2013 13th International Conference on Computer-Aided Design and Computer Graphics

High Quality Binocular Facial Performance Capture From Partially Blurred Image Sequence

Jian Jiang* Ming Zeng† Bojun Liang* Xinguo Liu*

*State Key Lab of CAD&CG, Zhejiang University
Hangzhou, China

†Software School of Xiamen University
Xiamen, China

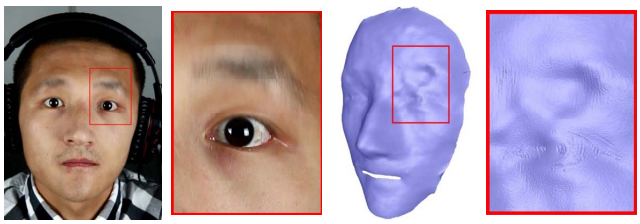
***Abstract*—Existing methods on passive facial performance capture assume that the input images are well captured. They merely consider how to deal with motion blurred images in the input sequence, which is very common in the image capture process.This paper presents a collection of novel algorithms and a thereby resulting system to reconstruct high quality facial dynamic geometry even from a partially blurred image sequence. In our method, we adopt binocular cameras to capture a stereo sequence. With this sequence,we first estimate depth map for each frame using a state-of-the-art stereo matching method.Then, based on the estimated depth map sequence, we track facial motion by leveraging constraints of both optical flow and geometry. In this step, a blur detection and regularization algorithm are devised to adaptively keep both shape and details. Finally, we synthesize temporal mesoscopic geometry on the blurred region from clear image texture of neighboring frames. We conduct extensive experiments on several sequences containing facial performance with blurred region, and the results demonstrate the effectiveness and robustness of our algorithms.**

***Keywords*-facial performance capture; motion blur; temporal detail synthesis**

I. Introduction

Over the past decade, facial performance capture have been playing a more and more important role in both games and movie industries. The high-end applications such as special effects for movies require the performance capture to produce high quality results in both spatial and temporal domains.

On the avenue to meet this demand, computer graphics and computer vision researchers have proposed various methods and systems by either improving data acquisition setup or exploring more information from the input data. Some use makers, paints[1], [2] or active light[3] to aid correspondence finding. As the increase of the resolution of digital cameras, a passive way to reconstruct facial geometry and motion from image sequence without additional marker or active lighting is gradually prevalent. Along this direction, Bradley et al.[4] utilized multiple camera to capture images of facial motion, and adopted 3D reconstruction and tracking algorithms to obtain the final facial dynamic geometry. Furthermore, Beeler et al.[5] developed a system featuring with much more spatial and temporal details based on their previous work on high quality face reconstruction method[6]. Most of these efforts, although their variety on techniques, rely on special hardware or carefully arranged cameras, lighting as well as experienced users to obtain high-quality input data. To bring facial performance techniques to usual users, Valgaerts et al.[7] recently proposed a lightweight binocular facial performance capture system which do not require a strict environment to acquire data.



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

Figure 1: (a) is a snapshot of a blurred sequence. (c) is the reconstruction of this frame (using method of [5]), where the blurred region is bad. (b) and (d) are their zoom-in views.

Although the above mentioned progresses exist, there is not a work yet concentrating on facial performance capture using blurred images. The blurred images will severely corrupt correspondence matching not only in stereo images but also in consecutive frames, leading to bad per-frame geometry and tracking results.

Aiming at addressing this problem, this paper investigates how to reconstruct facial performance from partially blurred image sequence while still keeping high-quality results. We take the motivation that the blurred regions merely supply geometry or temporal information, thereby these regions should have less influence than clear regions in the motion tracking procedure. Based on this intuition, we develop a novel facial performance capture system which is specially tailored to handle image sequence containing blurred regions, while still producing pore-level reconstruction results (see Fig. 1). This systematical innovation is composed of the following technical contributions:

- design a practical blurred region detection which generates a blurry map indicating the intensity of motion blur on each spatial position.

Corresponding author: Xinguo Liu, email:xgliu@cad.zju.edu.cn

978-1-4799-2576-6/13 $31.00 © 2013 IEEE
DOI 10.1109/CADGraphics.2013.33

- develop a novel face tracking algorithm combining optical flow, geometry constraints, and blur-aware regularization to accurately track facial geometry/motion on the blurred regions.
- synthesize pore-level details on blurred regions from the corresponding clear regions of neighboring frames.

II. Related Work

There are a great deal of work on facial performance capture. In this section, we review the related work on facial geometry and motion capture. First are methods using commodity RGB/RGBD sensors, followed by techniques using markers and active light, and finally are passive capture methods based on high resolution cameras.

Face fitting using commodity RGB/RGBD sensors. This kind of methods adopt a deformable face model to compute a set of parameters which best explain the color images or/and depth maps streamed from commodity RGB/RGBD sensors. For instance, Li et al.[8], Pighin et al.[9], Blanz et al.[10], Zeng et al.[11], and Cao et al.[12] capture facial motion from images, while Weise et al.[13] and Li et al.[14] utilize images together with corresponding depth data to improve robustness of motion capture. Since these methods track the face motion in a low-dimensional expression subspace, they can only produce low resolution face geometry and motion.

Facial capture using markers and active light. To robustly track face performance, markers and some active light are used to provide strong feature patterns. Bickel et al.[1], Furukawa et al.[2], and Huang et al.[15] manually places markers on the face, and robustly track face motion by localizing these makers. One main shortcoming of this kind of methods is the tedious manual maker placement. Zhang et al.[3], Weise et al.[16], Ma et al.[17], Alexander et al.[18], Wilson et al.[19], and Fyffe et al. [20] leverage projectors or controlled lighting system to project special illumination patterns on the face, so as to explore more feature correspondence or geometry details. However, the active intrusion hinders the simultaneous reconstruction of geometry and texture.

Passive capture using high-res cameras. Passive reconstruction requires no markers, structured light, or other specialized hardware, which has been attracting more and more interests from research community recently. Along this avenue, Beeler et al. [6] proposed to reconstruct the static face geometry with pore-level details. Then, Bradley et al.[4] built a studio with thousands of LED lights and a multiple-camera array to automatically reconstruct facial geometry and motion. Using a similar setup, Beeler et al.[5] further developed a system able to capture details both spatially and temporally. To bring such relative high-end systems to usual users, Valgaerts et al.[7] proposed a easy-to-deploy system which only consists of a binocular camera pair and can work well under uncontrolled lighting. This work lower the bar to facial performance capture. However, this work and other previous works do not concentrate the study on geometry/motion capture from a blurred image sequence, and they lack a thorough analysis on the reconstruction result in this case.

To fill this gap and make the facial performance capture technique more accessible to inexperienced users, we study to capture high-quality facial geometry and motion even from partially blurred image sequence. We use a binocular camera stereo under natural lighting as our acquisition setup, and take a similar system flow like works of Beeler et al.[5] and Bradely et al.[4]: build geometry for each frame, and track face to these per-frame geometry. For each stage of the flow, we adapt each part to handle motion blurred data without compromise on reconstruction quality.

III. Our Approach

Our approach takes a stereo video sequence as its input. As illustrated in Fig. 2, the flowchart of our approach consists of the following stages:

1) **Per-frame mesh reconstruction.** We reconstruct a mesh for each frame of the binocular sequence, and build the initial mesh from the first frame for motion tracking.
2) **Anti-blur facial motion tracking.** The task of this stage is to track the initial mesh to the following frames, according to the texture change of the images. In this stage, we adopt the anchoring scheme of Beeler et al.[5] to limit temporal drift. Meanwhile, to handle the blurred regions on the images, we design a optical flow and geometry based tracking and a blur-aware regularization to adaptively regularize face shape.
3) **Temporal detail synthesis.** This stage synthesize temporally coherent mesoscopic details across frames. Our methods take special care on the blurred regions by transferring details from neighboring frames.

Notations. We denote a frame by F^t, which contains a pair of images $(I^t_{left}, I^t_{right})$ at time t. Thus, the consecutive frames for time t are F^t and F^{t+1}. We further denote the time starting to capture by t_0.

IV. Anti-Blur Facial Performance Capture

A. Per-frame Mesh Reconstruction

Per-frame point cloud reconstruction. Each frame F^t in the recorded sequence is processed using the method of Beeler et al.[6]. The result is the corresponding per-frame 3D point cloud P^t of the face shape.

Initial mesh building. Notice that we have build a dense point cloud in which each 3D point corresponds to a 2D image pixel. We use the adjacency of the pixels to build the mesh M^t from the point cloud P^t for each frame at time t.

Now we obtain a mesh for each frame and the initial mesh M^{t_0}. The subsequent frame take M^{t_0} as the reference

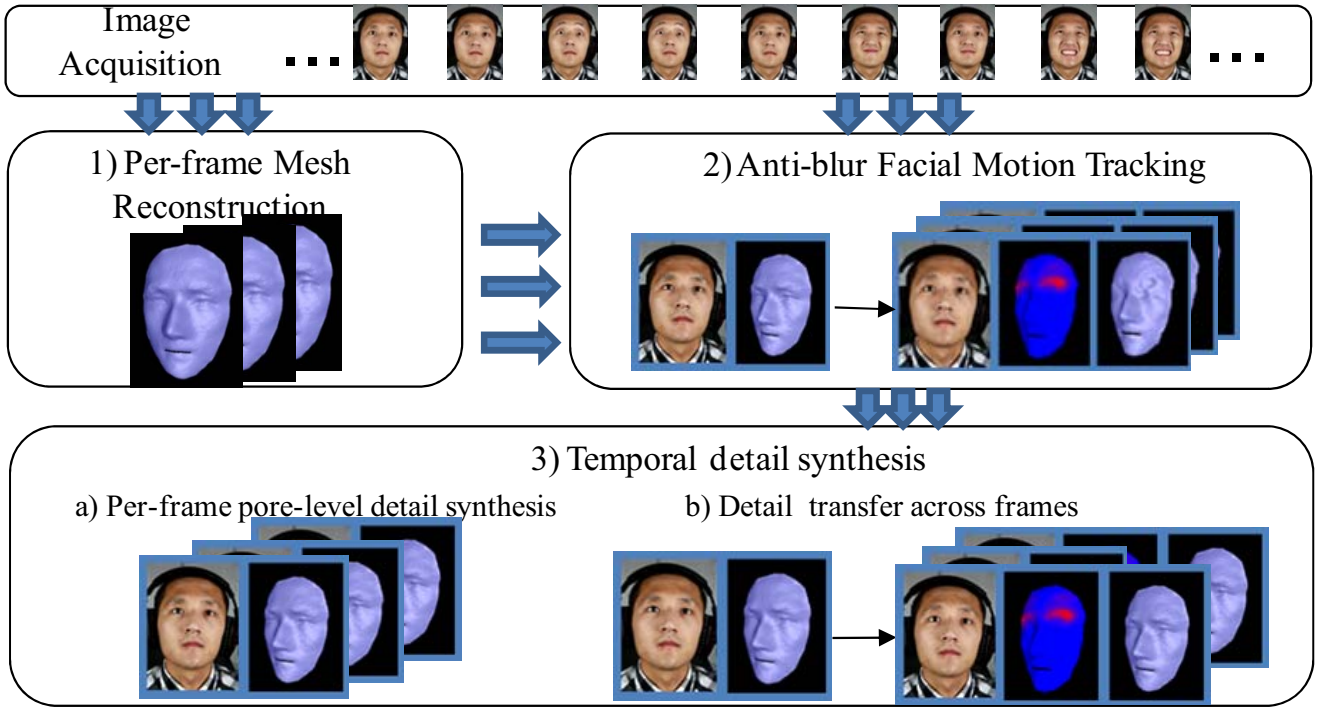


Figure 2: System overview

mesh, and all the reconstruction results share the same vertex connectivity.

B. Anti-Blur Facial Motion Tracking

On this stage, we track the initial template according to both the image stereo and the per-frame reconstructed mesh. Systematically, we follow the anchor-based tracking style proposed by Beeler et al.[5]. Based on this anchor-based framework, we take one step forward to investigate performance capture from blurred sequences. In such a blurred sequence, optical flow between two consecutive frames may not reliable, and the reconstructed meshes on the blurred regions may be wrong, see Fig. 1 for example. This flaw of the input data will lead to bad tracking results on blurred regions. To address this problem, we proposed to incorporate a blur-aware regularization to a mesh propagation procedure leveraging both optical flow and the reconstructed geometry. The combination of optical flow and geometry balances their tracking effects in a complementary way, leading to a drift-free tracking. While the blur-aware regularization impose shape constraints according to the blurry intensity, which retains more details on un-blurred regions and keeps a reasonable shape on the blurred regions.

Next we will first briefly review the anchor-based tracking framework, then describe the basic mesh tracking procedure, followed by the blur-aware regularization.

Anchor-based tracking framework. One facial performance sequence often contains repetitive subsequences. Based on this observation, Beeler et al.[5] proposed to identify anchor frames which contains similar facial expression to a manually chosen reference frame. The facial expression in the reference frame is usually in a rest pose, which is often the starting and ending pose of one expression subsequence. Thus, the anchor frames can serve as a boundary of the repetitive subsequences, so as to partition the whole sequence. In this spirit, the identified anchor frames is always in a neutral expression, and has no blur regions. In our method, we not only use this property to limit temporal drift, but also explore to leverage the anchor frames to detect blur regions of the other frames (refer to blur-aware regularization).

Optical-flow and geometry constrained mesh propagation. To propagate the tracked mesh M^t to the next frame, we take a 3D flow induced by 2D optical flow to find point correspondence between two neighboring frames. More concretely, we compute the optical flow for the left(right) image sequences, and denote it by $f^t_{left}(x)$ ($f^t_{right}(x)$), where x represents a pixel location. For each vertex v^t_i of mesh M^t, we project it onto I^t_{left} and I^t_{right}, thus to find the projection locations $x^t_{left,i}$ and $x^t_{right,i}$ respectively. Then we compute their tracked locations on the frames at $t+1$ by adding the optical flows to them:

$$\begin{aligned} x^{t+1}_{left,i} &= x^t_{left,i} + f^t_{left}(x) \\ x^{t+1}_{right,i} &= x^t_{right,i} + f^t_{right}(x) \end{aligned} \tag{1}$$

We then back-project $x_{left,i}^{t+1}$ and $x_{right,i}^{t+1}$ to the reconstructed geometry to get the hitting points $p_{left,i}^{t+1}$ and $p_{right,i}^{t+1}$. Ideally, $p_{left,i}^{t+1}$ and $p_{right,i}^{t+1}$ should be coincident, but this merely happens in practice due to errors coming from tracking or reconstruction. Here, we use a simple method to verify the reliability of these two points: the Euclidean distance between these two points should not exceeds a threshold $u = 2mm$. We then average these two points in both position and normal to obtain the correspondence point p_i^{t+1} of v_i^t.

Taking these valid correspondence pairs p_i^{t+1}, v_i^t, we minimize the following fitting energy to require the mesh to fit the point cloud:

$$E_{fitting} = \sum ||v_i^{t+1} - p_i^{t+1}||^2 \tag{2}$$

To constraint on the vertices which do not have valid correspondences, we adopt a Laplacian term to regularize the optimization problem:

$$E_{lap} = ||Lv^{t+1} - Lv^{t_0}||^2 \tag{3}$$

Adding the above two terms yields a total energy E_{tot}:

$$E_{tot} = E_{fitting} + \beta E_{Lap} \tag{4}$$

By minimizing this energy, we solve for the tracked positions of v_i^{t+1}.

Since the optical flow will cause tracking drift, this result may be not accurate. Therefore, we use this result as an initial value, and resolve a new energy incorporating a geometry term E_{geo} to refine the result:

$$\begin{aligned} E_{geo} = &\sum ||v_i^{t+1} - \tilde{p}_i^{t+1}||^2 \\ &+\rho \sum [\tilde{n}_i^{t+1} \cdot (v_i^{t+1} - \tilde{p}_i^{t+1})]^2 \end{aligned} \tag{5}$$

where $\tilde{p}_i^{t+1}$ is v_i^{t+1}'s nearest point on P^{t+1}, and $\tilde{n}_i^{t+1}$ is its normal. The first term accounts for the point-to-point distance, and the second term accounts for the point-to-plane distance. In our experiments, ρ is set to 0.1.

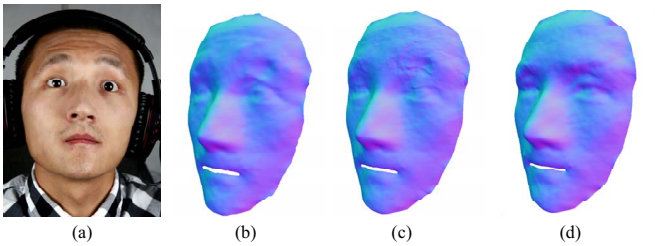
With the Laplacian term, we obtain the following updated total energy:

$$E_{tot}^* = E_{geo} + \beta E_{Lap} \tag{6}$$

We iteratively find the nearest points and solve Eq.6. Here, the maximal iteration count is set to 20.

In practice, due to the use of the geometry correlation between the initial template and the point cloud of the current frame, our method is less likely to suffer from temporal drift caused by optical flow.

Blur-aware regularization. The mesh propagation proposed in the previous section can work well on sequences without blurred images. However, for sequences with blurred images, in order to regularize the blurred region, the Laplacian term should be imposed. Unfortunately, the overall weight β in Eq. 6 is insufficient to adaptively control the effect of the regularization, which will therein incur over-smoothness in the other regions. Fig. 3 shows an example of this case.



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

Figure 3: Uniformly weighted regularization. The blurred image (a) leads to bad per-frame reconstruction (b). To regularize the blurred regions, the uniformly weighted regularization of Eq. 6 is used. (c) uses $\beta = 100$, which is too small to correct the blurred region. (d) therein increases the parameter to $\beta = 10000$, but the blurred regions are still not well corrected at the cost of smoothing out the wrinkles on the forehead.

Intuitively, we design a blur-aware regularization algorithm to adaptively impose different smooth effects on blur and un-blur regions. More specifically, we (1) compute a blurry map for each vertex on the current mesh, and (2) integrate the blurry map into the regularization term to achieve adaptive smoothing.

Blurry map computation. For each vertex, we compute its blurry intensity by comparing gradient map between the images of current frame and its nearest anchor frame. As listed in Algorithm. 1, we first compute the gradient maps of the left and the right images of the current frame and the anchor frame (line 4 - 7). Then, for each vertex v_i^t of the mesh M^t and v_i^{ach} of M^{ach}, we project them onto their corresponding images $(I_{left}^t, I_{right}^t)$ and $(I_{left}^{ach}, I_{right}^{ach})$, respectively (line 11 - 12). Since there are two images for each frame, we choose the projected image that has smaller angle between the camera axis and the vertex normal. Next, compare the gradients on a small window around P_i^{ach} and P_i^t (line 14). Notice that the $\succeq$ operation (line 14) is a strict component-wise comparison. It means all components of the left side are larger than that of the right side, thus to ensure a blurred point. If a point is blurred, then the blurry intensity of this point is set to the difference between the average gradient values of these two windows (line 15), otherwise the blurry intensity is set to zero (line 17). Finally, a post-processing step is applied to smooth the result. Since the blurry map is defined on the mesh, we take an average operation on 1-ring neighborhood of each vertex as the mean filter.

Blur-aware Laplacian constraint. Given the the blurry map $\mathcal{B}$, we construct a corresponding weighting matrix B to control the regularization weights. The relationship between the blurry map $\mathcal{B}$ and the weighting matrix B is designed

Algorithm 1 Blurry map computation

Input: The images of the anchor frame $I^{ach}_{left}, I^{ach}_{right}$, the mesh of the anchor frame M^{ach}, the images of the current frame $I^{t}_{left}, I^{t}_{right}$, the mesh of the anchor frame M^{t}.
Output: Blurry Map $\mathcal{B}$ of M^t

//Step1: Compute gradient map
$G(I^{ach}_{left}) \leftarrow G \bigotimes I^{ach}_{left}$
$G(I^{ach}_{right}) \leftarrow G \bigotimes I^{ach}_{right}$
$G(I^{t}_{left}) \leftarrow G \bigotimes I^{t}_{right}$
$G(I^{t}_{right}) \leftarrow G \bigotimes I^{t}_{left}$

//Step2: Estimate blur intensity
for each vertex of the mesh **do**
  /*Back-project v^t_i onto frame t, and v^{ach}_i onto anchor frame. Choose the image having smaller angle between camera axis and vertex normal. Denote the projected images as I^t and I^{ach} */
  $P^t_i \leftarrow$ BackProj(v^t_i)
  $P^{ach}_i \leftarrow$ BackProj(v^{ach}_i)
  //compare grads in s $\times$ s windows at P^t_i and P^{ach}_i
  if $G(I^{ach}) \bigotimes W_{s\times s} \succeq G(I_t) \bigotimes W_{s\times s}$ **then**
    $\mathcal{B}(t) = G(I^{ach}) \bigotimes W_{s\times s} - G(I^t) \bigotimes W_{s\times s}$
  else
    $\mathcal{B}(t) = 0$
  end if
end for

//Step3: Postprocessing
smooth the final result by a mean filter

as follows:

$$B(i,j) = \begin{cases} e^{\mathcal{B}(i)}(1+\mu\mathcal{B}(i)+\mu^2\mathcal{B}(i)^2), & i=j \\ 0, & i\neq j \end{cases} \quad (7)$$

where, μ is the average value of the blurry map $\mathcal{B}$.

By incorporating the weighting matrix B into Eq.3, we obtain the blur-aware Laplacian term:

$$E^*_{lap} = ||B(Lv^{t+1} - Lv^{t_0})||^2 \quad (8)$$

Thus, in the stage of the mesh propagation, we replace the uniform Laplacian term in Eq.4 and Eq.6 with the blur-aware Laplacian term of Eq.8. As shown in Fig. 4, the blur-aware regularization keeps a plausible shape on the blurred region without smoothing out the wrinkles on the forehead.

C. Detail Synthesis

This subsection describes how to synthesize pore-level detail across all frames of a sequence. Since the captured data may include blurred parts, where per-frame synthesis is impossible due to missing photometric information. Therefore, we develop a detail transfer algorithm to temporally synthesize the pore-level details on the blurred regions.

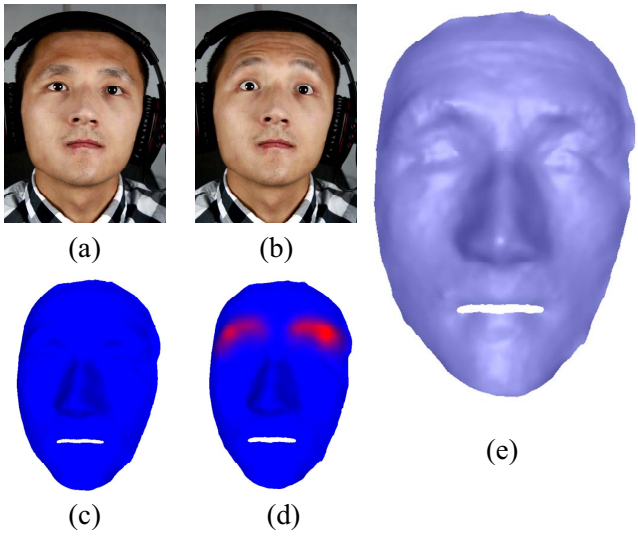


Figure 4: Blur-aware Regularization. (a) is a clear image, and (b) is blurred around the brows. (c) and (d) visualize the blurry maps of (a) and (b), respectively. The red indicates blurred regions. (e) is the reconstruction result of (b), with our blur-aware regularization. The result keeps a plausible shape while retaining wrinkles on the forehead.

Per-frame pore-level detail synthesis. As the first step on this stage, we take a state-of-the-art mesoscopic synthesis [6] to generate the pore-level details for each frame independently. Here, we denote by $\widetilde{M}^t$ the mesh after mesoscopic synthesis. Thereby, the mesoscopic details D^t can be represented by substraction of the detailed mesh $\widetilde{M}^t$ from the coarse mesh M^t:

$$D^t = \widetilde{M}^t - M^t \quad (9)$$

The mesoscopic synthesis relies on high-frequent spectrum of the image, which is lost on the blurred part of the images. An example is shown in Fig 5.

Detail transfer across frames. To generate pore-level details on the blurred regions, we develop a simple but practical method to transfer the details from the neighboring frames without blur. As we have gotten the blurry map for each mesh on the sequence, we are able to test whether a frame contains blurry regions or not. To do this, we sum up all blurry value of all vertices on the mesh. If the result is zero (note that the blurry value is non-negative), the corresponding frame is not blurry at every position, otherwise there are some blurry regions. Hereby, for the current mesh M^t, if it is clear, we retain the result of the per-frame synthesis; if it is blurry, we trace back to find the nearest clear frame before time t, and denote this time by

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

Figure 5: Detail synthesis. (a),(b). the region around the brow is blurred. (c). per-frame detail synthesis fails to generate promising pore-level details on this region. (d). the temporally coherent details synthesis produce much more realistic pore-level details than (c).

(a) (b)

Figure 6: Data Acquisition. (a) our hardware setup. (b) an example frame of a stereo sequence.

t_0, and similarly trace forward to find earliest time t_1 which is clear. Then, we can transfer the details of D^{t_0} and D^{t_1} to the blurry regions of M^t by linear interpolation:

$$\widetilde{M}^t = M^t + D^{t_0} + \frac{t - t_0}{t_1 - t_0}(D^{t_1} - D^{t_0}) \qquad (10)$$

We show the effectiveness of the detail transfer in Fig 5.

Temporal filtering on the whole sequence. To avoid the subtle but noticeable flickering caused by small differences between successive frames, we smooth the resulting sequence with a single pass of Gaussian filter in a [-1, +1] temporal window.

V. Experimental Results

A. Data Acquisition

Our acquisition setup consists of two Canon 5D MarkII SLRs in a indoor environment (See Fig. 6). Our method can work well on common lighting condition, and auxiliary lightings can be added when the environment light is too dark. These two SLRs are calibrated before capture using checkerboard [21]. These two SLRs capture HD video at a resolution of 1080×1920, and we use a LED light to temporally align the captured videos to achieve synchronization. Since our method can deal with blurred sequence, it is not necessary to require the actors to perform smoothly changing expressions.

B. Validation

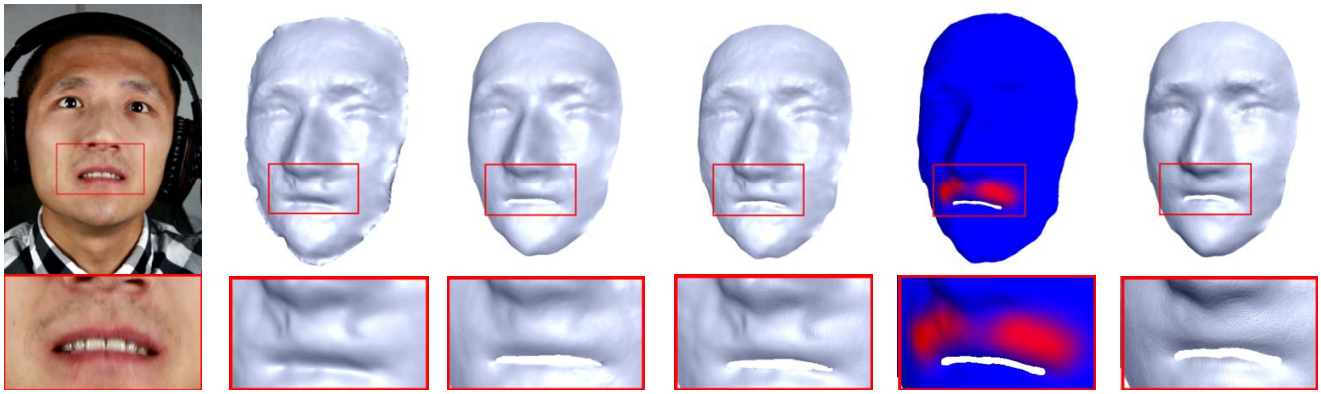
To validate the effectiveness of the proposed method, we conduct experiments to compare the results of our method with that of two state-of-the-art methods: Bradely et al.[4] and Beeler et al. [5], respectively. We capture one set of facial performance data using our acquisition setup. This data set contains 495 frames, and some regions of the images are blurred due to fast mouth motion.

Fig. 7 shows these comparisons on a frame which contains motion blur. In this frame, the mouth is blurred due to fast motion (a). Therefore the following per-frame reconstruction (b) is also bad due to stereo matching ambiguity on the blurred regions. (c) gives the result of the method of Bradely et al.[4]. It adopts uniform weighted regularization to fill up the missing (out-of-real-shape) regions, which can not adaptively balance between smoothness and feature-preserving. Here, we carefully tuned the parameters β (10000) to obtain much better results visually. However, it is clear that the result is prone to over-smoothness. (d) shows the results of Beeler et al.[5]. They take a locally regularization to cope with such cases, which fails to fill up such a large region due to its local nature. On the other hand, their per-frame detail synthesis is also unable to generate detail on the blurred regions, since the photometric information is lost there. (e) and (f) give the blurry map we detected and the result produced by our method. From (e), we see that the blurry map successfully detect the blurred regions. In (f), our blur-aware regularization "pull up" the bad reconstructed regions while still retain features on other regions. Meanwhile, the clear pore-level is also presented in the result, which shows the effectiveness of our temporally coherent detail transfer.

C. Results.

To further demonstrate the ability of our system to reconstruct realistic temporally coherent dynamic wrinkles and pore-level details, even if the input data contains blur, we show results of one sequences in Fig. 8.

Fig. 8 contains six frames of an image sequence, including some extreme expressions. The first row are the images. The second row are the reconstructed meshes. The third row shows the textured meshes, which are highly consistent with their corresponding images. The fourth row shows the reconstructed meshes with a grid pattern to enable the deformation evaluation, which demonstrate the low temporal drift of these results. In the second column, the wrinkle on the forehead is reliably reconstructed. In the second and fourth columns, our method handle the blurred regions (brow and mouth) well, producing plausible shape in the blurred region and adding high-fidelity pore-level details (for these



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

Figure 7: Compare our method with [4] and [5]. (a):the image. (b):the per-frame initial reconstructed mesh. (c)(d):the results of [4] and [5]. (e): the visualization of our blurry map. (f): our result.

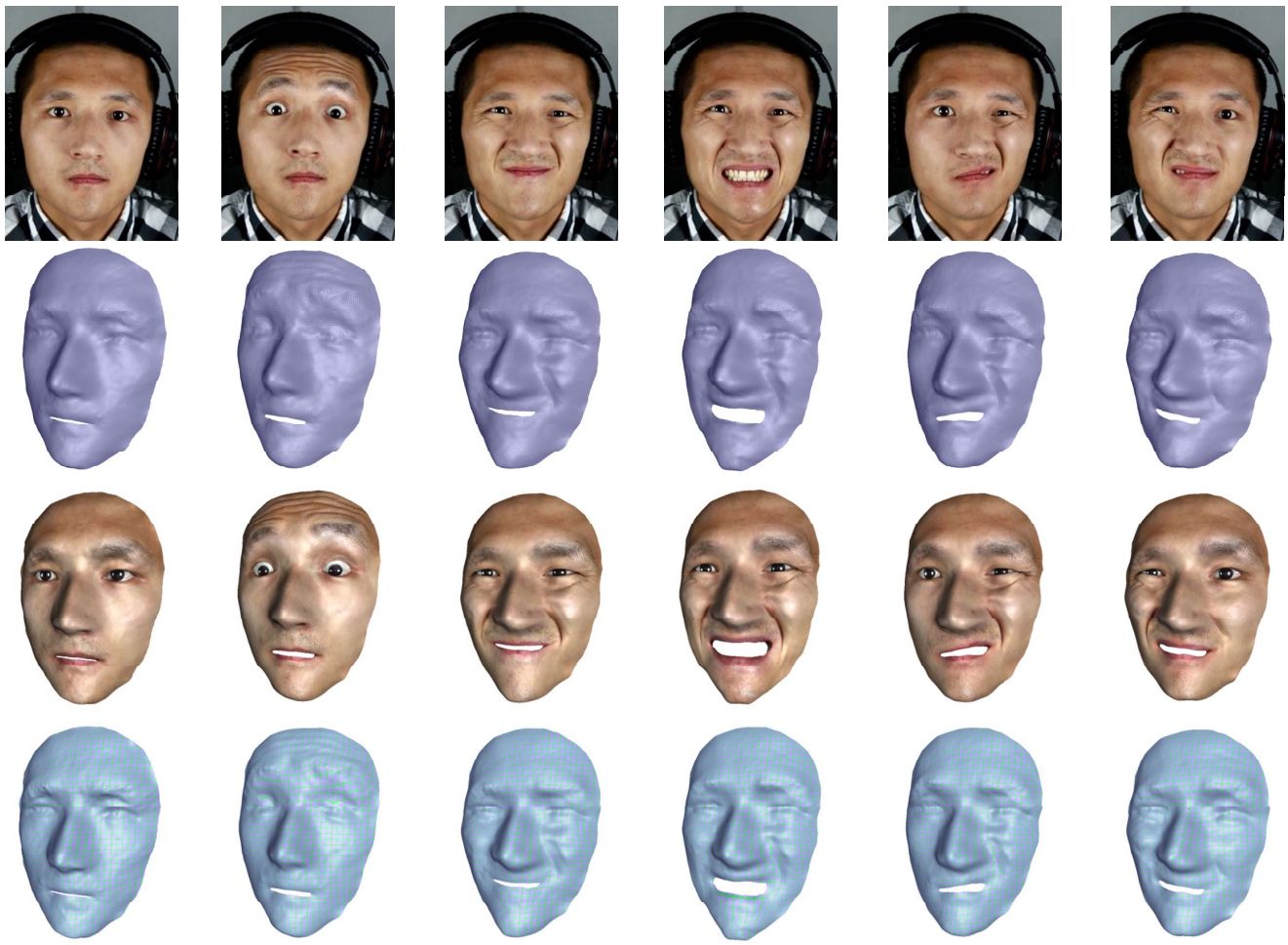


Figure 8: Frames taken from a single sequence of an actor. Top row: images from the video. Second row : reconstructed meshes. Third row: reconstructed meshes with texture. Bottom: reconstructed meshes rendered with a grid pattern demonstrating low temporal drift.

details, please see the electronic version of the paper).

D. Computational Time.

We implement our system in C++ on a desktop PC with Intel Core2 Duo E7400@2.80GHz CPU (use only one core) and 8GB memory. For a typical sequence containing about 500 frames and about 700K-vertex mesh resolution, our system takes about 3 hours for all each-frame mesh generation, about 5 hours for mesh propagation, and about 3 hours for pore-level detail synthesis. However, the system is very suitable to achieve parallelization due to anchor-frame used. Therefore, there is still a large space to reduce the computational time.

VI. Conclusion

We present a system to reconstruct high-fidelity 3D facial performance from binocular image sequences. Beyond the current state-of-the-art work, our system is able to cope with sequences with blurred images. To this end, we designed blur detection and blur-aware regularization to promisingly reconstruct the shape on the blurred region while retaining the details. We also develop a temporally coherent details synthesis to generate realistic pore-level details on the blurred regions. There are still limitations on current system. It is not appropriate to capture realistic movements of eyelid and lip.This requires special processing on these parts, which can take inspirations from work of Weise et al.[16] and Bradley et al.[4].

Acknowledgment

This work was partially supported by NSFC (No. 61379068).

References

[1] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister, M. Gross, Multi-scale capture of facial geometry and motion, ACM Trans. Graph. 26 (3).

[2] Y. Furukawa, J. Ponce, Dense 3d motion capture for human faces, in: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA, IEEE, 2009, pp. 1674–1681.

[3] L. Zhang, N. Snavely, B. Curless, S. M. Seitz, Spacetime faces: high resolution capture for modeling and animation, ACM Trans. Graph. 23 (3) (2004) 548–558.

[4] D. Bradley, W. Heidrich, T. Popa, A. Sheffer, High resolution passive facial performance capture, ACM Trans. Graph. 29 (4) (2010) 41:1–41:10.

[5] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, M. Gross, High-quality passive facial performance capture using anchor frames, ACM Trans. Graph. 30 (4) (2011) 75:1–75:10.

[6] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, M. Gross, High-quality single-shot capture of facial geometry, ACM Trans. Graph. 29 (2010) 40:1–40:9.

[7] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, C. Theobalt, Lightweight binocular facial performance capture under uncontrolled lighting, ACM Trans. Graph. 31 (6) (2012) 187:1–187:11.

[8] H. Li, P. Roivainen, R. Forcheimer, 3-d motion estimation in model-based facial image coding, IEEE Trans. Pattern Anal. Mach. Intell. 15 (6) (1993) 545–555.

[9] F. H. Pighin, R. Szeliski, D. Salesin, Resynthesizing Facial Animation through 3D Model-based Tracking, in: Proc. 7th International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 143–150.

[10] V. Blanz, C. Basso, T. Poggio, T. Vetter, Reanimating faces in images and video, Computer Graphics Forum 22 (3) (2003) 641–650.

[11] M. Zeng, L. Liang, X. Liu, H. Bao, Video-driven state-aware facial animation, Comput. Animat. Virtual Worlds 23 (3-4) (2012) 167–178.

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

[13] T. Weise, S. Bouaziz, H. Li, M. Pauly, Realtime performance-based facial animation, ACM Trans. Graph. 30 (4) (2011) 77:1–77:10.

[14] H. Li, J. Yu, Y. Ye, C. Bregler, Realtime facial animation with on-the-fly correctives, ACM Trans. Graph. 32 (4) (2013) 42:1–42:10.

[15] H. Huang, J. Chai, X. Tong, H.-T. Wu, Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition, ACM Trans. Graph. 30 (4) (2011) 74:1–74:10.

[16] T. Weise, H. Li, L. Van Gool, M. Pauly, Face/off: live facial puppetry, in: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '09, ACM, New York, NY, USA, 2009, pp. 7–16.

[17] W.-C. Ma, A. Jones, J.-Y. Chiang, T. Hawkins, S. Frederiksen, P. Peers, M. Vukovic, M. Ouhyoung, P. Debevec, Facial performance synthesis using deformation-driven polynomial displacement maps, ACM Trans. Graph. 27 (5) (2008) 121:1–121:10.

[18] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, P. Debevec, The digital emily project: photoreal facial modeling and animation, in: ACM SIGGRAPH 2009 Courses, SIGGRAPH '09, ACM, New York, NY, USA, 2009, pp. 12:1–12:15.

[19] C. A. Wilson, A. Ghosh, P. Peers, J.-Y. Chiang, J. Busch, P. Debevec, Temporal upsampling of performance geometry using photometric alignment, ACM Trans. Graph. 29 (2) (2010) 17:1–17:11.

[20] G. Fyffe, T. Hawkins, C. Watts, W.-C. Ma, P. E. Debevec, Comprehensive facial performance capture, Comput. Graph. Forum 30 (2) (2011) 425–434.

[21] Z. Zhang, A flexible new technique for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1330–1334.