead (Figure 12(b)) produces an incorrect ridge on the reconstructed facial geometry. Finally, our system cannot distinguish occlusions caused by facial hair, glasses, hands or other objects, *e.g.*, glasses shown in Figure 12(c). Though fine-scale details are still recovered, artifacts are introduced due to the occlusions caused by sun glasses. In the future, we would like to explore how to integrate shadow cues into the reconstruction framework and how to extend the current framework to handle non-Lambertian facial reflectance.

Acknowledgements

The authors would like to thank all of facial capture subjects: Tiana Zhang, Chenxi Yu and Weixiang Shi. This work is partially supported by the National Science Foundation under Grants No. IIS-1055046.

References

AMIT, Y., AND GEMAN, D. 1997. Shape quantization and recognition with randomized trees. *Neural Computation*. 9(7):1545–1588.

BASRI, R., AND JACOBS, D. W. 2003. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence. 25*, 2, 218–233.

BEELER, T., BICKEL, B., BEARDSLEY, P., SUMNER, B., AND GROSS, M. 2010. High-quality single-shot capture of facial geometry. *ACM Trans. Graph. 29*, 4, 40:1–40:9.

BEELER, T., HAHN, F., BRADLEY, D., BICKEL, B., BEARDSLEY, P., GOTSMAN, C., SUMNER, R. W., AND GROSS, M.

2011. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph. 30*, 4, 75:1–75:10.

BELHUMEUR, P. N., JACOBS, D. W., KRIEGMAN, D. J., AND KUMAR, N. 2011. Localizing parts of faces using a consensus of exemplars. In *Computer Vision and Pattern Recognition*, 545–552.

BICKEL, B., BOTSCH, M., ANGST, R., MATUSIK, W., OTADUY, M., PFISTER, H., AND GROSS, M. 2007. Multi-scale capture of facial geometry and motion. *ACM Trans. Graph. 26*, 3, 33:1–33:10.

BLANZ, V., BASSO, C., POGGIO, T., AND VETTER, T. 2003. Reanimating faces in images and video. In *Computer Graphics Forum*. 22(3):641–650.

BOUAZIZ, S., WANG, Y., AND PAULY, M. 2013. Online modeling for realtime facial animation. *ACM Trans. Graph. 32*, 4 (July), 40:1–40:10.

BRADLEY, D., HEIDRICH, W., POPA, T., AND SHEFFER, A. 2010. High resolution passive facial performance capture. *ACM Trans. Graph. 29*, 4, 41:1–41:10.

BUNNELL, M. 2005. Adaptive tessellation of subdivision surfaces with displacement mapping. In *GPU Gems 2*, M. Pharr, Ed. Addison-Wesley, 109–122.

CAO, C., WENG, Y., LIN, S., AND ZHOU, K. 2013. 3d shape regression for real-time facial animation. *ACM Trans. Graph. 32*, 4 (July), 41:1–41:10.

CAO, C., WENG, Y., ZHOU, S., TONG, Y., AND ZHOU, K. 2013. Facewarehouse: a 3d facial expression database for visual computing. *IEEE Trans. on Visualization and Computer Graphics. 20*, 3, 413–425.

COMANICIU, D., AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence. 24*, 5, 603–619.

DAI, Y., LI, H., AND HE, M. 2012. A simple prior-free method for non-rigid structure-from-motion factorization. In *Computer Vision and Pattern Recognition*, 2018–2025.

GARRIDO, P., VALGAERT, L., WU, C., AND THEOBALT, C. 2013. Reconstructing detailed dynamic face geometry from monocular video. *ACM Trans. Graph. 32*, 6 (Nov.), 158:1–158:10.

GUENTER, B., GRIMM, C., WOOD, D., MALVAR, H., AND PIGHIN, F. 1998. Making Faces. In *Proceedings of ACM SIGGRAPH 1998*, 55–66.

HORN, B. K., AND BROOKS, M. J. 1986. The variational approach to shape from shading. *Computer Vision, Graphics, and Image Processing. 33*, 2, 174–208.

HORN, B., AND BROOKS, M. 1989. *Shape from Shading*. MIT Press: Cambridge, MA.

HUANG, H., CHAI, J., TONG, X., AND WU, H.-T. 2011. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. *ACM Trans. Graph. 30*, 4, 74:1–74:10.

KEMELMACHER-SHLIZERMAN, I., AND BASRI, R. 2011. 3d face reconstruction from a single image using a single reference face shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence. 33*, 2, 394–405.

LAMDAN, Y., AND WOLFSON, H. 1988. Geometric hashing: A general and efficient model-based recognition scheme. *International Conference on Computer Vision*, 238–249.

LEPETIT, V., AND FUA, P. 2006. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 28(9): 1465–1479.

LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. 2009. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph. 28*, 5, 175:1–175:10.

LI, H., YU, J., YE, Y., AND BREGLER, C. 2013. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph. 32*, 4 (July), 42:1–42:10.

LOURAKIS, M. I. A., 2009. levmar: Levenberg-Marquardt nonlinear least squares algorithms in {C}/{C}++.

MA, W.-C., JONES, A., CHIANG, J.-Y., HAWKINS, T., FREDERIKSEN, S., PEERS, P., VUKOVIC, M., OUHYOUNG, M., AND DEBEVEC, P. 2008. Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Trans. Graph. 27*, 5, 121:1–121:10.

MATTHEWS, I., AND BAKER, S. 2004. Active appearance models revisited. *International Journal of Computer Vision. 60*, 2, 135–164.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Trans. Graph. 22*, 3 (July), 313–318.

ROSS, D. A., LIM, J., LIN, R.-S., AND YANG, M.-H. 2008. Incremental learning for robust visual tracking. *International Journal of Computer Vision. 77*, 1-3, 125–141.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (Aug.), 399–405.

SUWAJANAKORN, S., KEMELMACHER-SHLIZERMAN, I., AND SEITZ, S. M. 2014. Total moving face reconstruction. In *European Conference on Computer Vision*. 796–812.

VALGAERTS, L., WU, C., BRUHN, A., SEIDEL, H.-P., AND THEOBALT, C. 2012. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Trans. Graph. 31*, 6 (Nov.), 187:1–187:11.

VLASIC, D., BRAND, M., PFISTER, H., AND POPOVIĆ, J. 2005. Face transfer with multilinear models. *ACM Transactions on Graphics 24*, 3, 426–433.

WEDIN, P. Å. 1983. On angles between subspaces of a finite dimensional inner product space. In *Matrix Pencils*. 263–285.

WEISE, T., LI, H., VAN GOOL, L., AND PAULY, M. 2009. Face/off: live facial puppetry. In *Symposium on Computer Animation*, 7–16.

WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Realtime performance-based facial animation. *ACM Trans. Graph. 30*, 4, 77:1–77:10.

WU, C., VARANASI, K., LIU, Y., SEIDEL, H.-P., AND THEOBALT, C. 2011. Shading-based dynamic shape refinement from multi-view video under general illumination. In *International Conference on Computer Vision*, 1108–1115.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. 2004. Spacetime faces: high resolution capture for modeling and animation. *ACM Transactions on Graphics 23*, 3, 548–558.