Realtime Dynamic 3D Facial Reconstruction for Monocular Video In-the-Wild

Shuang Liu
Bournemouth University, UK

Zhao Wang
Bournemouth University, UK

Xiaosong Yang
Bournemouth University, UK

Jianjun Zhang
Bournemouth University, UK

Abstract

With the increasing amount of videos recorded using 2D mobile cameras, the technique for recovering the 3D dynamic facial models from these monocular videos has become a necessity for many image and video editing applications. While methods based parametric 3D facial models can reconstruct the 3D shape in dynamic environment, large structural changes are ignored. Structure-from-motion methods can reconstruct these changes but assume the object to be static. To address this problem we present a novel method for realtime dynamic 3D facial tracking and reconstruction from videos captured in uncontrolled environments. Our method can track the deforming facial geometry and reconstruct external objects that protrude from the face such as glasses and hair. It also allows users to move around, perform facial expressions freely without degrading the reconstruction quality.

1. Introduction

3D facial modeling is an essential technique for animation production in featured films and video games. Dedicated hardware such as depth sensors, laser scanners and camera arrays have been developed to acquire depth information for 3D model creation. However these can only be operated by trained professionals. In recent years, the wide spread availability of 2D RGB mobile cameras has sparked interest in 3D facial reconstruction from 2D input. Due to the increased interest of casual untrained users in applications such as image, video editing [34, 10], virtual makeup[29] and facial model creation [6].

Existing works based on parametric 3D facial model and shape-from-shading [32, 3, 14] are able to reconstruct minuscule detail while allowing the user to move around freely in the monocular setting. However, these methods cannot deal with structures such as hair and glasses (Fig. 2b). SFM methods [18, 11, 20], which estimate 3D structures from 2D images with different viewing angles, are able to handle these large variations. Nevertheless, the user is required to remain still while images from different angles are being taken. It involves separate capture and off-line processing phases, which is suboptimal and tedious because they require careful planning and possibly numerous trials. Moreover, feature point detection and matching on facial areas such as the cheek and forehead are more likely to fail due to the lack of highly distinctive texture pattern. Furthermore, without constraints such as controllable lighting, camera focus and limited motion, extra post-processing effort, like manual landmark adjustment, user specific model crafting and texture creation are inevitable even for state-of-the-art techniques [14, 15, 34]. Recently a method was proposed in [7], which is able to reconstruct hair but requires user input and interaction to specify 2D hair boundaries of images taken from different angles.

To address this challenge, we propose a novel realtime method that aims at automatically tracking the 3D facial performance and reconstructing the 3D geometry from monocular videos in uncontrolled environment. In order to reconstruct a dynamically deforming object, it is essential to define a rigid reference for the object. Although defining a canonical rigid reference for general object is not straightforward, facial deformation can be represented as facial expression variation. Therefore, we reconstruct the dynamic facial geometry by undoing the deformation caused by different expressions, which is made possible by a robust 3D facial tracking method. The flowchart of the proposed method is illustrated in Fig. 1.

2. Related Works

Existing 3D facial tracking and reconstruction works could be categorized as depth based and 2D image based. Although depth based methods are inherently less likely to suffer from depth ambiguity, they require special depth sensors, and therefore cannot process the vast majority of 2D recordings. Moreover, consumer grade depth sensors often fail to capture high frequency shape detail. Binocular and stereo vision systems are able to overcome the resolution

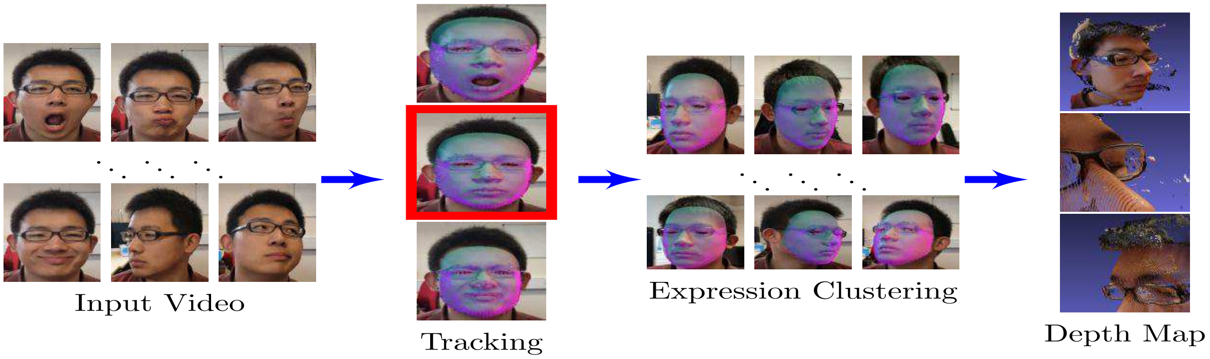


Figure 1: Given a video, a parametric 3D model [6] was fitted to noisy 2D landmarks produced from the off-the-shelf 2D landmark detector[21]. The fitted 3D model is used to refine the 3D position computed from the 2D landmarks via robust photometric tracking. The tracked 3D mesh are superimposed on the image. After clustering, video frames of similar facial expressions as seen at different viewing angles are used to compute a complete and smooth dense depth map. The glasses and hair are well reconstructed.

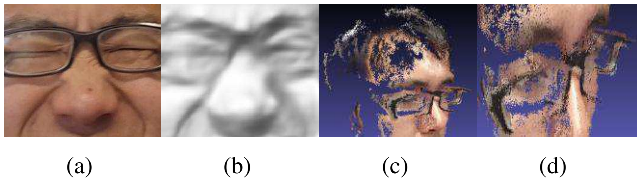


Figure 2: Shape from shading methods [32, 1] can recover minuscule details such as wrinkles, shown in 2a and 2b, however they cannot reconstruct larger geometry variation such as hair and glasses, which SFM methods [11, 12] can handle but fail to produce complete or smooth surface shown in 2c and 2d.

limits but require careful synchronization.

Ever since the release of consumer grade device such as Kinect, various methods that operate on noisy depth input have been proposed. A depth based method was introduced, which uses a parametric 3D facial model to robustly deal with the noisy depth input in [37]. Recently a state-of-the-art depth based tracking method with parametric 3D facial geometry and lighting model has been proposed for real-time facial expression transfer and reenactment in [33]. Due to the limited depth sensor resolution, RGB color input is used to supplement extra information to refine the tracking. An adaptive scheme was proposed to capture more detail with point-to-point deformation on top of blendshapes in [22]. To explicitly deal with outliers caused by occlusions, a method was proposed to segment the face and complete the occluded parts based on the blendshape in [19], which was later extended to RGB input in [28]. Binocular stereo system, on the other hand, can provide higher resolution and work in outdoor environments directly under sunlight, but are more prone to suffer from lighting variation. A robust method was introduced for a lightweight binocular system under uncontrolled lighting in [35]. Generally, for most of these state-of-the-art methods, a parametric 3D facial model is first fitted for the tracking target, which is later used for 3D tracking. These methods are quite stable as the combination of depth information the 3D facial model can effectively eliminate outliers and uncertainty.

2D image based methods are capable of processing existing footage without depth information, and are also more flexible in terms of hardware setup requirement. However, due to the lack of depth they are more likely to suffer from depth ambiguity and lighting variation. In Cao et al and Garido et al [14, 5] works personalized 3D facial models are first crafted for the tracking target semi-manually, which is later used to track the performance. Later a dynamic method was proposed to automatically track and generate personalized facial blendshape in realtime in [4]. Built on top of the robust tracking, more details were added to the person specific blendshape based on image input and user interaction in [3, 7]. The creation of photo-realistic person specific facial model can be useful for many application as seen in [34, 23], where the tracked facial performance was used to transfer the expression of source actor to the target. In order to create person specific model from a monocular rig, a method was proposed in [20], which produces facial mesh via multi-view stereo vision pipeline. To allow the geometry deformation and variations go beyond the blendshape and minuscule details, a method was proposed in [38], which physically models the anatomical structure of the face and deform the person specific model to match the monocular video input.

In summary, to the best of our knowledge, none of the reviewed works directly address the challenge of delivering a mobile-user-friendly tool that is capable of dynamically reconstructing full-scale 3D facial geometry in uncontrolled environments. To satisfy this need, we firstly propose a robust realtime 3D facial tracking method in Section 3, which obtains blendshape coefficient that is used to categorize facial deformation. After that, a novel depth reconstruction method is introduced in Section 4, where depth map is estimated for each individual expression. The performance of our method is examined in Section 5. We evaluate the quality of 3D tracking and depth reconstruction by comparing our method against popular methods [21, 4, 11, 12].

3. Robust Tracking

3.1. Parametric Model Fitting

To initialize, we first apply an off-the-shelf face detector [36] to obtain the bounding boxes. A 2D facial landmark detector [21] is trained on the dataset in [4], which has 73 landmarks that include essential facial features on the eyebrows, nose, mouth and the face contours, which are necessary in determining the neutral face shape and expression variation. In our experiments the landmark detector in [21] achieved the best trade-off between efficiency and accuracy, but for applications where real-time is not a priority the landmark detector could be swapped by more robust ones such as in [40, 41]. To reduce redundancy only representative landmarks are chosen as described in [24] as well. The landmarks in frame i is denoted as S_i, and the 3D parametric model from [6] is represented as:

$$T \times_2 U_{id}^T \times_3 U_{exp}^T = C, \tag{1}$$

where T is the data tensor and C is the core tensor. U_{id} and U_{exp} are orthonormal transform matrices, which contain the left singular vectors of the 2nd mode (identity) space and 3rd mode (expression) space respectively. In our setup we found that choosing 50 knobs for identity and 25 knobs for expression provides satisfactory approximation results.

The perspective projection operator is denoted as Π and the camera matrix is expressed as: $A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$. The intrinsic camera parameters are solved via the coordinate descent approach by iteratively fixing either the intrinsic camera parameters or the remaining parameters and solve for the others. The indices of inner facial landmarks such as nose, eyes, eyebrows and mouth are fixed, and the indices of face contours are updated in each iteration. The face contours are computed by uniformly sampling from the convex hull of projected facial contour vertices. The 3D to 2D alignment problem is solved by minimizing

$$\min_{I,E_i,R_i,t_i} \sum_i \left\|\Pi_A(I,E_i,R_i,t_i) - S_i\right\|_\epsilon, \tag{2}$$

where I, E_i, R_i and t_i denotes the identity coefficient for all frames, expression coefficients and the 3D rodrigues rotation and translation vector for frame i respectively. To minimize the effect of outliers in landmark detection, the robust Huber loss is applied, where ϵ controls the tolerance for outliers.

$$\|\delta\|_\epsilon = \begin{cases} \frac{1}{2}\delta^2 & \text{for}|\delta| \le \epsilon, \\ \epsilon\,|\delta| - \frac{1}{2}\epsilon^2 & \text{otherwise.} \end{cases} \tag{3}$$

The projection function is nonlinear but differentiable. Firstly the rotation and translation are solved via direct linear transform. Then all parameters are solved jointly via the Levenberg Marquardt algorithm [25]. Empirical experiments show that trust region optimization method converges to more natural expression, identity coefficient and smaller error, than line search and coordinate descent methods. Since the identity coefficient is the only parameter that affects all frames, the zero pattern in the normal equations is exploited to reduce the computational cost [9]. To keep the blendshape within valid range a box constraint is simulated which clamps the expression and identity coefficient parameter within the column-wise minimum l and maximum u of U_{id}^T and U_{exp}^T. The box constraint is achieved by variable transformation as:

$$f(x) = \frac{u+l}{2} + \frac{u-l}{2} \cdot \tanh((x - \frac{u+l}{2})/\frac{u-l}{2}), \tag{4}$$

and the corresponding transformed partial derivative is

$$f'(x) = \partial x - \partial x \cdot \tanh((x - \frac{u+l}{2})/\frac{u-l}{2})^2). \tag{5}$$

3.2. Photometric Tracking

To robustly track the object in new incoming frames, the photometric difference between the rendering and image is minimized, which greatly benefits the tracking quality because of automatic occlusion handling back faces culling. Given the parametric model computed from a few landmarks and images, realistic rendering is synthesized for previously unseen angles.

Due to the noisy environment and complex lighting in real world situations, we propose to simply use the median of extracted surface maps as a robust approximation of the face texture, which is updated during tracking. Empirically the experimental results show that such a low cost and straightforward approximation achieves similar performance to existing works that explicitly estimate the illumination and albedo of the face. The smoothness term is applied on a small window of 10 frames because longer sequences might not necessarily improve the accuracy, and slow down the computation. As a result, instant feedback of tracked 3D performance is provided since it is not essential for the 3D reconstruction.

Given the ith frame, the rendering function is defined as Φ and the target energy as:

$$\min_{P} \sum_{i} \|\Phi(\Pi_A(\mathbf{P_i})) - F_i\|_\epsilon + \beta \cdot \Theta(\mathbf{P}), \tag{6}$$

where $\mathbf{P}$ denotes the set of parameters E, R and t that are used to synthesize a virtual view given the texture map. A L2 smoothness term $\Theta(X)$ controlled by β is used on the expression and pose parameters is to exploit the temporal coherence, which is defined as

$$\Theta(X) = \|XO\|, \tag{7}$$

where $O \in \mathbb{R}^{n \times n}$ is a symmetrical matrix defined by

$$O = \begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix}. \tag{8}$$

The eigen-decomposition of $\beta \cdot \Theta(\mathbf{P})$ is actually the n-by-n type-2 discrete cosine transform (DCT) and inverse DCT (IDCT) matrices and can be directly solved [13].

Both the rendering and photometric evaluation function carry high computational cost. The search radius of E, R and t can be clamped to reduce the computational cost, which is calculated with respect to β as in Equation. (4, 5). To make intensity difference term $\rho = \|\Phi(\Pi_A(\mathbf{P})) - F_i\|_\epsilon$ differentiable, it is linearised via Taylor expansions approximation yielding the following equation:

$$\rho = \Phi(\Pi_A(\mathbf{P})) - \nabla F(x+u) \cdot (p-u) - F(x+p), \tag{9}$$

where u is image coordinate difference in F and p is the projected coordinate, ∇F is computed from 3×3 Sobel kernel convolution. The computational cost of evaluating the simulated Hessian matrix in trust-region methods at per-pixel level becomes a bottle neck. Hence we switch to line search method [2] with simulated Hessian matrix computed from previous gradient directions.

Even with reduced search radius, evaluating per-pixel is expensive when the input resolution is high. We take advantage of the high parallel capacity of GPU to achieve lower latency. The face reconstructed from the core C only contains points on the mesh, hence we render per-vertex smoothed coordinates and colour of the mesh with as a texture, then use CUDA/OpenGL interoperability to directly read from GPU memory and evaluate the cost function and derivative on GPU. It is only necessary to update the rendering in outer iteration to keep the line search stable and reduce data transfer. The native support of texture on GPU also allows fast sub-pixel interpolation, which provides higher accuracy.

The whole procedure of photometric tracking is summarized in Algorithm. 1. The error term is the accumulative photometric and smoothness penalty error. The smoothness penalty is shrunk by a factor between $[0, 1]$. We find that 3 iteration and a shrunk factor of 0.9 lead satisfactory results for most scenarios.

Algorithm 1 Photometric tracking

```
$I, E, R, t \leftarrow$ landmarks fitting
while error delta $>$ threshold do
    Texture $\leftarrow I, E, R, t$
    Smooth$(E, R, t)$
    Track$(E, R, t)$
    Shrink smoothness penalty
    iteration $\leftarrow$ iteration $+ 1$
    if iteration $>$ max iteration then break
```

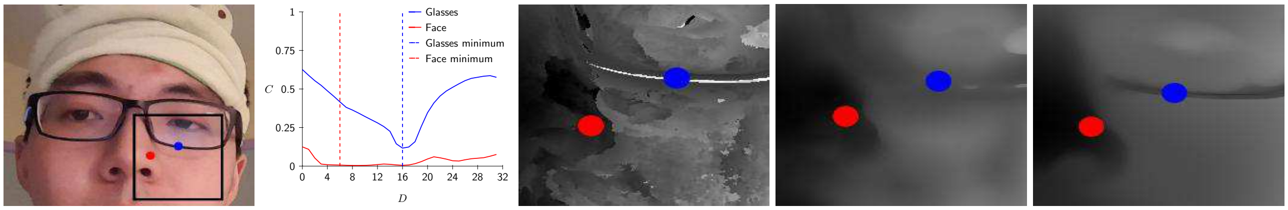
4. Depth Estimation

Considering facial deformation can be semantically represented by facial expression variation, we reduce the dynamic depth reconstruction problem to a series of static ones for each individual expression. Since facial deformation is expressed with blendshapes, the canonical rigid reference for each expression is established via clustering the blendshape coefficients. For each cluster we select a source frame with most visible facial area, and the incoming frames are assigned with their corresponding clusters and used as photometric measurement target frame. The number of clusters is updated during tracking to reflect the fact that more expressions have been observed, which is performed by ensuring the standard deviation of a cluster is lower than a threshold. The choice of this threshold would influence the number of views required for each expression and the reconstruction quality, where we found that for the 25 dimension expression coefficient a standard deviation of 0.1 is a good compromise.

We denote the inverse of rotation R^{-1} and translation t^{-1} from source R_s and T_s to target image R_t and t_t as $R^{-1} = R_t^T \times R_s$ and $t^{-1} = t_t - R_t^T \times t_s$. The average photometric error $C(u, d)$ of pixel u in reference image F_r and target image F_t with the inverse depth d is defined as

$$C_r(u, d) = \sum |F_r(u) - F_t(\Pi_A(A^{-1} \times [u, d], R^{-1}, t^{-1}))|. \tag{10}$$

Since the relative movement of face to the camera is small, the minimum average sum of the photometric error should correspond to the correct depth under the intensity



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

Figure 3: Take the blue and red points on the reference image 3a for example, the red point on smooth varying face surface has multiple local cost volume minimums while the blue point on the glasses that protrude from the face has one clear minimum close to the true value as seen in 3b. The low quality of pixel intensity as the matching feature leads to ambiguous noisy local minimums in 3c. Moreover, as observed in the plot, a large range of depth values are not useful and therefore should not be searched during iteration. Evidently, if a general scheme is used such as the one in [26], the reconstructed depth in 3d can only resemble the essences of true object, whereas the depth in 3e reconstructed by the proposed scheme is more accurate.

consistent assumption. However, the per-pixel minimum might not necessarily be accurate or lead to smooth surface due to factors such as specular reflection, self occlusion and intensity ambiguity in texture-less areas.

In order to solve this, previous methods [26, 17, 16] have employed a total variational minimization to remove noise. The goal is to minimize the gradient of depth map D to produce smooth surface and preserve depth discontinuity around edges, which is achieved via minimizing

$$\min_{D} \int_{\Omega} |\nabla D(u)| + \lambda C(u,d), \tag{11}$$

where ∇ is the distributional derivative and Ω is the image domain. The variational term $|\nabla D|$ is convex whereas the data fidelity term $\lambda|C(u,d)|$ is non-convex. A convex approximation of the data fidelity term controlled by λ can be obtained by linearizing the cost volume and solving the resulting approximation iteratively within a coarse-to-fine warping scheme. This would require keeping all the images thus significantly increasing the computational cost. Since the aim is to process long video sequences that contain as much expression and poses as possible, we follow the approach in [26], in which the energy functional is approximated by coupling the data and regularization terms through an auxiliary variable α.

$$\min_{D,\alpha} \int_{\Omega} G|\nabla D(u)|_{\epsilon} + \lambda C(u,\alpha(u)) + \frac{1}{2\theta}\|D(u) - \alpha(u)\|, \tag{12}$$

Although L1 total variation is robust to outliers, it suffers from the stair-case effect. One could alleviate this effect by applying Huber norm on the weighted variational term as $G|\nabla D|_{\epsilon}$, where $G = e^{-\nabla F}$ is the image gradient of the reference image computed from Equation 17, which is optionally normalized and scaled to reflect the smoothness regularization strength on edge boundaries. For continuous surface the Gaussian noise smaller than ϵ is smoothed by L2 norm while larger depth discontinuity are filtered by L1 norm.

Although the cost-volume is discrete, sub-sample refinement could be computed from performing a single Newton step using numerical differentiation of the coupling term $E(u,d,\alpha) = \lambda C(u,\alpha) + \frac{1}{2\theta}\|D - \alpha\|$.

$$\bar{\alpha} = \alpha - \frac{\nabla E(u,d,\alpha)}{\nabla^2 E(u,d,\alpha)} \tag{13}$$

To produce a smooth surface, one limitation of such approximation is that the cost volume needs to be sampled at a very high rate with every possible depth. Note that a rough model of the face is readily available from the parametric model, it is used as a prior to accelerate the iteration and generate more accurate results. Based on this, several modifications are introduced to the original update scheme, which significantly speeds up the optimization. The effectiveness of our proposed scheme is shown in Fig. 3.

1. The search radius is set according to the photometric tracking error. Because detail not included in the parametric model is less likely to be correctly captured in the median texture, a larger search radius is used for pixels with bigger error. The search radius s is set to be positive correlated to sum of intensity difference between the synthesized rendering and the real images, and the search range is centered around the depth of face model r,

$$\alpha \in [r - s, r + s]. \tag{14}$$

2. When solving the auxiliary variable α in each iteration, if the absolute difference $|C(u,\alpha) - C(u,r)| < \epsilon$, the auxiliary variable α is set to r instead of performing the single Newton step refinement.

3. Assuming there is no major facial modification, the search radius is limited to the visible range $d < r$,

where r is the depth value on the face model. For pixels not on the face model the search radius is set to the distance between the lowest and highest depth value of the face model.

As the coupling energy term θ becomes larger, the feasible search range of auxiliary variable is shrunk as well. Given the cost volume minimum and maximum of a pixel in current range, the coupling energy dictates that the solution should lie in the following bound,

$$C_u^{min} + \frac{1}{2\theta}\|D - \alpha\| \leq min(C_u^{max}, C(u, r)), \quad (15)$$

and the updated search radius is

$$s = 2\theta \cdot \lambda(min(C_u^{max}, C(u, r)) - C_u^{min}) \quad (16)$$

Following [31], the duality principles leads us to the primal-dual form of Equation 12, where the primal variable is α and denote the dual variable is denoted as q. It is essential for the gradient operation ∇ that operates on the dual variable q to be different from the one that operates on image in Equation 9, in order for the Stokes theorem to hold exactly. The gradient of depth map D is computed with forward differences with Neumann boundary condition. The divergence of the dual variable q, which is the adjoint of the gradient of D, is computed with backward differences. For image of size (W, H), the numerical scheme is detailed as follows:

$$\frac{\partial D(i,j)}{\partial x} = \begin{cases} D(i+1, y) - D(i,j) & \text{if} 1 \leq i \leq W, \\ 0 & \text{otherwise.} \end{cases}$$
$$\frac{\partial D(i,j)}{\partial y} = \begin{cases} D(i, j+1) - D(i,j) & \text{if} 1 \leq j \leq H, \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

$$div(p) = \begin{cases} q_x(i,j) - q_x(i-1,j) & \text{if} 1 \leq i \leq W, \\ q_x(i,j) & \text{if} i = 1, \\ -q_x(i-1,j) & \text{otherwise.} \end{cases}$$
$$+ \begin{cases} q_y(i,j) - q_y(i,j-1) & \text{if} 1 \leq j \leq H, \\ q_y(i,j) & \text{if} j = 1, \\ -q_y(i,j-1) & \text{otherwise.} \end{cases} \quad (18)$$

Following the duality-based algorithm in [8], σ is selected as $\sigma = \tau\frac{1}{\mathcal{L}^2}$, $\mathcal{L} = 8$, $\theta = 1$ and $\tau = 0.01$. The Huber norm control variable ϵ is set based on the search grid of the cost volume. The dual and primal variable is minimized in an alternating manner by fixing one while solving for the other:

1. Fixed α, solve

$$\min_D \int_\Omega |\nabla D|_\epsilon + \frac{1}{2\theta}\|D - \alpha\|. \quad (19)$$

The gradient ascent is performed on $\partial D = 0$, yielding

$$q^{n+1} = \Psi(\frac{q + \sigma G \nabla D}{1 + \sigma\epsilon}),$$
$$D^{n+1} = \frac{D^n + \tau(G \cdot div(q^{n+1}) + \frac{\alpha}{\theta})}{1 + \sigma\epsilon}. \quad (20)$$

where $\Psi(x) = \frac{x}{max(1, \|x\|)}$ is the resolvent operator that projects the gradient ascent step back onto the unit ball.

2. Fixed D, solve

$$\min_\alpha \int_\Omega \frac{1}{2\theta}\|D - \alpha\| + \lambda C(u, \alpha), \quad (21)$$

which is achieved via point-wise exhaustive search with the aforementioned scheme.

The point-wise search is independent of its neighbors and trivially parallelizable on modern GPU. The update for q and D on the other hand depends on its neighbors. Thus we use CUDA warp shuffles, which enable different processing units in the same wrap to share value through register and avoid reading/writing from global memory to reduce the overhead of syncing.

5. Experiments

In this section we detail the performance and implementation of our method. All of the experiments were done on a desktop PC with Intel Xeon (3.5 GHz), 32 GB RAM and GTX 980 graphics card. We designed two separate set of experiments to verify the effectiveness of our method. First we compare the facial performance tracking quality of our method to that of existing facial landmark tracking methods in uncontrolled setting. Next we compare the depth estimation accuracy to structure from motion methods where the person remain still and the camera position changes.

The 2D landmark detection takes one millisecond to compute as suggested by the title in [21]. The surface parametrization is only performed once when the parametric model was initially being fitted to the 2D landmarks, which takes 100ms. For 1080p videos, OpenGL rendering takes 5ms, error evaluation and derivative computation taking 2ms and the smoothing operation takes less than 1ms. For depth map with a size of 600 × 800 pixels, the cost volume aggregation takes 5ms to execute with a search grid resolution of 64 levels. The tracking and photometric error computation runs on average around 35 fps. The denoising

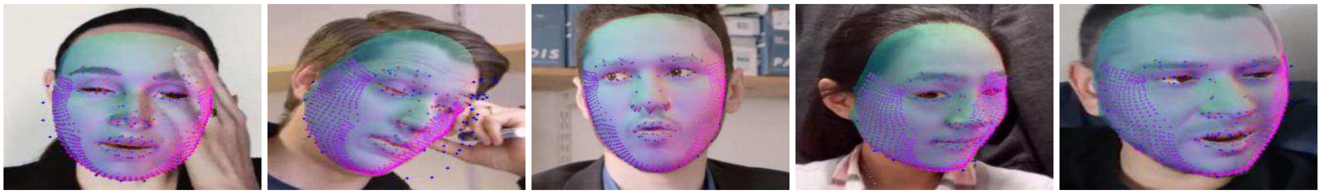


Figure 4: Our method is able to provide accurate 3D tracking which is crucial for successful depth estimation. Tracked 3D mesh is superimposed and the noisy 2D landmarks for initialization are shown in blue points.

takes 100 iterations that finish within 110ms, and as a result the denoised depth map is generated per user request instantly.

5.1. Robust Tracking

To evaluate the tracking performance, we compare our method with existing facial landmark detection methods on the video dataset in [30], as well as with our own recordings in tough situations, which are either downloaded from Youtube or recorded with a Samsung Galaxy S6 smart phone. Qualitative results are shown in Fig. 4, more of which can be found in the supplementary material.

The benchmark dataset (300 V-W) [30] consists of videos recorded in uncontrolled environment with manually labeled landmark ground truth. We redefine indices of the 3D parametric model the landmarks according to the protocol in [30]. Comparative results with existing methods are illustrated in Table 1. Although our landmark detector is based on [21], which did not achieve the best result, building on top of its output our method achieved the best result on the challenging subset and fullset.

5.2. Depth Estimation

To evaluate the proposed depth estimation method we compared the reconstructed depth map of the same recording to that of [26], which took a similar approach for real-time general object reconstruction. The quantitative result is shown in Fig. 3d and 3e. At first glance, the depth map produced by [26] roughly captured essences of the face. However, closer inspection revealed that it failed to produce an equally accurate representation as the proposed method.

Method	Common Subset	Challenging Subset	Fullset
ERT [21]	6.11	14.7	6.40
SDM [39]	6.12	14.1	6.14
LBP [27]	6.03	13.9	6.11
DDE [4]	5.45	11.9	6.32
Ours	**4.97**	**6.98**	**5.11**

Table 1: The qualitative comparison with existing methods measured in averaged errors on the 300 V-W [30], results taken from existing executable and literature. Note that both our method and [4] needs a few frames to start up, we excluded the results of first 3 seconds in each video.

Method	Average Error	Pose (s)	Depth (s)
PMVS [11]	1.4	25	283.4
GIPUMA [12]	1.1	25	95.9
Ours	**0.4**	**2.12**	**1.62**

Table 2: The average error is computed from the squared error of the facial area to the Kinect Fusion scan. Results of PMVS [11] and GIPUMA [12] are computed from 35 images, which are selected manually to cover most of the facial area and contain the least amount of motion blur. Results of our method are computed from 10s 30FPS short clips of the person. Example fused depth map of [11, 12] are shown in Fig. 2

For qualitative comparison between our method and SFM methods [11, 12], we measure the average computation time and the RMSE (cm) error of the reconstructed depth map compared to the real physical face. It is shown in Table 2 that our method achieved the lowest error and need the least amount of time. We showcase novel views generated from the depth map reconstructed by our method as well as the perspective-aware portrait photos manipulation results in Fig. 5, which is inspired by [10]. Today most photos are taken using mobile devices with fixed focal length. With the high quality depth map, images captured by fixed focal length camera can be modified to simulate results captured with different focal length. More comprehensive results and dynamic examples can be found in the supplementary material.

6. Conclusion

We have proposed a novel method for dynamic 3D facial reconstruction from monocular videos in uncontrolled environments. The key contribution is that we propose to reconstruct the depth of the face by undoing facial deforma-

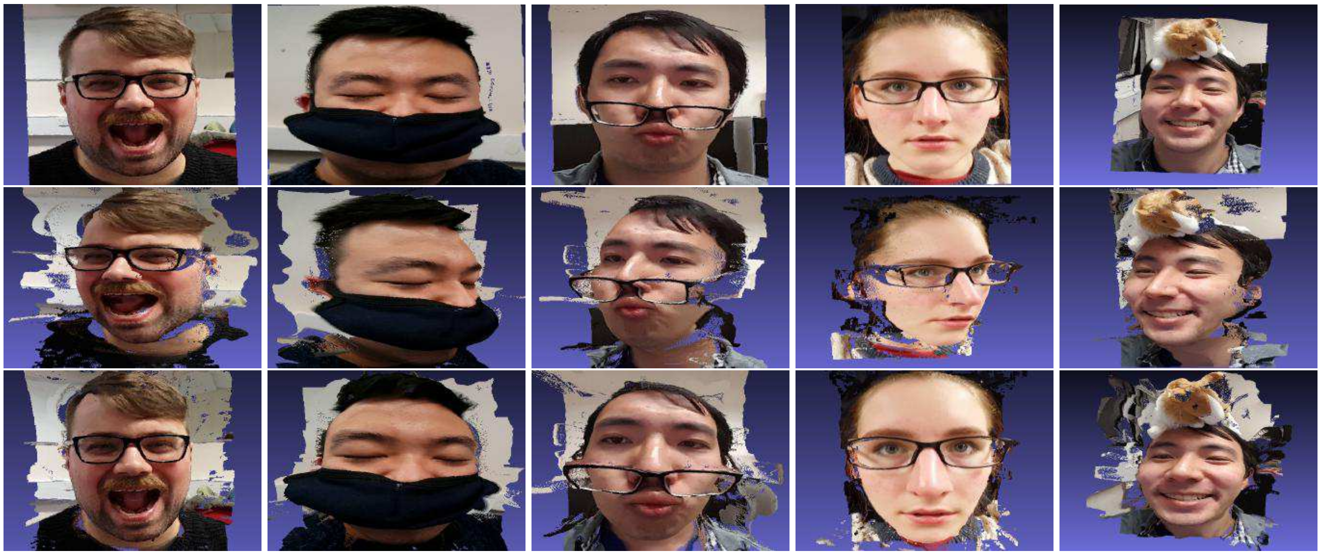


Figure 5: From top to bottom: reference view, novel unseen views and perspective-aware portrait photos manipulation based on the depth map obtained from our method.

tion, which is achieved by 3D facial performance tracking and expression coefficient clustering. Our method can be adapted to many applications that require 3D information of the face.

Experimental results show that our method is able to perform robust 3D facial tracking even from noisy output produced by the 2D landmark detector. Moreover, our method is able to produce realistic facial surface while preserving large facial geometry variation. Although our method only generates depth maps at the moment, we will investigate creating morphable 3D volumetric models for dynamic facial expression transfer and video retargeting in the future.

7. Acknowledgement

We would like to thank Kun Zhou for providing the FaceWarehouse dataset. The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement n° [612627].

References

[1] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2015.

[2] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, 2006.

[3] C. Cao, D. Bradley, K. Zhou, and T. Beeler. Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics (TOG)*, 34(4):46, 2015.

[4] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)*, 33(4):43, 2014.

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

[6] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014.

[7] C. Cao, H. Wu, Y. Weng, T. Shao, and K. Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Transactions on Graphics (TOG)*, 35(4):126, 2016.

[8] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[9] T. A. Davis. *Direct methods for sparse linear systems*. SIAM, 2006.

[10] O. Fried, E. Shechtman, D. B. Goldman, and A. Finkelstein. Perspective-aware manipulation of portrait photos. *ACM Transactions on Graphics (TOG)*, 35(4):128, 2016.

[11] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010.

[12] S. Galliani, K. Lasinger, and K. Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015.

[13] D. Garcia. Robust smoothing of gridded data in one and higher dimensions with missing values. *Computational statistics & data analysis*, 54(4):1167–1178, 2010.

[14] P. Garrido, L. Valgaerts, C. Wu, and C. Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Trans. Graph.*, 32(6):158–1, 2013.

[15] P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt. Reconstruction of personalized 3d face rigs from monocular video. *ACM Transactions on Graphics (TOG)*, 35(3):28, 2016.

[16] G. Graber, J. Balzer, S. Soatto, and T. Pock. Efficient minimal-surface regularization of perspective depth maps in variational stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–520, 2015.

[17] G. Graber, T. Pock, and H. Bischof. Online 3d reconstruction using convex optimization. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 708–711. IEEE, 2011.

[18] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[19] P.-L. Hsieh, C. Ma, J. Yu, and H. Li. Unconstrained realtime facial performance capture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1675–1683, 2015.

[20] A. E. Ichim, S. Bouaziz, and M. Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)*, 34(4):45, 2015.

[21] 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.

[22] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.*, 32(4):42–1, 2013.

[23] S. Liu, X. Yang, Z. Wang, Z. Xiao, and J. Zhang. Real-time facial expression transfer with single video camera. *Computer Animation and Virtual Worlds*, 27(3-4):301–310, 2016.

[24] S. Liu, Y. Zhang, X. Yang, D. Shi, and J. J. Zhang. Robust facial landmark detection and tracking across poses and expressions for in-the-wild monocular video. *Computational Visual Media*, 3(1):33–47, 2017.

[25] J. J. Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[26] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

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

[28] 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.

[29] K. Scherbaum, T. Ritschel, M. Hullin, T. Thormählen, V. Blanz, and H.-P. Seidel. Computer-suggested facial makeup. In *Computer Graphics Forum*, volume 30, pages 485–492. Wiley Online Library, 2011.

[30] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *Computer Vision Workshop (ICCVW), 2015 IEEE International Conference on*, pages 1003–1011. IEEE, 2015.

[31] J. Stühmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *Joint Pattern Recognition Symposium*, pages 11–20. Springer, 2010.

[32] S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. M. Seitz. Total moving face reconstruction. In *European Conference on Computer Vision*, pages 796–812. Springer, 2014.

[33] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, 34(6):183, 2015.

[34] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.

[35] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Trans. Graph.*, 31(6):187, 2012.

[36] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[37] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. In *ACM Transactions on Graphics (TOG)*, volume 30, page 77. ACM, 2011.

[38] 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.

[39] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 532–539, 2013.

[40] Y. Zhang, S. Liu, X. Yang, D. Shi, and J. J. Zhang. Sign-correlation partition based on global supervised descent method for face alignment. In *Asian Conference on Computer Vision*, pages 281–295. Springer, 2016.

[41] Y. Zhang, S. Liu, X. Yang, J. Zhang, and D. Shi. Supervised coordinate descent method with a 3d bilinear model for face alignment and tracking. *Computer Animation and Virtual Worlds*, 28(3-4), 2017.