Stabilized Real-time Face Tracking via a Learned Dynamic Rigidity Prior

CHEN CAO, Snap Inc.
MENGLEI CHAI, Snap Inc.
OLIVER WOODFORD, Snap Inc.
LINJIE LUO, Snap Inc.

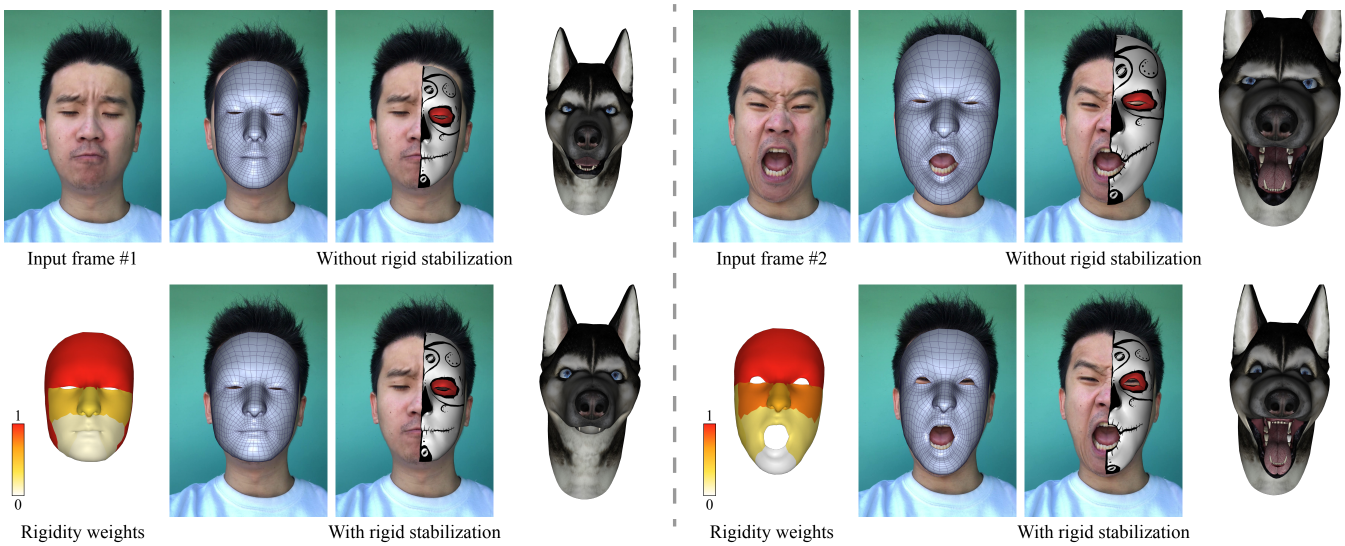


Fig. 1. Our rigid stabilization method produces stable head poses under exaggerated facial expressions. We compare the baseline method without rigid stabilization (*top*) to our method (*bottom*) by applying both to virtual makeup and avatar retargeting on face-squeezing (*left panel*) and face-enlarging (*right panel*) expressions. Notice the abrupt scale changes of the makeup and avatar in the baseline method due to the unstable estimates of head poses in depth. We also show the rigidity weights from our learned dynamic rigidity prior that provide per-face-region adaptivity in the rigid pose optimization.

Despite the popularity of real-time monocular face tracking systems in many successful applications, one overlooked problem with these systems is rigid instability. It occurs when the input facial motion can be explained by either head pose change or facial expression change, creating ambiguities that often lead to jittery and unstable rigid head poses under large expressions. Existing rigid stabilization methods either employ a heavy anatomically-motivated approach that are unsuitable for real-time applications, or utilize heuristic-based rules that can be problematic under certain expressions. We propose the first rigid stabilization method for real-time monocular face tracking using a dynamic rigidity prior learned from realistic datasets. The prior is defined on a region-based face model and provides dynamic region-based adaptivity for rigid pose optimization during real-time performance. We introduce an effective offline training scheme to learn the dynamic rigidity prior by optimizing the convergence of the rigid pose optimization to the ground-truth poses in the training data. Our real-time face tracking system is an optimization framework that alternates between rigid pose optimization and expression optimization. To ensure tracking accuracy, we combine both robust, drift-free facial landmarks and dense optical flow into the optimization objectives. We evaluate our system extensively against state-of-the-art monocular face tracking systems and achieve significant improvement in tracking accuracy on the high-quality face tracking benchmark. Our system can improve facial-performance-based applications such as facial animation retargeting and virtual face makeup with accurate expression and stable pose. We further validate the dynamic rigidity prior by comparing it against other variants on the tracking accuracy.

CCS Concepts: • **Computing methodologies** → **Motion capture**;

Additional Key Words and Phrases: Real-time Monocular Face Tracking

ACM Reference format:
Chen Cao, Menglei Chai, Oliver Woodford, and Linjie Luo. 2018. Stabilized Real-time Face Tracking via a Learned Dynamic Rigidity Prior. *ACM Trans. Graph.* 37, 6, Article 233 (November 2018), 11 pages.
https://doi.org/10.1145/3272127.3275093

1 INTRODUCTION

Recent advances in real-time monocular face tracking have catalyzed a wave of technical and product innovations that empowered facial-performance-based applications such as Snapchat Lens and similar products by Facebook, Apple and many others. Although existing real-time face tracking systems demonstrate impressive

0730-0301/2018/11-ART233 $15.00
https://doi.org/10.1145/3272127.3275093

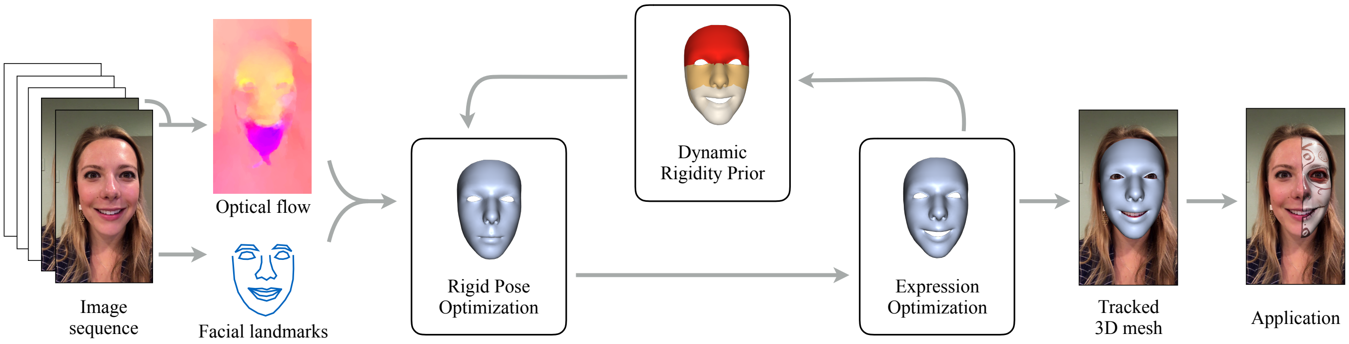


Fig. 2. Overview of our system. Our system takes image sequence as input and computes facial landmarks and dense optical flow as input constraints. We then alternate between rigid pose optimization (Sec. 5.1), expression optimization (Sec. 5.2) and using a learned dynamic rigidity prior to adjust per-region rigidity weights for pose optimization (Sec. 6). Our system outputs accurate tracked head poses and facial expressions for applications such as virtual makeup.

performance even in resource-constrained scenarios [Apple 2017; Bouaziz et al. 2013; Cao et al. 2014a; Tewari et al. 2017], most of them nevertheless exhibit problems that are becoming impediments to further improvement in tracking fidelity.

Rigid instability is a problem first addressed in [Beeler and Bradley 2014] that also occurs in most real-time monocular face tracking systems since they often employ a joint optimization for both rigid head pose and facial expression parameters. In such joint optimization, the input facial motion can be explained by either pose or expression parameters, creating ambiguities that often lead to jittery and unstable rigid head poses under large expressions (Fig. 1).

Beeler and Bradley [2014] proposed an anatomically-motivated approach to rigid stabilization by accurately modeling non-rigid skin deformations actuated by facial expression. Despite the high accuracy, their method operates on high-quality 3D facial scans and is therefore difficult to apply to monocular face tracking. Other existing approaches (e.g. [Weise et al. 2011]) prioritize on the relatively rigid part of the face (e.g. the upper part) in optimizing the rigid head pose. However, these heuristic-based approaches become problematic if the expression involves the relatively rigid part of the face (e.g. frown) or the additional input (e.g. depth) is not provided.

We propose a rigid stabilization method for real-time monocular face tracking, using a *dynamic rigidity prior* learned from realistic datasets. To the best of our knowledge, our rigid stabilization method is the first for real-time monocular face tracking applications. We define our dynamic rigidity prior on a *region-based face model* [Tena et al. 2011] to provide dynamic region-based adaptivity for rigid pose optimization during real-time performance. We introduce an effective offline training scheme to learn the hyper-parameters for the dynamic rigidity prior, by optimizing the convergence of the rigid pose optimization to the ground-truth poses in the training data. Our real-time face tracking system is an optimization framework that alternates between optimizing rigid pose with our dynamic rigidity prior, and optimizing expression parameters.

To ensure tracking accuracy, we combine both robust, drift-free facial landmark detection [Kazemi and Sullivan 2014] and dense, motion-guided correctives from fast optical flow [Kroeger et al. 2016] into the optimization objectives. Our region-based face model also allows additional expressiveness to faithfully model localized expressions (e.g. raising an eyebrow, opening mouth) compared to traditional holistic blendshapes.

We evaluate our system extensively by comparing, both quantitatively and qualitatively, to state-of-the-art monocular face tracking methods. We show that our system can faithfully capture rich expression while maintaining stable and accurate rigid head motion. This allows us to improve the quality of facial-performance-based applications such as facial animation retargeting. We further validate the dynamic rigidity prior by comparing it against other variants on the tracking accuracy.

In summary, our main contributions are:

- A novel rigid stabilization method using a learned dynamic rigidity prior to stabilize monocular real-time face tracking;
- A realistic facial performance dataset with ground-truth rigid pose and expression parameters;
- A novel scheme to learn the hyper-parameters of our dynamic rigidity prior from our facial performance dataset;
- An accurate monocular real-time face tracking system that combines an expressive region-based face model and the constraints of both sparse landmarks and dense optical flow.

2 RELATED WORK

High-Fidelity Facial Performance Capture. Capturing facial performance in high fidelity have long been an important research topic in computer graphics and vision. In this line of work, reconstruction quality is the primary concern and most practitioners adopt sophisticated capture setups with multiple cameras, special equipment or well-controlled environments. A number of methods have been proposed to capture high-quality static expressions [Beeler et al. 2010; Ghosh et al. 2011; Ma et al. 2007], and soon extended to capture dynamic facial expressions in live performances [Beeler et al. 2011; Bradley et al. 2010; Furukawa and Ponce 2009; Garrido et al. 2016b; Huang et al. 2011; Klaudiny and Hilton 2012; Zhang et al. 2004]. More recent approaches focus on simplifying capture setups to binocular inputs [Valgaerts et al. 2012] and even monocular camera [Garrido et al. 2013, 2016a; Shi et al. 2014a; Suwajanakorn et al. 2014]. However, all these methods leverage substantial amount of offline processing to achieve maximum reconstruction quality.

Real-Time Facial Performance Capture. Another line of research focuses on capturing and tracking live facial expressions in real-time. These real-time systems often adopt monocular setups to achieve maximum flexibility. The first approaches in this area leveraged depth information provided by custom-built structured light system [Weise et al. 2009] as well as off-the-shelf depth sensors [Apple 2017; Bouaziz et al. 2013; Chen et al. 2013; Li et al. 2013; Weise et al. 2011]. More recently, methods have started to emerge that operate only on a single monocular camera [Cao et al. 2014a, 2013; Chai et al. 2003; Rhee et al. 2011; Tewari et al. 2017; Thies et al. 2016; Wang et al. 2016]. Notably, these real-time methods typically employ strong global priors derived from simplified facial expression models such as morphable face model or linear blendshapes. While providing desired efficiency and robustness, these global facial expression priors also lead to limited expressiveness and rigid instability in their results. Chen et al. [2015] used sparse optical flow and mesh deformation to improve the tracking accuracy, but these components can only refine non-rigid expressions without improving rigid stability.

Region-Based Face Model. Previous work has shown segmenting the face model can increase the expressiveness for 3D morphable models [Blanz and Vetter 1999] and active appearance models [Peyras et al. 2007]. While segmentation can be performed by manual selection [Zhang et al. 2006], an automatic, physically-motivated approach [Joshi et al. 2003] can be also adopted. Neumann et al. [2013] proposed a method to automatically learn a localized deformation basis from performance capture data. Tena et al. [2011] introduced a joint optimization approach to simultaneously solve for all facial regions based on real facial performance datasets. More recently, Wu et al. [2016] proposed an anatomically-constrained local deformation model to further increase the fidelity of monocular facial animation.

3 OVERVIEW

The pipeline of our system is shown in Fig. 2. Like most real-time monocular face tracking systems, we take as input a sequence of images (typically videos) and perform facial landmark detection [Kazemi and Sullivan 2014] as well as efficient dense optical flow computation [Kroeger et al. 2016] to densify the target constraints. We then fit a region-based face model, learned from real facial performance datasets (Sec. 4), to the landmarks and optical flow constraints. This is done by alternating between two different optimizations (Sec. 5): the first solves for the rigid pose parameters of the model (Sec. 5.1); the second solves for the expression parameters of the model (Sec. 5.2).

We achieve rigid stabilization by incorporating *dynamic rigidity prior* during rigid pose optimization, which dynamically weights different regions of the face model as a function of their expression magnitudes (Sec. 6). In an offline training step, we learn the hyper-parameters of the dynamic rigidity prior for each region by optimizing the convergence of rigid pose optimization to the ground-truth poses in a realistic facial performance dataset (Sec. 6.3).

4 FACE MODEL FOR REAL-TIME TRACKING

In this section, we will first provide useful background of holistic multi-linear face models adopted by most real-time monocular face tracking systems and then extend the holistic models to the region-based model we used in our implementation. We will also introduce useful notations for the following sections.

4.1 Multi-Linear Face Model

Real-time face tracking methods [Cao et al. 2014a; Shi et al. 2014b; Thies et al. 2016] largely depend on the underlying face prior model to achieve tracking robustness and expressiveness. Multi-linear face models such as the one proposed in [Cao et al. 2014b] are widely adopted data-driven priors that expand the subspace of realistic face shapes under both identity and expression variations represented by a rank-3 tensor $\mathcal{B} \in \mathbb{R}^{3N_M \times N_I \times N_E}$, where N_M, N_I, N_E are the numbers of vertices in face model and the numbers of bases for identities and expressions respectively. Assuming the identity parameters η for the current subject and the camera matrix are already estimated using online adaptation methods such as [Cao et al. 2014a], the corresponding holistic expression blendshape bases can be extracted as: $\mathbf{B} = [\mathbf{B}_0, \mathbf{B}_1, \cdots, \mathbf{B}_{N_E}] = \mathcal{B} \otimes \eta$. Without loss of generality, $\mathbf{B}$ can be further converted into delta bases by: $\mathbf{B} = [\mathbf{B}_0, \Delta\mathbf{B}_1, \cdots, \Delta\mathbf{B}_{N_E}]$, where $\mathbf{B}_0$ is for the neutral expression or base shape, and $\Delta\mathbf{B}_i = \mathbf{B}_i - \mathbf{B}_0$. Given expression parameters $\beta = [\beta_1, \cdots, \beta_{N_E}]$, the 3D face model can be expressed as:

$$\mathbf{F}(\beta) = \mathbf{B}_0 + \sum_{i=1}^{N_E} \beta_i \Delta\mathbf{B}_i. \tag{1}$$

To fit the 3D face model to input constraints defined in image space, we need to define a rigid pose transformation $\mathbf{T} = [\mathbf{R}, \mathbf{t}] \in \mathbb{R}^{3\times4}$ consisting of rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$, parameterized by three Euler angles, and a translation vector $\mathbf{t} \in \mathbb{R}^3$. For convenience, we denote the Euler angles associated with $\mathbf{T}$ as $\mathbf{r}(\mathbf{T})$ and the translation vector as $\mathbf{t}(\mathbf{T})$. We also assume that camera intrinsics are known, and that the camera projection operator $\Pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$ maps from camera coordinate system to image coordinate system. We can then define the projected face shape $\mathbf{P}(\mathbf{T}, \beta) \in \mathbb{R}^{2\times N_M}$ under rigid pose $\mathbf{T}$ and expression parameters β:

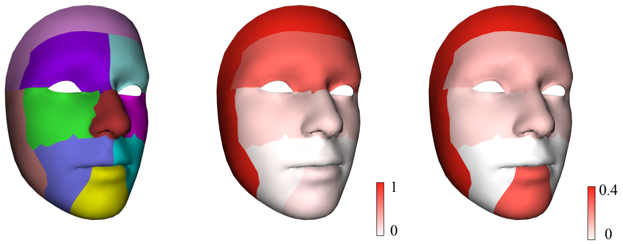
$$\mathbf{P}(\mathbf{T}, \beta) = \Pi\left(\mathbf{T}\underline{\mathbf{F}}(\beta)\right) \tag{2}$$

where $\underline{\mathbf{F}} \in \mathbb{R}^{4\times N_M}$ is $\mathbf{F}$ in homogeneous coordinates.

4.2 Region-Based Face Model

Tena et al. [2011] proposed a data-driven approach to learn the region-based face models based on motion-correlated local clusters in facial performance training data. Here we take a similar approach to learn our region-based face model from real facial performance datasets. To be specific, our region-based face model segments the entire face into K spatially-adjacent regions. To learn the segmentation, we collect around 3000 registered meshes from the FaceWarehouse database [Cao et al. 2014b], where each mesh contains N_M vertices. Our training data contains 150 different identities with each performing 20 different expressions, which covers a wide range of identity variations and most common expressions.

Following Tena et al. [2011], we compute the correlation matrix $\mathbf{C} \in \mathbb{R}^{N_M \times N_M}$ as well as the distance matrix $\mathbf{G} \in \mathbb{R}^{N_M \times N_M}$ over the entire training data for each pair of vertices in the mesh. The similarity matrix is the computed as: $\mathbf{S} = (1 - \phi)\mathbf{C} + \phi\mathbf{G}$ where ϕ



(a) (b) (c)

Fig. 3. Our region-based model (a) and the hyper-parameters α^k and $1/\sigma^k$ (b,c) of our dynamic rigidity prior defined on each region. Both face regions and hyper-parameters are learned. See Sec. 4 and Sec. 6.3 for more details.

is a weight to balance between correlation and mesh distances in the segmentation. We perform normalized spectral clustering [Ng et al. 2001] to $\mathbf{S}$, and we eventually get K clusters with each cluster representing a local face region $\Gamma^k \subset \{1, \cdots, N_M\}$. Note that we also ensure that shared vertices between regions are all included in their regions.

Our region-based face model is different from [Tena et al. 2011] in that, after performing segmentation, we do not perform Principal Component Analysis (PCA) in each region to derive a region-based PCA model. Instead, we use the segmentation results $\{\Gamma^k\}$ to directly segment the original multi-linear face model $\mathcal{B}$ into a region-based multi-linear model $\{\mathcal{B}^k\}$. The advantages of this approach are that the semantics of expression blendshapes are preserved so that explicit sparsity regularization on the expression semantics can be enforced, as we will see in Sec.6. Similar to holistic multi-linear models, we use online identity adaptation methods [Cao et al. 2014a] to compute identity coefficient vectors η^k for each region-based multi-linear model $\mathcal{B}^k$ to extract region-based blendshapes $\mathbf{B}^k = \mathcal{B}^k \otimes \eta^k \in \mathbb{R}^{4{N_M}^k \times N_E}$. Given $\Delta\mathbf{B}^k_i = \mathbf{B}^k_i - \mathbf{B}^k_0$ and $\beta^k = [\beta^k_1, \cdots, \beta^k_{N_E}]$, each region of the face model can be expressed independently as:

$$\mathbf{F}^k(\beta^k) = \mathbf{B}^k_0 + \sum_{i=1}^{N_E} \beta^k_i \Delta\mathbf{B}^k_i. \qquad (3)$$

The final face is then the combination of the regions: $\mathbf{F} = [\mathbf{F}^1, \cdots, \mathbf{F}^K]$. The positions of the shared vertices between the regions are simply averaged based on their positions in $\mathbf{F}^k$. In total our face mesh $\mathbf{F}$ contains 1,220 vertices and the numbers of shared vertices between regions range from 18 to 46. We can define similar region-based projected face shape $\mathbf{P}^k(\mathbf{T}, \beta^k) \in \mathbb{R}^{2 \times {N_M}^k}$ under rigid pose $\mathbf{T}$ and expression parameters β^k:

$$\mathbf{P}^k(\mathbf{T}, \beta^k) = \Pi\left(\mathbf{T}\underline{\mathbf{F}}^k(\beta^k)\right). \qquad (4)$$

5 OPTIMIZATION

In sections 5.1 and 5.2 we specify energy formulations for the rigid head pose and face expression model optimizations respectively. In section 5.3 we describe the implementation details.

5.1 Rigid Pose Optimization

We first optimize the rigid head pose $\mathbf{T}$, fixing the expression parameters $\beta = [\beta^1, \cdots, \beta^K]$.

Landmark energy. We employ the facial landmark detection method from [Kazemi and Sullivan 2014] to provide robust facial landmarks for the optimization. Formally, we denote a set of 2D facial landmark locations $\mathcal{L} = \{\mathbf{L}_1, \cdots, \mathbf{L}_{N_L}\}$ and their subsets $\mathcal{L}^k \subset \mathcal{L}$ defined on different regions. We also define a mapping $\ell(i)$ to map landmark $\mathbf{L}_i$ to its corresponding vertex on the face model. We introduce an energy term to minimize the L_2 norm of the landmark residuals $\mathbf{e}^k_{\text{land}}$ between the corresponding projected 3D vertex positions on the input image and the landmarks:

$$\mathcal{E}^{\text{pose}}_{\text{land}} = \sum_{k=1}^{K} w^k \left\| \mathbf{e}^k_{\text{land}}(\mathbf{T}, \beta^k) \right\|^2 \qquad (5)$$

$$\mathbf{e}^k_{\text{land}}(\mathbf{T}, \beta^k) = \left[\mathbf{P}^k(\mathbf{T}, \beta^k)_{\ell(i)} - \mathbf{L}_i\right]_{\forall i \in \mathcal{L}^k}. \qquad (6)$$

w^k are the per-region dynamic rigidity prior that weights different regions for rigid pose optimization based on their estimated rigidity during tracking, which we will detail in Sec. 6.

Dense flow energy. The detected landmarks are often too sparse to recover the complete motion of the face, especially in regions where landmarks are absent, e.g. cheek regions. Therefore, besides landmark locations, we need other denser motion cues to help extract true local motion and to correct landmark detection errors. We apply the fast optical flow estimation method [Kroeger et al. 2016] on input video stream inside the face region on-the-fly to extract the dense motion flow, and then map this motion flow to each face vertex projection in screen space through bilinear interpolation, annotated by $\mathbf{U}_i$. Given rigid pose $\mathbf{T}'$ and expression coefficients β' from previous frame, the L_2 norm of the flow residuals $\mathbf{e}^k_{\text{flow}}$ between the current projections of each face vertex i and the flow-predicted locations $\mathbf{P}^k(\mathbf{T}', \beta'^k)_i + \mathbf{U}_i$ should be minimized:

$$\mathcal{E}^{\text{pose}}_{\text{flow}} = \sum_{k=1}^{K} w^k \left\| \mathbf{e}^k_{\text{flow}}(\mathbf{T}, \beta^k) \right\|^2 \qquad (7)$$

$$\mathbf{e}^k_{\text{flow}}(\mathbf{T}, \beta^k) = \left[\mathbf{P}^k(\mathbf{T}, \beta^k)_i - \mathbf{P}^k(\mathbf{T}', \beta'^k)_i - \mathbf{U}_i\right]_{\forall i \in \Gamma^k} \qquad (8)$$

Temporal coherence energy. We also temporally regularize rigid parameters to further stabilize tracking results. However, traditional regularization with constant weights may not faithfully detect rigid motion. Instead, we use the dynamic rigidity weights w^k with current dense motion flow $\mathbf{U}$ to enforce stronger stabilization to still frames, while relaxing the restriction on fast moving frames. We define a rigid motion weight γ as follows:

$$\gamma = \exp\left(-\frac{1}{\sigma_\gamma^2} \sum_{k=1}^{K} \frac{w^k}{|\Gamma^k|} \sum_{i \in \Gamma^k} \|\mathbf{U}_i\|^2\right), \qquad (9)$$

where $\sigma_\gamma = 10.0$. Given w_a, we introduce the following temporal energy term to regularize pose optimization from the previous pose estimate $\mathbf{T}'$:

$$\mathcal{E}^{\text{pose}}_{\text{temp}} = \gamma\left(\left\|\mathbf{r}(\mathbf{T}) - \mathbf{r}(\mathbf{T}')\right\|^2 + \delta\left\|\mathbf{t}(\mathbf{T}) - \mathbf{t}(\mathbf{T}')\right\|^2\right), \qquad (10)$$

where $\delta = 0.01$.

Final pose energy. The final rigid pose optimization objective is defined as the linear combination of aforementioned energy terms to solve for $\hat{\mathbf{T}}$:

$$\hat{\mathbf{T}} = \arg\min_{\mathbf{T}} \sum_{\star} \lambda_{\star}^{\text{pose}} \mathcal{E}_{\star}^{\text{pose}}, \tag{11}$$

where $\lambda_{\text{land}}^{\text{pose}} = 1.0$, $\lambda_{\text{flow}}^{\text{pose}} = 0.8$, $\lambda_{\text{temp}}^{\text{pose}} = 2.0$ throughout the paper.

5.2 Expression Optimization

After estimating the rigid pose in each iteration, we optimize expression parameters β.

Landmark energy. We use a similar energy formulation as Eq. 5 to optimize expression parameters without the rigidity weights w^k:

$$\mathcal{E}_{\text{land}}^{\text{expr}} = \sum_{i=1}^{N_L} \sum_{i \in \mathcal{L}^k} \left\| \mathbf{e}_{\text{land}}^k (\mathbf{T}, \beta^k)_i \right\|^2 \tag{12}$$

Dense flow energy. Similar to Eq. 7, we would like to incorporate dense motion flow to improve expression parameter estimates. We define the dense flow energy term for expression as:

$$\mathcal{E}_{\text{flow}}^{\text{expr}} = \sum_{k=1}^{K} \sum_{i \in \Gamma^k} \left\| \mathbf{e}_{\text{flow}}^k (\mathbf{T}, \beta^k)_i \right\|^2 . \tag{13}$$

Note that the energy definition does not involve adaptive rigidity term compared to Eq. 7 since we want to minimize the residual energy with respect to all the region-based expression parameters.

Temporal coherence energy. In addition, we introduce a similar temporal energy term to regularize expression optimization from the previous expression estimates β':

$$\mathcal{E}_{\text{temp}}^{\text{expr}} = \left\| \beta - \beta' \right\|^2 . \tag{14}$$

L_1 sparsity energy. Since expression blendshapes are not linearly independent, favoring a sparse representation has been shown to reduce fitting errors [Bouaziz et al. 2013], and in addition, it enables higher-fidelity retargeting of face animations, since animators also choose sparse weights for these. We therefore use a robust L_1-norm regularization penalty, to encourage sparsity :

$$\mathcal{E}_{\text{l1}}^{\text{expr}} = \sum_{k=1}^{K} \sum_{i=1}^{N_E} \| \beta_i^k \|_1 . \tag{15}$$

Boundary consistency energy. One issue with region-based expression optimization is how to handle the boundary vertices between neighboring regions to preserve per-region structure while still achieving seamless blending across region boundaries. We take a similar approach to [Tena et al. 2011] that enforces a soft consistency term and merge these shared vertices into the average positions. The across-region consistency term is defined as:

$$\mathcal{E}_{\text{bound}}^{\text{expr}} = \sum_{i \in \Gamma^p \cap \Gamma^q} \left\| \mathbf{F}^p (\beta^p)_i - \mathbf{F}^q (\beta^q)_i \right\|^2 . \tag{16}$$

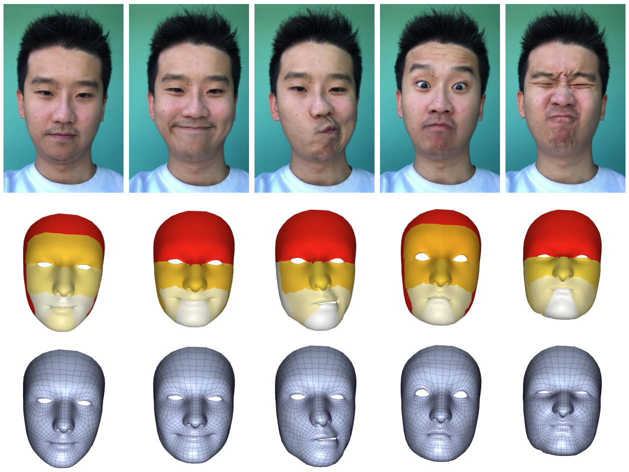


Fig. 4. Visualization of dynamic rigid weights of different frames. Regions with larger expression motion have lower weights, and vice versa.

Final expression energy. Finally, we minimize the linear combination of these energy terms for the expression parameters $\hat{\beta}$:

$$\hat{\beta} = \arg\min_{\beta} \sum_{\star} \lambda_{\star}^{\text{expr}} \mathcal{E}_{\star}^{\text{expr}}, \tag{17}$$

where we use $\lambda_{\text{land}}^{\text{expr}} = 1.0$, $\lambda_{\text{flow}}^{\text{expr}} = 0.3$, $\lambda_{\text{temp}}^{\text{expr}} = 5.0$, $\lambda_{\text{l1}}^{\text{expr}} = 2.0$ and $\lambda_{\text{bound}}^{\text{expr}} = 3.0$ throughout this paper.

5.3 Implementation Details

We use off-the-shelf non-linear least squares optimizer Ceres [Agarwal et al. 2016] in our implementation of the optimization framework. To implement the L_1-norm regularization penalty of Eq. 15, we use a residual which square-roots the expression weights. We alternate the optimizations, first the rigid pose energy of Eq. 11, then the expression energy of Eq. 17 for multiple passes. During each pass, we optimize the rigid pose until convergence and the expression energy of Eq. 17 for only one iteration. This is because the expression optimization is relatively expensive, we can get the most out of each pass by fully optimizing the rigid pose. Since the two optimizations minimize different cost functions, this formulation does not guarantee convergence. However, in practice we have found them to converge quickly to a stable solution and we adopt a fixed number of 3 passes throughout our experiments.

6 DYNAMIC RIGIDITY PRIOR

Rigid instability arises from the ambiguities to explain the observed facial motion by either the head pose or expression changes. For example, a face scrunching expression moves central face region landmarks in a similar way to moving the head back (Fig. 1); however, landmarks around the edge of the face do not suffer from the same ambiguity during this expression. To address this problem, we introduce a dynamic rigidity prior (section 6.1) to assign higher weights to regions at run-time which are more likely to give a reliable pose estimate during pose optimization. We also describe the training data (section 6.2) and objective function (section 6.3) to learn this prior in an offline training stage.

6.1 Formulation

To account for the varying reliability of different face regions for rigid pose optimization (Eq. 11), we formulate a dynamic rigidity prior $\{w^k\}$ that dynamically weights each region for more reliable rigid pose optimization (see Fig. 4). The prior is formulated based on the expression motion of each region $\mathbf{F}^k(\hat{\beta}^k)$ compared to its neutral expression base shape $\mathbf{B}_0^k$ after the expression optimization:

$$w^k = \alpha^k \exp\left(-\frac{\left\|\mathbf{F}^k(\hat{\beta}^k) - \mathbf{B}_0^k\right\|^2}{(\sigma^k)^2|\Gamma^k|}\right), \tag{18}$$

where α^k and σ^k are learned hyper-parameters (as described in section 6.3), and $|\Gamma^k|$ is the number of vertices in the k^{th} region. The intuition behind this formulation is that less neutral expressions tend to lead to greater pose instability, therefore regions with greater non-rigid deformations should be down-weighted more.

6.2 Training Data

We require data with which to learn the dynamic rigidity prior, consisting of landmark and flow measurements, with ground-truth rigid poses and expressions. Since real data with ground-truth rigid poses and expressions are hard to acquire, we generate our training data by employing artists to build 8 synthetic facial animation sequences, including talking and changing between different expressions. To each of these facial animation sequences we then apply 2 different rigid transformations, one captured from recorded video, the other a static head pose, creating 16 expression and pose sequences with 2668 frames in total. For each frame of these video sequences, we have its ground-truth expression coefficients, and rigid pose. Based on these data, we can generate the ground-truth facial landmarks and motion flow of each vertex.

It should be noted that since the ground-truth training data measurements are synthetically generated, they are noiseless; the pose and expression energies will evaluate to zero at the ground-truth parameters, which precludes any learning of hyper-parameters through directly minimizing these energies with respect to the hyper-parameters on the training data. Measurement noise to the facial landmarks and motion flow could be added to make the training data more realistic, but it is highly challenging to generate the noise close to that in the real case, with the additional risk of rendering the learning process dependent on a synthetic noise distribution. Therefore, we choose to optimize *the convergence* of the rigid pose optimization to the ground-truth poses from perturbed poses and expressions to emulate the real optimization scenarios. We apply small perturbations to the ground-truth expression coefficients and poses, generating 5 pose, expression pairs per frame, producing a final set of training data containing $S = 13,340$ samples.

6.3 Training Objective

Our goal in learning the dynamic rigidity prior is to find the hyper-parameters $\Theta = \{\alpha^k, \sigma^k\}_{k=1}^K$ in $\{w^k\}_{k=1}^K$ for each face region. As discussed, our training data measurements are noiseless, therefore we cannot simply find the parameters that minimize Eq. 11 at the ground truth solution—they will *all* generate *zero* cost. Instead, we phrase the training objective as finding Θ such that when Eq. 11 is minimized on real data from perturbed pose $\tilde{\mathbf{T}}^s$ and expression $\tilde{\beta}^s$ of each sample s, it converges as close to the ground-truth pose $\mathbf{T}^{*s}$ as possible:

$$\mathcal{E}_{\text{train}} = \sum_{s=1}^{S} \left\| d\left(\arg\min_{\tilde{\mathbf{T}}^s} \mathcal{E}_{\text{train}}^{\text{pose}}(\tilde{\mathbf{T}}^s, \tilde{\beta}^s),\ \mathbf{T}^{*s}\right)\right\|^2, \tag{19}$$

$$\begin{aligned}
\mathcal{E}_{\text{train}}^{\text{pose}}(\mathbf{T}, \beta) &= \left\|\mathbf{e}(\mathbf{T}, \beta)\right\|^2, \\
\mathbf{e}(\mathbf{T}, \beta) &= \begin{bmatrix} \mathbf{e}_{\text{land}}(\mathbf{T}, \beta) \\ \mathbf{e}_{\text{flow}}(\mathbf{T}, \beta) \end{bmatrix}, \\
\mathbf{e}_{\text{land}}(\mathbf{T}, \beta) &= \left[\sqrt{w^k \lambda_{\text{land}}^{\text{pose}}}\mathbf{e}_{\text{land}}^k(\mathbf{T}, \beta)\right]_{\forall k \in \{1,..,K\}}, \\
\mathbf{e}_{\text{flow}}(\mathbf{T}, \beta) &= \left[\sqrt{w^k \lambda_{\text{flow}}^{\text{pose}}}\mathbf{e}_{\text{flow}}^k(\mathbf{T}, \beta)\right]_{\forall k \in \{1,..,K\}},
\end{aligned}$$

where $d(\cdot, \cdot)$ computes a distance in pose space. This *is* possible to learn from our synthetic training data. Note that we dropped the temporal coherence energy $\mathcal{E}_{\text{temp}}^{\text{pose}}$ from the above cost so that the rigidity prior learns to generate the best pose possible, independent of previous frames.

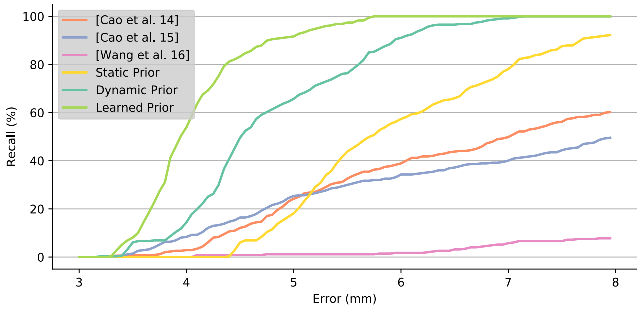
In order to make the minimization of Eq. 19 tractable, we approximate each minimization over pose within it by one Gauss-Newton step: $\delta\mathbf{T}^s = -\mathbf{J}^+\mathbf{e}(\tilde{\mathbf{T}}^s, \tilde{\beta}^s)$, where $\mathbf{J}$ is the Jacobian matrix of residual vector $\mathbf{e}$ differentiated with respect to $\tilde{\mathbf{T}}^s$, and $\mathbf{J}^+ = (\mathbf{J}^\top\mathbf{J})^{-1}\mathbf{J}^\top$ is the pseudo-inverse of $\mathbf{J}$. Letting $\Delta\mathbf{T}^s = \mathbf{T}^{*s} - \tilde{\mathbf{T}}^s$, this produces the following training energy:

$$\begin{aligned}
\mathcal{E}_{\text{train}}' &= \sum_{s=1}^{S} \left\|\delta\mathbf{T}^s - \Delta\mathbf{T}^s\right\|^2 \\
&= \sum_{s=1}^{S} \left\|\mathbf{J}^+\mathbf{e}(\tilde{\mathbf{T}}^s, \tilde{\beta}^s) + \Delta\mathbf{T}^s\right\|^2 \\
&= \sum_{s=1}^{S} \left\|\mathbf{J}^+\left(\mathbf{e}(\tilde{\mathbf{T}}^s, \tilde{\beta}^s) + \mathbf{J}\Delta\mathbf{T}^s\right)\right\|^2,
\end{aligned} \tag{20}$$

Intuitively, this objective encourages convergence to the ground-truth pose by enlarging the basin of convergence. The matrix $\mathbf{J}^+$ transforms the cost function from one minimizing measurement errors to one minimizing pose errors. However, it is desirable to drop the conditioning of $\mathbf{J}^+$ from Eq. 20 so that the optimization is formulated in the domain of measurement errors, which is in accordance with our run-time objective function in Eq. 11:

$$\begin{aligned}
\mathcal{E}_{\text{train}}'' &= \sum_{s=1}^{S} \left\|\mathbf{e}(\tilde{\mathbf{T}}^s, \tilde{\beta}^s) + \mathbf{J}\Delta\mathbf{T}^s\right\|^2, \\
&= \sum_{s=1}^{S}\sum_{k=1}^{K} w^k \Big(\lambda_{\text{land}}^{\text{pose}}\left\|\mathbf{e}_{\text{land}}^k(\tilde{\mathbf{T}}^s, \tilde{\beta}^s) + \mathbf{J}_{\text{land}}^k\Delta\mathbf{T}^s\right\|^2 + \\
&\qquad \lambda_{\text{flow}}^{\text{pose}}\left\|\mathbf{e}_{\text{flow}}^k(\tilde{\mathbf{T}}^s, \tilde{\beta}^s) + \mathbf{J}_{\text{flow}}^k\Delta\mathbf{T}^s\right\|^2\Big),
\end{aligned} \tag{21}$$

where $\mathbf{J}_\star^k$ are the parts of Jacobian specific to the $\star$ features of the k^{th} region. Note that the value in the braces is constant, making training very efficient.



Approaches	[Cao et al. 14]	[Cao et al. 15]	[Wang et al. 16]	Static Prior	Dynamic Prior	Learned Prior
Error (mm)	8.83	8.51	18.44	6.11	4.82	4.12

Fig. 5. The error-recall curves and average errors of different approaches using [Beeler et al. 2011] as the benchmark. We compare our learned dynamic rigidity prior (Learned Prior) to different variants (Static Prior and Dynamic Prior) for validation. Note that there is a base error of 3.6mm due to the difference of tracked mesh and ground-truth mesh. See Sec. 7 for details.

Finally, the hyper-parameters Θ are obtained by:

$$\Theta = \arg\min_{\Theta} \mathcal{E}''_{\text{train}}. \tag{22}$$

We minimize this objective (offline) until convergence using Ceres [Agarwal et al. 2016]. In addition, to avoid the trivial solution that $w^{k,s} = 0$, we enforce the normalization constraint that the weights should sum up to one for each sample: $\forall s, \sum_{k=1}^{K} w^{k,s} = 1$.

7 RESULTS

In this section, we present a validation of our proposed dynamic rigidity prior, a quantitative and qualitative evaluation of our face tracking system in comparison to existing solutions, a validation of the estimated head poses and discussion on the selection of the number of regions K. In our supplementary video, we present qualitative results of real-time facial animation retargeting and virtual face makeup applications using our method, which demonstrates superior quality as a real-time system. We also show the stability of our tracking by removing the rigid motion as well as an error analysis on the training data in the supplementary video.

Validation of Dynamic Rigidity Prior. We focus this validation on the key component of our method, the dynamic rigidity prior, to see how it affects the final tracking results. We compare the two variants to the full method (*Learned Prior*):

- *Static Prior. We assign a uniform rigid weight* $w^k = 1.0$ *to all regions in Eq. 7.*
- *Dynamic Prior. We assign uniform hyper-parameters* $\alpha^k = 1.0$ *and* $\sigma^k = 10.0$ *(chosen by hand to give the best results) to all regions in Eq. 18.*

To measure the tracking error numerically, we collected the reconstructed face data from the offline high-quality multi-view facial performance tracking system of [Beeler et al. 2011] as ground-truth, and apply each method on the frontal image sequence to generate the results. The error map is then computed by measuring the per-pixel depth distance between the tracking result and the ground-truth geometry. We present error-recall curves of the tracking errors as well as the average errors in Fig.5.

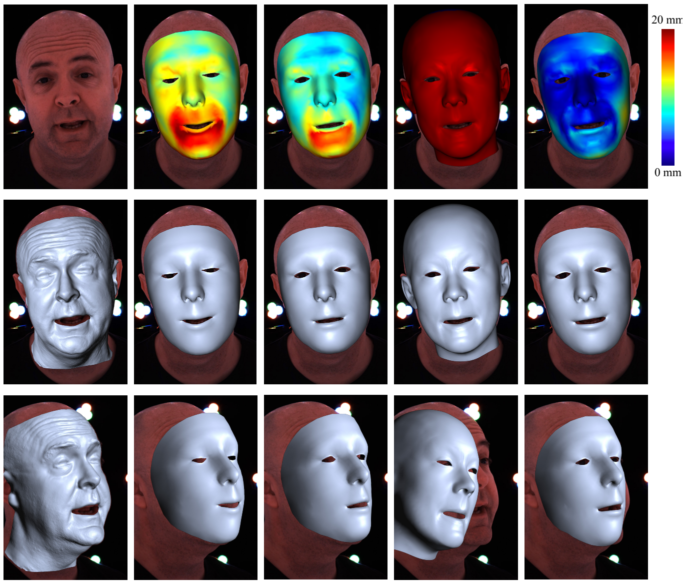


Fig. 6. Visual comparisons of different approaches. First column, from top to bottom: input image, reconstructed mesh from [Beeler et al. 2011] in frontal and side view respectively. The second to fifth columns, from right to left: results of [Cao et al. 2014a], [Cao et al. 2015], [Wang et al. 2016] and our method; from top to bottom: reconstructed mesh error, reconstructed mesh from frontal image, projected mesh to a side-view image.

As can be seen from the error curves and table, a dynamic prior achieves a lower error than the static prior, validating the conditioning of weights on region rigidity, while our full method achieves the lowest error, demonstrating the importance of learning the region weights in our dynamic rigidity prior. It is worth mentioning that due to the difference between the ground-truth detailed mesh and the coarser mesh used for tracking, there is a base error of *3.6mm* even with the optimal rigid pose fitting in neutral expression. The difference is therefore quite significant once the base error is discounted from the results.

Quantitative Comparisons. Following the quantitative evaluation framework introduced above, we further compare our method with three recent real-time face tracking methods [Cao et al. 2015, 2014a; Wang et al. 2016]. Fig.5 and Fig.6 show the comparison results. As we can see from the figures, our method achieves a significantly lower error than these other approaches.

The face tracking component of Wang et al. [2016] relies on only 2D face landmarks to optimize a 3D multi-linear face models [Cao et al. 2014b], which is similar to an optimization of Eq. 12 plus simple regularizations. Without denser photometric or motion-aware stabilization constraints, large rigid error and instability can be observed in the results.

Cao et al. [2014a] simultaneously regress both 2D landmarks and 3D expression coefficients from the input, then perform post-processing to estimate the final 3D shape. However, the rigid poses

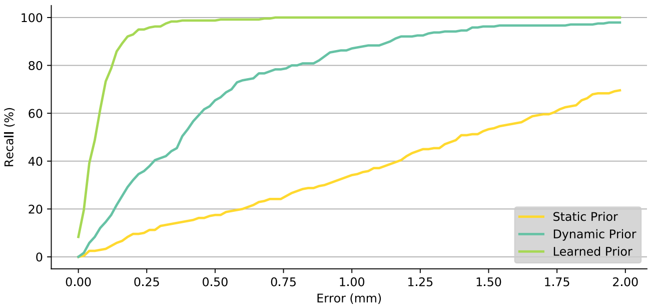


Fig. 7. The translational errors of our *Learned Prior* and different variants (*Static Prior* and *Dynamic Prior*) on a synthesized facial animation sequence.

are again still solely estimated based on sparse 2D landmarks, therefore the resulting 3D face shapes are often quite jittery and unstable, especially when the user is making exaggerated expressions, due to the coupling of rigid and non-rigid motions.

Their following work [Cao et al. 2015] tries to improve its tracking accuracy with sparse optical flow. However, this component is merely used to refine the non-rigid expressions to match the input video, and is not used in the rigid pose optimization. So although the projected mesh aligns well with the 2D video, this method does not correctly factor out the true rigid motion, which explains why the results remain unstable.

Validation on estimated head poses. To evaluate the accuracy of our estimated head poses, we synthesize another facial animation sequence which contains the ground truth rigid head pose for each frame. We then use the same method described in Sec. 6.2 to apply small perturbations to ground-truth expression coefficients and rigid pose parameters. We apply different variants (static prior, dynamic prior and learned prior) to track face from these perturbed initial parameters. Finally we compute the error between the estimated rigid poses and the ground-truth. While the estimated head rotational errors are similar, the translational errors are discriminative as shown in Fig. 7. Our learned rigidity dynamic prior achieves the lowest translational error.

Selection of the number of regions K. To validate our selection of the number of face regions, we apply our learned dynamic prior on different numbers of regions, and compute the translational errors on all our training data. Fig. 8 demonstrates the errors on different numbers of regions K. As K increases, the improvement to the training error diminishes. Throughout our experiments, we choose $K = 11$ regions as it offers a great trade-off between satisfactory expressiveness and computation complexity.

Qualitative Comparisons. Besides quantitative evaluations, we extend the comparisons to visually demonstrate our tracking quality on a rich set of in-the-wild input videos under various facial performances and environments captured with monocular web camera or cell phone.

First of all, we compare our work with an offline monocular face tracking method [Garrido et al. 2016a], which is able to generate high-fidelity face geometry over both coarse-level shape and fine-level detail (see Fig.9 top). However, despite its geometric quality, this method doesn't incorporate any rigid stabilization, which introduces obvious rigid drifting during tracking, especially along the depth axis. We note that, while modeling fine geometric details are not the focus of our work, our novel rigid stabilization framework can easily be incorporated into offline methods such as this, to further enhance the results.

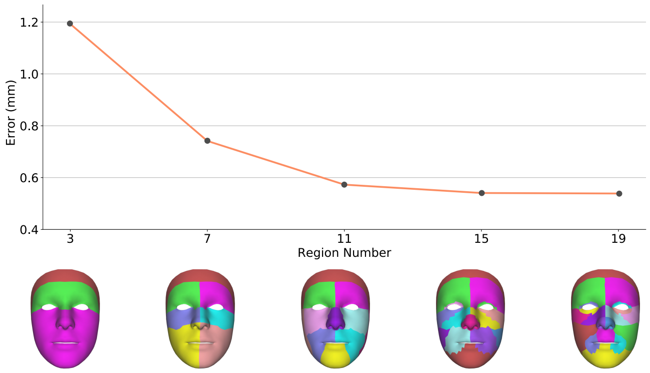


Fig. 8. The translational errors on the training data with different numbers of regions K and the corresponding region-based face models.

We then compare our approach with a real-time face tracking method *Face2Face* (Fig.9 middle) by Thies et al. [2016], which also adopts a dense photometric consistency measure to achieve pixel-accurate tracking. Even combining the sparse feature points and dense photo-consistency terms in their optimization, their results are not stable, suffering from pose drift, predominantly along the depth axis. In addition, the method needs to solve a complex nonlinear system, which is heavily optimized on GPU, making real-time performance on a mobile phone difficult.

We also compare our approach with a recent deep-learning-based single-view face reconstruction method [Tewari et al. 2017]. This end-to-end face regression network can extract 3D rigid poses and expression coefficients directly from single-view input images, without performing non-linear optimization each frame, which leads to extraordinary efficiency: as fast as 4ms/frame. However, the tracking accuracy cannot always compare to our optimization-based solution. Most significantly, since the approach works independently on each frame, the temporal inconsistency issue creates high rigid instability, even for gentle motions (Fig. 10).

Finally, we further evaluate our method by comparing with the face tracking system in Apple ARKit [Apple 2017], which stands as one of the best face tracking products and enables the popular performance-driven virtual character animation application of Apple Animoji. Since this face tracking system only works on a cellphone camera with depth information (used to capture face shape in the first frame only), we use an iPhone X to capture and track the face, and export the RGB video stream, which is then fed into our system (Fig.9 bottom). Although our method only relies on RGB input, it still achieves better-stabilized tracking results than ARKit, with comparable computation performance. Considering that ARKit is a highly optimized and tuned product, this comparison strongly demonstrates the robustness and accuracy of our method.

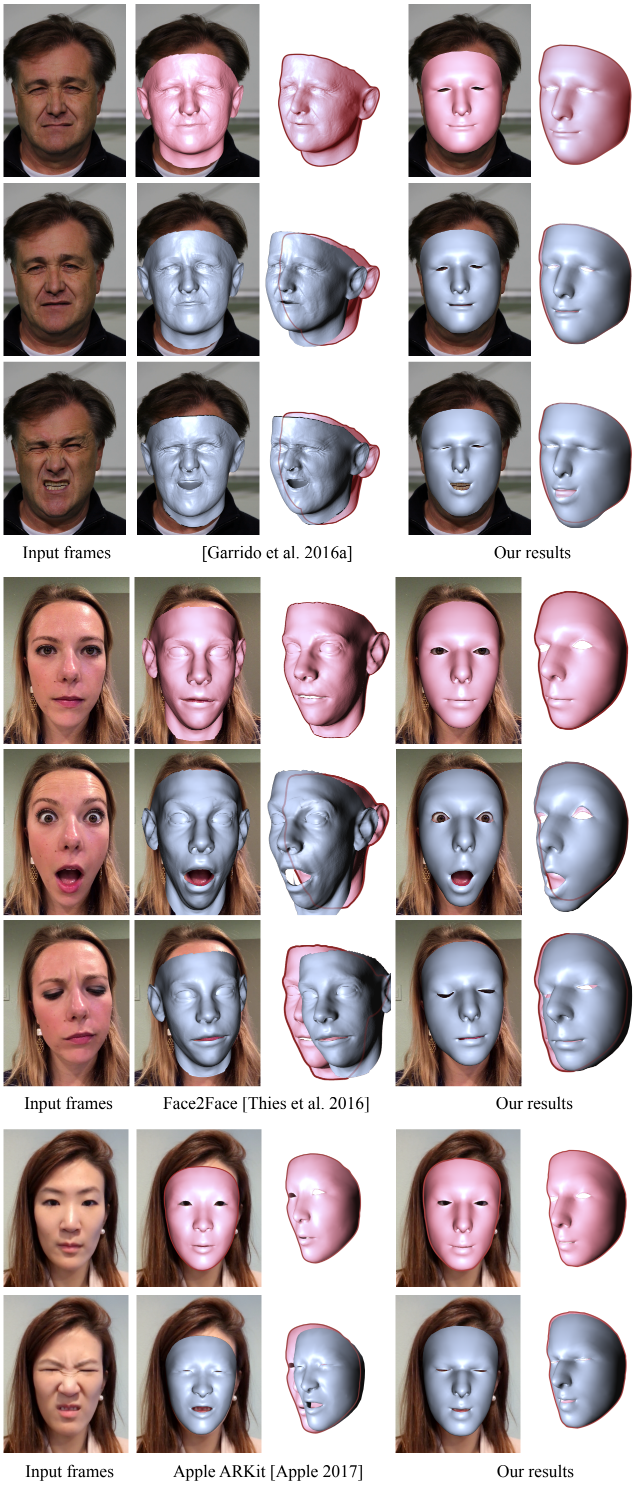


Fig. 9. Qualitative comparison between existing methods (middle) and ours (right). To highlight the differences, the resulting expressions (blue) are overlaid with the neutral expressions (red). Notice the abrupt head pose changes from existing methods.

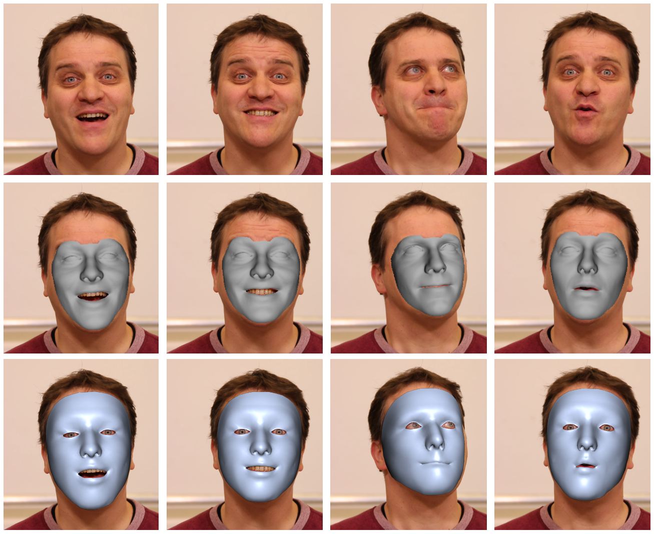


Fig. 10. Comparing the deep-learning-based method [Tewari et al. 2017] (second row) with ours (third row). Our method achieves better rigid stability; please refer to the supplementary video for clearer dynamic comparisons.

Performance. We implemented the described approach in C++, running in a single-thread on CPU; it does not use any multi-threading or GPU computation. We run all experiments on a MacBook Pro laptop with an Intel i7 (2.8 GHz) CPU and 16GB memory. For the live results, we directly use the built-in camera of the laptop, whose frame resolution is 1280x720. Our system takes about 5ms to track the 2D facial landmarks, 5ms to compute the optical flow, and another 10ms to go through the core optimizations. So in total our system takes around 20ms to process each frame. Since our method executes on a single CPU processor, it is trivial to port it to mobile platforms. We have implemented an unoptimized version on iOS, which is able to achieve equal quality results with more than 20 FPS on an iPhone 7. Please refer to the supplementary video for a recorded session interacting with the system.

Applications. Our stabilized face tracking system enables multiple facial-performance-driven applications, such as digital avatar retargeting and virtual face makeup. These applications are very sensitive to the rigid stabilization of face tracking. For digital avatar retargeting, unstable tracking results can lead to unstable movement and scale of avatar animation. For virtual face makeup, even subtle motion inconsistency can lead to significant relative drifting between the applied makeup sticker and the skin surface during tracking. Thanks to both accurate rigid motion and expression deformation tracked with our method, we can now accurately retarget digital avatars and apply virtual makeup to the face, to produce much more robust and realistic results (Fig.1 and Fig.11). Please refer to the supplementary video for more dynamic results.

8 CONCLUSION

We present a novel real-time monocular face tracking system, with improved rigid stability and model expressiveness. The dynamic rigidity prior we introduce, whose hyper-parameters are learned

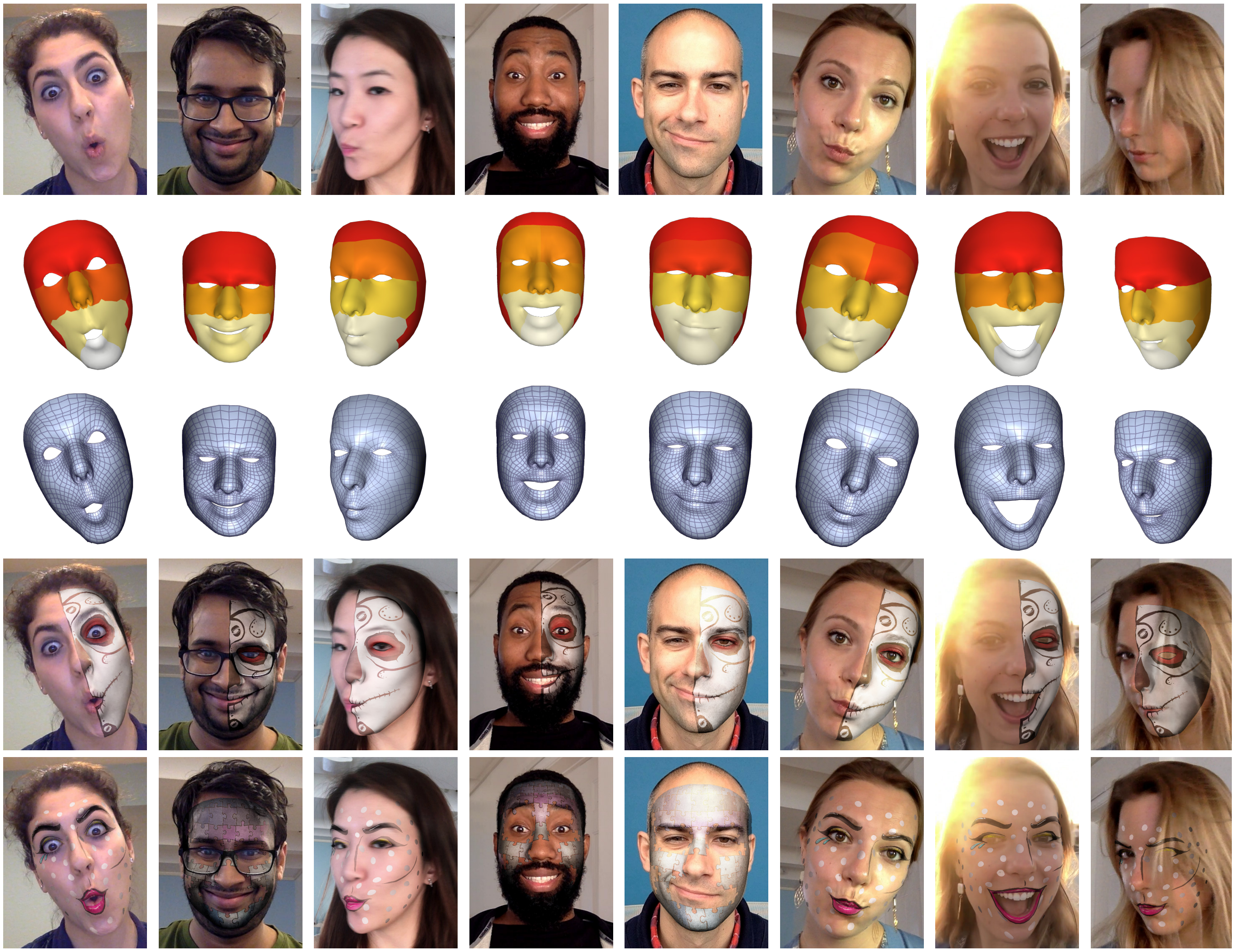


Fig. 11. More results. From top to bottom, each column shows the input image, rigidity weights, tracked mesh and two virtual makeup results. Our system can accurately track facial expressions from various performers even under challenging lighting and occlusions conditions. We encourage the reader to watch our supplementary video for more dynamic results.

from a new synthetic facial performance dataset, has been shown to significantly improve the rigid stability of head pose estimation over previous systems. This, combined with a region-based expression model and dense motion-guided correctives, enables greater tracking fidelity and model expressiveness. With our system, the quality of facial-performance-based applications, including virtual face make-up, can be greatly enhanced.

Although our system achieves superior tracking results on most inputs, it still shares some common limitations of monocular face tracking methods: 1) since our method partially depends on input facial landmarks, incorrect landmark locations caused by large face rotation, occlusion or poor lighting can affect the tracking accuracy; 2) too exaggerated or strange expressions, which are not well represented by the expression blendshapes, may not be perfectly tracked even with our region-based face models; 3) focusing on stability and expressiveness, our current results lack fine-scale facial details, which may impact the visual realism. However, our dynamic rigidity prior can be applied into any high-quality online or offline monocular facial performance capture method, enabling rigid stabilized results.

ACKNOWLEDGMENTS

We would like to thank Zehao Xue for his help on the assets and visualizations as well as our performers: Aletta Hiemstra, Jung Kim, Anisa Bashiri, Harsh Agrawal and Ian Wehrman. We thank Fuhao Shi, Michael Zollhoefer, Justus Thies, Pablo Garrido and Christian Theobalt for the comparisons. We also thank Thabo Beeler for kindly providing their high-quality face tracking results. Finally, we thank all the anonymous reviewers for their valuable comments.

REFERENCES

Sameer Agarwal, Keir Mierle, and Others. 2016. Ceres Solver. http://ceres-solver.org. (2016).

Apple. 2017. Animoji. A new way to get into character. (2017). https://www.apple.com/iphone-x/#truedepth-camera

T. Beeler, B. Bickel, R. Sumner, P. Beardsley, and M. Gross. 2010. High-Quality Single-Shot Capture of Facial Geometry. *ACM Trans. Graphics (Proc. SIGGRAPH)* (2010).

Thabo Beeler and Derek Bradley. 2014. Rigid Stabilization of Facial Expressions. *ACM Trans. Graph.* 33, 4, Article 44 (July 2014), 9 pages. https://doi.org/10.1145/2601097.2601182

Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W. Sumner, and Markus Gross. 2011. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graphics (Proc. SIGGRAPH)* 30, Article 75 (2011), 75:1–75:10 pages. Issue 4.

V. Blanz and T. Vetter. 1999. A morphable model for the synthesis of 3D faces. In *Proceedings of SIGGRAPH*. 187–194.

Sofien Bouaziz, Yangang Wang, and Mark Pauly. 2013. Online modeling for realtime facial animation. *ACM Trans. Graphics (Proc. SIGGRAPH)* 32, 4, Article 40 (2013), 40:1–40:10 pages.

Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. 2010. High Resolution Passive Facial Performance Capture. In *ACM SIGGRAPH 2010 Papers (SIGGRAPH '10)*. ACM, New York, NY, USA, Article 41, 10 pages. https://doi.org/10.1145/1833349.1778778

Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. 2015. Real-time High-fidelity Facial Performance Capture. *ACM Trans. Graph.* 34, 4, Article 46 (July 2015), 9 pages. https://doi.org/10.1145/2766943

Chen Cao, Qiming Hou, and Kun Zhou. 2014a. Displaced Dynamic Expression Regression for Real-time Facial Tracking and Animation. *ACM Trans. Graphics (Proc. SIGGRAPH)* 33, 4, Article 43 (2014), 43:1–43:10 pages.

Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 2013. 3D shape regression for real-time facial animation. *ACM Trans. Graphics (Proc. SIGGRAPH)* 32, 4, Article 41 (2013), 41:1–41:10 pages.

Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. 2014b. FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE Transactions on Visualization and Computer Graphics* 20, 3 (March 2014), 413–425. https://doi.org/10.1109/TVCG.2013.249

Jin-Xiang Chai, Jing Xiao, and Jessica Hodgins. 2003. Vision-based Control of 3D Facial Animation. In *SCA*.

Yen-Lin Chen, Hsiang-Tao Wu, Fuhao Shi, Xin Tong, and Jinxiang Chai. 2013. Accurate and Robust 3D Facial Capture Using a Single RGBD Camera. In *ICCV*.

Yasutaka Furukawa and Jean Ponce. 2009. Dense 3D Motion Capture for Human Faces. In *CVPR*.

Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. 2013. Reconstructing Detailed Dynamic Face Geometry from Monocular Video. In *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, Vol. 32. 158:1–158:10.

Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2016a. Reconstruction of Personalized 3D Face Rigs from Monocular Video. *ACM Trans. Graph.* 35, 3, Article 28 (May 2016), 15 pages. https://doi.org/10.1145/2890493

Pablo Garrido, Michael Zollhöfer, Chenglei Wu, Derek Bradley, Patrick Pérez, Thabo Beeler, and Christian Theobalt. 2016b. Corrective 3D Reconstruction of Lips from Monocular Video. *ACM Trans. Graph.* 35, 6, Article 219 (Nov. 2016), 11 pages. https://doi.org/10.1145/2980179.2982419

Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul Debevec. 2011. Multiview face capture using polarized spherical gradient illumination. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 30, 6, Article 129 (2011), 129:1–129:10 pages.

Haoda Huang, Jinxiang Chai, Xin Tong, and Hsiang-Tao Wu. 2011. Leveraging motion capture and 3D scanning for high-fidelity facial performance acquisition. *ACM Trans. Graphics (Proc. SIGGRAPH)* 30, 4, Article 74 (2011), 74:1–74:10 pages.

Pushkar Joshi, Wen C. Tien, Mathieu Desbrun, and Frédéric Pighin. 2003. Learning Controls for Blend Shape Based Realistic Facial Animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '03)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 187–192. http://dl.acm.org/citation.cfm?id=846276.846303

V. Kazemi and J. Sullivan. 2014. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 1867–1874. https://doi.org/10.1109/CVPR.2014.241

Martin Klaudiny and Adrian Hilton. 2012. High-detail 3D capture and non-sequential alignment of facial performance. In *3DIMPVT*.

Till Kroeger, Radu Timofte, Dengxin Dai, editor="Leibe Bastian Van Gool, Luc", Jiri Matas, Nicu Sebe, and Max Welling. 2016. *Fast Optical Flow Using Dense Inverse Search*. Springer International Publishing, Cham, 471–488. https://doi.org/10.1007/978-3-319-46493-0_29

Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. 2013. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graphics (Proc. SIGGRAPH)* 32, 4, Article 42 (2013), 42:1–42:10 pages.

Wan-Chun Ma, Tim Hawkins, Pieter Peers, Charles-Felix Chabert, Malte Weiss, and Paul Debevec. 2007. Rapid Acquisition of Specular and Diffuse Normal Maps from Polarized Spherical Gradient Illumination. In *Eurographics Symposium on Rendering*. 183–194.

Thomas Neumann, Kiran Varanasi, Stephan Wenger, Markus Wacker, Marcus Magnor, and Christian Theobalt. 2013. Sparse Localized Deformation Components. *ACM Trans. Graph.* 32, 6, Article 179 (Nov. 2013), 10 pages. https://doi.org/10.1145/2508363.2508417

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On Spectral Clustering: Analysis and an Algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01)*. MIT Press, Cambridge, MA, USA, 849–856. http://dl.acm.org/citation.cfm?id=2980539.2980649

Julien Peyras, Adrien Bartoli, Hugo Mercier, and Patrice Dalle. 2007. Segmented AAMs Improve Person-Independent Face Fitting. In *In BMVCâĂŹ07 - Proceedings of the 18th British Machine Vision Conference*.

Taehyun Rhee, Youngkyoo Hwang, James Dokyoon Kim, and Changyeong Kim. 2011. Real-time Facial Animation from Live Video Tracking. In *Proc. SCA*. 215–224.

Fuhao Shi, Hsiang-Tao Wu, Xin Tong, and Jinxiang Chai. 2014a. Automatic Acquisition of High-fidelity Facial Performances Using Monocular Videos. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 33 (2014). Issue 6.

Fuhao Shi, Hsiang-Tao Wu, Xin Tong, and Jinxiang Chai. 2014b. Automatic Acquisition of High-fidelity Facial Performances Using Monocular Videos. *ACM Trans. Graph.* 33, 6, Article 222 (Nov. 2014), 13 pages. https://doi.org/10.1145/2661229.2661290

Supasorn Suwajanakorn, Ira Kemelmacher-Shlizerman, and Steven M. Seitz. 2014. Total Moving Face Reconstruction. In *ECCV*.

J. Rafael Tena, Fernando De la Torre, and Iain Matthews. 2011. Interactive Region-based Linear 3D Face Models. *ACM Trans. Graph.* 30, 4, Article 76 (July 2011), 10 pages. https://doi.org/10.1145/2010324.1964971

Ayush Tewari, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Theobalt Christian. 2017. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *The IEEE International Conference on Computer Vision (ICCV)*.

J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. 2016. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2387–2395. https://doi.org/10.1109/CVPR.2016.262

L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt. 2012. Lightweight Binocular Facial Performance Capture under Uncontrolled Lighting. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 31, 6, Article 187 (2012).

Congyi Wang, Fuhao Shi, Shihong Xia, and Jinxiang Chai. 2016. Realtime 3D Eye Gaze Animation Using a Single RGB Camera. *ACM Trans. Graph.* 35, 4, Article 118 (July 2016), 14 pages. https://doi.org/10.1145/2897824.2925947

Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. 2011. Realtime Performance-Based Facial Animation. *ACM Trans. Graphics (Proc. SIGGRAPH)* 30, 4 (2011), 77:1–77:10.

Thibaut Weise, Hao Li, Luc Van Gool, and Mark Pauly. 2009. Face/Off: live facial puppetry. In *Proc. SCA*. 7–16.

Chenglei Wu, Derek Bradley, Markus Gross, and Thabo Beeler. 2016. An Anatomically-constrained Local Deformation Model for Monocular Face Capture. *ACM Trans. Graph.* 35, 4, Article 115 (July 2016), 12 pages. https://doi.org/10.1145/2897824.2925882

Li Zhang, Noah Snavely, Brian Curless, and Steven M. Seitz. 2004. Spacetime faces: high resolution capture for modeling and animation. *ACM Trans. Graphics (Proc. SIGGRAPH)* (2004), 548–558.

Qingshan Zhang, Z. Liu, Gaining Quo, D. Terzopoulos, and Heung-Yeung Shum. 2006. Geometry-driven photorealistic facial expression synthesis. *IEEE Transactions on Visualization and Computer Graphics* 12, 1 (Jan 2006), 48–60. https://doi.org/10.1109/TVCG.2006.9