To appear in ACM TOG 34(1).

Driving High-Resolution Facial Scans with Video Performance Capture

Graham Fyffe Andrew Jones Oleg Alexander Ryosuke Ichikari Paul Debevec *
USC Institute for Creative Technologies

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

Figure 1: *(a) High resolution geometric and reflectance information from multiple static expression scans is automatically combined with (d) dynamic video frames to recover (b) matching animated high resolution performance geometry that can be (c) relit under novel illumination from a novel viewpoint. In this example, the performance is recovered using only the single camera viewpoint in (d).*

Abstract

We present a process for rendering a realistic facial performance with control of viewpoint and illumination. The performance is based on one or more high-quality geometry and reflectance scans of an actor in static poses, driven by one or more video streams of a performance. We compute optical flow correspondences between neighboring video frames, and a sparse set of correspondences between static scans and video frames. The latter are made possible by leveraging the relightability of the static 3D scans to match the viewpoint(s) and appearance of the actor in videos taken in arbitrary environments. As optical flow tends to compute proper correspondence for some areas but not others, we also compute a smoothed, per-pixel confidence map for every computed flow, based on normalized cross-correlation. These flows and their confidences yield a set of weighted triangulation constraints among the static poses and the frames of a performance. Given a single artist-prepared face mesh for one static pose, we optimally combine the weighted triangulation constraints, along with a shape regularization term, into a consistent 3D geometry solution over the entire performance that is drift-free by construction. In contrast to previous work, even partial correspondences contribute to drift minimization, for example where a successful match is found in the eye region but not the mouth. Our shape regularization employs a differential shape term based on a spatially varying blend of the differential shapes of the static poses and neighboring dynamic poses, weighted by the associated flow confidences. These weights also permit dynamic reflectance maps to be produced for the performance by blending the static scan maps. Finally, as the geometry and maps are represented on a consistent artist-friendly mesh, we render the resulting high-quality animated face geometry and animated reflectance maps using standard rendering tools.

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

Keywords: Facial performance capture, temporal correspondence

*e-mail:{fyffe,jones,oalexander,debevec}@ict.usc.edu

1 Introduction

Recent facial geometry scanning techniques can capture very high resolution geometry, including high-frequency details such as skin pores and wrinkles. When animating these highly detailed faces, highly accurate temporal correspondence is required. At present, the highest quality facial geometry is produced by static scanning techniques, where the subject holds a facial pose for several seconds. This permits the use of high-resolution cameras for accurate stereo reconstruction and active illumination to recover pore-level resolution surface details. Such techniques also capture high-quality surface reflectance maps, enabling realistic rendering of the captured faces. Alternatively, static facial poses may be captured using facial casts combined with detail acquired from surface imprints. Unfortuately, *dynamic* scanning techniques are unable to provide the same level of detail as static techniques, even when high-speed cameras and active illumination are employed.

The classic approach to capturing facial motion is to use markers or face paint to track points on the face. However, such techniques struggle to capture the motion of the eyes and mouth, and rely on a high-quality facial rig to provide high-frequency skin motion and wrinkling. The best results are achieved when the rig is based on high-resolution static scans of the same subject. A second approach is to capture a performance with one or more passive video cameras. Such setups are lightweight as they use environmental illumination and off-the-shelf video cameras. As the camera records the entire face, it should be possible to recover eye and mouth motion missed by sparse markers. Still, by itself, passive video cannot match the resolution of static scans. While it is possible to emboss some video texture on the face [Bradley et al. 2010][Beeler et al. 2011][Valgaerts et al. 2012], many facial details appear only in specular reflections and are not visible under arbitrary illumination.

We present a technique for creating realistic facial animation from a set of high-resolution scans of an actor's face, driven by passive video of the actor from one or more viewpoints. The videos can be shot under existing environmental illumination using off-the-shelf HD video cameras. The static scans can come from a variety of sources including facial casts, passive stereo, or active illumination techniques. High-resolution detail and relightable reflectance properties in the static scans can be transferred to the performance using generated per-pixel weight maps. We operate our algorithm on a *performance flow graph* that represents dense correspondences between dynamic frames and multiple static scans, leveraging GPU-based optical flow to efficiently construct the graph. Besides a single artist remesh of a scan in neutral pose, our method requires no rigging, no training of appearance models, no facial feature detection, and no manual annotation of any kind. As a byproduct of our method we also obtain a non-rigid registration between the artist mesh and each static scan. Our principal contributions are:

- An efficient scheme for selecting a sparse subset of image pairs for optical flow computation for drift-free tracking.
- A fully coupled 3D tracking method with differential shape regularization using multiple locally weighted target shapes.
- A message-passing-based optimization scheme leveraging lazy evaluation of energy terms enabling fully-coupled optimization over an entire performance.

2 Related Work

As many systems have been built for capturing facial geometry and reflectance, we will restrict our discussion to those that establish some form of dense temporal correspondence over a performance.

Many existing algorithms compute temporal correspondence for a sequence of temporally inconsistent geometries generated by e.g. structured light scanners or stereo algorithms. These algorithms operate using only geometric constraints [Popa et al. 2010] or by deforming template geometry to match each geometric frame [Zhang et al. 2004]. The disadvantage of this approach is that the per-frame geometry often contains missing regions or erroneous geometry which must be filled or filtered out, and any details that are missed in the initial geometry solution are non-recoverable.

Other methods operate on video footage of facial performances. Methods employing frame-to-frame motion analysis are subject to the accumulation of error or "drift" in the tracked geometry, prompting many authors to seek remedies for this issue. We therefore limit our discussion to methods that make some effort to address drift. Li et al. [1993] compute animated facial blendshape weights and rigid motion parameters to match the texture of each video frame to a reference frame, within a local minimum determined by a motion prediction step. Drift is avoided whenever a solid match can be made back to the reference frame. [DeCarlo and Metaxas 1996] solves for facial rig control parameters to agree with sparse monocular optical flow constraints, applying forces to pull model edges towards image edges in order to combat drift. [Guenter et al. 1998] tracks motion capture dots in multiple views to deform a neutral facial scan, increasing the realism of the rendered performance by projecting video of the face (with the dots digitally removed) onto the deforming geometry. The "Universal Capture" system described in [Borshukov et al. 2003] dispenses with the dots and uses dense multi-view optical flow to propagate vertices from an initial neutral expression. User intervention is required to correct drift when it occurs. [Hawkins et al. 2004] uses performance tracking to automatically blend between multiple high-resolution facial scans per facial region, achieving realistic multi-scale facial deformation without the need for reprojecting per-frame video, but uses dots to avoid drift. Bradley et al. [2010] track motion using dense multi-view optical flow, with a final registration step between the neutral mesh and every subsequent frame to reduce drift. Beeler et al. [2011] explicitly identify anchor frames that are similar to a manually chosen reference pose using a simple image difference metric, and track the performance bidirectionally between anchor frames. Non-sequential surface tracking [Klaudiny and Hilton 2012] finds a minimum-cost spanning tree over the frames in a performance based on sparse feature positions, tracking facial geometry across edges in the tree with an additional temporal fusion step. Valgaerts et al. [2012] apply scene flow to track binocular passive video with a regularization term to reduce drift.

One drawback to all such optical flow tracking algorithms is that the face is tracked from one pose to another *as a whole*, and success of the tracking depends on accurate optical flow between images of the entire face. Clearly, the human face is capable of repeating different poses over different parts of the face asynchronously, which the holistic approaches fail to model. For example, if the subject is talking with eyebrows raised and later with eyebrows lowered, a holistic approach will fail to exploit similarities in mouth poses when eyebrow poses differ. In contrast, our approach constructs a graph considering similarities over multiple regions of the face across the performance frames and a set of static facial scans, removing the need for sparse feature tracking or anchor frame selection.

Blend-shape based animation rigs are also used to reconstruct dynamic poses based on multiple face scans. The company Image Metrics (now Faceware) has developed commercial software for driving a blend-shape rig with passive video based on active appearance models [Cootes et al. 1998]. Weise et al. [2011] automatically construct a personalized blend shape rig and drive it with Kinect depth data using a combination of as-rigid-as-possible constraints and optical flow. In both cases, the quality of the resulting tracked performance is directly related to the quality of the rig. Each tracked frame is a linear combination of the input blend-shapes, so any performance details that lie outside the domain spanned by the rig will not be reconstructed. Huang et al. [2011] automatically choose a minimal set of blend shapes to scan based on previously captured performance with motion capture markers. Recreating missing detail requires artistic effort to add corrective shapes and cleanup animation curves [Alexander et al. 2009]. There has been some research into other non-traditional rigs incorporating scan data. Ma et al. [2008] fit a polynomial displacement map to dynamic scan training data and generate detailed geometry from sparse motion capture markers. Bickel et al. [2008] locally interpolate a set of static poses using radial basis functions driven by motion capture markers. Our method combines the shape regularization advantages of blendshapes with the flexibility of optical flow based tracking. Our optimization algorithm leverages 3D information from static scans without constraining the result to lie only within the linear combinations of the scans. At the same time, we obtain per-pixel blend weights that can be used to produce per-frame reflectance maps.

3 Data Capture and Preparation

We capture high-resolution static geometry using multi-view stereo and gradient-based photometric stereo [Ghosh et al. 2011]. The scan set includes around 30 poses largely inspired by the Facial Action Coding System (FACS) [Ekman and Friesen 1978], selected to span nearly the entire range of possible shapes for each part of the face. For efficiency, we capture some poses with the subject combining FACS action units from the upper and lower half of the face. For example, combining eyebrows raise and cheeks puff into a single scan. Examples of the input scan geometry can be seen in Fig. 2. A base mesh is defined by an artist for the neutral pose scan.

The artist mesh has an efficient layout with edge loops following the wrinkles of the face. The non-neutral poses are represented as raw scan geometry, requiring no artistic topology or remeshing.

We capture dynamic performances using up to six Canon 1DX DSLR cameras under constant illumination. In the simplest case, we use the same cameras that were used for the static scans and switch to 1920×1080 30p movie mode. We compute a sub-frame-accurate synchronization offset between cameras using a correlation analysis of the audio tracks. This could be omitted if cameras with hardware synchronization are employed. Following each performance, we capture a video frame of a calibration target to calibrate camera intrinsics and extrinsics. We relight (and when necessary, repose) the static scan data to resemble the illumination conditions observed in the performance video. In the simplest case, the illumination field resembles one of the photographs taken during the static scan process and no relighting is required.

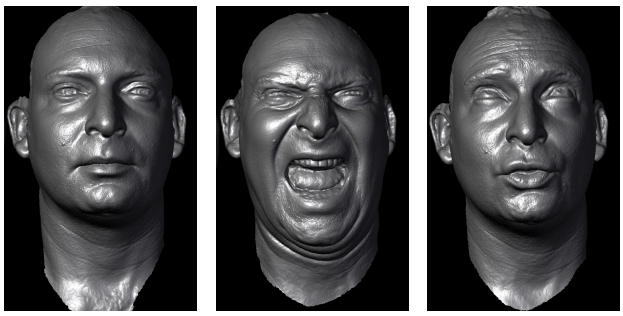


Figure 2: *Sample static scans (showing geometry only).*

4 The Performance Flow Graph

Optical-flow-based tracking algorithms such as [Bradley et al. 2010][Beeler et al. 2011][Klaudiny and Hilton 2012] relate frames of a performance to each other based on optical flow correspondences over a set of image pairs selected from the performance. These methods differ in part by the choice of the image pairs to be employed. We generalize this class of algorithms using a structure we call the *performance flow graph*, which is a complete graph with edges representing dense 2D correspondences between *all pairs* of images, with each edge having a weight, or confidence, of the associated estimated correspondence field. The graphs used in previous works, including anchor frames [Beeler et al. 2011] and non-sequential alignment with temporal fusion [Klaudiny and Hilton 2012], can be represented as a performance flow graph having unit weight for the edges employed by the respective methods, and zero weight for the unused edges. We further generalize the performance flow graph to include a dense *confidence field* associated with each correspondence field, allowing the confidence to vary spatially over the image. This enables our technique to exploit relationships between images where only a partial correspondence was able to be computed (for example, a pair of images where the mouth is similar but the eyes are very different). Thus our technique can be viewed as an extension of anchor frames or minimum spanning trees to minimize drift independently over different regions of the face.

A performance capture system that considers correspondences between all possible image pairs naturally minimizes drift. However, this would require an exorbitant number of graph edges, so we instead construct a graph with a reduced set of edges that approximates the complete graph, in the sense that the correspondences are *representative* of the full set with respect to confidence across the regions of the face. Our criterion for selecting the edges to include in the performance flow graph is that any two images having a high

Figure 3: *performance flow graph showing optical flow correspondences between static and dynamic images. Red lines represent optical flow between neighboring frames within a performance. Blue, green, and orange lines represent optical flow between dynamic and static images. Based on initial low-resolution optical flow, we construct a sparse graph requiring only a small subset of high resolution flows to be computed between static scans and dynamic frames.*

confidence correspondence between them in the complete graph of possible correspondences ought to have a *path* between them (a concatenation of one or more correspondences) in the constructed graph with nearly as high confidence (including the reduction in confidence from concatenation). We claim that correspondences between temporally neighboring dynamic frames are typically of high quality, and no concatenation of alternative correspondences can be as confident, therefore we always include a graph edge between each temporally neighboring pair of dynamic frames. Correspondences between frames with larger temporal gaps are well-approximated by concatenating neighbors, but decreasingly so over larger temporal gaps (due to drift). We further claim that whenever enough drift accumulates to warrant including a graph edge over the larger temporal gap, there exists a path with nearly as good confidence that passes through one of the predetermined *static scans* (possibly a different static scan for each region of the face). We justify this claim by noting the 30 static poses based on FACS ought to span the space of performances well enough that any region of any dynamic frame can be corresponded to some region in some static scan with good confidence. Therefore we do not include any edges between non-neighboring dynamic frames, and instead consider only edges between a static scan and a dynamic frame as candidates for inclusion (visualized in Fig. 3). Finally, as the drift accumulated from the concatenation described above warrants additional edges only sparsely over time, we devise a coarse-to-fine graph construction strategy using only a sparse subset of static-to-dynamic graph edges. We detail this strategy in Section 4.1.

4.1 Constructing the Performance Flow Graph

The images used in our system consist of one or more dynamic sequences of frames captured from one or more viewpoints, and roughly similar views of a set of high-resolution static scans. The nodes in our graph represent static poses (associated with static scans) and dynamic poses (associated with dynamic frames from one or more sequences). We construct the performance flow graph by computing a large set of static-to-dynamic optical flow correspondences at a reduced resolution for only a single viewpoint, and

then omit redundant correspondences using a novel voting algorithm to select a sparse set of correspondences that is representative of the original set. We then compute high-quality optical flow correspondences at full resolution for the sparse set, and include all viewpoints. The initial set of correspondences consists of quarter-resolution optical flows from each static scan to every n^{th} dynamic frame. For most static scans we use every 5^{th} dynamic frame, while for the eyes-closed scan we use every dynamic frame in order to catch rapid eye blinks. We then compute normalized cross correlation fields between the warped dynamic frames and each original static scan to evaluate the confidence of the correspondences. These correspondences may be computed in parallel over multiple computers, as there is no sequential dependency between them. We find that at quarter resolution, flow-based cross correlation correctly assigns low confidence to incorrectly matched facial features, for example when flowing disparate open and closed mouth shapes. To reduce noise and create a semantically meaningful metric, we average the resulting confidence over twelve facial regions (see Fig. 4). These facial regions are defined once on the neutral pose, and are warped to all other static poses using rough static-to-static optical flow. Precise registration of regions is not required, as they are only used in selecting the structure of the performance graph. In the subsequent tracking phase, per-pixel confidence is used.

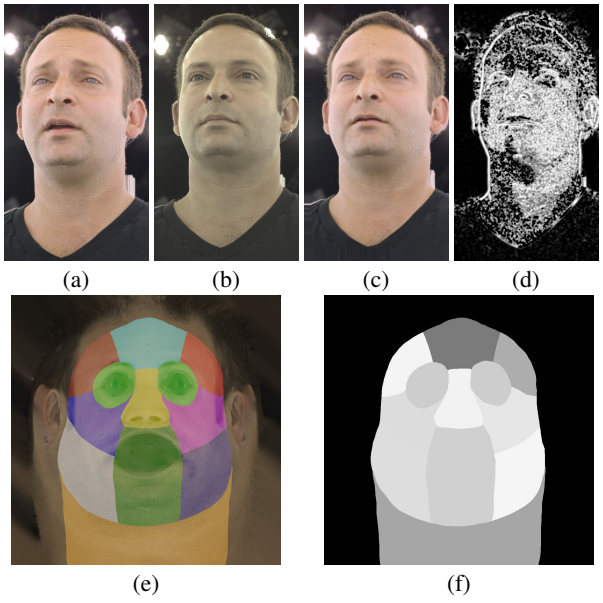


Figure 4: *We compute an initial low-resolution optical flow between a dynamic image (a) and static image (b). We then compute normalized cross correlation between the static image (b) and the warped dynamic image (c) to produce the per-pixel confidence shown in (d). We average these values for 12 regions (e) to obtain a per-region confidence value (f). This example shows correlation between the neutral scan and a dynamic frame with the eyebrows raised and the mouth slightly open. The forehead and mouth regions are assigned appropriately lower confidences.*

Ideally we want the performance flow graph to be sparse. Besides temporally adjacent poses, dynamic poses should only connect to similar static poses and edges should be evenly distributed over time to avoid accumulation of drift. We propose an iterative greedy voting algorithm based on the per-region confidence measure to identify good edges. The confidence of correspondence between the dynamic frames and any region of any static facial scan can be viewed as a curve over time (depicted in Fig. 5). In each iteration we identify the maximum confidence value over all regions, all scans, and all frames. We add an edge between the identified dynamic pose

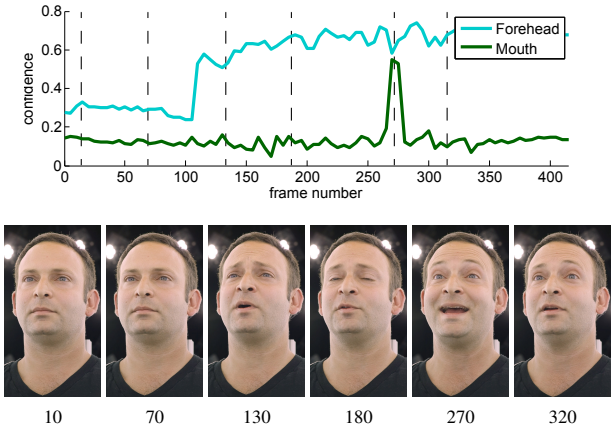


Figure 5: *A plot of the per-region confidence metric over time. Higher numbers indicate greater correlation between the dynamic frames and a particular static scan. The cyan curve represents the center forehead region of a brows-raised static scan which is active throughout the later sequence. The green curve represents the mouth region for an extreme mouth-open scan which is active only when the mouth opens to its fullest extent. The dashed lines indicate the timing of the sampled frames shown on the bottom row.*

and static pose to the graph. We then adjust the recorded confidence of the identified region by subtracting a hat function scaled by the maximum confidence and centered around the maximum frame, indicating that the selected edge has been accounted for, and temporal neighbors partly so. All other regions are adjusted by subtracting similar hat functions, scaled by the (non-maximal) per-region confidence of the identified flow. This suppresses any other regions that are satisfied by the flow. The slope of the hat function represents a loss of confidence as this flow is combined with adjacent dynamic-to-dynamic flows. We then iterate and choose the new highest confidence value, until all confidence values fall below a threshold. The two parameters (the slope of the hat function and the final threshold value) provide intuitive control over the total number of graph edges. We found a reasonable hat function falloff to be a 4% reduction for every temporal flow and a threshold value that is 20% of the initial maximum confidence. After constructing the graph, a typical 10-20 second performance flow graph will contain 100-200 edges between dynamic and static poses. Again, as the change between sequential frames is small, we preserve all edges between neighboring dynamic poses.

After selecting the graph edges, final HD resolution optical flows are computed for all active cameras and for all retained graph edges. We directly load video frames using nVidia's h264 GPU decoder and feed them to the FlowLib implementation of GPU-optical flow [Werlberger 2012]. Running on a Nvidia GTX 680, computation of quarter resolution flows for graph construction take less than one second per flow. Full-resolution HD flows for dynamic-to-dynamic images take 8 seconds per flow, and full-resolution flows between static and dynamic images take around 23 seconds per flow due to a larger search window. More sophisticated correspondence estimation schemes could be employed within our framework, but our intention is that the framework be *agnostic* to this choice and *robust* to imperfections in the pairwise correspondences. After computing optical flows and confidences, we synchronize all the flow sequences to a primary camera by warping each flow frame forward or backward in time based on the sub-frame synchronization offsets between cameras.

We claim that an approximate performance flow graph constructed in this manner is more representative of the complete set of possible

correspondences than previous methods that take an all-or-nothing approach to pair selection, while still employing a number of optical flow computations on the same order as previous methods (i.e. temporal neighbors plus additional sparse image pairs).

5 Fully Coupled Performance Tracking

The performance flow graph is representative of all the constraints we could glean from 2D correspondence analysis of the input images, and now we aim to put those constraints to work. We formulate an energy function in terms of the 3D vertex positions of the artist mesh as it deforms to fit all of the dynamic and static poses in the performance flow graph in a *common head coordinate system*, as well as the associated head-to-world rigid transforms. We collect the free variables into a vector $\theta = (\mathbf{x}_i^p, \mathbf{R}_p, \mathbf{t}_p | p \in \mathcal{D} \cup \mathcal{S}, i \in \mathcal{V})$, where $\mathbf{x}_i^p$ represents the 3D vertex position of vertex i at pose p in the common head coordinate system, $\mathbf{R}_p$ and $\mathbf{t}_p$ represent the rotation matrix and translation vector that rigidly transform pose p from the common head coordinate system to world coordinates, $\mathcal{D}$ is the set of dynamic poses, $\mathcal{S}$ is the set of static poses, and $\mathcal{V}$ is the set of mesh vertices. The energy function is then:

$$\mathbf{E}(\theta) = \sum_{(p,q)\in\mathcal{F}} (\mathbf{E}_{\text{corr}}^{pq} + \mathbf{E}_{\text{corr}}^{qp}) + \lambda \sum_{p\in\mathcal{D}\cup\mathcal{S}} |\mathcal{F}_p| \mathbf{E}_{\text{shape}}^p + \zeta \sum_{p\in\mathcal{S}} |\mathcal{F}_p| \mathbf{E}_{\text{wrap}}^p + \gamma |\mathcal{F}_g| \mathbf{E}_{\text{ground}}, \quad (1)$$

where $\mathcal{F}$ is the set of performance flow graph edges, $\mathcal{F}_p$ is the subset of edges connecting to pose p, and g is the ground (neutral) static pose. This function includes:

- dense correspondence constraints $\mathbf{E}_{\text{corr}}^{pq}$ associated with the edges of the performance flow graph,
- shape regularization terms $\mathbf{E}_{\text{shape}}^p$ relating the differential shape of dynamic and static poses to their graph neighbors,
- "shrink wrap" terms $\mathbf{E}_{\text{wrap}}^p$ to conform the static poses to the surface of the static scan geometries,
- a final grounding term $\mathbf{E}_{\text{ground}}$ to prefer the vertex positions in a neutral pose to be close to the artist mesh vertex positions.

We detail these terms in sections 5.2 - 5.5. Note we do not employ a stereo matching term, allowing our technique to be robust to small synchronization errors between cameras. As the number of poses and correspondences may vary from one dataset to another, the summations in (1) contain balancing factors (to the immediate right of each Σ) in order to have comparable total magnitude (proportional to $|\mathcal{F}|$). The terms are weighted by tunable term weights λ, ζ and γ, which in all examples we set equal to 1.

5.1 Minimization by Lazy DDMS-TRWS

In contrast to previous work, we consider the *three-dimensional coupling* between all terms in our formulation, over all dynamic and static poses simultaneously, thereby obtaining a robust estimate that gracefully fills in missing or unreliable information. This presents two major challenges. First, the partial matches and loops in the performance flow graph preclude the use of straightforward mesh propagation schemes used in previous works. Such propagation would produce only partial solutions for many poses. Second (as a result of the first) we lack a complete initial estimate for traditional optimization schemes such as Levenberg-Marquadt.

To address these challenges, we employ an iterative scheme that admits partial intermediate solutions, with pseudocode in Algorithm 1. As some of the terms in (1) are data-dependent, we adapt the outer loop of Data Driven Mean-Shift Belief Propagation (DDMSBP) [Park et al. 2010], which models the objective function in each iteration as an increasingly-tight Gaussian (or quadratic) approximation of the true function. Within each DDMS loop, we use Gaussian Tree-Reweighted Sequential message passing (TRW-S) [Kolmogorov 2006], adapted to allow the terms in the model to be constructed lazily as the solution progresses over the variables. Hence we call our scheme *Lazy DDMS-TRWS*. We define the ordering of the variables to be pose-major (i.e. visiting all the vertices of one pose, then all the vertices of the next pose, etc.), with static poses followed by dynamic poses in temporal order. We decompose the Gaussian belief as a product of 3D Gaussians over vertices and poses, which admits a pairwise decomposition of (1) as a sum of quadratics. We denote the current belief of a vertex i for pose p as $\bar{\mathbf{x}}_i^p$ with covariance $\mathbf{\Sigma}_i^p$ (stored as inverse covariance for convenience), omitting the i subscript to refer to all vertices collectively. We detail the modeling of the energy terms in sections 5.2 - 5.5, defining $\bar{\mathbf{y}}_i^p = \mathbf{R}_p \bar{\mathbf{x}}_i^p + \mathbf{t}_p$ as shorthand for world space vertex position estimates. We iterate the DDMS loop 6 times, and iterate TRW-S until 95% of the vertices converge to within 0.01mm.

Algorithm 1 Lazy DDMS-TRWS for (1)

$\forall_{p,i} : (\mathbf{\Sigma}_i^p)^{-1} \leftarrow \mathbf{0}$.
for DDMS outer iterations **do**
  // Reset the model:
  $\forall_{p,q} : \mathbf{E}_{\text{corr}}^{pq}, \mathbf{E}_{\text{shape}}^p, \mathbf{E}_{\text{wrap}}^p \leftarrow$ undefined (effectively 0).
  for TRW-S inner iterations **do**
    // Major TRW-S loop over poses:
    for each $p \in \mathcal{D} \cup \mathcal{S}$ in order of increasing $o(p)$ **do**
      // Update model where possible:
      for each $q | (p,q) \in \mathcal{F}$ **do**
        if $(\mathbf{\Sigma}^p)^{-1} \neq \mathbf{0}$ and $\mathbf{E}_{\text{corr}}^{pq}$ undefined **then**
          $\mathbf{E}_{\text{corr}}^{pq} \leftarrow$ model fit using (2) in section 5.2.
        if $(\mathbf{\Sigma}^q)^{-1} \neq \mathbf{0}$ and $\mathbf{E}_{\text{corr}}^{qp}$ undefined **then**
          $\mathbf{E}_{\text{corr}}^{qp} \leftarrow$ model fit using (2) in section 5.2.
      if $(\mathbf{\Sigma}^p)^{-1} \neq \mathbf{0}$ and $\mathbf{E}_{\text{wrap}}^p$ undefined **then**
        $\mathbf{E}_{\text{wrap}}^p \leftarrow$ model fit using (8) in section 5.4.
      if $\exists_{(p,q)\in\mathcal{F}} | (\mathbf{\Sigma}^q)^{-1} \neq \mathbf{0}$ and $\mathbf{E}_{\text{shape}}^p$ undefined **then**
        $\mathbf{E}_{\text{shape}}^p \leftarrow$ model fit using (5) in section 5.3.
      // Minor TRW-S loop over vertices:
      Pass messages based on (1) to update $\bar{\mathbf{x}}^p$, $(\mathbf{\Sigma}^p)^{-1}$.
      Update $\mathbf{R}_p, \mathbf{t}_p$ as in section 5.6.
    // Reverse TRW-S ordering:
    $o(s) \leftarrow \|\mathcal{D} \cup \mathcal{S}\| + 1 - o(s)$.

5.2 Modeling the Correspondence Term

The correspondence term in (1) penalizes disagreement between optical flow vectors and projected vertex locations. Suppose we have a 2D optical flow correspondence field between poses p and q in (roughly) the same view c. We may establish a 3D relationship between $\mathbf{x}_i^p$ and $\mathbf{x}_i^q$ implied by the 2D correspondence field, which we model as a quadratic penalty function:

$$\mathbf{E}_{\text{corr}}^{pq} = \frac{1}{|\mathcal{C}|} \sum_{\substack{c\in\mathcal{C} \\ i\in\mathcal{V}}} \sum (\mathbf{x}_i^q - \mathbf{x}_i^p - \mathbf{f}_{\substack{pq \\ i}}^c)^{\mathsf{T}} \mathbf{F}_{\substack{pq \\ i}}^c (\mathbf{x}_i^q - \mathbf{x}_i^p - \mathbf{f}_{\substack{pq \\ i}}^c), \quad (2)$$

where $\mathcal{C}$ is the set of camera viewpoints, and $\mathbf{f}_{\substack{pq \\ i}}^c, \mathbf{F}_{\substack{pq \\ i}}^c$ are respectively the mean and precision matrix of the penalty, which we estimate from the current estimated positions as follows. We first project $\bar{\mathbf{y}}_i^p$ into the image plane of view c of pose p. We then warp the 2D image position from view c of pose p to view c of pose q

using the correspondence field. The warped 2D position defines a world-space view ray that the same vertex i ought to lie on in pose q. We transform this ray back into common head coordinates (via $-\mathbf{t}_q$, $\mathbf{R}_q^\mathsf{T}$) and penalize the squared distance from $\mathbf{x}_i^q$ to this ray. Letting $\mathbf{r}_{\substack{pq\\i}}^c$ represent the direction of this ray, this yields:

$$\mathbf{f}_{\substack{pq\\i}}^c = (\mathbf{I} - \mathbf{r}_{\substack{pq\\i}}^c \mathbf{r}_{\substack{pq\\i}}^{c\,\mathsf{T}})(\mathbf{R}_q^\mathsf{T}(\mathbf{c}_q^c - \mathbf{t}_q) - \bar{\mathbf{x}}_i^p), \tag{3}$$

where $\mathbf{c}_q^c$ is the nodal point of view c of pose q, and $\mathbf{r}_{\substack{pq\\i}}^c = \mathbf{R}_q^\mathsf{T}\mathbf{d}_{\substack{pq\\i}}^c$ with $\mathbf{d}_{\substack{pq\\i}}^c$ the world-space direction of the ray in view c of pose q through the 2D image plane point $f_{pq}^c[\mathrm{P}_p^c(\bar{\mathbf{y}}_i^p)]$ (where square brackets represent bilinearly interpolated sampling of a field or image), f_{pq}^c the optical flow field transforming an image-space point from view c of pose p to the corresponding point in view c of pose q, and $\mathrm{P}_p^c(\mathbf{x})$ the projection of a point $\mathbf{x}$ into the image plane of view c of pose p (which may differ somewhat from pose to pose). If we were to use the squared-distance-to-ray penalty directly, $\mathbf{F}_{\substack{pq\\i}}^c$ would be $\mathbf{I} - \mathbf{r}_{\substack{pq\\i}}^c \mathbf{r}_{\substack{pq\\i}}^{c\,\mathsf{T}}$, which is singular. To prevent the problem from being ill-conditioned and also to enable the use of *monocular* performance data, we add a small regularization term to produce a non-singular penalty, and weight the penalty by the confidence of the optical flow estimate. We also assume the optical flow field is locally smooth, so a *large* covariance $\mathbf{\Sigma}_i^p$ inversely influences the precision of the model, whereas a *small* covariance $\mathbf{\Sigma}_i^p$ does not, and weight the model accordingly. Intuitively, this weighting causes information to propagate from the ground term outward via the correspondences in early iterations, and blends correspondences from all sources in later iterations. All together, this yields:

$$\mathbf{F}_{\substack{pq\\i}}^c = \min(1, \det(\mathbf{\Sigma}_i^p)^{-\frac{1}{3}}) v_{\substack{pq\\i}}^c \tau_{pq}^c [\mathrm{P}_p^c(\bar{\mathbf{y}}_i^p)](\mathbf{I} - \mathbf{r}_{\substack{pq\\i}}^c \mathbf{r}_{\substack{pq\\i}}^{c\,\mathsf{T}} + \epsilon\mathbf{I}), \tag{4}$$

where $v_{\substack{pq\\i}}^c$ is a soft visibility factor (obtained by blurring a binary vertex visibility map and modulated by the cosine of the angle between surface normal and view direction), τ_{pq}^c is the confidence field associated with the correspondence field f_{pq}^c, and ϵ is a small regularization constant. We use $\det(\mathbf{\Sigma})^{-1/3}$ as a scalar form of precision for 3D Gaussians.

5.3 Modeling the Differential Shape Term

The shape term in (1) constrains the differential shape of each pose to a spatially varying convex combination of the differential shapes of the neighboring poses in the performance flow graph:

$$\mathbf{E}_{\mathrm{shape}}^p = \sum_{(i,j)\in\mathcal{E}} \left\| \mathbf{x}_j^p - \mathbf{x}_i^p - \mathbf{l}_{ij}^p \right\|^2, \tag{5}$$

$$\mathbf{l}_{ij}^p = \frac{\epsilon(\mathbf{g}_j - \mathbf{g}_i) + \sum_{q|(p,q)\in\mathcal{F}} \mathrm{w}_{ij}^{pq}(\bar{\mathbf{x}}_j^q - \bar{\mathbf{x}}_i^q)}{\epsilon + \sum_{q|(p,q)\in\mathcal{F}} \mathrm{w}_{ij}^{pq}}, \tag{6}$$

$$\mathrm{w}_{ij}^{pq} = \frac{w_i^{pq} w_j^{pq}}{w_i^{pq} + w_j^{pq}}, \tag{7}$$

where $\mathcal{E}$ is the set of edges in the geometry mesh, $w_i^{pq} = \det(\frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \mathbf{F}_{\substack{pq\\i}}^c + \mathbf{F}_{\substack{qp\\i}}^c)^{1/3}$ (which is intuitively the strength of the relationship between poses p and q due to the correspondence term), $\mathbf{g}$ denotes the artist mesh vertex positions, and ϵ is a small regularization constant. The weights w_i^{pq} additionally enable trivial synthesis of high-resolution reflectance maps for each dynamic frame of the performance by blending the static pose data.

5.4 Modeling the Shrink Wrap Term

The shrink wrap term in (1) penalizes the distance between static pose vertices and the raw scan geometry of the same pose. We model this as a regularized distance-to-plane penalty:

$$\mathbf{E}_{\mathrm{wrap}}^p = \sum_{i\in\mathcal{V}} (\mathbf{x}_i^p - \mathbf{d}_i^p)^\mathsf{T} \mathrm{g}_i^p (\mathbf{n}_i^p \mathbf{n}_i^{p\mathsf{T}} + \epsilon\mathbf{I})(\mathbf{x}_i^p - \mathbf{d}_i^p), \tag{8}$$

where $(\mathbf{n}_i^p, \mathbf{d}_i^p)$ are the normal and centroid of a plane fitted to the surface of the static scan for pose p close to the current estimate $\bar{\mathbf{x}}_i^p$ in common head coordinates, and g_i^p is the confidence of the planar fit. We obtain the planar fit inexpensively by projecting $\bar{\mathbf{y}}_i^p$ into each camera view, and sampling the raw scan surface via a set of precomputed rasterized views of the scan. (Alternatively, a 3D search could be employed to obtain the samples.) Each surface sample (excluding samples that are occluded or outside the rasterized scan) provides a plane equation based on the scan geometry and surface normal. We let $\mathbf{n}_i^p$ and $\mathbf{d}_i^p$ be the weighted average values of the plane equations over all surface samples:

$$\mathbf{n}_i^p = \sum_{c\in\mathcal{C}} \omega_{\substack{p\\i}}^c \mathbf{R}_p^\mathsf{T} \hat{\mathbf{n}}_p^c [\mathrm{P}_p^c(\bar{\mathbf{y}}_i^p)] \text{ (normalized)}, \tag{9}$$

$$\mathbf{d}_i^p = \Big(\sum_{c\in\mathcal{C}} \omega_{\substack{p\\i}}^c\Big)^{-1} \sum_{c\in\mathcal{C}} \omega_{\substack{p\\i}}^c \mathbf{R}_p^\mathsf{T} (\hat{\mathbf{d}}_p^c [\mathrm{P}_p^c(\bar{\mathbf{y}}_i^p)] - \mathbf{t}_p), \tag{10}$$

$$\mathrm{g}_i^p = \min(1, \det(\mathbf{\Sigma}_i^p)^{-\frac{1}{3}}) \sum_{c\in\mathcal{C}} \omega_{\substack{p\\i}}^c, \tag{11}$$

where $(\hat{\mathbf{n}}_p^c, \hat{\mathbf{d}}_p^c)$ are the world-space surface normal and position images of the rasterized scans, and $\omega_{\substack{p\\i}}^c = 0$ if the vertex is occluded in view c or lands outside of the rasterized scan, otherwise $\omega_{\substack{p\\i}}^c = v_{\substack{p\\i}}^c \exp(-\|\hat{\mathbf{d}}_p^c[\mathrm{P}_p^c(\bar{\mathbf{y}}_i^p)] - \bar{\mathbf{y}}_i^p\|^2)$.

5.5 Modeling the Ground Term

The ground term in (1) penalizes the distance between vertex positions in the ground (neutral) pose and the artist mesh geometry:

$$\mathbf{E}_{\mathrm{ground}} = \sum_{i\in\mathcal{V}} \left\| \mathbf{x}_i^g - \mathbf{R}_g^\mathsf{T} \mathbf{g}_i \right\|^2, \tag{12}$$

where $\mathbf{g}_i$ is the position of the vertex in the artist mesh. This term is simpler than the shrink-wrap term since the pose vertices are in one-to-one correspondence with the artist mesh vertices.

5.6 Updating the Rigid Transforms

We initialize our optimization scheme with all $(\mathbf{\Sigma}_i^p)^{-1} = 0$ (and hence all $\bar{\mathbf{x}}_i^p$ *moot*), fully relying on the lazy DDMS-TRWS scheme to propagate progressively tighter estimates of the vertex positions $\mathbf{x}_i^p$ throughout the solution. Unfortunately, in our formulation the rigid transforms $(\mathbf{R}_p, \mathbf{t}_p)$ enjoy no such treatment as they always occur together with $\mathbf{x}_i^p$ and would produce non-quadratic terms if they were included in the message passing domain. Therefore we must initialize the rigid transforms to some rough initial guess, and update them after each iteration. The neutral pose is an exception, where the transform is *specified by the user* (by rigidly posing the artist mesh to their whim) and hence not updated. In all our examples, the initial guess for all poses is simply the same as the user-specified rigid transform of the neutral pose. We update $(\mathbf{R}_p, \mathbf{t}_p)$ using a simple scheme that aligns the neutral artist mesh to the current result. Using singular value decomposition, we compute the closest rigid transform minimizing $\sum_{i\in\mathcal{V}} r_i \left\| \mathbf{R}_p \mathbf{g}_i + \mathbf{t}_p - \bar{\mathbf{R}}_p \bar{\mathbf{x}}_i^p - \bar{\mathbf{t}}_p \right\|^2$, where r_i is a rigidity weight value (high weight around the eye sockets and temples, low weight elsewhere), $\mathbf{g}_i$ denotes the artist mesh vertex positions, and $(\bar{\mathbf{R}}_p, \bar{\mathbf{t}}_p)$ is the previous transform estimate.

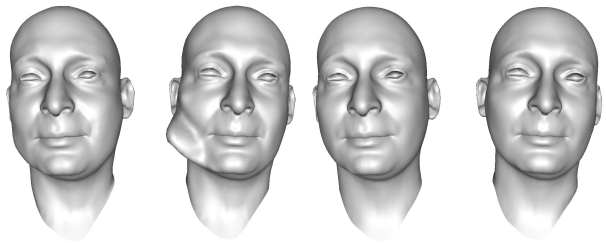
5.7 Accelerating the Solution Using Keyframes

Minimizing the energy in (1) over the entire sequence requires multiple iterations of the TRW-S message passing algorithm, and multiple iterations of the DDMS outer loop. We note that the performance flow graph assigns static-to-dynamic flows to only a sparse subset of performance frames, which we call *keyframes*. Correspondences among the spans of frames in between keyframes are reliably represented using concatenation of temporal flows. Therefore to reduce computation time we first miminize the energy at *only the keyframes and static poses*, using concatenated temporal flows in between keyframes. Each iteration of this reduced problem is far cheaper than the full problem, so we may obtain a satisfactory solution of the performance keyframes and static poses more quickly. Next, we keep the static poses and keyframe poses *fixed*, and solve the spans of in-between frames, omitting the shrink-wrap and grounding terms as they affect only the static poses. This subsequent minimization requires only a few iterations to reach a satisfactory result, and each span of in-between frames may be solved *independently* (running on multiple computers, for example).

6 Handling Arbitrary Illumination and Motion

Up to now, we have assumed that lighting and overall head motion in the static scans closely matches that in the dynamic frames. For performances in uncontrolled environments, the subject may move or rotate their head to face different cameras, and lighting may be arbitrary. We handle such complex cases by taking advantage of the 3D geometry and relightable reflectance maps in the static scan data. For every 5$^{\text{th}}$ performance frame, we compute a relighted rendering of each static scan with roughly similar rigid head motion and lighting environment as the dynamic performance. These renderings are used as the static expression imagery in our pipeline. The rigid head motion estimate does not need to be exact as the optical flow computation is robust to a moderate degree of misalignment. In our results, we (roughly) rigidly posed the head by hand, though automated techniques could be employed [Zhu and Ramanan 2012]. We also assume that a HDR light probe measurement [Debevec 1998] exists for the new lighting environment, however, lighting could be estimated from the subject's face [Valgaerts et al. 2012] or eyes [Nishino and Nayar 2004].

The complex backgrounds in real-world uncontrolled environments pose a problem, as optical flow vectors computed on background pixels close to the silhouette of the face may confuse the correspondence term if the current estimate of the facial geometry slightly overlaps the background. This results in parts of the face "sticking" to the background as the subject's face turns from side to side (Fig. 6). To combat this, we weight the correspondence confidence field by a simple soft segmentation of head vs. background. Since head motion is largely rigid, we fit a 2D affine transform to the optical flow vectors in the region of the current head estimate. Then, we weight optical flow vectors by how well they agree with the fitted transform. We also assign high weight to the region deep inside the current head estimate using a simple image-space erosion algorithm, to prevent large jaw motions from being discarded. The resulting soft segmentation effectively cuts the head out of the background whenever the head is moving, thus preventing the optical flow vectors of the background from polluting the edges of the face. When the head is not moving against the background the segmentation is poor, but in this case the optical flow vectors of the face and background agree and pollution is not damaging.



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

Figure 6: *(a, b) Two frames of a reconstructed performance in front of a cluttered background, where the subject turns his head over the course of ten frames. The silhouette of the jaw "sticks" to the background because the optical flow vectors close to the jaw are stationary. (c, d) A simple segmentation of the optical flow field to exclude the background resolves the issue.*

7 Results

We ran our technique on several performances from three different subjects. Each subject had 30 static facial geometry scans captured before the performance sessions, though the performance flow graph construction often employs only a fraction of the scans. An artist produced a single face mesh for each subject based on their neutral static facial scan.

7.1 Performances Following Static Scan Sessions

We captured performances of three subjects directly following their static scan sessions. The performances were recorded from six camera views in front of the subject with a baseline of approximately 15 degrees. Our method produced the performance animation results shown in Fig. 19 without any further user input.

7.2 Performances in Other Locations

We captured a performance of a subject using four consumer HD video cameras in an office environment. An animator rigidly posed a head model roughly aligned to every 5$^{\text{th}}$ frame of the performance, to produce the static images for our performance flow graph. Importantly, this rigid head motion does not need to be very accurate for our method to operate, and we intend that an automated technique could be employed. A selection of video frames from one of the views is shown in Fig. 7, along with renderings of the results of our method. Despite the noisy quality of the videos and the smaller size of the head in the frame, our method is able to capture stable facial motion including lip synching and brow wrinkles.

7.3 High-Resolution Detail Transfer

After tracking a performance, we transfer the high-resolution reflectance maps from the static scans onto the performance result. As all results are registered to the same UV parameterization by our method, the transfer is a simple weighted blend using the cross-correlation-based confidence weights w_i^{pq} of each vertex, interpolated bilinearly between vertices. We also compute values for w_i^{pq} for any dynamic-to-static edge pq that was not present in the performance flow graph, to produce weights for every frame of the performance. This yields detailed reflectance maps for every performance frame, suitable for realistic rendering and relighting. In addition to transferring reflectance, we also transfer geometric details in the form of a displacement map, allowing the performance tracking to operate on a medium-resolution mesh instead of the full scan resolution. Fig. 8 compares transferring geometric details

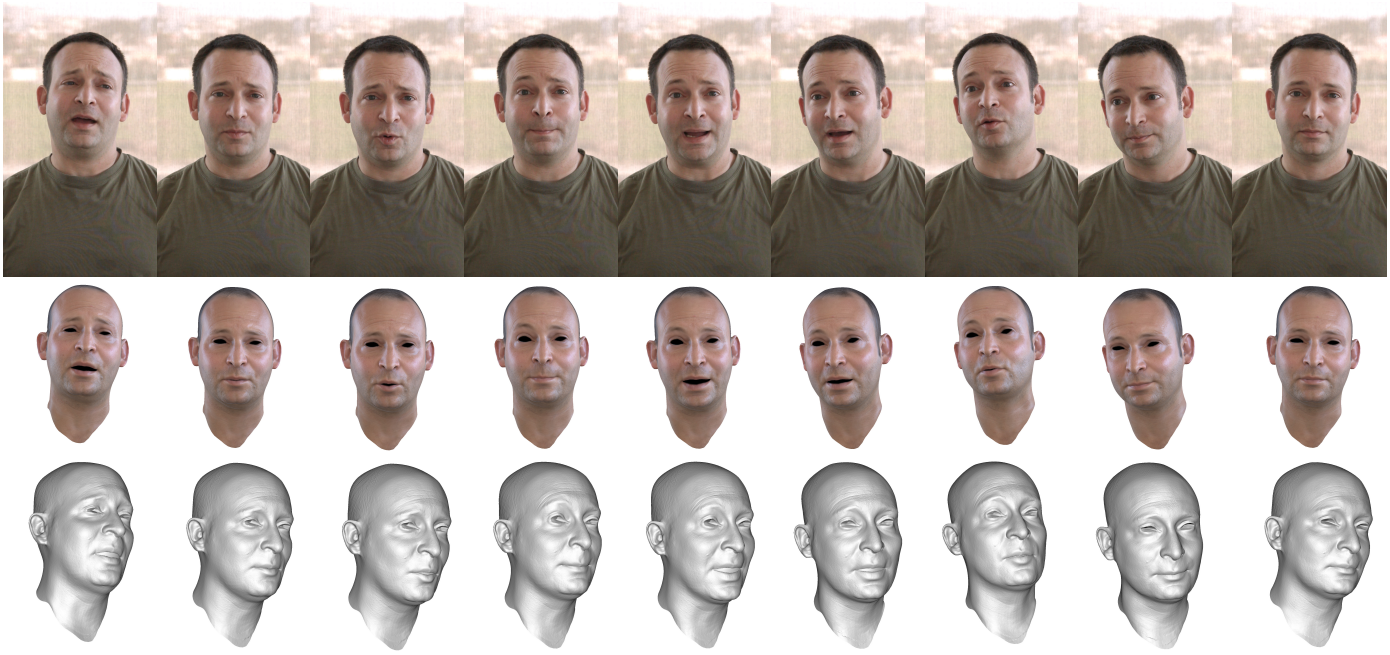
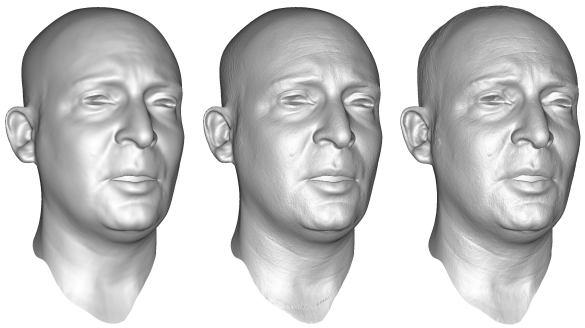


Figure 7: *A performance captured in an office environment with uncontrolled illumination, using four HD consumer video cameras and seven static expression scans. Top row: a selection of frames from one of the camera views. Middle row: geometry tracked using the proposed method, with reflectance maps automatically assembled from static scan data, shaded using a high-dynamic-range light probe. The reflectance of the top and back of the head were supplemented with artist-generated static maps. The eyes and inner mouth are rendered as black as our method does not track these features. Bottom row: gray-shaded geometry for the same frames, from a novel viewpoint. Our method produces stable animation even with somewhat noisy video footage and significant head motion. Dynamic skin details such as brow wrinkles are transferred from the static scans in a manner faithful to the video footage.*



(a) (b) (c)

Figure 8: *High-resolution details may be transferred to a medium-resolution tracked model to save computation time. (a) medium-resolution tracked geometry using six views. (b) medium-resolution geometry with details automatically transferred from the high-resolution static scans. (c) high-resolution tracked geometry. The transferred details in (b) capture most of the dynamic facial details seen in (c) at a reduced computational cost.*

from the static scans onto a medium-resolution reconstruction to directly tracking a high-resolution mesh. As the high-resolution solve is more expensive, we first perform the medium-resolution solve and use it to prime the DDMS-TRWS belief in the high-resolution solve, making convergence more rapid. In all other results, we show medium-resolution tracking with detail transfer, as the results are satisfactory and far cheaper to compute.

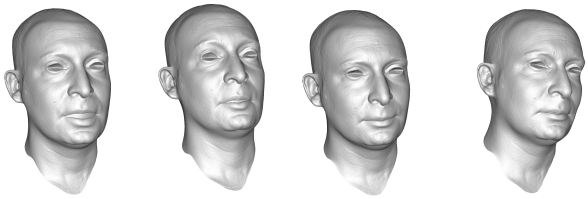


Figure 9: *Results using only a single camera view, showing the last four frames from Fig. 7. Even under uncontrolled illumination and significant head motion, tracking is possible from a single view, at somewhat reduced fidelity.*

7.4 Monocular vs. Binocular vs. Multi-View

Our method operates on any number of camera views, producing a result from even a single view. Fig. 9 shows results from a single view for the same uncontrolled-illumination sequence as Fig. 7. Fig. 10 shows the incremental improvement in facial detail for a controlled-illumination sequence using one, two, and six views. Our method is applicable to a wide variety of camera and lighting setups, with graceful degradation as less information is available.

7.5 Influence of Each Energy Term

The core operation of our method is to propagate a known facial pose (the artist mesh) to a set of unknown poses (the dynamic frames and other static scans) via the ground term and correspondence terms in our energy formulation. The differential shape term and shrink wrap term serve to regularize the shape of the solution. We next explore the influence of these terms on the solution.

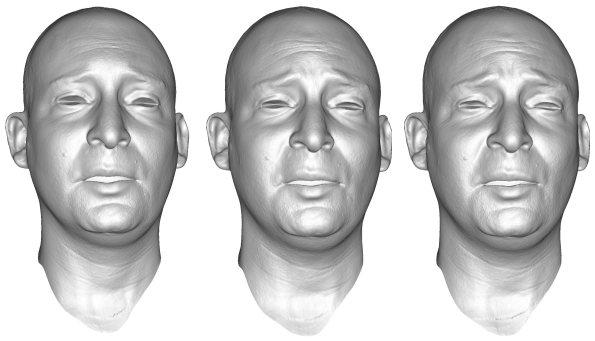


Figure 10: *Example dynamic performance frame reconstructed from (a) one view, (b) two views and (c) six views. Our method gracefully degrades as less information is available.*

Figure 11: *The artist mesh is non-rigidly registered to each of the other static expression scans as a byproduct of our method. The registered artist mesh is shown for a selection of scans from two different subjects. Note the variety of mouth shapes, all of which are well-registered by our method without any user input.*

Correspondence Term The correspondence term produces a consistent parameterization of the geometry suitable for texturing and other editing tasks. As our method computes a coupled solution of performance frames using static poses to bridge larger temporal gaps, the artist mesh is non-rigidly registered to each of the static scans as a byproduct of the optimization. (See Fig. 11 for examples.) Note especially that our method automatically produces a complete head for each expression, despite only having static facial scan geometry for the frontal face surface. As shown in Fig. 12, this consistency is maintained even when the solution is obtained from a different performance. Fig. 13 illustrates that the use of multiple static expression scans in the performance flow graph produces a more expressive performance, with more accentuated facial expression features, as there are more successful optical flow regions in the face throughout the performance.

Differential Shape Term In our formulation, the differential shape of a performance frame or pose is tied to a blend of its neighbors on the performance flow graph. This allows details from multiple static poses to propagate to related poses. Even when only one static pose is used (i.e. neutral), allowing temporal neighbors to influence the differential shape provides temporal smoothing without overly restricting the shape of each frame. Fig. 13 (c, d) illustrates the loss of detail when temporal neighbors are excluded from the differential shape term (compare to a, b).

Figure 12: *Top row: neutral mesh with checker visualization of texture coordinates, followed by three non-rigid registrations to other facial scans as a byproduct of tracking a speaking performance. Bottom row: the same, except the performance used was a series of facial expressions with no speaking. The non-rigid registration obtained from the performance-graph-based tracking is both consistent across expressions and across performances. Note, e.g. the consistent locations of the checkers around the contours of the lips.*

Shrink Wrap Term The shrink wrap term conforms the static poses to the raw geometry scans (Fig. 14). Without this term, subtle details in the static scans cannot be propagated to the performance result, and the recovered static poses have less fidelity to the scans.

7.6 Comparison to Previous Work

We ran our method on the data from [Beeler et al. 2011], using their recovered geometry from the first frame (frame 48) as the "artist" mesh in our method. For expression scans, we used the geometry from frames 285 (frown) and 333 (brow raise). As our method makes use of the expression scans only via image-space operations on camera footage or rasterized geometry, any point order information present in the scans is entirely ignored. Therefore in this test, it is as if the static scans were produced individually by the method of [Beeler et al. 2010]. We constructed a simple UV projection on the artist mesh for texture visualization purposes, and projected the video frames onto each frame's geometry to produce a per-frame UV texture map. To measure the quality of texture alignment over the entire sequence, we computed the temporal variance of each pixel in the texture map (shown in Fig.15 (a, b)), using contrast normalization to disregard low-frequency shading variation. The proposed method produces substantially lower temporal texture variance, indicating a more consistent alignment throughout the sequence, especially around the mouth. Examining the geometry in Fig.15 (c-f), the proposed method has generally comparable quality as the previous work, with the mouth-closed shape recovered more faithfully (which is consistent with the variance analy-

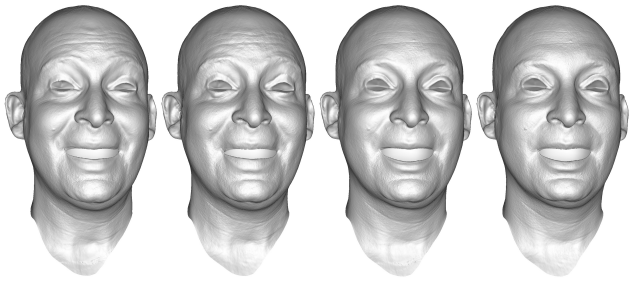


Figure 13: *Using multiple static expressions in the performance flow graph produces more detail than using just a neutral static expression. Multiple static expressions are included in the performance flow graph in (a, c), whereas only the neutral expression is included in (b, d). By including temporal neighbors and static scans in determining the differential shape, details from the various static scans can be propagated throughout the performance. Differential shape is determined by the static expression(s) and temporal neighbors in (a, b), whereas temporal neighbors are excluded from the differential shape term in (c, d). Note the progressive loss of detail in e.g. the brow region from (a) to (d).*

sis). We also compared to [Klaudiny and Hilton 2012] in a similar manner, using frame 0 as the artist mesh, and frames 25, 40, 70, 110, 155, 190, 225, 255 and 280 as static expressions. Again, no point order information is used. Fig. 16 again shows an overall lower temporal texture variance from the proposed method.

7.7 Performance Timings

We report performance timings in Fig. 17 for various sequences, running on a 16-core 2.4 GHz Xeon E5620 workstation (some operations are multithreaded across the cores). All tracked meshes have 65 thousand vertices, except Fig. 8(c) and Fig. 15 which have one million vertices. We report each stage of the process: "Graph" for the performance graph construction, "Flow" for the high-resolution optical flow calculations, "Key" for the performance tracking solve on key frames, and "Tween" for the performance tracking solve in between key frames. We mark stages that could be parallelized over multiple machines with an asterisk (*). High-resolution solves (Fig. 8(c) and Fig. 15) take longer than medium-resolution solves. Sequences with uncontrolled illumination (Fig. 7 and Fig. 9) take longer for the key frames to converge since the correspondence tying the solution to the static scans has lower confidence.

7.8 Discussion

Our method produces a consistent geometry animation on an artist-created neutral mesh. The animation is expressive and lifelike, and the subject is free to make natural head movements within a certain degree. Fig. 18 shows renderings from such a facial performance rendered using advanced skin and eye shading techniques as described in [Jimenez et al. 2012]. One notable shortcoming of our performance flow graph construction algorithm is the neglect of eye blinks. This results in a poor representation of the blinks in the final animation. Our method requires one artist-generated mesh per subject to obtain results that are immediately usable in production pipelines. Automatic generation of this mesh could be future work, or use existing techniques for non-rigid registration. Omitting this step would still produce a result, but would require additional cleanup around the edges as in e.g. [Beeler et al. 2011][Klaudiny and Hilton 2012].

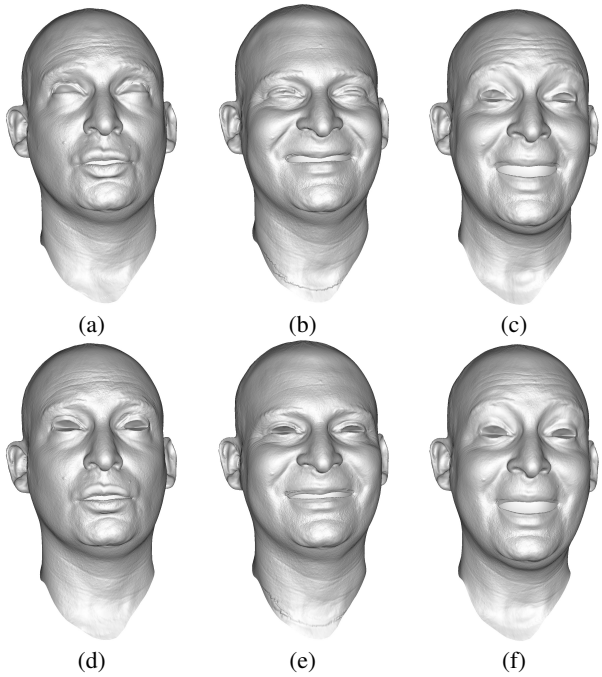


Figure 14: *The shrink wrap term conforms the artist mesh to the static scan geometry, and also improves the transfer of expressive details to the dynamic performance. The registered artist mesh is shown for two static poses in (a) and (b), and a dynamic pose that borrows brow detail from (a) and mouth detail from (b) is shown in (c). Without the shrink wrap term, the registration to the static poses suffers (d, e) and the detail transfer to the dynamic performance is less sucessful (f). Fine-scale details are still transferred via displacement maps, but medium-scale expressive details are lost.*

8 Future Work

One of the advantages of our technique is that it relates a dynamic performance back to facial shape scans using per-pixel weight maps. It would be desirable to further factor our results to create multiple localized blend shapes which are more semantically meaningful and artist friendly. Also, our algorithm does not explicitly track eye or mouth contours. Eye and mouth tracking could be further refined with additional constraints to capture eye blinks and more subtle mouth behavior such as "sticky lips" [Alexander et al. 2009]. Another useful direction would be to retarget performances from one subject to another. Given a set of static scans for both subjects, it should be possible to clone one subject's performance to the second subject as in [Seol et al. 2012]; providing more meaningful control over this transfer remains a subject for future research. Finally, as our framework is agnostic to the particular method employed for estimating 2D correspondences, we would like to try more recent optical flow algorithms such as the top performers on the Middlebury benchmark [Baker et al. 2011]. Usefully, the quality of our performance tracking can be improved any time that an improved optical flow library becomes available.

Acknowledgements

The authors thank the following people for their support and assistance: Ari Shapiro, Sin-Hwa Kang, Matt Trimmer, Koki Nagano, Xueming Yu, Jay Busch, Paul Graham, Kathleen Haase, Bill Swartout, Randall Hill and Randolph Hall. We thank the authors of [Beeler et al. 2010] and [Klaudiny et al. 2010] for graciously

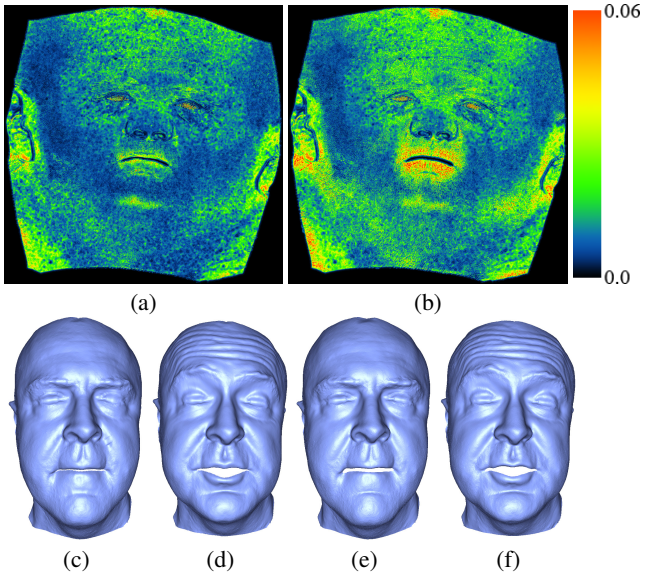


Figure 15: *Top row: Temporal variance of contrast-normalized texture (false color, where blue is lowest and red is highest), with (a) the proposed method and (b) the method of [Beeler et al. 2011]. The variance of the proposed method is substantially lower, indicating a more consistent texture alignment throughout the sequence. Bottom row: Geometry for frames 120 and 330 of the sequence, with (c, d) the proposed method and (e, f) the prior work.*

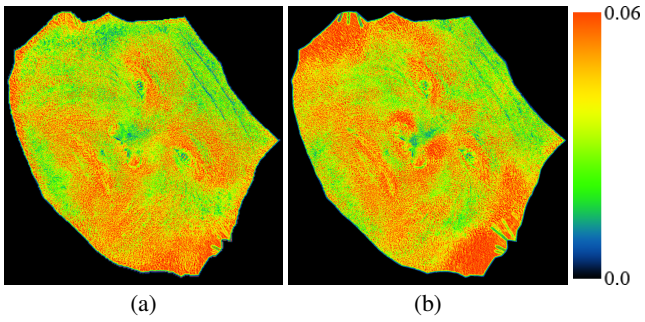


Figure 16: *Temporal variance of contrast-normalized texture (false color, where blue is lowest and red is highest), with (a) the proposed method and (b) the method of [Klaudiny et al. 2010]. As in Fig.15, the variance of the proposed method is generally lower.*

Sequence	Frames	Graph*	Flow*	Key	Tween*
Fig. 7	170	0.5 hr	8.0 hr	5.2 hr	1.2 hr
Fig. 8(b)	400	1.1 hr	24 hr	4.3 hr	4.3 hr
Fig. 8(c)	400	1.1 hr	24 hr	24 hr	26 hr
Fig. 9	170	0.5 hr	2.0 hr	3.6 hr	0.9 hr
Fig. 15	347	0.1 hr	15 hr	16 hr	17 hr
Fig. 16	300	0.2 hr	12 hr	3.0 hr	3.0 hr
Fig. 19 row 2	600	1.6 hr	36 hr	6.5 hr	7.0 hr
Fig. 19 row 4	305	0.8 hr	18 hr	3.3 hr	3.5 hr
Fig. 19 row 6	250	0.7 hr	15 hr	2.6 hr	2.8 hr

Figure 17: *Timings for complete processing of the sequences used in various figues, using a single workstation. A * indicates an operation that could trivially be run in parallel across many machines.*

providing the data for the comparisons in Figs. 15 and 16, respectively. We thank Jorge Jimenez, Etienne Danvoye, and Javier von der Pahlen at Activision R&D for the renderings in Fig. 18. This work was sponsored by the University of Southern California Office of the Provost and the U.S. Army Research, Development, and Engineering Command (RDECOM). The content of the information does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

References

ALEXANDER, O., ROGERS, M., LAMBETH, W., CHIANG, M., AND DEBEVEC, P. 2009. Creating a photoreal digital actor: The digital emily project. In *Visual Media Production, 2009. CVMP '09. Conference for*, 176–187.

BAKER, S., SCHARSTEIN, D., LEWIS, J. P., ROTH, S., BLACK, M. J., AND SZELISKI, R. 2011. A database and evaluation methodology for optical flow. *International Journal of Computer Vision 92*, 1 (Mar.), 1–31.

BEELER, T., BICKEL, B., BEARDSLEY, P., SUMNER, B., AND GROSS, M. 2010. High-quality single-shot capture of facial geometry. *ACM Trans. on Graphics (Proc. SIGGRAPH) 29*, 3, 40:1–40:9.

BEELER, T., HAHN, F., BRADLEY, D., BICKEL, B., BEARDSLEY, P., GOTSMAN, C., SUMNER, R. W., AND GROSS, M. 2011. High-quality passive facial performance capture using anchor frames. In *ACM SIGGRAPH 2011 papers*, ACM, New York, NY, USA, SIGGRAPH '11, 75:1–75:10.

BICKEL, B., LANG, M., BOTSCH, M., OTADUY, M. A., AND GROSS, M. 2008. Pose-space animation and transfer of facial details. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '08, 57–66.

BORSHUKOV, G., PIPONI, D., LARSEN, O., LEWIS, J. P., AND TEMPELAAR-LIETZ, C. 2003. Universal capture: image-based facial animation for "the matrix reloaded". In *SIGGRAPH*, ACM, A. P. Rockwood, Ed.

BRADLEY, D., HEIDRICH, W., POPA, T., AND SHEFFER, A. 2010. High resolution passive facial performance capture. In *ACM SIGGRAPH 2010 papers*, ACM, New York, NY, USA, SIGGRAPH '10, 41:1–41:10.

COOTES, T. F., EDWARDS, G. J., AND TAYLOR, C. J. 1998. Active appearance models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Springer, 484–498.

DEBEVEC, P. 1998. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '98, 189–198.

DECARLO, D., AND METAXAS, D. 1996. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, IEEE Computer Society, Washington, DC, USA, CVPR '96, 231–238.

EKMAN, P., AND FRIESEN, W. 1978. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto.

GHOSH, A., FYFFE, G., TUNWATTANAPONG, B., BUSCH, J., YU, X., AND DEBEVEC, P. 2011. Multiview face capture using

Figure 18: *Real-time renderings of tracked performances using advanced skin and eye shading [Jimenez et al. 2012].*

polarized spherical gradient illumination. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, ACM, New York, NY, USA, SA '11, 129:1–129:10.

GUENTER, B., GRIMM, C., WOOD, D., MALVAR, H., AND PIGHIN, F. 1998. Making faces. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '98, 55–66.

HAWKINS, T., WENGER, A., TCHOU, C., GARDNER, A., GÖRANSSON, F., AND DEBEVEC, P. 2004. Animatable facial reflectance fields. In *Rendering Techniques 2004: 15th Eurographics Workshop on Rendering*, 309–320.

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

JIMENEZ, J., JARABO, A., GUTIERREZ, D., DANVOYE, E., AND VON DER PAHLEN, J. 2012. Separable subsurface scattering and photorealistic eyes rendering. In *ACM SIGGRAPH 2012 Courses*, ACM, New York, NY, USA, SIGGRAPH 2012.

KLAUDINY, M., AND HILTON, A. 2012. High-detail 3d capture and non-sequential alignment of facial performance. In *3DIMPVT*.

KLAUDINY, M., HILTON, A., AND EDGE, J. 2010. High-detail 3d capture of facial performance. In *3DPVT*.

KOLMOGOROV, V. 2006. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell. 28*, 10, 1568–1583.

LI, H., ROIVAINEN, P., AND FORCHEIMER, R. 1993. 3-d motion estimation in model-based facial image coding. *IEEE Trans. Pattern Anal. Mach. Intell. 15*, 6 (June), 545–555.

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

NISHINO, K., AND NAYAR, S. K. 2004. Eyes for relighting. *ACM Trans. Graph. 23*, 3, 704–711.

PARK, M., KASHYAP, S., COLLINS, R., AND LIU, Y. 2010. Data driven mean-shift belief propagation for non-gaussian mrfs. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 3547 –3554.

POPA, T., SOUTH-DICKINSON, I., BRADLEY, D., SHEFFER, A., AND HEIDRICH, W. 2010. Globally consistent space-time reconstruction. *Computer Graphics Forum (Proc. SGP)*.

SEOL, Y., LEWIS, J., SEO, J., CHOI, B., ANJYO, K., AND NOH, J. 2012. Spacetime expression cloning for blendshapes. *ACM Trans. Graph. 31*, 2 (Apr.), 14:1–14:12.

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

WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Realtime performance-based facial animation. In *ACM SIGGRAPH 2011 papers*, ACM, New York, NY, USA, SIGGRAPH '11, 77:1–77:10.

WERLBERGER, M. 2012. *Convex Approaches for High Performance Video Processing*. PhD thesis, Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. 2004. Spacetime faces: high resolution capture for modeling and animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, 548–558.

ZHU, X., AND RAMANAN, D. 2012. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2879–2886.

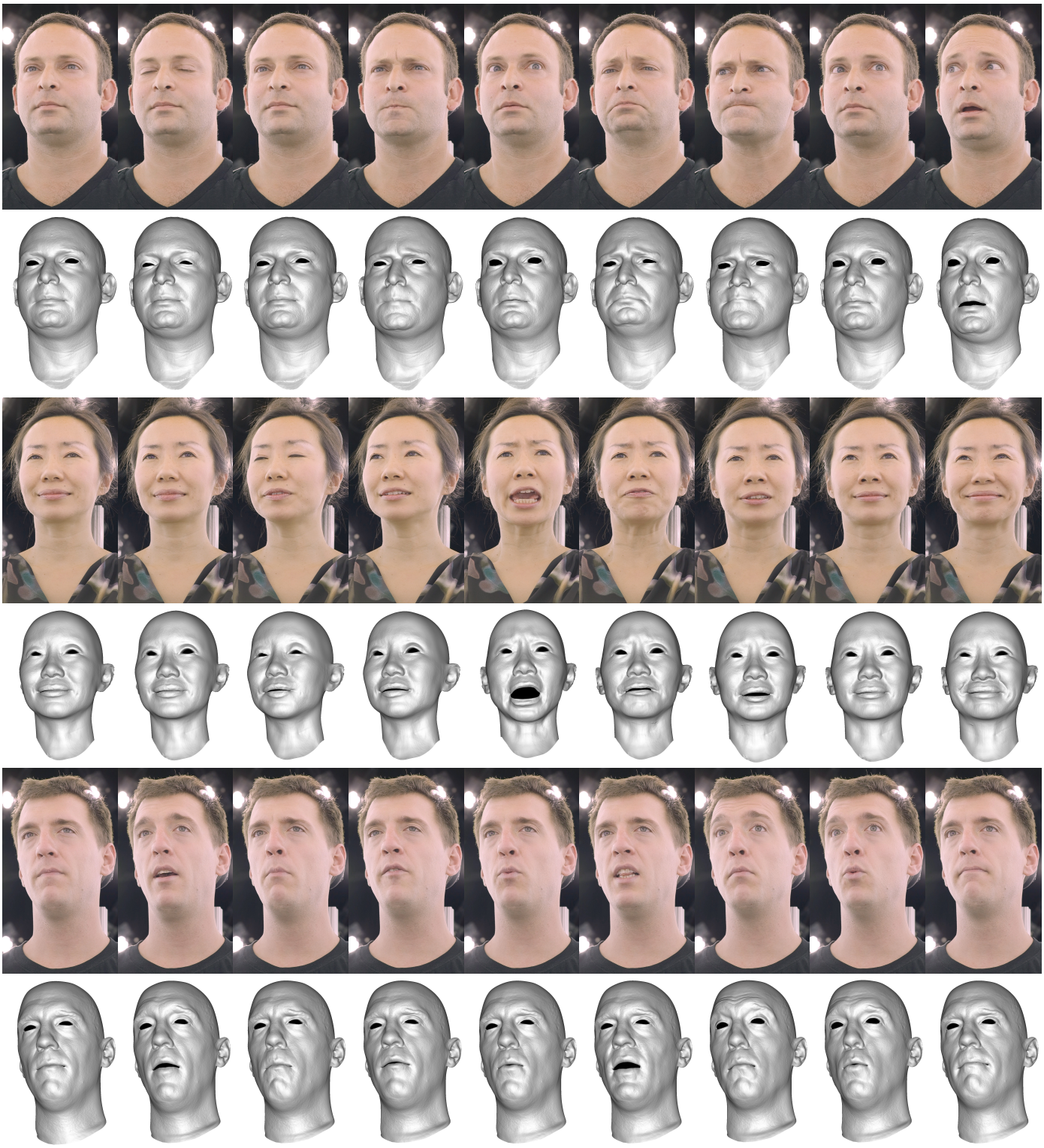


Figure 19: *Three tracked performances with different subjects, using six camera views and six to eight static expression scans per subject. Shown are alternating rows of selected frames from the performance video, and gray-shaded tracked geometry for the same frames. Our method produces a consistent geometry animation on an artist-created neutral mesh. The animation is expressive and lifelike, and the subject is free to make natural head movements within a certain degree.*