Fully Automatic Generation of Anatomical Face Simulation Models

Matthew Cong* Stanford University Industrial Light + Magic
Michael Bao* Stanford University
Jane L. E* Stanford University
Kiran S. Bhat† Industrial Light + Magic
Ronald Fedkiw* Stanford University Industrial Light + Magic

Figure 1: *Twelve anatomical face simulation models automatically generated using our procedure. We activated the levator palpabrae muscles to open the eyes and the zygomatic major and orbicularis oculi muscles to produce the smiles. These meshes were selected to represent a variety of features and characteristics. Asimov, Demon, Goblin, Artec Human, Kiran, Lincoln, Matthew, Ogre Head, Old Man, Orc, Spielberg and Yoda.*

Abstract

We present a fast, fully automatic morphing algorithm for creating simulatable flesh and muscle models for human and humanoid faces. Current techniques for creating such models require a significant amount of time and effort, making them infeasible or impractical. In fact, the vast majority of research papers use only a floating mask with no inner lips, teeth, tongue, eyelids, eyes, head, ears, etc.—and even those that build the full visual model would typically still lack the cranium, jaw, muscles, and other internal anatomy. Our method requires only the target surface mesh as input and can create a variety of models in only a few hours with no user interaction. We start with a symmetric, high resolution, anatomically accurate template model that includes auxiliary information such as feature points and curves. Then given a target mesh, we automatically orient it to the template, detect feature points, and use these to bootstrap the detection of corresponding feature curves. These curve correspondences are used to deform the surface mesh of the template model to match the target mesh. Then, the calculated displacements of the template surface mesh are used to drive a three-dimensional morph of the full template model including all interior anatomy. The resulting target model can be simulated to generate a large range of expressions that are consistent across characters using the same muscle activations. Full automation of this entire process makes it readily available to a wide range of users.

CR Categories: I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: facial animation, model creation, muscles

1 Introduction

High-quality facial modeling and animation is in high demand and remains a challenging problem. This is especially true in the entertainment industry; [Bhat et al. 2013; Fyffe et al. 2015] illustrate some of the challenges faced in recent blockbuster films and research efforts, respectively. Even with advances in technology, high-quality facial animation remains inaccessible to the vast majority of computer graphics practitioners, including both artists and researchers, partly due to the complexity involved in authoring the facial rig.

Two widely adopted techniques for high-quality facial rigging are blendshapes (see [Lewis et al. 2014] for an overview) and skinning-based rigs [Autodesk 2011; 3Lateral 2015]. Typical production quality rigs utilize several hundred input shapes corresponding to different expressions which can either be sculpted by an artist or acquired by scanning an actor's face during a range-of-motion facial exercise. These facial rigs are usually animated via keyframe animation or with performance capture [Beeler et al. 2011; Shi et al. 2014]. To further improve the results, researchers have proposed adding procedural correctives in conjunction with the blendshapes [Bhat et al. 2013; Garrido et al. 2013] as well as using an estimated skull to stabilize scanned facial expressions [Beeler and Bradley 2014]. These techniques have produced compelling facial anima-

*e-mail: {mdcong,mikebao,ejane,rfedkiw}@stanford.edu
†e-mail: kbhat@ilm.com

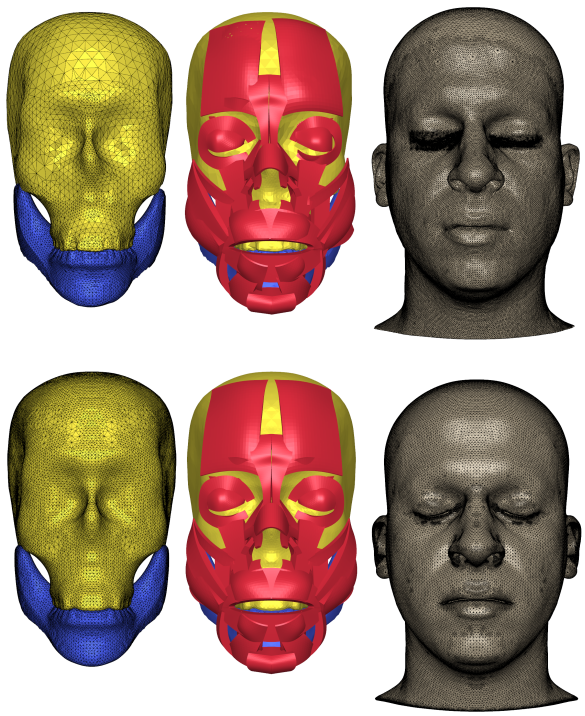


Figure 2: *We utilize a high-fidelity template consisting of a cranium (yellow), jaw (blue), muscles (red), and tetrahedralized flesh mesh (right) as a prior in order to aid in the creation of our facial models. To remove bias, we symmetrize the cranium, jaw, muscles, and flesh mesh in the original asymmetric model (top) to obtain a symmetric cranium, jaw, muscles, and flesh mesh (bottom).*

tion, but require a significant amount of setup, processing, and artistic adjustment.

Alternatively, one could utilize a physically based animation approach using an anatomically accurate model of facial structures including soft tissue, musculature, and the underlying skeletal structure (see e.g. [Sifakis et al. 2005; Sifakis et al. 2006]). The benefits of this approach include the straightforward handling of contact and collision as well as the ability to actuate individual muscles for animation. However, building an anatomically accurate model for physical simulation requires MRI or CT scans of the actor, which are very hard to acquire in practice. Moreover, such data doesn't exist for creature models. Other factors specific to physically based simulation (and not required in blendshape approaches) such as mesh quality, self-intersections, etc. also need to be considered.

In this paper, we propose a framework for automatically building anatomically accurate facial rigs which is based on a three-dimensional Poisson equation based morph from an existing symmetric template model. Our approach uses sparse contour features that are automatically detected on the target mesh. The three-dimensional contour detection algorithm, which uses a "fixed kernel" feature detector that is independent of the underlying mesh topology, works across a wide range of human and humanoid faces.

2 Related Work

High-quality facial performance capture can be obtained through a variety of techniques. [Beeler et al. 2010] used multiple cameras and pairwise stereo reconstruction to obtain high-quality per-frame face geometry. The resulting per-frame face geometry was subsequently used in [Beeler et al. 2011] with videos to generate a single face triangulated surface that was propagated through an entire performance using anchor frames. [Fyffe et al. 2015] also combined static scans (including both geometry and reflectance) with videos to obtain a facial performance that could be better rendered under novel illumination.

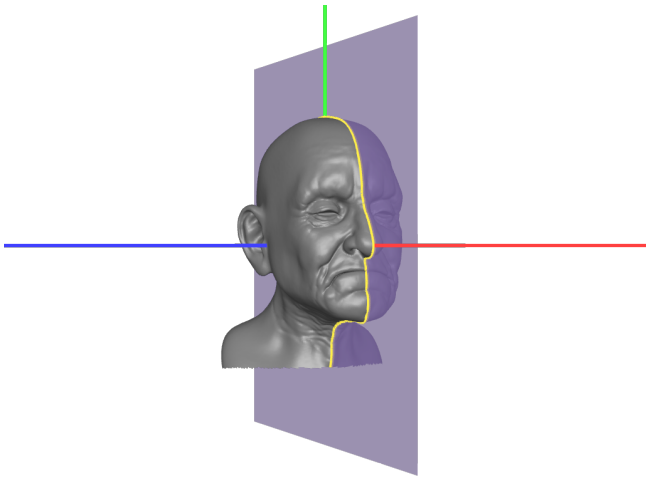


Figure 3: *Given an input target mesh, we automatically orient it such that the positive X-axis is aligned with the nose tip, the Y-axis goes through the top of the head, and the Z-axis goes through the right-ear. We bootstrap the process by using a rigid body inertia tensor to get an approximate set of axes and then refine this by adjusting the symmetry plane (purple) to achieve optimal symmetry as well as using feature analysis of a profile curve (yellow).*

Other techniques do not assume the existence of static scans and reconstruct the geometry by deforming a template surface mesh. [Akimoto et al. 1993] detected landmarks on images from two viewpoints and used the landmarks to deform a template mesh. Expression cloning [Noh and Neumann 2001] used radial basis functions and a cylindrical projection to map a template mesh to a target mesh and also proposed a method for retargeting an animation from the template to the target. More recently, [Zhang et al. 2004] captured a facial performance by deforming a template mesh using depth maps computed from multiple video cameras and structured light. This enabled them to fill in missing data and maintain a vertex correspondence through the entire performance. Template meshes were also used by [Kemelmacher-Shlizerman and Basri 2011], which leveraged similarity across surfaces to reconstruct a face mesh from a single image, and [Shi et al. 2014], which automatically acquired facial performances from a single monocular video, and have been used for surface and motion reconstruction of scanned shapes [Li et al. 2009]. A large number of techniques such as iterative closest point [Besl and McKay 1992], deformable iterative closest point [Pauly et al. 2005; Brown and Rusinkiewicz 2007], as-rigid-as possible surface modeling [Sorkine and Alexa], and deformation transfer [Sumner and Popović 2004] have been developed to align and deform a target mesh to a template mesh.

3 Template Model

Following the approach of [Sifakis et al. 2005], we started by constructing a high resolution, anatomically and biomechanically accurate flesh and muscle model of a subject's face. The flesh was represented by a conforming tetrahedralized mesh. The cranium and the jaw were represented using both an explicit triangulated surface and an implicit level set surface. The model contains 41 facial muscles that were each represented using a B-spline solid and embedded into the tetrahedralized flesh mesh following the approach of [Teran et al. 2003]. The jaw joint was defined by the endpoints of the left and right condyle sliding tracks and can rotate around the axis that

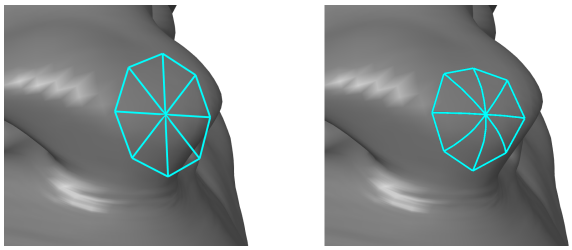


Figure 4: *We use a mesh independent fixed size stencil for curvature estimation. In this example, we move* .005 *units along eight equally spaced directions to collect sample data for the stencil. (left) The stencil placed at a point. (right) The stencil wrapped around the mesh so that its shape can be used for curvature estimation.*

connects the jaw along those two tracks. Since this model was built to match the anatomy of the subject as closely as possible, it encodes the (natural) asymmetries of the subject. These asymmetries are unique to the subject and would introduce additional unnatural bias if transferred to another model. In order to avoid biasing due to asymmetry, we symmetrized our model to obtain a symmetric template (see Figure 2). Note that this symmetrization operation only needs to be performed once and therefore was accomplished manually. Symmetrization of the muscles was straightforward since the control points of the B-spline were already symmetric in kind and number and we merely had to use averaging to make them symmetric in value.

We manually estimated the symmetry plane of the model, before converting the cranium and jaw triangulated surfaces to implicit signed distance fields on a symmetric grid aligned with the symmetry plane. A symmetric level set implicit surface was obtained by averaging the signed distance values with their symmetric counterparts. Then, the fast marching method [Osher and Fedkiw 2002] was used to convert the averaged values into a signed distance function. Symmetric triangulated surfaces were then obtained as the surface of a tetrahedralized volume generated using the method of [Molino et al. 2003] with modifications to preserve symmetry. We initialized the method with a symmetric BCC lattice aligned with the symmetry plane and enforced symmetry on all topological operations such as red-green refinement. Likewise, we symmetrized the force/velocity for physically-based compression and/or carried out Gauss-Seidel sweeping using pairs of nodes on opposite sides of the symmetry plane for optimization-based compression.

The process for generating the symmetric tetrahedralized flesh mesh was similar to the process used to generate the symmetric cranium and jaw. However, the eyelids and lips on the original mesh are in close enough proximity that the asymmetries cause them to merge together during averaging. Thus, using a quasistatic simulation, we activated the eyelid muscles to lift the eyelids and opened the lips to separate the lips obtaining a deformed asymmetric mesh which was rasterized and symmetrized as above. Then, for each node in this mesh, we used its barycentric weights in the deformed asymmetric mesh in order to interpolate a displacement. If a node was outside the deformed asymmetric mesh, we first projected it to the surface. The displacements were averaged across the symmetry plane and used to reverse the displacement of the deformed symmetric mesh to obtain a rest state with closed eyes and lips.

Due to the averaging of displacements, the resulting symmetric mesh typically had low-quality/degenerate/inverted tetrahedra. Thus, we again opened the jaw and eyelids, but this time using the symmetric mesh while also enforcing symmetry during the quasistatic flesh simulation. This mouth/eye open state was converted into a symmetric level set and remeshed symmetrically as above.

Figure 5: *A color scale representing the sum of the absolute values of the principal curvatures. (left) A standard curvature calculation using the one-ring of the underlying mesh. (right) The improved curvature estimation using our fixed size stencil shown in Figure 4. We found this smoother curvature estimate to be much more helpful for detecting landmarks.*

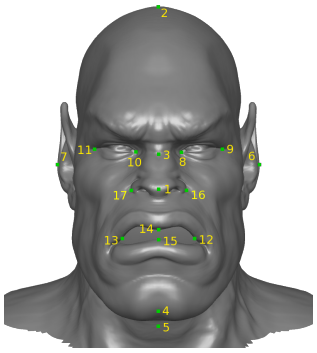
This undeformed state is of higher quality then that of the original symmetric mesh because its undeformed target state was symmetric as opposed to asymmetric. Then, again using barycentric interpolation of displacements, each surface node was returned to an undeformed state. Finally, the interior nodes were relaxed into a rest state using a quasistatic mass-spring simulation consisting of edge springs and tetrahedral altitude springs [Selle et al. 2008] while the rewound surface nodes were constrained to not move.

4 Target Meshes

In order to demonstrate our algorithm's versatility, we aimed to use a wide variety of targets ranging from humans to humanoid creatures. We decided on 12 meshes representative of four categories: human models, creature models, human scans, and scans of statues. Our method is robust enough to handle some asymmetry, but it is assumed that the ears and eyes lie on opposite sides of the mesh and that the nose and mouth lie along the center of the mesh.The Artec Human scan was downloaded from the Artec 3D Gallery website. The Lincoln scan was obtained from the Smithsonian X 3D website. Matthew, Kiran, and Yoda were scanned using an Artec Eva scanner and subsequently processed using Artec Studio. With the exception of the Yoda scan which contained the entire head, the Matthew, Kiran, Artec Human, and Lincoln scans only contained the front of the face. Thus we used a variation of our surface morphing algorithm to reconstruct the remainder of the face (see Section 8.1). Other meshes were obtained from online resources such as TurboSquid and TF3DM. After acquiring the models, we removed any extraneous parts of the mesh including the torso and shoulders. We removed triangles with areas less than $1e^{-9}\text{m}^2$ to prevent degeneracies, and we removed occluded triangles using MeshLab to get the outer shell. We then closed the surface and generated shrink-wrapped level set surfaces for each closed target mesh using the method of [Zhao et al. 2001].

5 Orientation and Symmetry

Since we make no assumptions about the initial orientation of the mesh, we bootstrap a set of coordinate axes by diagonalizing the rigid body inertia tensor calculated using a thin shell approximation. Then, for each pair of axes, we calculate a symmetry error by reflecting each point across the symmetry plane and computing the distance between its reflected position and the closest point on the mesh. The plane with the least symmetry error is designated as the left/right symmetry plane. A profile curve is created by intersecting the symmetry plane with the mesh. Every point on the profile curve can be assigned a value based on its distance from the center of mass, and we identify all local extrema of this func-



1	Pronasale
2	Top of the Head
3	Sellion
4	Ganthion
5	Neck Start
6	Left Extrema
7	Right Extrema
8	Left Endocanthion
9	Left Exocanthion
10	Right Endocanthion
11	Right Exocanthion
12	Left Mouth Corner
13	Right Mouth Corner
14	Mouth Upper Lip
15	Mouth Lower Lip
16	Left Alar
17	Right Alar

Figure 6: *Using the coordinate axes and profile curve from Figure 3, the curvature from Figure 5, and a number of robust heuristics discussed in Section 6, we automatically detect 17 key landmarks on an input face mesh.*

tion. For each local maxima, we find the two closest local minima on either side of the maxima, and note that these correspond to the nose bridge and the subnasale when the maxima is at the tip of the nose. Then, for each local maxima, the coordinate system is rotated about the symmetry plane normal until the X-axis coincides with the local maxima. Using this coordinate system, we can compute a number of geometric properties concerning this candidate nose defined by the triangle connecting the bridge, base, and tip non-dimensionalized by the size of the bounding box. For example, we consider how far forward the tip of the nose is from the bridge and base, how far up and down the bridge and base are from the tip, the angle between the bridge and the base, etc. Finally, after scoring all relative maxima, the one with the highest score is chosen as the tip of the nose.

At this point, we have a candidate coordinate system which has its origin at the center of mass. Next, we refine the symmetry plane and coordinate system using only a subset of the geometry which is deemed most reliable. Using a tight bounding box, we only consider the symmetry error of the vertices in the forward 30% of the bounding and within 20% of the bounding box height both up and down from the origin. Roughly speaking, this is the region around the nose and cheeks but can vary based on the target mesh. Note that the symmetry plane is invariant to rotation about its normal or translation within the plane, and thus we consider the remaining three degrees of freedom during optimization: rotation about the two in-plane axes as well as translation normal to the plane. Optimizing over these three degrees of freedom gives us our new coordinate system and symmetry plane from which we generate a new profile curve (see Figure 3).

6 Landmarks

We automatically detect 17 landmarks as shown in Figure 6. Many of the same landmarks are also automatically selected in expression cloning [Noh and Neumann 2001]. A number of landmarks lie on the profile curve and are readily detected. The tip of the nose is detected by intersecting the X-axis with the mesh. The top of the head is the first visible point along the center of the face when moving from top to bottom, assuming the camera is placed at $(1, 0, 0)$. The first minima along the profile curve above the pronasale is labeled as the nose bridge. We detect the chin by looking for a point below the nose in the forward 30% of the bounding box whose vertex normal points in a downward direction. From the set of candidate points, we choose the point that has the greatest Euclidean distance from the origin. The 'neck start' point is then a 0.02 unit drop from the chin. Although not on the profile curve, we also find and detect the left and right extrema points which are the left-most and right-most points on the mesh above the chin.

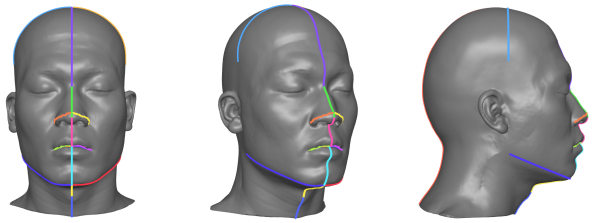


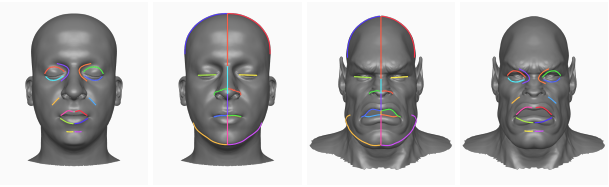
Figure 7: *The profile curve is split into segments at important landmark points. Using the coordinate axes from Figure 3, we place the camera at different locations and compute occlusion curves for the nose and mouth. A cutting plane is used to compute the curves around the head and chin, which outline boundaries of the face when viewed from the front. Colors illustrate how a curve is split into multiple component curves.*

The remaining landmarks around the eyes, mouth, and nose utilize a robust estimation of curvature. We tested a variety of methods for computing curvature on a wide variety of target meshes and discovered that the results were unreliable for feature detection. The problem stems from the fact that humans interpret curvature of various features on the face using a scale invariant to the size of the mesh. Since we are looking for features as opposed to mesh abnormalities, we take a relatively novel approach to estimating the curvature. To estimate the curvature at a vertex, we do not use the typical one ring of vertices which utilizes an arbitrary subset of a face mesh dependent on the mesh resolution, but rather use a fixed-size stencil which is invariant of the mesh. We chose a fixed-size stencil that uses 8 equally spaced points on a circle with a radius of 0.005. We place this stencil on a vertex and fold it about the mesh using the advection method of [Qiu et al. 2015] to determine where each of the 8 points would lie on the mesh as shown in Figure 4. Then using these 8 triangles folded around the mesh, we use the curvature estimation from [Taubin 1995] to estimate the curvature for this vertex. One could obviously use a different number of points or a different radius, but similar results should be achievable as long as a fixed-size stencil is used (see Figure 5).

To detect the four feature points at the corners of the eyes, we create an axis aligned bounding box centered at the nose bridge which extends a fixed fraction of the bounding box size in each of the three axial directions. Each eye is considered independently using half of the bounding box. We only consider vertices whose principal curvature absolute value sum is within the top 10% of all vertices on the front of the face (i.e. $X > 0$). From these, we choose the points closest to the bottom rear corners of each half bounding box.

To detect the landmarks around the mouth, we use an axis aligned bounding box that stretches from the nose to the chin in the Y-direction and from the inside of the eye to a fraction of the way to the outside of the eye in the Z-direction. In the X direction, the box is initialized from the nose to the chin but is then expanded by $1/6$ of the bounding box thickness in both directions. We eliminate points whose vertical distance is too close to the nose and identify points above a threshold curvature. If there are no points that satisfy these conditions, we expand the bounding box incrementally in the Z-direction towards the outer eye until we find at least one candidate point for the mouth corner. If there are multiple points that are close candidates, we take the candidate point that is furthest towards the outside of the bounding box. If there are multiple points that are close, we choose the point which has the smallest Y value.

We detect an occlusion curve for the mouth by placing the camera at $(.7, 0, 0)$ and identifying the boundary of the completely visible



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

Figure 8: *We transfer a number of curves from the template to the target using 2D image based techniques.* ***(a)*** *2D curves on a 2D image of the template.* ***(b)*** *2D correspondence curves on a 2D image of the template.* ***(c)*** *Same as (b) except as determined for a target mesh using the curves from Figure 7, landmarks, and various heuristics.* ***(d)*** *Curves from (a) mapped to a target mesh using the correspondence between (b) and (c). These 2D curves are then projected back into 3D.*

triangles. The upper and bottom lip points are determined within the mouth search region by intersecting this occlusion curve with the symmetry plane. In the case of scanned meshes, the occlusion curve can fail to detect a mouth curve. In that case, we estimate the mouth lip positions to be halfway between the two corner points and slightly offset in the up and down direction.

To identify the left and right alar curvature points (valleys at the side of the nose), we also use an occlusion curve. For this curve, we place the camera on the symmetry plane offset from the chin looking upwards at the center of the mesh, considering only intersections from above a horizontal plane slightly below the level of the nose tip to avoid obstruction from the mouth. In order to filter the points in the full occlusion curve to just relevant points near the nose, we make the assumption that the alar points must reside between the midpoints of the eyes. We identify candidate alar points along the occlusion curve by computing the dot product between the position and a direction of $(0.2, 0, \pm 1)$ to determine relative local minima in distance, capturing the valleys to the sides of the nose tip. From this set of candidates, we find the candidate that is closest to a point that we approximate based on detected landmarks and typical face construction/proportions, while also being of high curvature and resulting in a reasonable nose curve arc length. We approximate the nose alar points as points horizontally on the same plane as the nose tip, at the depth of the inner eye point and of a distance .2 of the way between the inner eye point and the outer eye point from the symmetry plane.

7 Target Curves

We use the landmarks to bootstrap the detection of feature curves. We chose to use curves because the Poisson equation provides better results with boundary conditions of lower codimension (see Section 8). Furthermore, many of the feature curves that we detect are aligned with muscles underneath the surface, and modifying these curves allows for control over how a muscle is morphed. The curves we detect are stored such that they are easily editable allowing a user to add, delete, or modify curves to increase the quality of the morph (for use in high-end production such as motion pictures). However, this has not been done for any of the examples presented in this paper.

We segment our curves to obtain more accurate correspondences between the target mesh and the template model. The profile curve is segmented using the profile curve landmarks. A curve is found at the top of the head by intersecting the mesh with the plane centered at the top of the head landmark with a normal pointing along the X-axis. Only the points above the eyes are kept, and the curve is

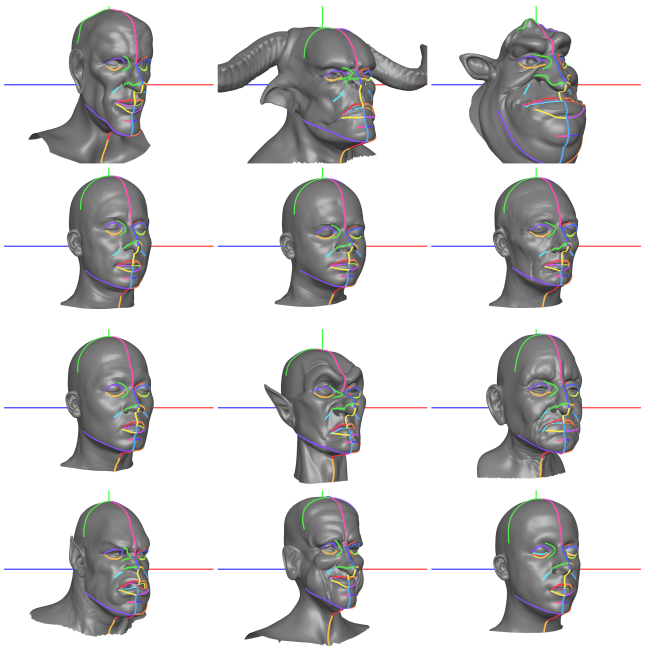


Figure 9: *All the target meshes with their final curve correspondences from Figures 7 and 8. These form the basis of correspondences for the surface morph.*

then split into a left and right section using the symmetry plane. We perform a similar method for the lower jaw curves; however, the plane we use is angled such that it goes through the chin and the jaw origin, which is the point on the negative Y-axis with the same Y value as the mouth left corner. The mouth occlusion curve from Section 6 is split it into four sections using the symmetry plane and feature points. The nose occlusion curve wraps across the front of the nose, and is also split along the symmetry plane into left and right nose curves.

We found a number of curves difficult to robustly identify using the aforementioned methods. Thus, we transfer the more difficult curves using a two-dimensional image-based method to bootstrap correspondences. We project the curves and landmarks of the target mesh onto the image plane using a perspective projection with the camera placed at $(1, 0, 0)$. The visible portions of the profile, contour, and occlusion curves along with two additional curves connecting the inner and outer corners of the eyes are used as correspondences between the two-dimensional template and target images (see Figures 8b and 8c). We overlay the two images and calculate a displacement for each vertex of the segmented two-dimensional correspondence curves. Then, we solve two two-dimensional Poisson equations $\nabla^2 u = 0$ where u is either the x or y component of the displacement. If the Poisson stencil intersects with a correspondence curve, we apply a Dirichlet boundary condition using [Gibou et al. 2002] with a value obtained by linearly interpolating vertex displacements to the intersection point. The results of the Poisson solve are used to interpolate displacements to the vertices of the template curves in Figure 8a so that they can be transferred to the target image in Figure 8d. These points are then projected back onto the three-dimensional target mesh by tracing rays from the same camera used for the perspective projection.

8 Morphing

Due to the rapid falloff in the solution kernels of the Poisson equation, we had difficulty obtaining acceptable results when using boundary conditions of codimension higher than one. Thus, we use

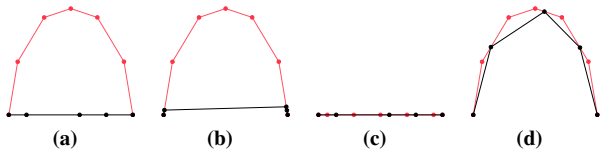
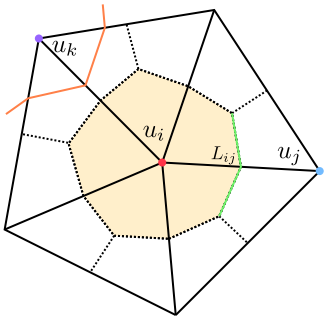


Figure 10: ***(a)*** *The Poisson solve tends to underestimate regions of high curvature as it stretches between boundary conditions.* ***(b)*** *Projecting each point from the Poisson solve (black) onto the closest point on the target mesh (red) results in stretching.* ***(c)*** *Instead we first project the higher curvature target mesh onto the result of the Poisson solve, and then* ***(d)*** *use the corresponding locations in order to map from the template mesh onto the target mesh.*

the curves shown in Figure 9 as boundary conditions for morphing the two-dimensional surface mesh of the template to the target mesh, before subsequently using this newly acquired correspondence between surface meshes as boundary conditions for a fully three-dimensional morph. Alternatively, one could estimate the surface correspondence using techniques such as deformable iterative closest point [Pauly et al. 2005; Brown and Rusinkiewicz 2007].

Aligning the two meshes using the common coordinate systems (Figure 3), we can look at each target curve (Figure 9), and calculate the Euclidean distance between the vertices from the surface of the template model to the target mesh. These displacements can then be linearly interpolated to the interiors of the curves. Note that the displacements are three-dimensional vectors $\vec{d} = (d_x, d_y, d_z)$ and thus we solve three codimension-one Poisson equations, $\nabla^2 u = 0$ where u is either d_x, d_y, or d_z, on the template triangulated surface in order to find a displacement for every vertex.

In order to obtain a symmetric system, we discretize the area-weighted Laplacian using the finite volume method. The yellow control area for a vertex is defined as one-third the area of all incident triangles. The length-weighted flux F_{ij} between two adjacent vertices i and j is equal to $L_{ij}(u_j - u_i)/ \parallel \vec{x_j} - \vec{x_i} \parallel$ where L_{ij} is highlighted in green. The area-weighted Laplacian at node i is then equal to $A_i \nabla^2 u = \sum_{j \in N_i} F_{ij}$ where N_i are the neighbor nodes of node i. The resulting system is symmetric positive definite and is solved using modified incomplete Cholesky preconditioned conjugate gradients. If the segment connecting two nodes intersects a correspondence curve (e.g. the segment connecting u_i and u_k which intersects the orange curve), we apply a Dirichlet boundary condition using the second order accurate method of [Gibou et al. 2002]. Note that since the discretization is carried out on the template mesh, we can precompute everything except for the values of the displacements used in the Dirichlet boundary conditions—allowing for a number of optimizations. The Poisson equation has a smoothing effect and therefore although our boundary conditions will map curves to curves, the portions of surface between curves will tend to flatten out. We have chosen the particular curves in Figure 9 in order to ensure that the solution to the Poisson equation displaces the surface of the template model close enough to the target mesh with the desired proportions within the desired subsections of the mesh.



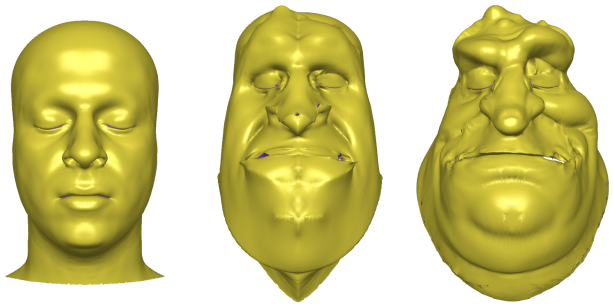
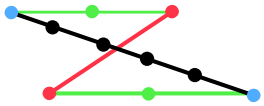


Figure 11: *Using the curves from Figure 9 as Dirichlet boundary conditions, we solve three codimension-one Poisson equations on the triangulated surface in order to map the template mesh to the target mesh. (left) Template mesh, (middle) template mesh deformed based on the Poisson solve, (right) result after performing the snap.*

After morphing the template surface mesh so that it is approximately aligned with the target mesh, we snap the morphed template mesh to the target mesh as follows (see Figure 10). First, we snap the target mesh to the morphed template mesh in order to find correspondences (i.e. barycentric weights) that are then used to snap the morphed template mesh to the target mesh. Note that we do not snap the template mesh to the target mesh directly since the smoothed out template mesh is lacking features. One drawback of snapping, especially over longer distances, is that triangles of the template mesh may invert in regions of higher curvature. We identify a triangle as inverted (red) if the dot product of its face normal and the shrink-wrapped level set normal at its barycenter is negative. Then, we identify a subregion (green) around the inverted triangles and relax this region into a non-inverted state (black) using zero-length springs while constraining the nodes on the boundaries of this region (blue). After relaxation, the vertices of the distorted template surface mesh can be resnapped. This process can be run multiple times to improve the quality of the resultant mesh (see Figure 11).



At this point, every vertex in the template surface mesh has a corresponding morph location on the target mesh providing per-vertex displacement values. Next, we solve three three-dimensional Poisson equations $\nabla^2 u = 0$ where u is either d_x, d_y, or d_z in order to find a displacement field on the entire three-dimensional space following the approach of [Ali-Hamadi et al. 2013]. If the stencil intersects with the template surface mesh, we apply a Dirichlet boundary condition again using [Gibou et al. 2002] with a value obtained by barycentrically interpolating triangle vertex displacements to the intersection point.

Once again, since this Poisson equation is discretized on a template model, everything except for the Dirichlet boundary conditions can be precomputed for the sake of optimization. The resulting three-dimensional displacement field is used for a final morph of the entire template model. The template model, triangulated surfaces, and tetrahedralized volumes are morphed by linearly interpolating the displacement from the grid for each vertex. Implicit surfaces are morphed by first converting them to a tetrahedralized volume, morphing the tetrahedralized volume, and then re-rasterizing the tetrahedralized volume onto a three-dimensional grid.

B-spline solid muscles are similarly morphed by converting them to a tetrahedralized volume, morphing the tetrahedralized volume, and reconstructing the B-spline solid. One could morph the control points corresponding to the B-spline solid directly; however, some

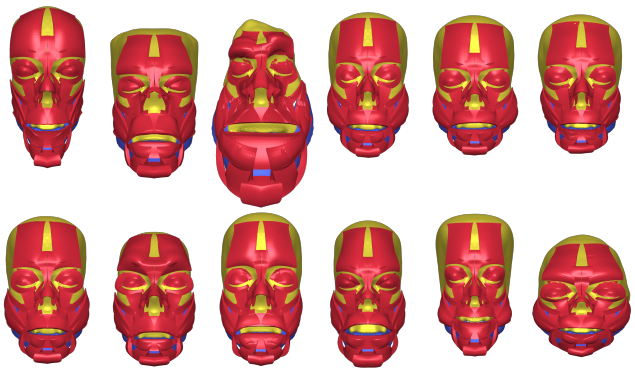


Figure 12: *Morphed craniums, jaws, and muscles for the twelve anatomical face simulation models automatically generated using our procedure.*

control points for the B-spline solid muscles lie outside of the flesh volume of the template model and the derivative discontinuity in the displacement field at the template surface mesh can often result in displacements that are less accurate for these points. The morphed cranium, jaw, and muscles for each of our target meshes are shown in Figure 12.

When displacing the tetrahedralized flesh mesh, the exterior surface may collide with the interior surface. We apply [Bridson et al. 2002] or [Baraff et al. 2003] to detect and resolve these collisions to obtain a collision-free flesh triangulated surface. Note that we only modify the displacements of the interior surface during collision processing. Then, we recompute the displacement field using the displacements of the collision-free flesh triangulated surface as boundary conditions and use this new displacement field to morph the rest of the interior anatomy.

8.1 Scan Reconstruction

Our technique for morphing the template surface mesh to the target mesh was used to reconstruct plausible full head geometry and ears for the frontal scans of Matthew, Kiran, Artec Human, and Lincoln since our automated morphing algorithm assumes that the target mesh is a full face mesh (including ears, back of the head, etc.). We began by drawing the correspondence curves on the partial scan which were then used to deform the template surface mesh to approximately match the target mesh as above. Then, we snapped nodes with a corresponding location on the partial frontal scan. This created discontinuities at the boundaries of the partial frontal scan and was resolved by repeating the Poisson solves on the triangulated surface with the displacements of the snapped nodes as Dirichlet boundary conditions.

9 Remeshing

The morphed tetrahedralized flesh mesh can often contain low-quality/degenerate/inverted tetrahedra which we resolve by remeshing using [Molino et al. 2003]. However, if the mouth and/or eyes are closed on the target mesh, converting the surface of the tetrahedralized flesh mesh to a level set implicit surface representation will merge the mouth/eyes shut. To prevent this, we first open the mouth and lift the eyelids using a quasistatic flesh simulation. The mouth is opened by slightly opening the jaw after initializing a mass-spring system consisting of edge springs and tetrahedral altitude springs. The eyelids are opened simply by shrinking the rest lengths of the springs lying within the region of the morphed eyelid muscles. After remeshing, we interpolate displacements of surface

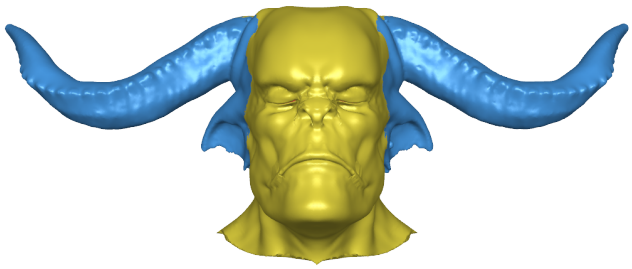


Figure 13: *Using a flood fill algorithm on a shrink-wrapped level set, we extract the portion of the level set corresponding to the ears and horns (blue) which is not captured by our morph. This region is converted into a tetrahedralized mesh and hard bound to the morphed tetrahedralized flesh mesh (yellow) which allows the ears (and horns) to move as the mouth and eyes are opened. This also allows the ears (and horns) to be properly represented in our final model.*

nodes returning them to their undeformed state before using a mass-spring simulation to relax the interior of the mesh (see Section 3).

9.1 Ears and Horns

It can be quite difficult to identify salient features for correspondences when morphing the ears. However, since the ears require no collision geometry or internal musculature, we can fully automatically generate ears and horns from the shrink-wrapped level set. We begin by detecting bounding boxes around the ears and horns. Starting from the left and right extrema landmarks, we detect a tight bounding box around the points which have a sum absolute curvature value > 140 and have a path returning to the left/right extrema point through other high curvature points. After finding the initial region, we increase the height of the bounding box such that the bottom corners have the same Y value as the chin and perform another search for connected high curvature regions. We discard regions that are too small and then merge this region with the initial region to get a bounding box around the ears and horns. Since we obtain the ears from the target shrink-wrapped level set, they are no longer needed on the template and thus we remove them unless we have a target mesh without ears such as a three-dimensional scan. In addition, as pertains to Figure 10c, we also do not snap any target mesh nodes in the ear bounding boxes. Likewise, the snap is also ignored for morphed template surface nodes inside the bounding box. This results in a discontinuity at the boundary of the target ear bounding boxes which is resolved by repeating the Poisson solves on the triangulated surface (see Section 8) with the displacements of the snapped nodes as Dirichlet boundary conditions.

We identify the ears and horns with a flood fill algorithm. First, we identify the initializing seeds by finding all cells inside the shrink-wrapped level set as well as the ear bounding box, which are also adjacent to and outside the morphed template surface mesh. Starting from these initial seeds, we flood fill the shrink-wrapped level set to obtain the ears and horns. This flood fill region is converted into a level set, meshed into a tetrahedralized volume using [Molino et al. 2003], and hard bound to the morphed tetrahedralized volume as shown in Figure 13. After opening the mouth and eyes (as discussed above), we convert both deformed tetrahedralized volumes into a single level set implicit surface for remeshing (again as above) forming a contiguous mesh that includes the ears and horns.

10 Discussion and Conclusions

We simulate our models using the the quasistatic framework of [Teran et al. 2005] to obtain the expressions shown in Figures 1,

Figure 14: *Here we show a selection of expressions on Ogre Head. (left) Ogre Head with eyes open. (middle) We activate the frontalis muscles to raise the Ogre Head's eyebrows. (right) We activate the left levator labii superioris, levator anguli oris, zygomatic minor, llsan lateral, and buccinator muscles to raise the left side of the Ogre Head's upper lip and the left depressor anguli oris, depressor labii inferioris, and mentalis to lower the left side of his lower lip.*

14, and 15 (see video for additional expressions). The muscle activations are driven by anatomically motivated keyframes and the same muscle activations are used across all of the models. One could alternatively use a dynamic simulation framework without any modifications to the model to obtain ballistic effects such as secondary motion of the cheeks in laughter and speech.

We have presented a fast, fully automatic algorithm for creating simulatable flesh and muscle models for human and humanoid faces. We accomplish this by morphing a symmetric template model to match a target mesh based on automatically detected landmarks and feature curves. The resulting target models contain all internal anatomy and are able to exhibit the full range of motion available to the high visual fidelity template model. Since our algorithm requires only a sculpted or scanned target shape as input (e.g. the rest/neutral shape), building a target model requires no additional time or cost on top of a traditional capture session.

Ongoing work consists of quantifying and minimizing errors in the target model in comparison to ground truth. Even a model constructed from scratch with known constitutive properties and simulated with large amounts of computational resources can produce a result that differs somewhat from reality due to individual abnormalities/characterizations in the actor's anatomy. Thus, there may be differences in the muscle activations that need to be considered to produce certain expressions. This could potentially be compensated for by using correctives [Bhat et al. 2013]. There may also be differences in the underlying muscle geometry and thus part of our current work is focused on solving a series of fully automated optimization problems in order to determine adjustments to the underlying three-dimensional anatomical model in order to more precisely match the actor's range of motion.

Acknowledgements

Research supported in part by the Intel Science and Technology Center for Visual Computing. Computing resources were provided in part by ONR N00014-05-1-0479. M.C. was supported in part by an NDSEGF. We would like to thank Christos Kozyrakis and Mark Horowitz for additional computing resources. The authors would like to thank Brice Criswell, Scott Jones, Michael Koperwas, and Cary Phillips for helpful discussions.

References

Figure 15: *Here we show a different selection of expressions on Yoda. (left) Yoda with eyes open. (middle) We open Yoda's jaw. (right) We slightly close the eyes and activate the zygomatic major and orbicularis oculi muscles to show Yoda with a sleepy, content smile.*

3LATERAL, 2015. 3Lateral Studio. `3lateral.com`.

AKIMOTO, T., SUENAGA, Y., AND WALLACE, R. S. 1993. Automatic creation of 3d facial models. *IEEE Comput. Graph. Appl. 13*, 5 (Sept.), 16–22.

ALI-HAMADI, D., LIU, T., GILLES, B., KAVAN, L., FAURE, F., PALOMBI, O., AND CANI, M.-P. 2013. Anatomy transfer. In *ACM SIGGRAPH Asia 2013 papers*, SIGGRAPH ASIA '13, 188:1–188:8.

AUTODESK, 2011. Face robot. `http://softimage.wiki.softimage.com/xsidocs/face_cover.htm`.

BARAFF, D., WITKIN, A., AND KASS, M. 2003. Untangling cloth. *ACM Trans. Graph. (SIGGRAPH Proc.) 22*, 862–870.

BEELER, T., AND BRADLEY, D. 2014. Rigid stabilization of facial expressions. *ACM Trans. Graph. 33*, 4 (July), 44:1–44:9.

BEELER, T., BICKEL, B., BEARDSLEY, P., SUMNER, B., AND GROSS, M. 2010. High-quality single-shot capture of facial geometry. *ACM Trans. Graph. (SIGGRAPH Proc.) 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. *ACM Trans. Graph. (SIGGRAPH Proc.) 30*, 4, 75:1–75:10.

BESL, P. J., AND MCKAY, N. D. 1992. Method for registration of 3-d shapes. In *Robotics-DL tentative*, International Society for Optics and Photonics, 586–606.

BHAT, K. S., GOLDENTHAL, R., YE, Y., MALLET, R., AND KOPERWAS, M. 2013. High fidelity facial animation capture and retargeting with contours. In *ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 7–14.

BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. (SIGGRAPH Proc.) 21*, 3, 594–603.

BROWN, B. J., AND RUSINKIEWICZ, S. 2007. Global non-rigid alignment of 3-d scans. In *ACM TOG*, vol. 26, ACM, 21.

FYFFE, G., JONES, A., ALEXANDER, O., ICHIKARI, R., AND DEBEVEC, P. 2015. Driving high-resolution facial scans with video performance capture. *ACM TOG 34*, 1.

GARRIDO, P., VALGAERTS, L., WU, C., AND THEOBALT, C. 2013. Reconstructing detailed dynamic face geometry from monocular video. *ACM Trans. Graph. 32*, 6, 158.

GIBOU, F., FEDKIW, R., CHENG, L.-T., AND KANG, M. 2002. A second order accurate symmetric discretization of the Poisson equation on irregular domains. *J. Comput. Phys. 176*, 205–227.

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

LEWIS, J. P., ANJYO, K., RHEE, T., ZHANG, M., PIGHIN, F., AND DENG, Z. 2014. Practice and Theory of Blendshape Facial Models. In *Eurographics 2014 - State of the Art Reports*, The Eurographics Association, S. Lefebvre and M. Spagnuolo, Eds.

LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. 2009. Robust single-view geometry and motion reconstruction. In *ACM TOG*, vol. 28, ACM, 175.

MOLINO, N., BRIDSON, R., TERAN, J., AND FEDKIW, R. 2003. A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In *12th Int. Meshing Roundtable*, 103–114.

NOH, J., AND NEUMANN, U. 2001. Expression cloning. In *Proc. of ACM SIGGRAPH*, ACM Press, E. Fiume, Ed., 277–288.

OSHER, S., AND FEDKIW, R. 2002. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag. New York, NY.

PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. J. 2005. Example-based 3d scan completion. In *Proceedings of the Third Eurographics Symposium on Geometry Processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SGP '05.

QIU, L., YU, Y., AND FEDKIW, R. 2015. On thin gaps between rigid bodies two-way coupled to incompressible flow. *J. Comput. Phys. 292*, 0, 1 – 29.

SELLE, A., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. *ACM Trans. Graph. (SIGGRAPH Proc.) 27*, 3 (Aug.), 64.1–64.11.

SHI, F., WU, H.-T., TONG, X., AND CHAI, J. 2014. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM Trans. Graph. 33*, 6 (Nov.), 222:1–222:13.

SIFAKIS, E., NEVEROV, I., AND FEDKIW, R. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph. (SIGGRAPH Proc.) 24*, 3.

SIFAKIS, E., SELLE, A., ROBINSON-MOSHER, A., AND FEDKIW, R. 2006. Simulating speech with a physics-based facial muscle model. *ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 261–270.

SORKINE, O., AND ALEXA, M. As-rigid-as-possible surface modeling.

SUMNER, R., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. In *ACM Trans. on Graph. (Proc. ACM SIGGRAPH)*, vol. 23, 399 – 405.

TAUBIN, G. 1995. Estimating the tensor of curvature of a surface from a polyhedral approximation. *ICCV*, 902–907.

TERAN, J., BLEMKER, S., NG, V., AND FEDKIW, R. 2003. Finite volume methods for the simulation of skeletal muscle. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 68–74.

TERAN, J., SIFAKIS, E., IRVING, G., AND FEDKIW, R. 2005. Robust quasistatic finite elements and flesh simulation. *Proc. of the 2005 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 181–190.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. 2004. Spacetime faces: High resolution capture for modeling and animation. In *ACM Trans. Graph. (Proc. ACM SIGGRAPH)*, ACM Press, vol. 23, 548–558.

ZHAO, H.-K., OSHER, S., AND FEDKIW, R. 2001. Fast surface reconstruction using the level set method. In *1st IEEE Wrkshp. on Variational and Level Set Meth., 8th Int. Conf. on Comput. Vis.*, 194–202.