Interactive Sculpting of Digital Faces Using an Anatomical Modeling Paradigm

A. Gruber[1,2]   M. Fratarcangeli[1,3]   G. Zoss[1,2]   R. Cattaneo[1]   T. Beeler[1]   M. Gross[1,2]   D. Bradley[1]

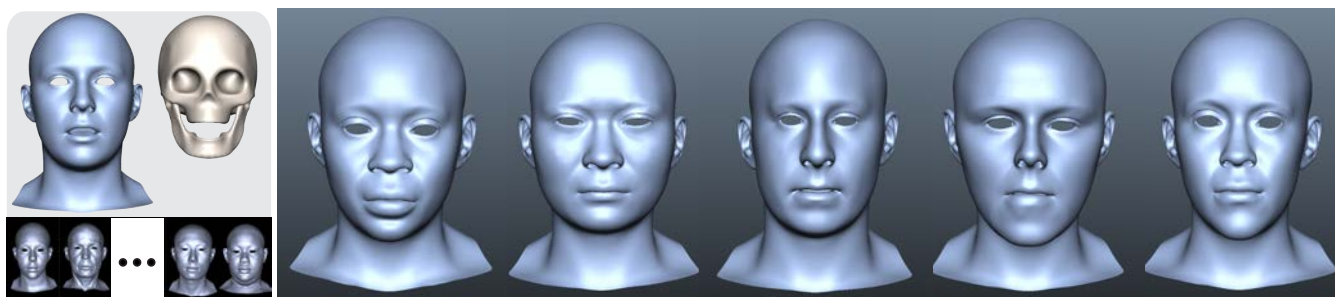[1]DisneyResearch|Studios   [2]ETH Zurich   [3]Chalmers University of Technology

Figure 1: *We build a local anatomical model from a corpus of 3D face scans (left) and create a novel anatomically-driven digital face sculpting system where novice users can efficiently create realistic 3D face identities (right).*

Abstract

Digitally sculpting 3D human faces is a very challenging task. It typically requires either 1) highly-skilled artists using complex software packages for high quality results, or 2) highly-constrained simple interfaces for consumer-level avatar creation, such as in game engines. We propose a novel interactive method for the creation of digital faces that is simple and intuitive to use, even for novice users, while consistently producing plausible 3D face geometry, and allowing editing freedom beyond traditional video game avatar creation. At the core of our system lies a specialized anatomical local face model (ALM), which is constructed from a dataset of several hundred 3D face scans. User edits are propagated to constraints for an optimization of our data-driven ALM model, ensuring the resulting face remains plausible even for simple edits like clicking and dragging surface points. We show how several natural interaction methods can be implemented in our framework, including direct control of the surface, indirect control of semantic features like age, ethnicity, gender, and BMI, as well as indirect control through manipulating the underlying bony structures. The result is a simple new method for creating digital human faces, for artists and novice users alike. Our method is attractive for low-budget VFX and animation productions, and our anatomical modeling paradigm can complement traditional game engine avatar design packages.

CCS Concepts

• *Computing methodologies → Mesh geometry models;*

1. Introduction

Digital humans are an ever more important part of our reality, and have become central elements in modern film and video game productions, social media and communication applications, and augmented- and virtual reality experiences, to name just a few areas. Creating realistic digital humans in general and faces in particular is an extremely challenging task, as it is paramount to master every aspect such as to not fall into the infamous uncanny valley.

A typical approach to model digital humans and to avoid the uncanny valley is to digitize real humans using scanning systems. This, however, is only feasible in cases where the goal is to get an exact digital copy of a human who can be captured. For entertainment use cases, exact digital doubles are often less relevant, in particular when designing novel characters artistically. At the moment, for high-end productions artists create such characters using free-

form sculpting tools, such as ZBrush[†], Mudbox[‡] or Maya[§]. These applications are powerful design tools that offer a lot of functionality and artistic freedom, and thus require great skill in the art of sculpting human faces. As a consequence, creating even a single digital human asset can easily take expert artists weeks to months to complete, restricting the use of such assets for hero characters only, even in high budget productions. On the other end of the spectrum, several video game engines[¶] offer consumer-level avatar design tools, which greatly simplify the design process through heavy constraints, e.g. artistically-created blendshape models and predetermined libraries of example faces. While easy to use, these methods have limited flexibility due to the imposed constraints. Taking instead a data-driven approach, we propose a new interactive face sculpting method that is both easy to use for novice users, and also flexible in sculpting capability—combining key benefits from both high-end and consumer-level facial editing tools available today.

Data-driven face modeling is an attractive approach to both ease the burden of manually sculpting digital characters from scratch, and also remove the need for limited artistically-created constraints. The general idea is to sculpt a face while constraining the result to be a plausible human face. Domain-specific modeling tools, such as MakeHuman[‖], ensure that the resulting face is plausible by enforcing modifications to reside within a subspace of face shapes learned from data. Such tools are typically controlled globally and indirectly via a low dimensional parameterization, for example a PCA basis. While this ensures robustness, it makes the creation of specific local features rather difficult and often impossible, since any change affects the face globally. Furthermore, this indirect control, while effective for changing semantically meaningful properties like gender and age, does not allow for the fine-grained control that direct sculpting methods offer, where the user can, for example, grab and move a certain part of the skin surface.

Our novel, data-driven approach facilitates artistic sculpting of 3D faces, allowing even novice users to intuitively and efficiently create plausible digital humans. We focus on the problem of geometry creation for the neutral expression or *identity* of the character, leaving other challenges such as texture, appearance and expression rigging outside the scope of this work. Our method unifies indirect (semantic features) and direct control of the geometry, as well as local and global editing paradigms. In order to generate plausible identities, even under local direct manipulation, we propose to use anatomical constraints that relate the skin surface to the underlying bony structure, inspired by Wu et al. [WBGB16]. This allows for a completely novel modeling paradigm—*anatomical modeling*—where the shape and position of the underlying bones may be altered to properly impact the skin surface at an anatomical level. For example, in order to create a character with a narrow jaw line it is sufficient with our system to scale the width of the jaw bone and the anatomical constraints will deform the skin appropriately (see

Fig. 6). This is likely the most intuitive way to create such a facial shape.

In order to provide this anatomical modeling paradigm, we use a data-driven approach and build a specialized anatomical local model (ALM) from a dataset of scanned 3D faces with underlying bony structures estimated. The ALM model, introduced by Wu et al. [WBGB16], allows local control regularized globally by the skull and mandible. While originally designed for performance capture of one specific face, we show how to extend this model in the context of digital sculpting, combining geometric and anatomical information from many individuals into a single model. Furthermore, as solving an ALM model is a time-consuming optimization, we extend recent GPU-based optimization schemes specifically designed for nonlinear optimization problems [FBG*20] to achieve an interactive sculpting interface. As a result, we provide an intuitive, simple and efficient method for digital face sculpting that allows both direct local and indirect semantic control, all while ensuring that the resulting face remains plausible through our anatomical modeling paradigm. Specifically, our interaction allows:

- *Local-global control* - Alter the entire face or manipulate only parts of it.
- *Indirect control* - Indirectly alter the face shape by changing the underlying parameters, such as increasing BMI or age of the character.
- *Direct control* - Directly change the facial shape by moving surface points.
- *Anatomical control* - Change the facial shape indirectly by manipulating the underlying bones.

Our approach is an ideal solution for low-budget CG productions where time-consuming and expensive asset creation using high-end production tools is infeasible. The core component of our method, i.e. the data-driven anatomical modeling paradigm, would also complement traditional consumer-level game avatar design packages by reducing artist-mandated constraints and allowing more editing freedom. We will demonstrate a wide range of 3D faces created with our method, and validate our approach through a user study of both novice and expert artists.

2. Related Work

In this section we first present related work in human face synthesis, grouped by methodology and conclude by motivating our own approach.

2.1. Morphable Model Based Approaches

Morphable models build on the assumption that human shape geometries lie on a manifold, which can be explored in order to synthesize identities and motions. A linear statistical model based upon the identity space of a population of 3D digital humans can be used to represent bodies (e.g., [ACP03, LMR*15]), or specifically faces (e.g., [BV99]). Such models are often built by applying principal component analysis (PCA) to a dataset composed of hundreds [PKA*09], or even thousands [BRP*18, LBB*17] of 3D scans. In a similar way, facial configurations can also be manually edited by using linear blend shapes [LWP10, TZN*15, BWP13].

[†] http://pixologic.com/features/about-zbrush.php
[‡] https://www.autodesk.com/products/mudbox/overview
[§] https://www.autodesk.com/products/maya/overview
[¶] e.g. Black Desert Online, Fallout 4
[‖] http://www.makehumancommunity.org/

However, the set of blend shapes of a reasonably expressive model can be very large, which may become unmanageable when blending between identities of different people instead of expressions of just one individual. A notable exception is the use of *local* linear models as proposed by [TDlTM11], which can allow greater expressivity with fewer blend shapes. We also employ a local model approach for direct face manipulation, however with superior performance over [TDlTM11] on account of our anatomical component that aims to respect physical properties of faces during the edits. Our method is also more flexible as our ALM model inherently separates local motion from local deformation, yielding a larger space of sculptable faces.

Morphable models offer straightforward ways to implement direct and indirect control. However, the basis transform induced by applying PCA or a similar method, entirely removes the semantic meaning of the basis. This makes the indirect control via coefficient manipulation neither useful nor intuitive for artists. To address the issue, [NVW*13] present a way to extract a small number of intuitive base shapes but without guaranteeing orthogonality. A semantic approach for indirect shape editing has been proposed in [YCHK15], which allows the user to manipulate the geometry of an object in a continuous way by tuning semantic attributes (for example, it allows to make the shape of a shoe *more comfortable*). This approach is more generic than ours but the semantic attributes require a large number of user tests to be defined (several hundreds for each attribute). We avoid this by targeting specifically the editing of human faces and, by doing so, we can embed anatomical knowledge in our model and, at the same time, rely on predetermined data for each learned subject (e.g., BMI, age, etc.).

In fact, our method is based on the Anatomical Local Model (ALM) introduced by [WBGB16], which decomposes the face into many small local patches, overlapping and not necessarily aligned into a grid. Each patch gets their own set of rigid motion and non-rigid deformation parameters. Patches are coupled with global parameters that define the rigid transformation of the underlying bony structure. The facial configuration is computed by solving a nonlinear least-squares optimization problem considering the anatomy of the face, achieving expressivity without sacrificing stability and coherence.

2.2. Physics-based Face Models

Physics-based approaches to facial animation are able to synthesize motions by modeling passive flesh, active muscles, and bone structures while simulating their physical interactions, including collisions. After the seminal work by [TW90], several approaches have been proposed over the years with different trade-offs with respect to interactivity, generality, realism and easiness to build and use, [KHS03, SNF05, Fra13, CBE*15, IKKP17] just to name a few.

While these models can effectively capture facial dynamics, they lack some of the features needed for modeling identities, for example it is very difficult, if not impossible, to change gender, BMI, ethnicity, or make local changes to some facial parts (e.g., nose or eyes).

2.3. Sketch-Based Approaches

Direct manipulation interfaces allow artists to manipulate handles on the face surface changing its geometry. Sketch-based approaches provide a number of different methods to place handles in a natural way, often similar to how a painter uses a brush on canvas. In [LCXS07], a stroke-based interface, namely *Face Poser*, is combined with a model prior to constrain obtained results to plausible facial expressions. [SILN11] propose a direct manipulation technique for large blendshape models. They expand on the work by [AL10] to present the artist with a more predictable interaction experience. [YK14] present a shape editing method to deform objects in ways that make certain edits more preferable over others. Such preferences are not semantic, however, but learned from a set of input geometric models. [NVW*13] describe an automatic decomposition method to extract localized deformation components (essentially localized blendshapes), from a given set of shapes. Moreover, they enable the artist to specify new components and allow for direct as well as indirect control. To extend indirect blendshape activation interfaces, [COL15] describe a technique to allow the artist to place direct control manipulators on the model on the fly based on screen-space sketch strokes. They find the corresponding blendshape weights for the manipulators with a technique inspired by [AL10]. In contrast to these methods, our work is the first to allow interactive facial manipulation with an anatomical modeling paradigm, which provides additional interactive controls over bone structures, and allows a high degree of editing freedom while remaining anatomically plausible through the ALM model optimization.

2.4. Generative Models

Recently, Generative Adversarial Networks [GPAM*14] have been trained to model 3D faces based on UV maps [MPN*19], which can be mixed with PCA and correlate both geometry and appearance at the same time [GLP*19]. These approaches are promising both in terms of quality and usability; they are particularly effective when the training dataset is defined according to the geometry, appearance and motion of a single individual. When used in a broader context, however, they may lead to either unwanted visual results or lack of detailed control, making them not useful in the context of mass production of digital faces.

In contrast to the aforementioned methods, our approach provides an easy and precise tool for interactive face manipulation that converts manual edits into anatomically-based constraints in a transparent way to the user. This novel, unified approach includes both direct and indirect semantic, as well as anatomical, face control providing artistic freedom while leading to 3D realistic faces in a matter of a few minutes, even when used by novice users.

3. Background and Model Construction

In this section, we give an overview of the anatomical local face model (Section 3.1), which forms the basis of our plausible sculpting technique, and we highlight our novel extensions to the model. We also briefly outline the GPU-based nonlinear solver (Section 3.2), which provides solves of the anatomical model at interactive framerates, and finally we describe the 3D face dataset and

pre-processing steps required to create our modeling system (Section 3.3).

3.1. Anatomical Local Face Model

Wu et al. [WBGB16] introduced an anatomical local face model (ALM) for robust facial performance capture from monocular videos. The idea is to divide the face into many local patches and model individual local deformation subspaces (one per patch). A key component is to separate the local *motion* of a patch from its local *deformation*. The model is regularized by anatomical constraints, specifically a skin thickness subspace using the underlying skull and mandible bones. An illustration of the model used in this work is given in Fig. 2. The face mesh has a resolution of 31361 vertices, the jaw and skull bones are tesselated with 2652 and 5375 vertices, respectively. While the bone resolution has no immediate impact, the face mesh resolution and patch layout directly affect the performance of the system. We use a patch layout with 165 patches, which provides a reasonable trade-off between local deformation and solve time.
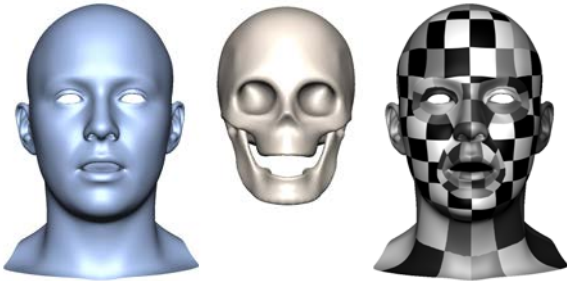

Figure 2: *The anatomical local model consists of a face divided into local patches and underlying bone geometry.*

In the case of Wu et al., the model was built from a sparse number of 3D facial expressions of the specific actor they wished to track. A new pose of the target subject's face can then be formed by solving for the parameters of the model given the following constraints: 2D projection constraints from an input video, anatomical constraints from the bony structures, patch overlap constraints that ensure neighboring patches align well, and a temporal consistency constraint. We refer to Wu et al. for detailed derivation of the model and fitting procedure, but for completeness we include their model parameters: the rigid local patch motions $\{M_i\}$, the local patch deformation coefficients $\{\alpha_i\}$, the rigid motion of skull M_s and relative rigid motion of the jaw Θ, as well as the optimization energy E given as

$$E = E_M + E_O + E_A + E_T, \tag{1}$$

where E_M, E_O, E_A, and E_T are the 2D projected motion, overlap, anatomical and temporal energy terms mentioned above.

Our scenario is to model the neutral face of a fictional subject, rather than predict a novel expression of a known subject. To this end, we construct an ALM model from a corpus of 3D scanned neutral faces from many different people (see Section 3.3 for a description of the dataset). In this case, the local patch deformation subspaces encode identity deformations rather than expression deformations. Note that each input shape to the ALM model now has different skull and jaw geometry. Our model is built with the average bone shapes, and it is important to note that we will allow changes to the skull and jaw shapes during modeling operations. To accomplish this, we define parametric skull and jaw shape models, and the corresponding shape coefficients are added as new parameters to the ALM model. Since we focus on the neutral expression only, the original jaw motion parameter Θ is removed from our model. To allow direct 3D control over the vertices, we introduce an additional energy term, E_P, for 3D positional constraints which is similar to the 2D motion constraint E_M in Wu et al. E_P is defined over the set of constrained vertices $\mathbf{V}$ as

$$E_P = \lambda_P \sum_{v \in V} \sum_{i \in \Omega(v)} w_v \| x_{v,i} - p_v \|^2, \tag{2}$$

where $x_{v,i}$ is the unknown 3D position of vertex v in patch i, $\Omega(v)$ is the set of patches which contain vertex v, and p_v is the user-defined constraint. λ_P is a weighting factor for the positional constraints as a whole, and w_v defines a per-vertex weight for the constraint.

In our implementation, we remove the temporal consistency term E_T as we are not working with temporal information. The final energy for our optimization is thus

$$E = E_M + E_P + E_O + E_A. \tag{3}$$

Note that in our setting, E_M is the energy term corresponding to projection constraints which are obtained from user interaction.

3.2. GPU Nonlinear Solver

The energy described above is minimized using the Gauss-Newton method. Wu et al. employed a custom implementation, however several off the shelf solvers are also readily available (e.g. the Ceres solver [AM10]). Unfortunately, most existing solvers are too slow for the interactive requirements of our system. One exception is the recently-proposed GPU-based solver of Fratarcangeli et al. [FBG*20], which is specifically designed for large-scale semi-sparse problems, and has been demonstrated on facial reconstruction using the ALM model. Therefore, it is a natural choice for us to employ this GPU solver in our application.

3.3. 3D Face Dataset

The ALM model used in this work is built from a dataset of registered 3D face scans with corresponding bone structures. The face geometries were acquired with the method of Beeler et al. [BBB*10], and a template mesh topology was semi-manually aligned using iterative laplacian mesh deformation [SCOL*04]. A template skull [BB14] and jaw [ZBGB19] were fitted to the resulting face shapes. The dataset contains 226 individual people, spanning gender, ethnicity, age and body mass index (BMI). Fig. 3 illustrates a small subset of the scanned subjects in common topology.

Since we focus on modeling the neutral identity of individuals, the dataset consists purely of neutral faces devoid of expression. It is worth noting, however, that our method can also be applied in the context of expression modeling, with the goal of creating a rigged character, provided that a similar expression dataset also exists.
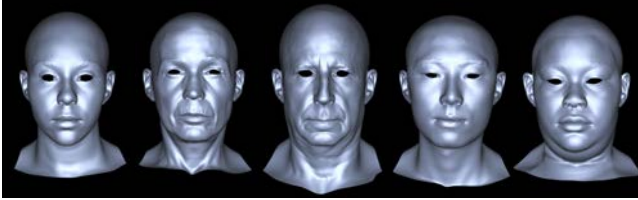

Figure 3: *Our method is data-driven, using a face model built from a dataset of 226 subjects spanning different ages, genders, ethnicities and BMI. A selection of dataset samples are illustrated here.*

3.3.1. Model Compression

Creating the ALM model from all 226 input shapes yields a large parameter-space which slows down solve times. We therefore compress the model to a tractable size. Specifically, we perform principal components analysis (PCA) on the deformation subspace of each patch and choose the top five eigenvectors as a low-dimensional representation of the deformation subspace. While for a global model, five eigenvectors would be an insufficient basis by far, for a local model this basis captures on average 70% of the shape variation, since each patch is compressed separately. We employ such aggressive compression to obtain a minimal basis for faster solve times. It is important to note that the deformation subspace of a patch is related one-to-one with the anatomical subspace. For patches that are anatomically constrained, each shape in the deformation subspace has a corresponding location on the bone surface, a normal direction, and skin thickness value. These three components form the anatomical constraints (see Wu et al. [WBGB16] for more details). When compressing the deformation subspace, we must also compress the anatomical subspace to the same dimensionality. To achieve this, we first express each of the five eigenvectors as a linear combination of the original 226 shapes. As this is a highly ambiguous problem, we use the L_1 norm to compute the weights, yielding a sparse solution. The resulting weights are then used to form a weighted combination of the original bone points, normals and skin thicknesses, providing an anatomical subspace that is now compatible with the corresponding shape subspace. With the compressed shape subspaces and corresponding updated anatomical subspaces, the ALM model is ready for interactive sculpting. Note that since the compression operation occurs separately per-patch, individual patch subspaces no longer have global semantic meaning. This is not an issue, however, since patch parameters will be solved individually during sculpting, and the vertex position overlap constraint E_O will ensure a smooth and consistent mesh result, despite the inconsistent semantics of the subspaces.

4. Interactive Face Sculpting

We now outline our method for interactive face sculpting, using the anatomical local model described in the previous section as a core component. We begin with a generic face, computed as the average of the face dataset our model is built on. The user can then perform a sequence of interactive edits, which are converted by the system to optimization constraints used to solve for ALM model parameters, specifically the rigid local patch motions and the local patch deformations. Note that we do not allow the rigid motion of the skull to change during sculpting. Given a set of solved model parameters, we reconstruct the corresponding 3D face mesh using the patch blending approach of Wu et al. [WBGB16], and present the result to the user for additional edits. By using the GPU-based non-linear solver described in Section 3.2, we achieve interactive performance. Due to the data-driven ALM model, the resulting face remains realistic even for a sparse number of input constraints. For example, with just two strokes to drag the sides of the nose outwards, the system will create a face with a realistic wide nose.

Our approach unifies the two major types of interactions in any modeling system, *direct manipulation* and *indirect semantic manipulation*, and introduces as a third control paradigm *anatomical manipulation*. In the following, we expand on how these methods of control are implemented as ALM constraints.

4.1. Direct Manipulation

In the context of 3D sculpting, direct edits through brush strokes is typically the mode of control most often used. Our method supports direct control by clicking and dragging surface points to desired locations (see Fig. 4). Technically speaking, if a user moves a vertex to a new position, this creates a single 3D positional constraint (Eq. 2) for the ALM solve. Multiple position constraints can be combined to yield the desired face surface. Alternatively, the user has the option to specify 2D projection constraints given the current view matrix, allowing the point to lie anywhere along the 3D ray instead of providing an exact 3D location.

To control the locality of the manipulation, our system allows to specify a region of influence for each edit. Vertices inside that region are allowed to move freely, while vertices outside the defined region are constrained to their current position using positional constraints. More specifically, we constrain only 5% of the vertices, which are chosen at random. Applying a gradient falloff to the positional constraints at the region boundary produces smooth edits. This is accomplished by defining an inner and outer sphere for the region of influence. Direct manipulation with a region of influence is illustrated in Fig. 4. First the region is specified, then a point is moved to create a constraint, and finally the face is updated. Note that the resulting deformation is immediately presented to the user and we split it in three steps (shown in Figure 4) for illustrative purposes.

4.2. Indirect Semantic Manipulation

The idea behind indirect face manipulation is to allow semantic control of the skin shape. For example, we construct controls over age, gender, ethnicity or BMI, in a data-driven way leveraging the
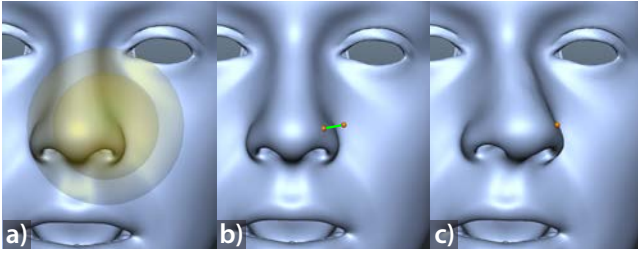
Figure 4: *Direct Manipulation - a) The user specifies a region of influence. b) The user directly moves a point on the face surface. c) The system solves the ALM model with the new constraint. Note this update happens at interactive framerates.*

Figure 5: *Indirect Semantic Manipulation - a) Global control allows to holistically modify characteristics, such as increasing BMI and Male qualities. b) This can be applied also within a region of influence, such as changing from a more European nose shape to more African American, without affecting the rest of the face.*

3D face dataset. Since semantics are user defined, we manually label the dataset with the desired properties. For each category, we compute the *characteristic differential* shape vector. These shape vectors provide a parametric control of the global shape along semantically meaningful dimensions. Dialing in a characteristic applies the corresponding differential to the current edited face, and then the modified vertex positions are used to create positional constraints for an ALM solve. Note that we cannot just use the parametrically modified mesh directly, as we must remain in the anatomical modeling paradigm to allow further sculpting edits. In practice, it is not necessary to create a positional constraint for each modified vertex as the parametric model is linear and locally smooth, so a random subset of 5% of the vertices is used as positional constraints for the ALM solve, yielding faster solve times. To compute the characteristic differential vector, for the numerical variables BMI and age we employ linear regression to compute the best fit line through our data samples, and then the differential vector is taken as the orientation of this line. For the categorical variable gender, we simply regress through the male and female group mean. For ethnicity, we first compute the mean per category and then define the differential as the difference between those means and the global mean of all subjects in the dataset. As with direct manipulation, indirect manipulation can be applied within a region of influence, defined the same way as in Section 4.1. Fig. 5 demonstrates semantic indirect control both globally and locally within a region of influence. When dialing in a semantic feature globally (i.e. without a region of influence), we also automatically modify the skull and jaw bones to match the semantic skin changes, using corresponding characteristic differentials for the bone geometry. Modifying the underlying anatomy is a new form of anatomical manipulation, which can also be performed independently as described next.

4.3. Anatomical Manipulation

We introduce anatomical manipulation as a novel and intuitive means of sculpting human faces. It combines the flavors of direct and indirect manipulation, since a user may directly (or indirectly) manipulate the bones, which in turn indirectly affects the skin surface due to the ALM model. Moving the jaw bone forward, for example, will cause the skin to naturally follow. In addition to direct manipulation of the bones, we also provide indirect control in a
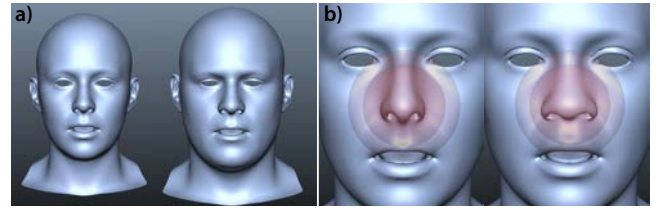
data-driven way, similar in spirit to indirect face control. This time, however, we opted to automatically parameterize the shape space of the bones in the dataset using PCA, applied on each bone independently. We found that the PCA modes provide a meaningful shape subspace, since they define large, linear modes of change. Manipulating the bones by changing the PCA weight coefficients yields new vertex positions and normals for the bone geometry, which in turn prescribes anatomical constraints for solving the ALM. As described in Wu et al. [WBGB16], these anatomical constraints take the form of surface position constraints provided by a given bone point with known normal direction and skin thickness. In our case, we assume the thickness does not change during sculpting, but the bone points and normals can be edited using the PCA subspace. As a result, modifying the underlying bones will indirectly modify the skin surface through the ALM model.

We demonstrate examples of anatomical manipulation in Fig. 6, where we first dial in a deformation mode on the skull which makes it shorter and wider, and then decrease two deformation modes of the jaw that correspond to width and overall scale. Changing the underlying anatomy results in interactive indirect control of the face in a natural way. Please note that, while we only experimented with this manipulation method in one direction (i.e. modifying the skin via the bones), changing the bone shape and pose via skin manipulation could be added as future work.

4.4. Baking

As edits are made, more and more constraints are generally added to the ALM solve. After some time, the constraints may begin to fight against each other, and new user edits might not have the desired effect. For this reason, we introduce the concept of baking the current result into the model, allowing to remove all existing constraints and continue editing with a 'blank slate' of constraints. Furthermore, it can be difficult to simultaneously perform direct, indirect and anatomical editing as the various different constraints may become invalid, thus we require to bake the current result when switching between editing modes.

Baking is the process of applying the current artist-provided edits to the ALM model in such a way that the resulting baked model with those constraints removed produces the same geometry as the unbaked model with the constraints applied. Technically speaking,
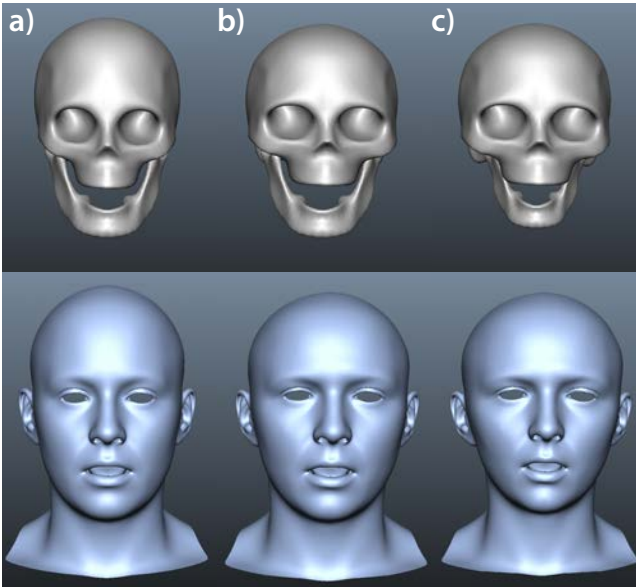
Figure 6: *Anatomical Manipulation - a) Starting from the average face and bone shapes, b) the skull can be made shorter and wider, then c) the jaw can be made smaller and narrower. The anatomical manipulation indirectly effects the shape of the face.*

we need to update the model such that without user-defined constraints, the optimization energy in Eq. 3 is zero without changing the face shape. Analyzing the different terms in the energy function, E_M and E_P will trivially resolve to zero as we will remove all the user-specified projection and position constraints. The overlap energy E_O can be made zero by replacing the mean shape of each patch with the corresponding local deformation of the current face shape, and zeroing out the deformation coefficients $\{\alpha_i\}$. We then also update the deformation subspace differentials to be the original subspace shape minus the new mean, such that the model provides the same subspace variation as before. For the anatomical energy, E_A, note that the underlying bones in the model are continuously updated to the current user-edited bone shapes, so all that remains is to compute a 3D offset for each anatomical constraint such that when evaluated with the current bone, the residual with respect to the baked shape is zero.

Baking can be performed as frequently as the user desires, from continuously baking after each edit to baking only once after finishing the current edit mode. We left the decision on baking frequency to the user rather than automatically baking after every edit, since baking incurs a small overhead in processing time. In practice, we found the most natural workflow to include baking after sufficiently completing edits to each local region of the face. Thanks to our baking procedure, users may perform edits in any order (direct, indirect, anatomical), and arbitrarily switch back and forth between editing modes.

4.5. User Interface

We provide a simple and intuitive interface for face modeling, as depicted in Fig. 7. For indirect and anatomical manipulation we provide linear sliders on the left side of the screen. Direct manipulation can be performed by editing the current face on the right side of the 3D viewport. Next to the face we show the current anatomical bones for better feedback when performing anatomical editing. The bones and the face are continuously in sync, in terms of shape as well as pose (as the face is oriented by the user, the bones will be oriented likewise). Changing the region of influence for editing operations is implemented with the mouse wheel, and visualized as two transparent spheres indicating the inner and outer region of influence (i.e. defining the gradient fall-off for the effect).
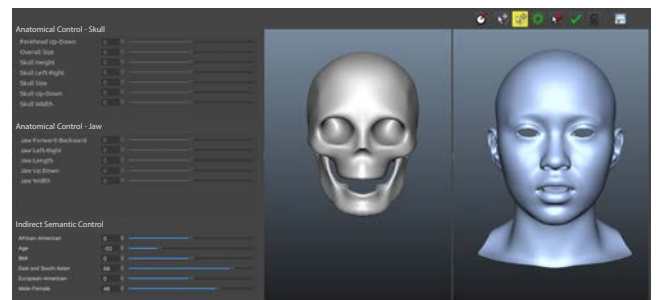

Figure 7: *User Interface - We provide a simple and intuitive GUI for face modeling, allowing direct control of the face surface as well as indirect semantic and anatomical control through sliders.*

Table 1 summarizes all the interactions in our user interface and how they form optimization constraints, as described in detail in Sections 4.1 - 4.3.

Interaction	Mode	Constraint
Vertex Dragging	Direct (Section 4.1)	3D Positional
Vertex Dragging	Direct (Section 4.1)	2D Projection
Skin Sliders	Indirect (Section 4.2)	3D Positional
Bone Dragging	Anatomical (Section 4.3)	Anatomical
Bone Sliders	Anatomical (Section 4.3)	Anatomical

Table 1: *Summary of interactions and corresponding constraints.*

In Section 5 we detail a study of novice users who are able to create interesting 3D digital faces—testifying to the simplicity and usability of our interface.

4.6. Implementation Details

4.6.1. System

All results were produced and the user study conducted on a system equipped with an Intel Core i7-7700K CPU @ 4.20GHz with 4 physical cores, 32 GB of RAM and a Nvidia GeForce RTX 2080 Ti graphics card.

4.6.2. Performance

The GPU based solver implementation computes 6.5 frames per second. While this already enables interaction, there is room for

improvement. In timings, data transfer between host (CPU) and device (GPU) takes a signification portion of the overall execution time and GPU utilization is below 25%. Since much of the data could be reused for subsequent solves, optimizing data transfer as well as improving GPU initialization could provide a significant speedup. Furthermore, to boost our framerate, we currently reduce the maximal number of iterations our solver may take in any given solve. While this improves the perceived interactivity, it negatively impacts the ratio between data transfer and computation time.

4.6.3. Weight Parameters

As described in Section 4.4, we enforce baking before switching editing modes. On top of the given motivation, this also allows us to control certain optimization parameters better. In particular, we can tune the weights for the different energy terms on a per mode basis. As an example, indirect modifications benefit from larger weights for overlap constraints to ensure smooth results. Direct modifications on the other hand are then overly restricted. See table Table 2 for the exact weights we use for each editing mode.

Mode	E_M	E_P	E_O	E_A
Direct Manipulation	10	30	0.8	500
Indirect Semantic Manipulation	0	20	2.2	1000
Anatomical Manipulation	0	0	0.85	50

Table 2: *The values we use for the weight parameters in the different editing modes. Note that we set weights to zero if they are not used in a given mode. Please refer to Section 3.1 for the definition of the energy terms.*

5. Results and User Study

We now present results of our interactive digital face sculpting method, starting with a demonstration of the proposed workflow given an example task, followed by an illustration of the variance our method can produce in a result gallery, and ultimately presenting a user study including both novice users and skilled artists.

5.1. Workflow

Fig. 8 illustrates a typical workflow in our system. The example task is to create a caucasian male, approximately student-aged, who appears content and attentive. Starting with the mean face (Fig. 8a), the user employs indirect face control to increase the characteristics of European American and male (Fig. 8b). Then using anatomical control, the user thins and scales down the jaw, while scaling up the back of the skull (Fig. 8c). Finally, the user employs direct manipulation with region of influence to close and widen the mouth (using 5 brush strokes), and raise and focus the eyes (using 8 brush strokes—Fig. 8d). The entire process took less than 5 minutes.

5.2. Result Gallery

To highlight the variety of plausible faces that can be quickly achieved with our method, we illustrate a gallery of results in Fig. 9. Each of these faces took only minutes to create by a novice user with no artistic background.
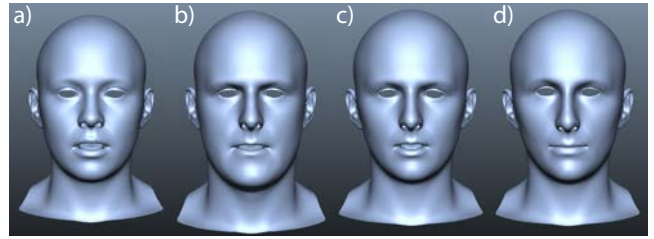

Figure 8: *Workflow - a) Starting mean mesh. b) Indirect face control to increase European American and male characteristics. c) Anatomical control to widen the skull and narrow the jaw. d) Direct control with region of influence to close and widen the mouth, and raise and focus the eyes.*

Figure 9: *Result Gallery - A variety of digital faces created with our system by a novice user.*

5.3. User Study

In order to evaluate our method, we performed the following user study. Six subjects with no digital modeling experience were given two sculpting tasks to accomplish using our system. The subjects consisted of three males and three females, between the ages of 23 and 31. All subjects use computers on a daily basis. The first task was to create a digital character with the following qualities: *female, European, young, pleasant, thin lips, small mouth*. The second task, arguably more challenging, was to create a character described as: *male, asian, middle-aged, weathered (e.g. tired of life), skinny*. Each participant was given a 5-minute tutorial of the application and was then allowed to experiment freely for another 5 minutes before given the tasks. The order of the two tasks was randomized for each participant. The resulting faces from the 6 study participants are shown in Fig. 10 (Subject 1 through Subject 6), and the respective time to completion is given for each result. On average, across all subjects and both tasks, it took approximately 9 minutes for novice users to create 3D faces that match the given descriptions. While some results contain minor artifacts and vary in quality, overall the sculpted faces successfully meet the given criteria. As a baseline, we asked a skilled artist to perform the same two
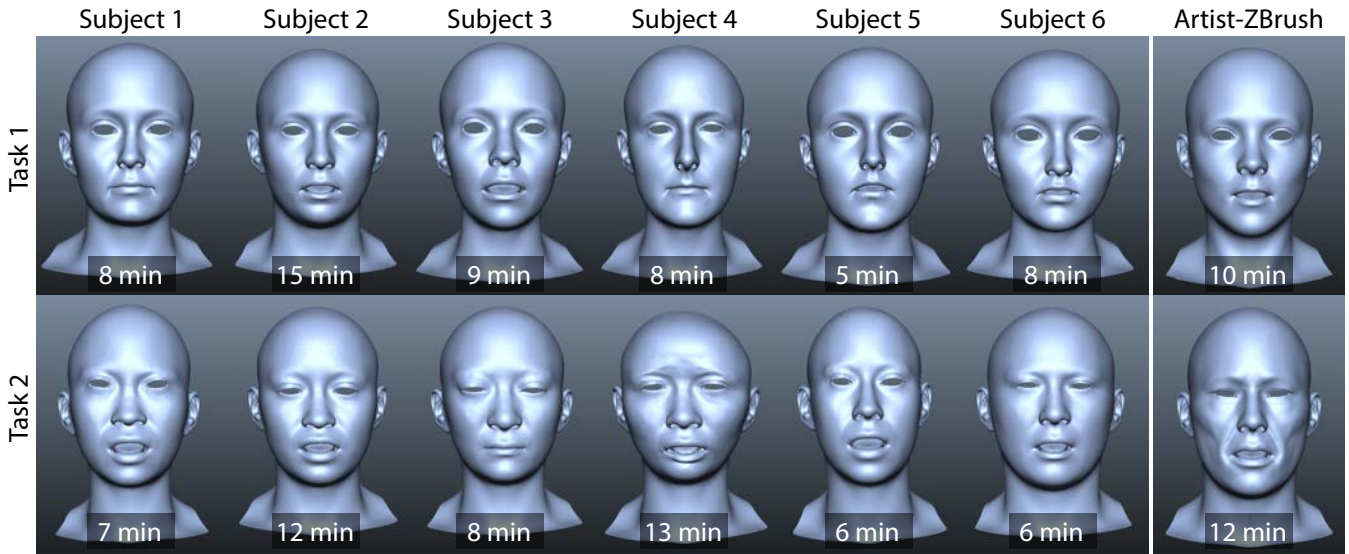
Figure 10: *User Study - The sculpting results from 6 novice users for two different tasks are illustrated along with the time it took to create each face. As a baseline, the results of a skilled artist using professional software are given on the right. Our method allows inexperienced users to create plausible digital faces that match a given criteria within the same amount of time as a professional artist.*

tasks in ZBrush, a free-form sculpting tool, starting with the same mean face mesh as the study participants. The results are given in Fig. 10 (right). While there is a noticeable improvement in quality from the skilled artist (in particular for the more challenging task 2), novice users were still able to create comparable face shapes in the same (or less) amount of time.

After the tasks were complete, the participants were asked to complete a short survey of four questions to better understand their experience with our system. Each question was answered on a 5-point Likert scale. The results of the survey are given in Table 3. Generally-speaking, participants were happy with the outcome of their digital face sculpting tasks and found the tool easy to learn as well as easy to use.

Question	Avg.	St.Dev.
The tool is easy to use	4.167	0.408
The tool is easy to learn	4.333	1.211
The tool allowed me to create what I wanted	3.667	0.516
I am happy with what I created with the tool	4.167	0.408

Table 3: *User Study - Questionnaire results for 4 questions, scored on a 5 point Likert scale (1 - strongly disagree, 2 - somewhat disagree, 3 - neither agree nor disagree, 4 - somewhat agree, 5 - strongly agree). In general, participants agree that our method is easy to learn and use, and were happy with the results they created.*

To the best of our knowledge, our method presents the first approach for interactive face sculpting based on a data-driven anatomical modeling paradigm. Our method allows novice users to create realistic 3D digital faces, and in the same or less time that a skilled artist requires with free-form sculpting software. Additionally, we received feedback from the artist after examining our method, that our tool would be very useful as a starting point for hero characters,

where the result would be exported to a free-form sculpting tool for finalizing fine-scale details. This could cut down on time spent in digital character modeling, even in professional production scenarios with skilled artists.

6. Conclusions and Future Work

In this work we propose a new interactive method for creating digital human faces using an anatomical modeling paradigm. At the core of our approach is an anatomical local face model, built from a dataset of 3D face scans with estimated bone geometry. We provide a tool for face manipulation that converts user edits to optimization constraints for the ALM model. We show how to implement traditional methods of manipulation in this framework, including both direct and indirect semantic face control, as well as a new form of manipulation—anatomical control, where the user can manipulate the underlying bone structures to indirectly modify the face surface. Our method is intuitive and easy to use for novice users, and the 3D face results remain in the realm of realistic anatomical faces thanks to our data-driven approach. The anatomical modeling paradigm presented in this work could also be employed in traditional video game character design packages.

In the future, we feel that incorporating additional constraints during user edits, for example left/right symmetry, would greatly improve the user experience. Additionally, this work only addressed the problem of creating facial geometry for the neutral expression of individuals—a complementary problem is to create or synthesize corresponding facial texture and reflectance maps, which we leave as future work. A simple extension of our method would be to swap the multi-identity compressed ALM model with an actor-specific ALM model built with a sparse set of facial expressions, and then our approach could analogously be used for

facial expression modeling. Finally, as a consequence of the mesh resolution we use to maintain interactive manipulation, the user is not able to model fine scale details such as wrinkles and blemishes. One avenue for enriching the results with high resolution geometry would be to synthesize a displacement or normal map, in tandem with the surface geometry. Such a *detail map* could then be applied on top of the face geometry in real-time during sculpting.

References

[ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: Reconstruction and parameterization from range scans. In *ACM SIGGRAPH 2003 Papers* (2003), ACM, pp. 587–594. 2

[AL10] ANJYO K., LEWIS J.: Direct manipulation blendshapes. *IEEE Computer Graphics and Applications 30*, 04 (jul 2010), 42–50. 3

[AM10] AGARWAL S., MIERLE K.: Ceres solver. http://ceres-solver.org, 2010. 4

[BB14] BEELER T., BRADLEY D.: Rigid stabilization of facial expressions. *ACM Trans. Graph. 33*, 4 (2014), 44. 4

[BBB*10] BEELER T., BICKEL B., BEARDSLEY P., SUMNER B., GROSS M.: High-quality single-shot capture of facial geometry. In *ACM Trans. Graph.* (2010), vol. 29, ACM, p. 40. 4

[BRP*18] BOOTH J., ROUSSOS A., PONNIAH A., DUNAWAY D., ZAFEIRIOU S.: Large scale 3d morphable models. *International Journal of Computer Vision 126*, 2 (2018), 233–254. 2

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *Computer Graphics and Interactive Techniques* (USA, 1999), SIGGRAPH '99, pp. 187–194. 2

[BWP13] BOUAZIZ S., WANG Y., PAULY M.: Online modeling for realtime facial animation. *ACM Trans. Graph. 32*, 4 (2013). 2

[CBE*15] CONG M., BAO M., E J. L., BHAT K. S., FEDKIW R.: Fully automatic generation of anatomical face simulation models. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2015), pp. 175–183. 3

[COL15] CETINASLAN O., ORVALHO V., LEWIS J.: Sketch-Based Controllers for Blendshape Facial Animation. In *EG 2015 - Short Papers* (2015), Bickel B., Ritschel T., (Eds.), The Eurographics Association. 3

[FBG*20] FRATARCANGELI M., BRADLEY D., GRUBER A., ZOSS G., BEELER T.: Fast Nonlinear Least Squares Optimization of Large-Scale Semi-Sparse Problems. *Computer Graphics Forum (Eurographics) 2*, 39 (2020). 2, 4

[Fra13] FRATARCANGELI M.: Position-based facial animation synthesis. *Computer Animation and Virtual Worlds 23*, 3-4 (2013), 457–466. 3

[GLP*19] GECER B., LATTAS A., PLOUMPIS S., DENG J., PAPAIOAN-NOU A., MOSCHOGLOU S., ZAFEIRIOU S.: Synthesizing Coupled 3D Face Modalities by Trunk-Branch Generative Adversarial Networks. *arXiv e-prints* (2019), 1–11. URL: http://arxiv.org/abs/1909.02215, arXiv:1909.02215. 3

[GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 2672–2680. 3

[IKKP17] ICHIM A.-E., KADLEČEK P., KAVAN L., PAULY M.: Phace: Physics-based face modeling and animation. *ACM Trans. Graph. 36*, 4 (2017). 3

[KHS03] KÄHLER K., HABER J., SEIDEL H.-P.: Reanimating the dead: Reconstruction of expressive faces from skull data. *ACM Trans. Graph. 22*, 3 (2003), 554–561. 3

[LBB*17] LI T., BOLKART T., BLACK M. J., LI H., ROMERO J.: Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph. 36*, 6 (2017). 2

[LCXS07] LAU M., CHAI J., XU Y. Q., SHUM H. Y.: Face poser: Interactive modeling of 3D facial expressions using model priors. *ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA* (2007), 161–170. 3

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 34*, 6 (Oct. 2015), 248:1–248:16. 2

[LWP10] LI H., WEISE T., PAULY M.: Example-based facial rigging. *ACM Trans. Graph. 29*, 4 (2010). 2

[MPN*19] MOSCHOGLOU S., PLOUMPIS S., NICOLAOU M., PA-PAIOANNOU A., ZAFEIRIOU S.: 3dfacegan: Adversarial nets for 3d face representation, generation, and translation, 2019. arXiv:1905.00307. 3

[NVW*13] NEUMANN T., VARANASI K., WENGER S., WACKER M., MAGNOR M., THEOBALT C.: Sparse localized deformation components. *ACM Trans. Graph. 32*, 6 (2013). 3

[PKA*09] PAYSAN P., KNOTHE R., AMBERG B., ROMDHANI S., VET-TER T.: A 3d face model for pose and illumination invariant face recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance* (Sep. 2009), pp. 296–301. 2

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Proc. SGP* (2004), pp. 175–184. 4

[SILN11] SEO J., IRVING G., LEWIS J. P., NOH J.: Compression and direct manipulation of complex blendshape models. In *SIGGRAPH Asia* (New York, NY, USA, 2011), SA '11, ACM. 3

[SNF05] SIFAKIS E., NEVEROV I., FEDKIW R.: Automatic determination of facial muscle activations from sparse motion capture marker data. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH 05, Association for Computing Machinery, pp. 417–425. 3

[TDlTM11] TENA J. R., DE LA TORRE F., MATTHEWS I.: Interactive region-based linear 3d face models. *ACM Trans. Graphics (Proc. SIG-GRAPH) 30*, 4 (2011), 76:1–76:10. 3

[TW90] TERZOPOULOS D., WATERS K.: Physically-based facial modelling, analysis, and animation. *The Journal of Visualization and Computer Animation 1*, 2 (1990), 73–80. 3

[TZN*15] THIES J., ZOLLHÖFER M., NIEUNDEFINEDNER M., VAL-GAERTS L., STAMMINGER M., THEOBALT C.: Real-time expression transfer for facial reenactment. *ACM Trans. Graph. 34*, 6 (2015). 2

[WBGB16] WU C., BRADLEY D., GROSS M., BEELER T.: An anatomically-constrained local deformation model for monocular face capture. *ACM Trans. Graph. 35*, 4 (2016), 115. 2, 3, 4, 5, 6

[YCHK15] YUMER M. E., CHAUDHURI S., HODGINS J. K., KARA L. B.: Semantic shape editing using deformation handles. *ACM Trans. Graph. 34*, 4 (2015). 3

[YK14] YUMER M. E., KARA L. B.: Co-constrained handles for deformation in shape collections. *ACM Trans. Graph. 33*, 6 (2014). 3

[ZBGB19] ZOSS G., BEELER T., GROSS M., BRADLEY D.: Accurate markerless jaw tracking for facial performance capture. *ACM Trans. Graph. 38*, 4 (2019), 1–8. 4