Fat Pad Cages for Facial Posing

Anonymous Author(s)

ABSTRACT

We introduce Fat Pad cages for posing facial meshes. It combines cage representation and facial anatomical elements, and enables users with limited or no artistic skill to quickly sketch realistic and believable facial expressions. The model relies on one or several cage(s) that deform(s) the facial mesh following the human fat pads map. To follow this map, mesh deformations are based on a new function that filters Green Coordinates using geodesic distances. It prevents global deformation and computes, for each vertex, an attenuation matrix based on pad shapes and geodesic distances to ensure smooth deformations at the pads’ borders. Specific borders such as lips, nostrils and eyelids are processed slightly differently to allow folding up and opening. Cages have been automatically created and adapted to fit any new unknown facial mesh. Thus, users can quickly create facial expressions without having to either individually sculpt facial shapes or to use generic sculpting tools leading to undesired facial artifacts. We show that non-experts can produce complex poses in only few interactions. To validate our approach, we present a user study comparing our Fat Pad cages to regular Green Coordinates. Results show that Fat Pad cages bring a significant improvement in reproducing existing facial expressions and are unanimously preferred.

KEYWORDS

Facial Posing, Fat Pads, Mesh Modeling, Cage Deformation, Human-Computer Interaction

1 INTRODUCTION

Facial expressions are key elements in the realistic characters creation process [OBP*12]. All of the hero characters of the most recent blockbusters have facial rigs composed of thousands of blend shapes. These blend shapes are a predominant and massively used technique to animate faces. They represent facial elementary displacements and are, most of the time, related to facial muscles [JTDP03]. Modeling them requires strong modeling and sculpting skills (like studio artists) and a lot of time (several weeks for a hero character). Today, posing faces can be achieved through linear piece-wise modeling approaches [TDITM11], physics-based approaches [IKKP17] or any free-form deformation (FFD) techniques [SCOA*04]. In the first case, it requires a lot of facial performances motion capture data. In the second, physics-based approaches require an high-end polyhedral anatomical representation of the human face to perfectly simulate bones, muscles and fat. And in the last case, FFD mesh deformers are too generic and do not take into account facial semantics. They can lead to severe uncanny artifacts for non-experts as a face is deformed as any other meshes. To sum up, posing facial meshes with all current mesh modeling approaches require today skills or data. They all come with a technical cost for non-experts and do not prevent artistic artifacts. These non-expert users may be professionals, like stage directors who want to sketch facial expressions on a tablet to give guidelines, or beginners, such as digital art school students.
Understanding of underlying anatomy to create realistic characters have been used for hundreds of years by drawers, sculptors and any art masters. The final shape of a facial expression is a complex combination of the skull, the muscles and the fat [Zar17]. Skin is almost negligible as it is soft and thin. It may rather be considered as a protective and aesthetic layer than a dynamic and active one. The skull gives the macroscopic shape of the face, and except for the jaws, it should not move. Actually, muscles are the cause of the action, and fat is the main visual element of the final shape (wrinkles, dimples, gaps and folds). Fat on the face is divided in vertices of the cage but also the normals to the faces of the cage. The computation time is similar to the MVC one. GC have several positive aspects compared to MVC or HC. They are better at maintaining the details on a surface. They do not generate negative coordinates, which allows the articulation of bone structures with concave cages without producing artifacts. This can be useful with the jaw. They tend to preserve the volume of an object. It enables the deformations to be more believable and for the model to go through the cage. And they allow extended deformation which enable to build a local cage, with a decreasing deformation outside the cage.

Cage generation is the process used to build a cage around a mesh (i.e. defining the cage topology), Jacobson et al. give three main rules to maintain during cage generation [JDKL14]. It needs to be low resolution so the amount of manual efforts in the mesh deformation process is reduced. It must fully and often tightly bind the enveloped model. It should respect the topology of the enveloped model for the manipulation to be intuitive enough.

Several types of cage generation processes can be found, all of them having interesting aspect to consider. In their paper, Yang et al. show an example of the use of template based cage generation type [YCSZ13]. They use a template model to automatically construct a muscle structure on any new model. The “nested cage” by Sachl [SVJ15], uses the character mesh to build more and more detailed cage mesh around the model. This way, the cage follows the model topology and surrounds it tightly. With his bounding shape based approaches, Xian proposes to generate oriented bounding boxes (OBB) for each mesh part and then registering the OBB together [XLG12]. This approach is useful for complex models with multiple articulations that can be divided in blocks. However, this appears to be hard to apply on faces due to the lack of articulation. The paper of Chen et al. gives an example of a skeleton based approach, where the adaptable cage is generated to the skeleton of the model and then automatically refined to reconstruct time-varying details more faithfully [CF14]. Linking the cage topology to the skeleton is an interesting idea but, unlike the body, facial movements are not led by the skeleton. Le et al.’s cage relies on the user’s inputs [LTW95]. The user has to cut the model in parts, and the slides set is used to construct an initial cage via the automated orientation optimization of cage cross sections and cage meshing. The cage is refined by fixing the coverage issue and providing a controllable tightness. Our target is a fully automatic approach to generate cages, with no user interaction.

### 2 RELATED WORK

We review most related literature on cages as well as works concerned with facial posing.

**Cage-based Deformation and Generation** - Cages are low polygon-count polyhedron with barely the same shape as the mesh. All vertices of the mesh (in the cage) are linear sums of the cage’s vertices multiplied by special weight functions (coordinates). Deforming the cage by manipulating its vertices induces smooth deformation of the mesh. This approach is suitable for interactive applications since the coordinates are pre-computed. Floater et al. [Flo03] are the first to propose a way to use cages with a random geometry with the Mean Values Coordinates (MVC). These coordinates, based on barycentric coordinates, allow smooth deformations of the polygons of a mesh in the cage. The drawbacks of MVC are the counter-intuitive results caused by negative coordinates and the loss of details during the deformations. Joshi et al. then proposed the Harmonic Coordinates (HC) that resolve the negative coordinates issues but require a long computation time [JMD*07]. Lipman et al. put forward the Green Coordinates (GC) that do not have negativity issues and preserve the details on a mesh [LLCO08]. These coordinates come from the Green Functions theory and take into account not only the position of the vertices of the cage but also the normals to the faces of the cage.

The computation time is similar to the MVC one. GC have several positive aspects compared to MVC or HC. They are better at maintaining the details on a surface. They do not generate negative coordinates, which allows the articulation of bone structures with concave cages without producing artifacts. This can be useful with the jaw. They tend to preserve the volume of an object. It enables the deformations to be more believable and for the model to go through the cage. And they allow extended deformation which enable to build a local cage, with a decreasing deformation outside the cage.
Blender 3. However, designing blend shapes is a tedious and time-consuming task and requires strong artistic skills. As facial expressions are complex combinations of multiple blend shapes, high-end facial rigs may contain hundreds of them. A character like Siren 4 contains more than a thousand of blend shapes.

To ease the design of facial expressions, Tena et al. [TDITM11] propose a region-based linear model, that uses PCA. This method separates the mesh in several models and the formulation restricts the model solutions to have semi-consistent boundaries while simultaneously enforcing user-defined constraints in a least squares sense. To build the area, they used a training set of pictures on different people and calculated the PCA for each. The idea to use model dislocation in several sub meshes is interesting, but the use of PCA to separate the model can lead to multiples artifacts and non-realistic results. And while the region-based linear model can be applied to multiple models, this method does not allow to model other expressions than the ones from the training set. In a similar way to deform faces, it is also possible to rely on eigenbases of the Laplacian to generate low-frequency bases as proposed by Bouaziz et al. [BWIP13] to track human face using RGB-D camera. An eigen decomposition of the face makes full sense when it is performed on a dynamic facial model (e.g. user on a video). It enables to get a decomposition close to a blend shape model. It appears to be inappropriate in our case as we only have one static facial mesh with no apriori knowledge.

Physical models may lead to realistic facial expressions. Water proposed a muscle model that uses muscle vectors and radial functions derived from linear and spherinder muscles to deform a skin mesh [Wat87]. Lee et al. emphasized on realistic modeling for facial animation and presented an algorithm that automatically constructs functional models of human heads, subjected to laser-scanned range and reflectance data [LTW95]. Contractile muscles at anatomically correct positions within a dynamic skin model are then inserted and rooted in an estimated skull structure with a hinged jaw. Kähler et al. [KHS01] proposed a model for muscle-based facial animation composed of three layers: a skin layer representing the epidermis and subcutaneous fatty tissue, a layer of muscles attached to the skull, and inserted into the skin, and the underlying bone structure, composed of immovable skull and rotating jaw. Actual muscle-based models are often too complex and require numerous computations. More recently, Ichim et al. have presented a novel physics-based approach to model faces [IKKP17]. As their approach is based on energy minimisation of a system simulating facial physical interactions, it cannot lead to undesired artifacts as a blend shapes rig can create but building such a full anatomical facial rig is a very tedious and complex task.

This analysis of the literature highlights a lack between the intuitive cage deformer tools, not used to pose faces, and dedicated professional tools, requiring time and strong artistic skills. Fat Pad cages can be filled to this gap by proposing a novel intuitive, interactive and accurate modeling approach usable by anyone (from beginner to expert) to model facial meshes.

### 3 OVERVIEW
Cages are an easy and user-friendly approach to deform a mesh, but they are not adapted to facial deformation. Two issues prevent them from being used to pose faces. First, the cage topology has an important influence on the mesh deformation. The design of the cage must be adapted to facial deformation. Second, a face has a particular deformation pattern due to the combination of the skull, the muscles and the fat. Cages are a generic tool that do not take into account these constraints. Deforming a facial mesh with current cage deformation techniques may lead to undesired artifacts and thus, place the result into the Uncanny Valley. We propose to limit and better localize the cage deformations within local areas (fat pads) of a facial mesh (see Fig. 2).

Figure 1 presents an overview of our pipeline. Here and all along the paper, we represent the input as a whole human head. Any new input mesh goes first through a retopologization process [SP04] to follow our template mesh topology (15k vertices and 30k triangles). Fat pads are defined according to anatomical data [Zar17]. A template fat pad map with handles’ positions per handle are specified once on the template mesh (Section 4.1). After the retopologization step, the attenuation matrix is computed for each pad (Section 4.2) based on their adapted shapes onto the unique facial morphology (Section 4.3). The cage is then automatically computed using, for the vertices, a specific extrapolation of the handles along the surface normal, and a Delaunay triangulation coupled with a convex hull extraction for the topology (Section 5). At the end, the fat pads and the cage are combined enabling to pose any facial expressions (Section 6).

### 4 FAT PADS
In-depth study of face design and modeling reveals that the distortions of a face is more defined by its fat than its bone structure [Zar17]. The face is a stack of bones (cranial box and jaw), muscles and fat. The fat is divided in several pads all around the face. The fat pads and their movements are the source of the wrinkles, dimples, gaps and folds of the face.

We propose to divide a facial mesh in smaller areas called “fat pads” (see Figure 2). On each pad, some vertices are manually labeled as handles. They are used in the next step (Section 5) to build the vertices of the cage, and their displacement induces the smooth

---

1. https://www.blender.org
3. Blender
4. Siren
deformation of the corresponding area. This method represents the
way facial muscles would deform the skin. The fat pads approach
limits the deformation, induced by the cage, to a local area. For in-
stance, one could deform a cheek without seeing the nose moving.

Green Coordinates (GC) [LLC08] are used here but the con-
cept is applicable to other coordinates system [JMD*07, NS13]. We
decided to use GC because they better preserve the details of the
mesh and they are faster to compute during the interaction than
HC. Another very important feature of GC is that the cage does
not need to contain entirely the mesh, as opposed to most of other
cage coordinates systems. The mesh deformation is thus computed
using GC. However, the vertices of the cage (i.e. the handles) only
affect the vertices of their own pad. To ensure a smooth and con-
tinuous deformation between the pads over the entire model, we
need to ensure that the deformation is attenuated at the border of
the pads. Hence, a weight between 0 and 1 smooths the GC. This
weight is associated to each vertex of the facial mesh according to
its position in a pad. While the vertices close to the handle move
normally, the ones close to the border area remain still or almost
still.

4.1 Template Map
Fat pads are designed following a reference book used in modeling
departments in VFX studios [Zar17]. As Cong et al. have proposed
for simulatable flesh and muscles [CBE*15], we use a template map
on which we manually painted fat pads. This operation has only
been performed once. The template fat pad map is then used for all
new input meshes. Figure 3 shows a subset of the fat pads we used.
Only the right and middle pads are shown on the figure, the left
ones being symmetrically identical to the right ones. Painted fat
can then be automatically adapted to others meshes thanks to the
retopologization process. If a character requires specific fat motion
behaviors, the template map is fully re-paintable and handles freely movable. In real life, fat pads are symmetric
on the face and most of the time do not overlap (there are some exceptions around the lips). The human face has smaller fat pads
or tissues in-between main fat pads. To avoid the modeling to be-
come too complex, our model do not take them into account. To
fill these gaps between the pads, we propose to overlap them as
shown on Figure 2. It avoids sharp borders and ensure smooth de-
fractions at the borders. A custom plugin in Autodesk Maya has
been developed for this purpose where fat pads can be painted on
a facial mesh, and where handles can be specified per pad.

4.2 Attenuation Matrix
A weight \( w_{v;i} \) is computed for each vertex \( v \) of a pad and for
each handle \( h \), in a way that the closer the vertex is to the border,
the higher the attenuation is. To avoid any border aberration, the
weight must decrease slowly to zero at the border. We propose to
use the following function that decreases according to the geodesic
distance between the vertex position and the border.

\[
w_{v;i} = \frac{(d(v, h) - d(i, h))^2}{d(i, h)^2}
\]  
(1)

with \( d \) the geodesic distance between two vertices, and \( i \) the vertex
located at the intersection of the border of the pad and the line \( v/h \)

\[
V' = V + (V_{ge} - V)W_h
\]  
(2)

(see Figure 5 and 6). The use of a quadratic radial basis function
(RBF) kernel appears to be an appropriate candidate as it smoothly
goes from 1 (at the handle) to 0 (at the boundaries), thus restricting
the influence on the fat pads. We use this function to compute the
weight matrix \( W_h \) for each vertex of each fat pad. The fact that this
type of kernel has a zero derivative at the boundaries is not an issue
in our case as we set the boundaries’ vertices fixed and unmovable
(weight is 0).

Specific Borders’ Vertices. Even if most of fat pads borders
are supposed to remain still, there are some specific cases where
it should not. The lips are a good example where fat pads must be
processed differently. Indeed, it has to follow the deformation of
the fat pads to allow folding up, opening, and closing. Such bor-
ders can be specified in our model as an exception not to set the
attenuation at the border to zero (see the red stroke on Figure 4).
This is applied to the nostrils and the eyelids as well. In our posing
process, we have set the weight for this specific pad borders’ ver-
tices to 1. This way it moves freely according to the handle motion
as if it was not linked to another adjacent pad.

The final positions \( V' \) of all vertices after manipulating a handle
\( h \) is then defined as:

\[
V' = V + (V_{ge} - V)W_h
\]  
(2)
with $V$ the current vertices’ positions, $V_{gc}$ the GC positions and $W_h$ the weight matrix associated to the handle $h$.

### 4.3 Pad Shape Processing

For each fat pad the weight matrix $W_h$ is computed. This has to be performed for any new facial mesh as fat pads have different shapes. $d(i, h)$, the distance between the handle and the border passing through vertex $i$, is a key value to compute the matrix weight $W_h$. It is different for each vertex and each handle. To compute $d(i, h)$, we set a plane $P$, defined by the line $L$ between the handle and the vertex and the normal $N$ on the mesh at the handle (see Figure 6 - left).

![Figure 6: Left - Given an handle and a vertex, the plane $P$ intersects the pad’s border (red sphere). Right - left nostril fat pad has a concave shape.](image)

$P$ may intersect the border at several points, especially if the fat pad is non-convex (see Figure 6 - right). Figure 5 shows the three cases $A$, $B$ and $C$ on a non-convex fat pad with vertices $v_A$, $v_B$ and $v_C$, and the handle $h$. The plane $P$ for the three cases is the same, and therefore the list of intersection points $I = [i_A, i_B, i_C, i_F]$. The first way to remove some intersection candidates is to filter according to the direction: $(\hat{h}, \hat{v}) \leq 0$ (case A). Then to filter according to the distance. A vertex $v_n$ must be between $h$ and $i_n$: $d(h, v) < d(h, i)$ (cases B and C). The minimum distance is kept $i_{selected} = \text{idx}(d(v, i))$. The weight matrix is computed offline. We use Surazhsky’s algorithm to compute the geodesic distances [SSK’05].

### 5 CAGE CONSTRUCTION

Cages have been selected as deformation technique for several reasons. First, the shape preservation property of the cage combined with GC highly reduces undesired artifacts while preserving face volume. Second, the intuitiveness, simplicity of usage and direct manipulation given by cages suit our needs for non-experts. There is no parameter to set, no action radius to define, no modeling tool to pick-up. And last but not least, cages ensure smooth deformation on encapsulated meshes.

The cage is built using the position of the handles defined for each fat pad. To get the cage topology, we compute a Delaunay triangulation on all the handles. It generates a tetrahedral mesh from which we extract the outer surface (convex hull) by only keeping faces shared once by tetrahedrons. This insures an homogeneous connection with almost all the nearest neighbors for each handle.

As we want to model facial poses, we need the face to be dynamic and the rest (scalp, neck, ears etc.) to remain static. Thus, we do not want the cage to entirely encapsulate the head mesh but only the face. As underlined by Lipman et al. [LLCO08], a GC cage does not need to contain entirely the mesh, as opposed to most of other cage coordinates systems. This enables to build local cages. Several conditions are thus required: 1/ the vertices at the border of the cage need to be fixed, 2/ the cage must be scaled so that the edges of the triangulation do not intersect the model, and 3/ the cage needs to be completely closed.

- **Fixed vertices.** To respect condition 1/ cage borders’ vertices are duplicated, scaled and then fixed (see right Figure 8). As cage vertices are based on pads’ handles, fixing the initial borders’ vertices do not make sense as it strongly limits the interaction with the pads. The duplication prevents this issue.

- **Scaled cage.** To respect condition 2/, cage vertices are moved away from the mesh along the surface normal to ensure no intersection between the mesh and the cage. First, we apply a uniform scaling to all the cage vertices. A scaling of 3 for the upper part of the face and 6 for the lower part due to smaller and tighter number of handles on the jaws. Then, based on their position on the face, a specific non-uniform adjustment scaling is applied to the vertices. Keeping the uniform scaling would have misaligned some handles with their specific part of the face and disturbed the user. For example, assuming that the face has been aligned to the template face, nose handles are only scaled according to $z$-axis (towards the image plane) to keep them in the middle of the face.

- **Closed cage.** To respect condition 3/, two new vertices are positioned at the back of the head of the model (see left Figure 8).

![Figure 7: Example of a vertex $v_A$ in a fat pad with a handle $h$ (case A). $d(i, h)$ is the geodesic distance between $h$ and $i_A$, the intersection between the pad’s border and $h\hat{v}_A$. In a case of a non-convex fat pad, multiple intersection points may exists for a handle $h$ (cases B and C).](image)

Figure 7 shows results of automatic cage creation on seven different cases. Our solution enable to fit all types of faces from skinny (G) to large (E), with larger lips (D) or smaller nose (F). From the side view, all cage vertices remain visible and accessible.

This would lead however to uncanny results around the mouth. Indeed, GC are meant to affect all the vertices within the cage. For instance if the upper lip handle is moved up to open the mouth, and the lower lip handle is moved down, vertices of the lower lip fat
Figure 7: Examples of cages adaptation to seven different facial morphologies.

Figure 8: The cage is closed using two additional vertices in the back of the head. These additional vertices are not displayed to the user during the interaction.

pad are moved relatively to the new position of those of the upper lip. The expected behavior is that they should be independent. To tackle this issue, two cages are built, one with the upper part of the head and one with the lower part as shown on Figure 8.

6 RESULTS

We present some posing results obtained through our Fat Pad model. The application has been implemented in Python 2.7 with the Mayavi library\(^5\) for the 3D visualization and Qt\(^6\) for the user interface (see Figure 9). The full cage is not displayed to the user, only the parts with the movable handles (i.e. not the ones behind the mesh, added to close the cage). The mesh is deformed in real-time when a handle is moved. When a handle is selected, the associated fat pad is colored in light blue to show the user the impacted area. The center of the deformation (actual position of the handle on the fat pad) is also highlighted with a red sphere.

Facial Poses. Figure 15 presents posing results, obtained by a non-expert user on seven different facial meshes, compared to a professional facial modeling artist (top row). The artist has used generic mesh deformers [SCOA04] whereas the non-expert has used our Fat Pad cage. We have obtained the seven meshes through a digital double creation pipeline\([DGO]^{19}\). The seven meshes (from A to G) share the same mesh topology as the reference one (top row). This enables to transfer the template fat pads from the reference character. The cages are also automatically computed based on the new positions of the pad handles due to facial morphology variations between people. The left column shows the input facial meshes of three females (A, D, F and G) and three males (B, C and E). The three adjacent columns show the result of the facial poses we modeled. We selected fundamental action units (AU), namely AU-2 (Outer Brow Raiser), AU-9 (Nose Wrinkler) and AU-12 (Lip Corner Puller). They are involved in many facial human expressions (e.g. Happiness, Sadness, Disgust, Surprise, Fear etc.). For each modeling, deformation amplitude is highlighted by a heat map visualization. It shows the difference between the initial mesh and the final modeling. Dark blue is for pure static regions where red is for the regions that move the most. We can notice how close to the ground-truth the deformations are. It also emphasizes that our new Fat Pad cage prevents undesired mesh deformation due to global deformation (recomputation of the whole mesh when a cage vertex is moved) and ensures smooth deformation at the borders of

\(^5\)https://pypi.org/project/mayavi/
\(^6\)https://www.qt.io/
the pads. With our cage, most of the mesh remains static, deformations only occur in the specific locations the user wanted to model.

Unrealistic Character. Figure 10 shows results of posing faces using our Fat Pad model on three unrealistic characters. We use the same human template fat pad map as defined previously, thus the facial layout remains human (e.g. two eyes, one nose and one mouth). For each unrealistic character, from left to right, we present the neutral face, its computed cage in front and side view, and a random expression obtained using our model. We can notice that our Fat Pad model is fully adaptable to any other facial morphology as soon as it respects the human facial layout. There were some initial normal artifacts on some meshes (e.g. the eyelids of the last row character) which may affect the results perception.

7 EVALUATION
A user study has been conducted to evaluate our approach. Two cages systems are compared, the classical GC cage and the Fat Pad cage. Our hypothesis is that the user would be more effective with the Fat Pad cage to create a facial expression.

7.1 Experimental Data
The template facial mesh, presented above, is a part of a rig containing blend shapes designed by professional CG artists. In this study, two expressions were used as reference to reproduce: "sadness" and "disgust" (see Figure 11). Sadness is meant to be an easy expression to copy (not too different from the neutral one) while disgust is more complicated with an open mouth and a wrinkled nose. Two cages were set up to manipulate the neutral mesh: the Fat Pad cage and the GC cage. They both have the same topology from the participant’s point of view (although the Fat Pad cage is composed of two sub-cages). For each of the two cages, participants had thus to realize the two expressions, leading to four tasks. We have chosen to compare our method to a single GC cage since this is the way it is used in the literature, and because a two-parts GC cage led to strong artifacts around the mouth. A "local" GC cage has an influence on the whole mesh.

7.2 Measures
Objective and subjective measures were taken from the experiment. Objective measures were the time to perform a task, the number of movements (manipulation of the handles), the number of "undos", and the root mean square (RMS) of the Hausdorff distances between each participant’s result and the reference model [ASEE02]. Subjective measures were the following questionnaire evaluated on a 5-point Likert scale, from 1 (totally disagree) to 5 (totally agree):

- Q1 The result is similar to the model
- Q2 The task was easy
- Q3 The manipulation was easy
- Q4 I am happy with my result

These questions aimed at understanding how participants evaluate their work and the use of the cages. At the end of the study, an informal interview was conducted to discuss their overall preference between the two cages.

7.3 Protocol
Participants had two screens. The one in front of them displayed the UI they had to interact with (see Figure 9). Reference meshes were displayed on a second screen on their right. Before the study, participants were asked a set of questions regarding their expertise in 3D viewing and modeling. The study started with an exploration phase, where they get used to the user interface, for approximately...
10 minutes. Then the four tasks were randomly presented. During a task participants were invited to talk in a sort of "thinking out loud" process. They stopped a task whenever they were satisfied with the result, and the meshes were saved for further analysis. The duration of the study was about 40 minutes.

7.4 Results
17 participants, 15 males and 2 females, took part in the study (age $\bar{x} = 33.75 \sigma = 12.920$). They were all used to 3D visualization and navigation ($\bar{x} = 3.65 \sigma = 1.069$ on a Likert scale 1 to 5) but had no expertise in 3D modeling ($\bar{x} = 1.75 \sigma = 0.958$). Examples of produced expressions can be seen on Figure 12. Results produced with the Fat Pad cage are visually similar to those created with the GC cage for the sadness expression, both close to the reference (Figure 11). However, for the disgust expression, the Fat Pad cage is closer to the reference because of the possibility to open the mouth.

Figure 12: Examples of results. From left to right: Sadness with the GC cage, Sadness with the Fat Pad cage, Disgust with the GC cage, Disgust with the Fat Pad cage.

Figure 13 shows the mean of the RMS of the Hausdorff distance between the participants’ meshes and their reference. Meshes created with the Fat Pad cages are significantly closer to the reference than those created with the GC cages (Wilcoxon signed rank tests: $p = 0.0021$ for sadness, and $p = 3.05e^{-05}$ for disgust). However, no significant differences were found for the time, the number of moves and of undos. Participants did not spend more time or made more mistakes with Fat Pad cages compared to GC cages.

The answers to the questionnaire are depicted on Figure 14. In the case of the sadness task, we did not observe statistical differences between the two cages ($p > 0.05$). It seems that, for a simple task, the performance with the Fat Pad cages is not perceived differently from the one with the GC cages. This is in line with the visual analysis (Figures 12). However, in the case of the disgust task, all assertions were better rated for the Fat Pad cages ($Q1: p = 0.001$, $Q2: p = 0.001$, $Q3: p = 0.03$, and $Q4: p = 0.002$). This preference is also visible from the produced meshes, and was confirmed by the post-test interviews.

7.5 Discussion
The opinion observed from the questionnaire let us think that the expression “sadness” was equivalently appreciated by the participants whether it was done with the Fat Pad cage or the GC cage. While sadness being the easiest expression, participants were unfamiliar to blend shapes and animation, and thus do not always capture all the subtleties of a facial expression. Some did not notice the frown up or the changes in the eyes, and they mainly focused on the mouth. The potential of the Fat Pad cages for these subtle movements was not fully exploited here.

However, results of the questionnaire show a significant difference of preference between Fat Pad and GC on the disgust expression. This may be explained by the fact that the deformation in disgust is important. Hence participants needed more precision, which is hard to achieve with the GC cage. The Fat Pad cage only allowed a nose wrinkle or to open the mouth. All participants were frustrated when they faced the mouth opener in GC cage, which is actually impossible to make and essential for posing this expression. Some participants, when they could not open the mouth, stopped the task right away. Some decided to focus on the cheeks, the frown or other attributes of the expression.

The impossibility to open the mouth was also one of the reasons why the Hausdorff distance is very high for the task GC and disgust. Nevertheless, the distance is also more important for GC with sadness. This supposes that the Fat Pad cage is a more efficient and accurate tool than the GC cage. During the post-test interviews, 14/17 participants reported preferring the Fat Pad cage. The precision, the possibility to open the mouth and the blue area...
feedback (fat pad areas) were really appreciated. Two participants had no preference between the two cages and one selected the GC cage.

All the participants noticed that the interface was easy to manipulate and to get used to. The participants quickly understood how the interface worked, although they were sometimes surprised by the deformation of the mesh. It was too exaggerated or opposed to what they were expecting. This feedback was reported more often with the GC cage as the deformation is applied on the whole mesh.

Ten participants complained about handles selection. When the mesh is turned sideways (see Figure 8), the selection of a handle might be tricky as two handles may overlapped. Six people proposed to add an automatic symmetry feature as they had trouble reproducing the same deformation for the other half of the face. Five people said it was difficult with the Fat Pad cage to exactly model the lips corners for the disgust expression. It can be explained by the fact that the corner pad is at the border of the two cages.

8 CONCLUSIONS AND PERSPECTIVES

We presented Fat Pad cages for facial posing. A first combination between cage-based deformation and facial anatomical model. The paper has described three main contributions: the Fat Pads concept enabling an interactive mesh deformation that respects facial anatomical constraints, a new automatic way of creating personalized cages for any facial mesh, and a user study validating the high interest and preference of Fat Pad cages. The proposed filter function, based on fat pad shapes and geodesic distances, to adapt the coordinate system appears to be more suitable to pose faces than Green Coordinates. It prevents global deformation and ensures smooth deformation at the borders of the pads. The generated cages closely fits the shape of the mesh and can be considered as an adaptive extension or extrusion of the head.

The user study confirmed the interest of our approach and also provided valuable insight to improve our system. The cage could be adapted to a specific task. Depending on the precision required, the number of handles could be increased. It would be interesting to allow the user to dynamically add handles to a fat pad. For instance, to design a subtle expression around the mouth, having more handles could be increased. It would be interesting to apply the symmetry feature as they had trouble reproducing the same deformation for the other half of the face. Five people said it was difficult with the Fat Pad cage to exactly model the lips corners for the disgust expression. It can be explained by the fact that the corner pad is at the border of the two cages.

REFERENCES


[SCOA04] SORKINE O., COHEN-OR D., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. 179–188.


Figure 15: Results of modeling three primary facial poses using our Fat Pad cage on seven different facial meshes. The references (top) are made by a professional artist. The heat map shows the difference between the initial mesh and the final modeling (from dark blue (static) to red).