Interactive 3d caricature from harmonic exaggeration

THOMAS LEWINER[1], THALES VIEIRA[2],
DIMAS MARTINEZ[2], ADELAILSON PEIXOTO[2], VINÍCIUS MELLO[3] AND LUIZ VELHO[4]

[1] Department of Matematics — Pontifícia Universidade Católica — Rio de Janeiro — Brazil
[2] Institute of Mathematics — Universidade Federal de Alagoas — Maceió — Brazil
[3] Institute of Mathematics — Universidade Federal da Bahia — Salvador — Brazil
[4] Visgraf Laboratory — IMPA — Rio de Janeiro — Brazil

Abstract. A common variant of caricature relies on exaggerating characteristics of a shape that differs from a reference template, usually the distinctive traits of a human portrait. This work introduces a caricature tool that interactively emphasizes the differences between two three-dimensional meshes. They are represented in the manifold harmonic basis of the shape to be caricatured, providing intrinsic controls on the deformation and its scales. It further provides a smooth localization scheme for the deformation. This lets the user edit the caricature part by part, combining different settings and models of exaggeration, all expressed in terms of harmonic filter. This formulation also allows for interactivity, rendering the resulting 3d shape in real time.
Keywords: ***Caricature. Manifold Harmonics. Geometry Processing. GPU. 3d Faces. Shape Modeling.***

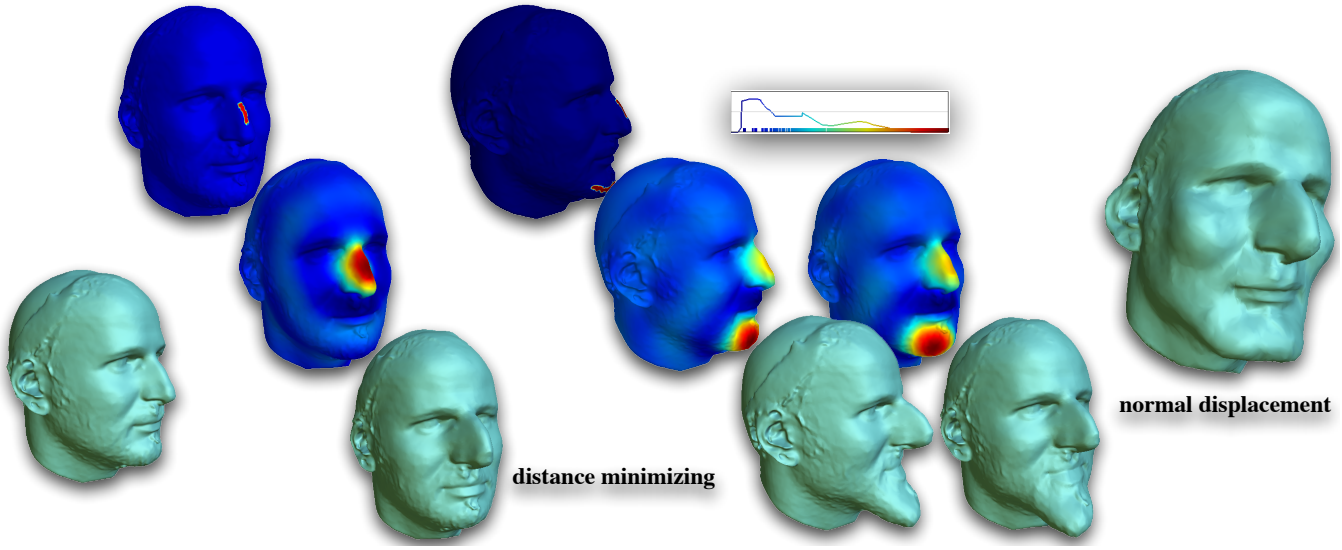

Figure 1: *Progressive caricature of a scanned head: the user paints the regions to be caricatured, and chooses the amplitude (here $\mu = -6$), scale selection as curve and filter model to use for each part. The differences are obtained by registration with Figure 8 as template.*

1 Introduction

Caricature is an illustration technique that exaggerates specific characteristic traits in a portrait of a human subject. Its main goal is to reveal the essence of a person by emphasizing particular aspects that visually identifies the individual. In this way, some features are magnified while other features are attenuated creating a non-realistic personalized impression of the subject [13].

In the western culture, caricature has a long tradition, way back to the Renaissance. Early examples can be found in the works of Leonardo da Vinci. Subsequently, other great artists such as Honoré Daumier specialized in this form of expression. Nowadays, caricatures are present in many types of media, ranging from newspapers and magazines to film and television. It has been traditionally used in political cartoons and then became a powerful means of entertainment in general. Many public figures, such as politicians and movie stars are associated with their caricatured depiction.

Preprint MAT. 13/10, communicated on December 10^{th}, 2010 to the Department of Mathematics, Pontifícia Universidade Católica — Rio de Janeiro, Brazil. The corresponding work was published in the proceedings of Shape Modeling International, Computers & Graphics 35(3), 2011..

As a practice, caricature can be considered an art form, not only because the production of an effective caricature requires skill and talent, but mainly because it essentially depends on an interpretation of the reality. In that sense, each caricaturist has a particular style that marks his/her work.

The social importance and appeal of caricature motivates the investigation of this topic in Computer Graphics. In its own way, it configures a remarkably rich area of research because it can be seen both as modeling and as a visualization problem, where perceptual and semantic issues play a fundamental role.

The main challenge in this area is to develop models that are able to sensibly take into account subjective parameters and to implement systems that allow simple intuitive expression. In this perspective, this work introduces an *interactive* system to create and model three-dimensional caricatures by exaggerating the harmonic differences from a template.

Historically, the research in computer-based caricature has its origins in the master's thesis of Susan Brennan in 1982 [6]. Since then, the area experiences significant development as reflected by the large number of publications devoted to this subject. Despite of those advances, the basic problems are far from being completely solved and still motivate intense research efforts.

Related Work Computer-based caricature methods can be broadly subdivided into three main categories depending on the principles adopted to model the problem: *template-based*, *extrapolation*, and *style learning* methods. These methods produce caricatures automatically or semi-automatically, but most of them rely on user interaction.

Template-based methods employ a reference facial shape, which contains specific features that can be emphasized or de-emphasized to different levels in order to produce a caricature. In general, the shape is defined geometrically and warping techniques are used to deform the template geometry interactively. Most of such systems work in two dimensions directly with photographs or with illustrations of a face [1, 8, 12] or parametric three dimensional shapes [11, 20]. The proposed techniques work directly on 3d shapes.

Extrapolation methods assume that a caricature exaggerates the traits distinguishing the face from the normal one. They resort to an average face that serves as the basis for extrapolating specific features. This kind of techniques works by amplifying the difference of the input face from the mean face. The previously mentioned seminal system of Brennan [6] was based on such principles, also known by illustrators [22]. In recent years, several systems of this type have been proposed [5, 14, 27]. A common way to model the face in such systems is through principal component analysis (PCA) that provides a representation in which the mean face is explicitly defined. These systems also specify exaggeration rules that allow the user to control the caricature effects [9, 21, 30]. The present work formulates the differences in harmonic space, which performs at interactive rates on general shapes.

Style-Learning methods, instead of modeling the caricature process by itself, attempt to recreate the mechanisms used by caricature artists. This is typically done using statistical inference techniques that construct a probabilistic model from examples [10, 17, 18, 19, 25], capturing the style of a specific caricaturist.

Contribution In this work, we propose to use spectral representation of the differences between a template and the 3d shape to be caricatured. We build on the Manifold Harmonics geometry processing [28, 24, 23], since it provides a very intuitive framework for mesh edition through a reduced set of parameters. In particular it admits a direct GPU implementation [16] that performs at interactive rate.

We derive the extrapolation principle within the harmonic space of the shape to be caricatured. We propose three different harmonic deformation filters, extending usual 3d caricatures beyond human faces, e.g. using animal shapes. We also use the harmonic basis to provide interactive controls to tune the caricature: a localization control, which specifies which parts of the face should be caricatured; and a scale control to exaggerate a subset of the harmonics for the caricature.

Our formulations adapt to a GPU implementation that correctly computes the per-vertex normals. This allows a WYSIWYG interface that gives feedback on the user's control in real time. Moreover, a partial result can be quickly used in place of the initial shape, letting the user caricature each feature with different settings (Figure 1).

2 Basics of Manifold Harmonics

In this section, we quickly review the basics of manifold harmonics filtering [28, 23].

(a) Laplace harmonics

On a discrete mesh M with vertex set V, manifold harmonics provide a linear basis H_k for the space of discrete scalar functions $f : V \to \mathbb{R}$ defined at the mesh vertices. This basis is built from the eigenvectors of a discrete Laplace operator, H_k being interpreted as the k^{th} fundamental oscillation mode of the mesh. Furthermore, the basis is sorted by increasing frequency, the first eigenvectors corresponding to large scale deformation and the last ones interpreted as fine details or noise.

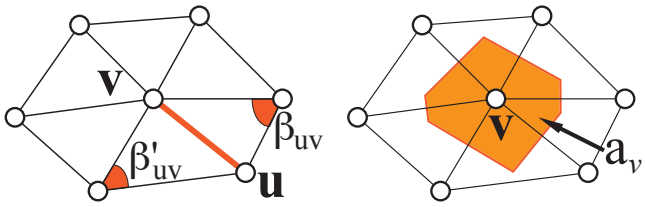

Figure 2: *Geometric elements for the discrete Laplace operator.*

In order to construct H_k, Vallet and Lévy use the Laplace-De Rham operator derived from Discrete Exterior Calculus, which is given by an $N \times N$ matrix Δ, where $N = \#V$ is the number of vertices of the mesh. Its coefficients Δ_{uv} are

zero if vertices u and v are different and not adjacent, and otherwise:

$$\Delta_{uv} = -\frac{cot\,(\beta_{uv}) + cot\,(\beta'_{uv})}{\sqrt{a_u \cdot a_v}}\ , \quad \Delta_{uu} = -\sum_v \Delta_{uv}\ ,$$

where a_v is the area of the circumcentric dual of vertex v, and angles β_{uv}, β'_{uv} are opposed to edge uv (Figure 2).

(b) Harmonic transform

The above definition of the discrete Laplace operator is symmetric, guaranteeing the existence of real-valued orthogonal eigenvectors $H_k : V \to \mathbb{R}$, satisfying $\Delta H_k(v) = \lambda_k H_k$ and, using Kronecher's δ notation, for all vertices u,v and all frequencies k,l, we have:

$$\sum_{v\in V} H_k(v)\cdot H_l(v) = \delta_{k,l}, \quad \sum_{k=0}^{N-1} a_v \cdot H_k(u)\cdot H_k(v) = \delta_{u,v}.$$

Since $(H_k, 0 \le k < N)$ is a basis, any scalar function $f : V \to \mathbb{R}$ has coordinates on this basis, which we write $\widetilde{f_k}$. With the orthogonality properties, we obtain those coordinates by simple projections [28]:

$$f(v) = \sum_{k=0}^{N-1} H_k(v)\cdot \widetilde{f_k}, \quad \widetilde{f_k} = \sum_{u\in V} a_u \cdot H_k(u)\cdot f(u)\ . \quad (1)$$

(c) Scalar Filtering

Using the analogy with Fourier analysis, the square root of the eigenvalue $\sqrt{\lambda_k}$ is interpreted as the frequency of harmonic H_k. GIven a scalar signal on the mesh $f : V \to \mathbb{R}$, the amplitudes $\widetilde{f_k}$ of each of its frequencies of can be filtered by amplifying each of them by $\varphi_k \in \mathbb{R}$. The filtered signal ${}^{\varphi}f$ is then given by:

$${}^{\varphi}f(v) = \sum_{k=0}^{N-1} H_k(v)\cdot\left(\varphi_k \cdot \widetilde{f_k}\right)\ .$$

As observed in the original work [28], the high frequencies (i.e. for $k > n$ with $n \ll N$) are expensive to compute, increase the complexity of the filter control, and do not provide significant modeling power at the global scale. We thus use a single factor φ_d to amplify all those high frequencies:

$$\begin{aligned}{}^{\varphi}f(v) &= \sum_{k=0}^{n-1} H_k(v)\cdot\left(\varphi_k \cdot \widetilde{f_k}\right) + \varphi_d \cdot \left(\sum_{k=n}^{N-1} H_k(v)\cdot \widetilde{f_k}\right)\\ &= \sum_{k=0}^{n-1} H_k(v)\cdot\left(\varphi_k \cdot \widetilde{f_k}\right) + \varphi_d \cdot df(v)\ . \quad (2)\end{aligned}$$

The residual $df(v)$ of f at each vertex is computed at preprocessing using the complementary expression: $df(v) = f(v) - \sum_{k=0}^{n-1} H_k(v)\cdot\widetilde{f_k}$. This way, the last eigenvectors of the basis are never used, and only the restricted basis $(H_k, 0 \le k < n)$ needs to be computed.

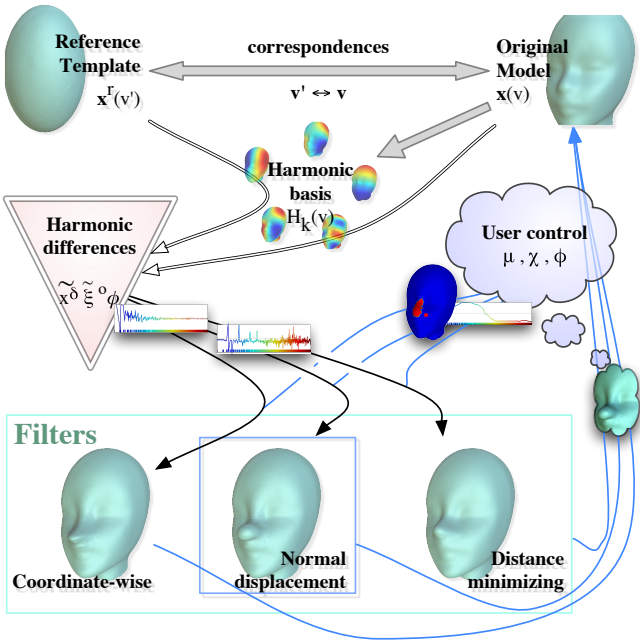

Figure 3: *Workflow of our interactive 3d caricature system: during preprocessing, the correspondences between the shape and the template and the harmonic decomposition of the shape are computed, and then the harmonic representations of their differences. The user then interacts by selecting morphing, localization and scale controls μ, χ, φ, visualizing the results of those controls in realtime. The user then validates one of the three harmonic exaggeration models, and its resulting caricature is quickly reused in place of the new shape.*

3 Interactive 3D Caricature Overview

We propose a caricature system in the extrapolation category, which starts with a reference 3d *template* (e.g., a normal face) and a 3d *shape* (e.g., the face to be caricatured), both represented as triangular meshes (Figure 4). We first compute, for each vertex v of the shape, a corresponding point $\mathbf{x}^r(v)$ in the template (section 6). We decompose their differences in the frequency domain, using the harmonic basis of the shape, since we want to preserve and exaggerate its features, and not those of the template. We propose three different representations of the differences between the shape and the template: coordinate-wise $\widetilde{\mathbf{x}^{\delta}}$, as a normal displacement $\widetilde{\xi}$ or as a distance minimizing filter ϕ, which are described in the next section.

The caricature is created by the user through several controls (section 5): a morphing parameter μ, which amplifies the differences; a localization function $\chi(v)$, which specifies the region of the shape to be deformed; and a scale control Φ_k, which selects and eventually amplifies the frequencies used for the caricature. The localization function $\chi(v)$ serves as a blending between the shape and the deformed region, and is obtained from the characteristic function of that region through filtering, also using the harmonics basis.

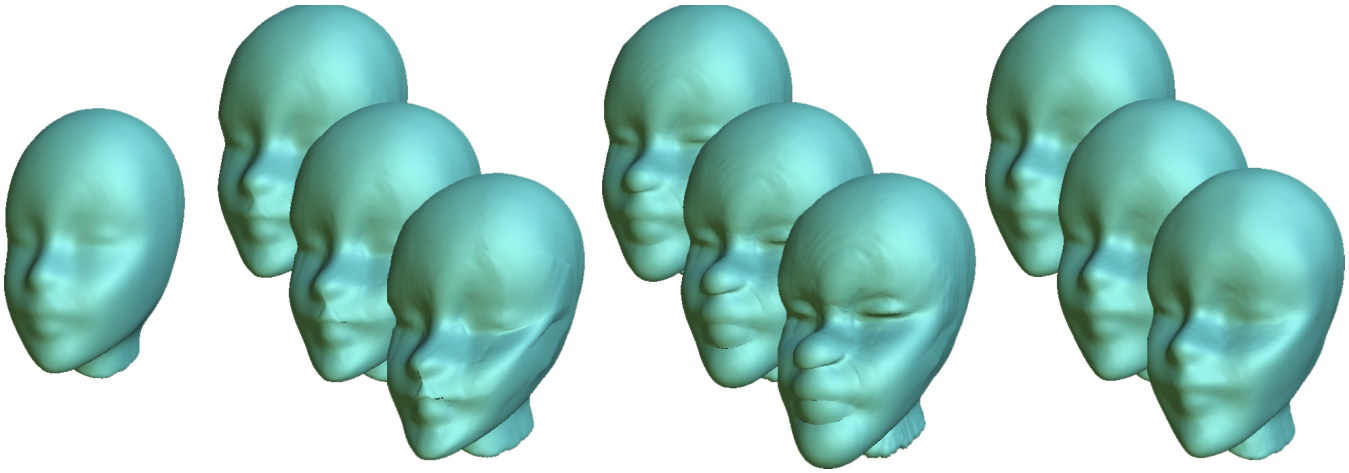

Figure 5: *Caricatures obtained with an ellipsoid as reference template, varying morphing parameter μ from -0.3 (top), -0.4 (middle) to -0.6 (bottom), for the coordinate-wise filter (left), normal displacement filter (center) and distance minimizing filter (right).*

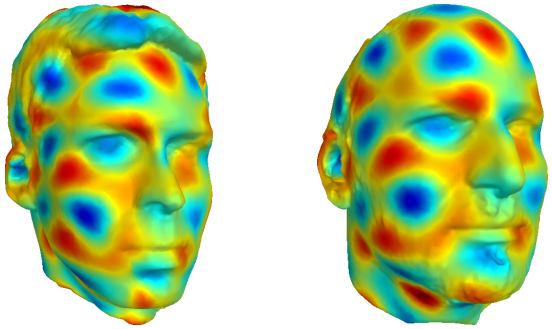

Figure 4: *Vertex-wise correspondences between a reference template (left) and the original shape of Figure 1 to be caricatured (right), colored according to the value of H_{170} on each vertex.*

The computation of the caricature for each representation is performed on the GPU, with an exact normal computation (section 6), allowing the display of the three resulting caricatures in real time with high quality. Finally, an optimized reprocessing allows using the result of one caricature in place of the shape, caricaturing each part independently with different parameters [2], in a layer-by-layer manner. The schematic representation of the whole process is illustrated in Figure 3.

4 Harmonic Exaggeration

In this section, we propose three harmonic representations for the differences between the shape and the template. We denote by $\mathbf{x}(v) = (x(v), y(v), z(v)) \in \mathbb{R}^3$ the position of vertex v of the shape, and $\mathbf{x}^r(v)$ the corresponding point in the template. The k-th harmonic of the shape is denoted $H_k : V \to \mathbb{R}$.

Given a morphing parameter μ, and a representation $\widetilde{\mathbf{X}}_k$ of the differences, we compute the coordinates ${}_\mu\mathbf{x}(v)$ of our exaggerated shape using the generic form:

$$ {}_\mu\mathbf{x}(v) = \mathbf{x}(v) + \mu \cdot \left(\sum_k H_k(v) \cdot \widetilde{\mathbf{X}}_k \right) . \qquad (3) $$

This ensures that for $\mu = 0$, ${}_0\mathbf{x}(v) = \mathbf{x}(v)$, and that for $\mu = 1$, $\mathbf{x}^r(v)$ is approximated by ${}_1\mathbf{x}(v)$. The caricature is obtained for $\mu < 0$, exaggerating away from the template (Figure 5).

(a) Coordinate-wise filter

The simplest approach looks for a strict morphing: ${}_1\mathbf{x}(v) = \mathbf{x}^r(v)$, which translates in our framework by $\mathbf{X}(v) \equiv \mathbf{x}^\delta(v) = \mathbf{x}^r(v) - \mathbf{x}(v)$. The coordinate-wise exaggeration is written in coordinates as:

$$ {}^c_\mu\mathbf{x}(v) = \mathbf{x}(v) + \mu \cdot \left(\sum_{k=0}^{n-1} H_k(v) \cdot \widetilde{\mathbf{x}^\delta}_k + \mathbf{dx}^\delta(v) \right) \qquad (4) $$

This ensures that for $\mu = 0$, ${}^c_0\mathbf{x} = \mathbf{x}$ and for $\mu = 1$, ${}^c_1\mathbf{x} = \mathbf{x}^r$. Without further control, this corresponds to linear morphing as obtained by direct correspondences [15].

Observe that there is no single filter φ that achieves mapping the three scalar signals $x(v)$, $y(v)$ and $z(v)$ of the shape to the template through Equation (2). This is only possible in general with three different filters φ^x_k, φ^y_k and φ^z_k, which are defined here by:

$$ \left.\begin{array}{l} \varphi^x_k = 1 + \mu \cdot \frac{\widetilde{x^\delta}_k}{\widetilde{x}_k} \\ \varphi^y_k = 1 + \mu \cdot \frac{\widetilde{y^\delta}_k}{\widetilde{y}_k} \\ \varphi^z_k = 1 + \mu \cdot \frac{\widetilde{z^\delta}_k}{\widetilde{z}_k} \end{array}\right\} \Longrightarrow {}_\mu\,\mathbf{x}(v) = \begin{bmatrix} x_{\varphi_x}(v) \\ y_{\varphi_y}(v) \\ z_{\varphi_z}(v) \end{bmatrix} . $$

Such triple filtering allows mapping any mesh to any mesh with the same connectivity, and does not use much of the intrinsic geometry of the shape (Figure 5).

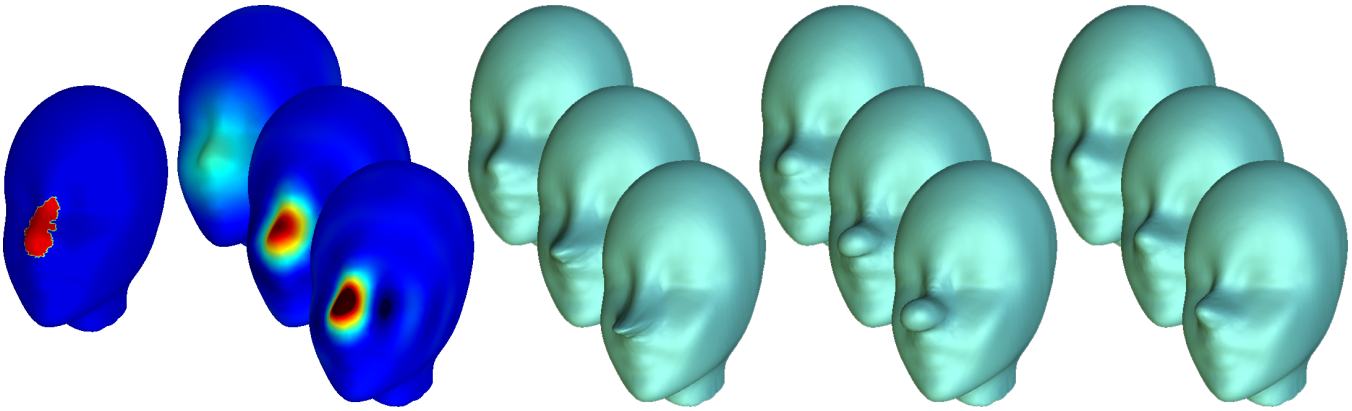

Figure 6: *Localization control: by a simple pick on the mesh (left), the user sketches a region to deform χ_{01}. This region is smoothed by cutting the high frequencies $k \geq n\chi$ of its characteristic function χ_{01}, here with $n\chi = 32$ for the top row, 96 for the middle one and 192 for the lowest row, out of $n = 512$ computed frequencies. The results of the three filters with $\mu = -0.8$, in the same order as Figure 5.*

(b) Normal displacement filter

In order to model the shape to template differences through a single scalar signal, we must give up the exact morphing ${}_1\mathbf{x}(v) \approx \mathbf{x}^r(v)$. We propose to approximate the difference $\mathbf{x}^\delta(v)$ by its projection $\xi(v)$ along the shape normal $\mathbf{n}(v) \in \mathbb{S}^2$ at vertex v: $\xi(v) = \langle\, \mathbf{x}^\delta(v) \mid \mathbf{n}(v) \,\rangle$. This normal displacement $\xi : V \to \mathbb{R}$ is a scalar signal, directly representable in the low and high frequencies of the shape by $\widetilde{\xi}_k, d\xi(v)$ through Equation (2). Interpreting $\mathbf{X}(v) \cong \xi(v) \cdot \mathbf{n}(v)$, the normal displacement exaggeration is derived from Equation (3):

$$ {}^n_\mu\mathbf{x}(v) = \mathbf{x}(v) + \mu \cdot \left(\sum_{k=0}^{n-1} H_k(v) \cdot \widetilde{\xi}_k + d\xi(v) \right) \cdot \mathbf{n}(v) \quad (5) $$

On perfect conditions [7], $\mathbf{x}^r(v) = \mathbf{x}(v) + \xi(v) \cdot \mathbf{n}(v)$, guaranteeing the morphing. The specificity of those conditions means that this filter restricts the deformation, preserving more of the original geometry of the shape (Figure 5). However, this formulation generates auto-intersections in the caricature for large $|\mu|$.

(c) Distance minimizing filter

Our third filter models the caricature directly as a spectral deformation [24], looking for a single scalar filter φ, φ_d that, equally applied to each of the three coordinate signals $x(v)$, $y(v)$ and $z(v)$ would best approximate the template. Using a 2-norm, pair-wise distance for approximation measure, this error is expressed as $\sum_{v \in V} \| {}^\varphi\mathbf{x}(v) - \mathbf{x}^r(v) \|^2$, where the optimization variables φ, φ_d appear in ${}^\varphi\mathbf{x} = \sum_{k=0}^{n-1} H_k(v) \cdot (\varphi_k \cdot \widetilde{\mathbf{x}}_k) + \varphi_d \cdot \mathbf{dx}(v)$. Subtracting 1 to each φ_k and to φ_d is equivalent to replacing $\mathbf{x}^r$ by $\mathbf{x}^\delta$ (see appendix):

$$ {}^o\phi, {}^o\phi_d = \operatorname{argmin}_{\varphi,\varphi_d} \frac{1}{2} \sum_{v \in V} \| {}^\varphi\mathbf{x}(v) - \mathbf{x}^\delta(v) \|^2 $$

This minimization actually simplifies due to the orthogonality of the harmonic basis, and its solution is explicit:

$$ {}^o\phi_k = \frac{\left\langle \widetilde{\mathbf{x}}_k \mid \widetilde{\mathbf{x}^\delta}_k \right\rangle}{\|\widetilde{\mathbf{x}}_k\|^2}, \quad {}^o\phi_d = \frac{\sum_{v \in V} \left\langle \mathbf{dx}(v) \mid \mathbf{dx}^\delta(v) \right\rangle}{\sum_{v \in V} \|\mathbf{dx}(v)\|^2} . $$

This optimal filter is then exaggerated by morphing parameter μ using Equation (3):

$$ {}^o_\mu\mathbf{x}(v) = \mathbf{x}(v) + \mu \cdot \sum_{k=0}^{n-1} H_k(v) \cdot {}^o\phi_k \cdot \widetilde{\mathbf{x}}_k + \mu \cdot {}^o\phi_d \cdot \mathbf{dx}(v). \quad (6) $$

This filter uses few extrinsic geometry of the shape, and thus further restricts the deformation possibilities, for small $|\mu|$ (Figure 5). In positive terms, it better preserves the intrinsic geometry of the shape even for large $|\mu|$.

5 Localization and Scale Controls

The above filters are already controlled by the morphing parameter μ, with the generic formulation of the caricatured coordinates ${}_\mu\mathbf{x}(v) = \mathbf{x}(v) + \mu \cdot \left(\sum_k H_k(v) \cdot \widetilde{\mathbf{X}}_k \right)$, with $\widetilde{\mathbf{X}}_k$ according to the chosen filter (Equations (4), (5) and (6)). We enhance the user control by letting her/him specify which part of the shape (e.g. nose, chick) should be deformed, and how much each scale (i.e. frequency) should contribute to the deformation. Those controls are injected in the generic form of ${}_\mu\mathbf{x}$.

(a) Localization control

The user specifies the region to be deformed simply by painting on the shape (Figure 6). The vertices picked by the pointer defines a characteristic function on the mesh: $\chi_{01} : V \to \mathbb{R}$. This function is typically 1 for the painted vertices and 0 otherwise, but intensity values lower than 1 may be specified before picking. The deformation is then restricted to the painted region by: ${}_\mu\mathbf{x}(v) = \mathbf{x}(v) + \mu \cdot \chi_{01}(v) \cdot \left(\sum_k H_k(v) \cdot \widetilde{\mathbf{X}}_k + \mathbf{dX}(v) \right)$.

If restricting the deformation directly to the painted region, the blending by χ_{01} would lead to cracks. We decompose χ_{01} to obtain a smoother localization function χ. Since we already computed the harmonics decomposition on the mesh, we obtain a surface Gaussian smoothing χ interact-

ively by a low-pass filter on χ_{01}:

$$\chi(v) = \sum_{k=0}^{n\chi-1} H_k(v) \cdot \widetilde{\chi_{01}}_k \,.$$

The cutoff frequency $n\chi$ is specified by the user. The caricature is then smoothly localized by:

$${}_\mu\mathbf{x}(v) = \mathbf{x}(v) + \mu \cdot \chi_(v) \cdot \left(\sum_k H_k(v) \cdot \widetilde{\mathbf{X}}_k + \mathbf{dX}(v) \right) .$$

The user can caricature independently several regions by using result on one region as the new shape, and specify a new region on that shape.

(b) Scale control

Our caricature system offers controls to exaggerate rather fine details or larger scale differences, by restricting or amplifying the difference representation to certain frequencies. This is simply done through an independent filter Φ_k, Φ_d (Figure 7):

$${}_\mu\mathbf{x}(v) = \mathbf{x}(v) + \mu\, \chi(v) \left(\sum_k H_k(v)\, \Phi_k\, \widetilde{\mathbf{X}}_k + \Phi_d\, \mathbf{dX}(v) \right) .$$

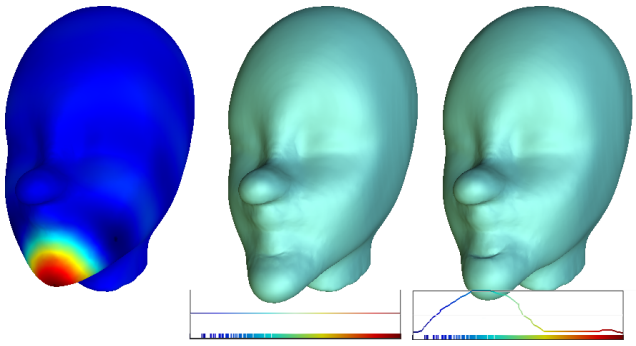

Figure 7: *The result of Figure 6 with $n\chi = 96$ and the normal displacement filter is used in place of the original shape to continue the caricaturing on the chin (left). The frequencies can be used equally (middle) or selecting and amplifying some of them through Φ, here amplifying intermediate frequencies (right) to mark the chin wrinkle.*

Combining with the morphing, localization and scale control, we get the final formulations for each filter:

- coordinate-wise filter (Equation (4)):

$${}^c_\mu\mathbf{x}(v) = \mathbf{x}(v) + \mu\, \chi(v) \left(\sum_{k=0}^{n-1} H_k(v)\, \Phi_k\, \widetilde{\mathbf{x}^\delta}_k + \Phi_d\, \mathbf{dx}^\delta(v) \right) .$$

- normal displacement filter (Equation (5)):

$${}^n_\mu\mathbf{x}(v) = \mathbf{x}(v) + \mu\, \chi(v) \left(\sum_{k=0}^{n-1} H_k(v)\, \Phi_k\, \widetilde{\xi}_k + \Phi_d\, d\xi(v) \right) \mathbf{n}(v).$$

- distance minimizing filter (Equation (6)):

$${}^o_\mu\mathbf{x}(v) = \mathbf{x}(v) + \mu\, \chi(v) \left(\sum_{k=0}^{n-1} H_k(v)\, \Phi_k\, {}^o\phi_k\, \widetilde{\mathbf{x}}_k + \Phi_d\, {}^o\phi_d \mathbf{dx}(v) \right) .$$

6 Implementation

The above formulations allow for interactive implementation, offering to the user an immediate return from the chosen setting of each control. This is achieved through a GPU implementation of the filters, leaving to the CPU the low-pass filter for the localization control χ, which is very fast since $n\chi$ is small. The picking for the scale control Φ and the definition of χ_{01} is also handled in CPU. Finally, to correctly illuminate the caricatured shapes, we propose a GLSL program to compute the vertex normals. This geometry shader is not specific to our filters and would apply to any triangular mesh.

(a) Shape / template correspondences

The correspondences computation is a challenging problem, mainly because geometries may differ considerably between the template and the shape. In addition, our filters presume that meshes coordinates are represented in a common global coordinate system, requiring a previous registration step.

A first option for the correspondences relies on semi-automatic cross-parameterization [15]. A set of matching vertices is obtained by user interaction, and a common, coarse base mesh is constructed on those vertices. The connectivity of the shape is then mapped onto the template through the parameterizations obtained on each element of the base mesh. Then, we align the shape to the template through a rigid-body transformation, computed by minimizing a *point-to-point* error metric between a set of correspondences. We use the correspondences obtained by the cross-parameterization, and a fast registration based on singular value decomposition [3] (Figures 10, 11 and 14).

Since the parameterization step is closely related to the registration step, as the quality of both results is dependent on the quality of the correspondences the reverse approach is feasible. A robust registration actually defines (or may help improving) the correspondences between the shape and the template. We illustrate this approach by using a single ICP [4] with automatic pre-alignment on some of the results (Figures 1 and 8).

As a final step of the caricature, the user may magnify all the previously edited exaggerations at once. A simple way to obtain such effect is to consider as template a simplified or scaled down version of the shape (Figures 1 and 8). The vertices of the versions of the mesh then naturally correspond.

(b) GPU feeding and normal computation

We opted for GLSL as programming language for its portability, and follow previous GPU Manifold Harmonic filter implementation [16]. It uses a fragment shader for the summation of Equation (2) and a render-to-vertex-buffer copy to obtain ${}_\mu\mathbf{x}$ for each filter. The only difference resides in the larger number of textures: $\widetilde{\mathbf{x}^\delta}$, $\mathbf{dx}^\delta$, $\widetilde{\xi}$, $d\xi$, $\mathbf{n}$ and ϕ, complementing the basis H_k, original position $\widetilde{\mathbf{x}}$, $\mathbf{dx}$ and the controls μ, χ and Φ.

The corresponding work was published in the proceedings of Shape Modeling International, Computers & Graphics 35(3), 2011..

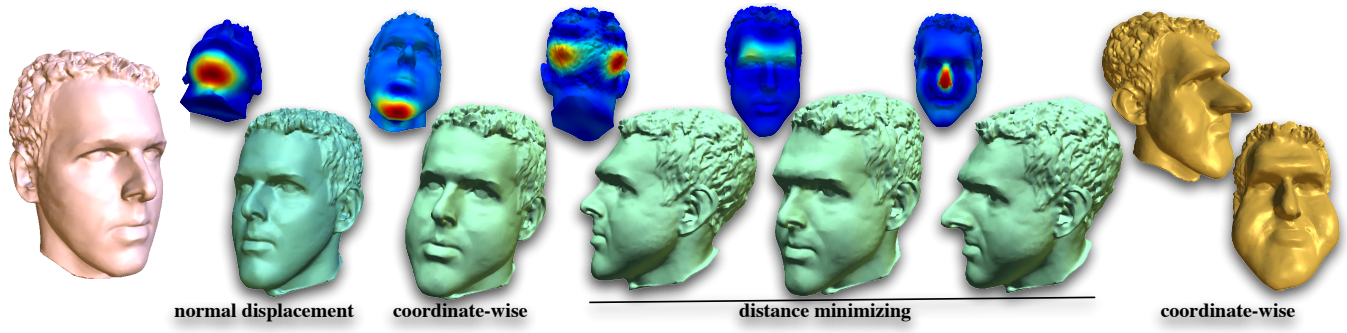

Figure 8: *Progressive caricature of a face model registered to the face of Figure 5. The last step uses a scaled down model as template.*

```
uniform float w,h ;                                  // buffer dimensions
uniform float max ;                                  // maximal area
varying in float id[], d^-1[];                       // vertex id and inverse degree
void main()
    vec3 e0 = gl_PositionIn[1].xyz - gl_PositionIn[0].xyz ;
    vec3 e1 = gl_PositionIn[2].xyz - gl_PositionIn[0].xyz ;
    vec3 nT = max * cross(e0, e1) + vec3(0.5,0.5,0.5) ;
    for (int i=0; i< gl_VerticesIn; i++)
        gl_FrontColor.rgb = d^-1[i] * nT ;
        gl_Position.x = 2.0 * mod( id[i], w ) / w - 1.0 ;
        gl_Position.y = 2.0 * floor( id[i]/w ) / h - 1.0 ;
        EmitVertex() ;
        EndPrimitive() ;
```

Algorithm 9: *Geometry shader for the vertex normal computation.*

We introduce a vertex normal computation using the same GLSL framework. We render the triangle mesh once with the updated vertices coordinates, and a fixed texture containing the vertex id $i(v)$ and the inverse of its degree $d^{-1}(v)$. The geometry shader (Algorithm 9) computes the triangle normal $\mathbf{n}_T$ and, for each triangle vertex v, emits a point at the 2d position in the normal array corresponding to its id $i(v)$, with a color encoding $d^{-1}(v)\ \mathbf{n}_T$. Using a $1/1$ blending function performs the final sum, the frame buffer ends up storing, for each vertex, the average of the normals of its adjacent faces. This sum is easily improved by weighting each normal by the area of the triangle, i.e. avoiding to normalize the cross product. The normal is color coded with a small workaround to the normalization. The normal is finally used after a render-to-normal-buffer copy.

(c) Fast use of a result as a new shape

The versatility of the proposed controls may lead the user to prefer one filter for some part of the caricature and different filters or control values for other parts. This is possible by using a partial caricature result in place of the original shape, in a layer-by-layer process. In particular, this permits caricaturing one feature at a time, as recommended by professional [2]. The first CPU preprocessing computes the template/shape vertex-wise correspondences, the manifold harmonics basis and the filter elements $\widetilde{\mathbf{x}^\delta}$, $\mathbf{dx}^\delta$, $\widetilde{\xi}$, $d\xi$ and $\not{\phi}$. To reduce the computations, we keep the correspondences and harmonic basis, preserving the original features of the shape. Only the filter elements are updated and re-sent to the GPU.

model	fig.	template	verts $\#V$	H_k *secs*	$\widetilde{\mathbf{X}}$ *secs*	user *fps*
Face 1	1,4	Face 3	49k	429	4.4	8.6
Simple	5-7	Ellipsoid	50k	422	3.8	8.5
Face 2	8	Simple	33k	217	2.5	8.8
Lion	10	Beethoven	30k	171	1.1	12.1
Face 3	11	Dog	36k	191	1.7	10.6
Planck	12	Egea	49k	402	3.9	9.0
Egea	13	Planck	31k	183	2.4	9.6

Table 1: *Performance tests on a $2.8GHz$ processor with a GeForce GT 9600M with $512MB$ of RAM. The interaction speed includes the model display, picking, the three filters and the vertex normal update. It is measured in frame per second (fps), while the pre-computation times are expressed in seconds.*

7 Results

We experiment our interactive caricaturing tool, letting the user choose the filters that were more expressive to her/him. We first observe that each of the three filters was used, with a preference on the normal displacement filter for smooth shapes (Figure 7), distance minimizing for isolated parts (Figures 1, 8, 12 and 13) and coordinate-wise for the final exaggeration (Figure 8).

We also test different correspondences computation methods: cross-parameterization [15], as illustrated in Figures 10, 11 and 14, registration only and scaled down model, both used in the examples of Figures 1 and 8.

We set the cut-off frequency to $n\chi = 96$. We can use strong morphing parameters for the distance minimizing filters: $\mu = -8.0$, while much lower ones $-0.8 > \mu > -2.4$ suit better for the other filters. The scale control Φ was set by hand, with amplifications from 0 to 2.

Interaction performances Our harmonic formulation allows interactive rates (Table 1), and we take advantage of this capability by introducing our WYSIWYG interface for 3d caricature. This feature let the user quickly tune expressive caricatures even with a single iteration of the parameters (Figures 11 and 10). The fast re-use of the result as template also permits more sophisticated effects, while keeping the interactivity of the interface (Figures 1, 8, 13 and 12).

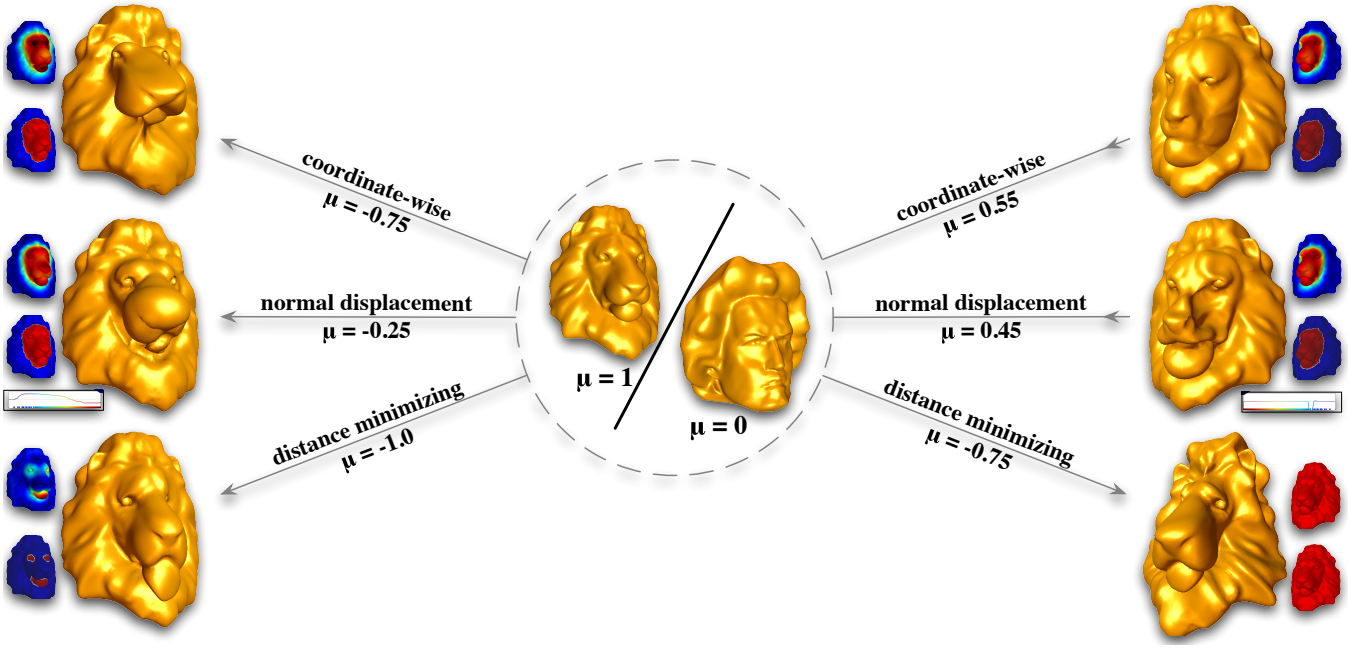

Figure 10: *Independent caricatures obtained using different filters, regions and morphing parameter values, using Beethoven model as template for the lion head. For each caricature, the vignettes represent the selected region χ_{01} (bottom) and blending function χ (top) all obtained with $n\chi = 96$. The scale control has been used for the middle row, and is displayed below the vignettes.*

Indeed, the interface achieves above 8 frames per second when displaying the three filters simultaneously, even with the picking and smoothing of χ which takes in average 39 milliseconds on the models we use (Table 1). The substitution of the shape by one of the results lasts in average 1.2 seconds for the filter computation on CPU and up to 4 seconds for the reprocessing.

Filter comparisons We can compare the filters introduced here based on some preliminary experiments. The coordinate-wise filter allows for blended exaggeration or reduction (as Egea's mouth in Figure 13). Among the three filters, it is the most appropriate for generating sharp features, such as the nose in Figure 6, although with large μ it can induce self-intersections of the model (Figure 14). It is also appropriate for the final exaggeration with a scaled down template. The normal displacement filter must be used with small μ to keep a coherent model, but suits very well to inflate regions without thin features, such as the chin in Figure 8 or large noses in Figures 1 and 12. Finally, the distance minimizing filter, by its least squares nature, offers subtle, diffusive but shape preserving deformations. For example in Figure 12, the ears are exaggerated but the inner cartilage shape is preserved, similarly to Egea's scar in Figure 13.

The left group of Figure 14 illustrates the use of the coordinate-wise filter alone, varying the morphing parameter μ without other control. This morphing exactly corresponds to the only cross-parameterization approach [15] if $\chi \equiv 1$, and we can observe the more expressive results obtained in Figures 12 and 13.

Finally, our formulation does not rely on a specific human face model, and thus allows for other caricature styles, such as using animals (Figure 11) or even caricaturing animals (Figure 10).

Discussion The use of manifold harmonics to model the exaggeration as a filter restricts the deformation to operate on non-localized basis. The reduction of the filters to not too high frequencies further limits the ability to capture details such as wrinkles, while a more careful treatment of high frequencies may [26].

The correspondence accuracy may also impact the result. In particular, using only continuous cross-parameterization reveals the C^1-discontinuity of the correspondences for large $|\mu|$ (in particular the left Egea in the last row of Figure 14).

8 Conclusion

In this work, we propose an interactive tool for 3d caricature modeling. It derives the extrapolation principle [6] in terms of harmonic filtering, introducing interactivity and providing morphing, localization and scale controls. This work may be improved in several directions, among which enhancing the interface with automatic segmentation to select significant parts to be exaggerated, adding subjective learning in the interface with intelligent galleries [29], and including texture and non-photorealistic rendering.

Acknowledgement We would like to thank the Max-Planck-Institut für Informatik and Cyberware for the 3d models and the persons who let us use their 3d faces. This work is financed by CNPq, FAPERJ and FAPEAL.

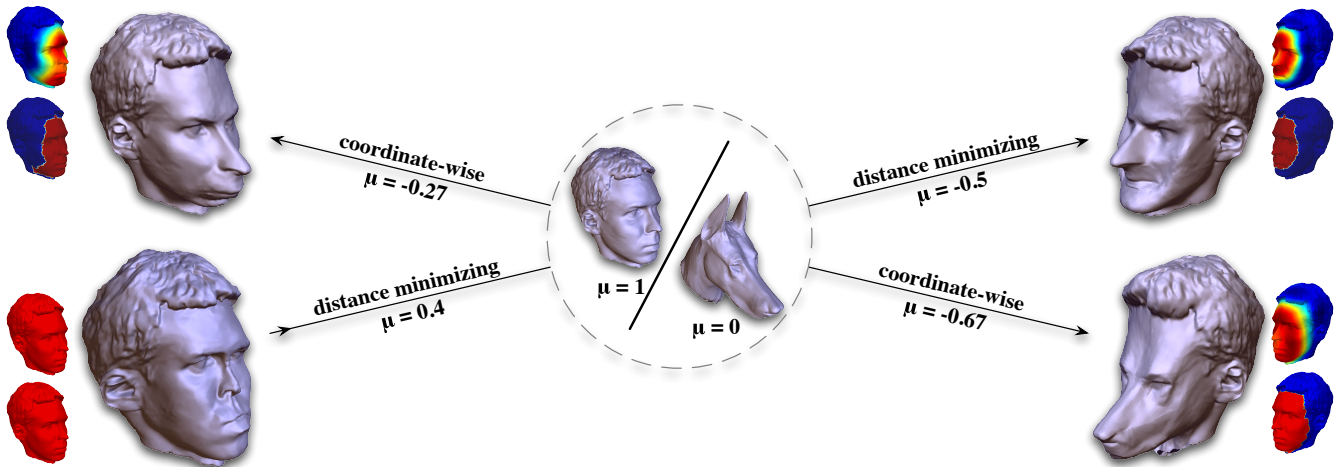

Figure 11: *Independent caricatures obtained using different regions and morphing parameter values, using a dog head as template. For each caricature, the vignettes represent the selected region χ_{01} (bottom) and blending function χ (top) all obtained with $n\chi = 96$.*

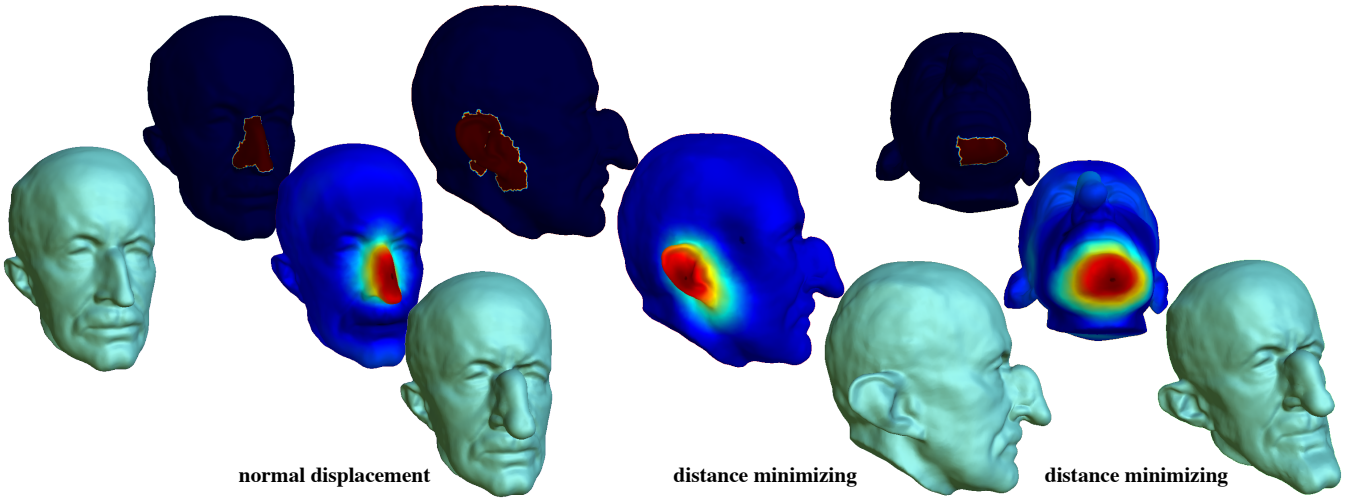

Figure 12: *Progressive caricature of the Max Planck model with template from Egea (Figure 14).*

References

[1] E. Akleman. Making caricatures with morphing. In *Siggraph Visual Proceedings*, page 145, 1997.

[2] E. Akleman and J. Reisch. Modeling expressive 3D caricatures. In *Siggraph Sketches*, page 61, 2004.

[3] K. Arun, T. Huang and S. Blostein. Least-squares fitting of two 3D point sets. *Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[4] P. Besl and N. McKay. A method for registration of 3D shapes. *Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[5] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Siggraph*, pages 187–194, 1999.

[6] S. E. Brennan. Caricature generator: the dynamic exaggeration of faces by computer. *Leonardo*, 18(3):170–178, 1985.

[7] F. Chazal, A. Lieutier and J. Rossignac. Normal-map between normal-compatible manifolds. *International Journal of Computational Geometry and Applications*, 17(5):403–421, 2007.

[8] H. Chen, N.-N. Zheng, L. Liang, Y. Li, Y.-Q. Xu and H.-Y. Shum. Pictoon: a personalized image-based cartoon system. In *Multimedia*, pages 171–178, 2002.

[9] W. Chen, H. Yu and J. J. Zhang. Example based caricature synthesis. In *Computer Animation and Social Agents*, pages 11–18, 2009.

[10] L. Clarke, M. Chen and B. Mora. Automatic generation of 3d caricatures based on artistic deformation styles. *Transactions on Visualization and Computer Graphics*, 2010.

[11] G. Fu, Y. Chen, J. Liu, J. Zhou and P. Li. Interactive expressive 3D caricatures design. In *Multimedia*, pages 965–968, 2008.

[12] B. Gooch, E. Reinhard and A. Gooch. Perception-driven black-and-white drawings and caricatures, 2002.

[13] F. Grose. *Rules for drawing caricatures : with an essay on comic painting*. Hooper Wigstead, 1796.

[14] H. Koshimizu, M. Tominaga, T. Fujiwara and K. Murakami. On KANSEI facial processing for computerized facial caricaturing system. In *Systems, Man, and Cybernetics*, volume 6, pages 294–299, 1999.

[15] V. Kraevoy and A. Sheffer. Cross-parameterization and compatible remeshing of 3d models. In *Siggraph*, pages 861–869, 2004.

[16] T. Lewiner, T. Vieira, A. Bordignon, A. Cabral, C. Marques, J. Paixão, L. Custódio, M. Lage, M. Andrade, R. Nascimento, S. de Botton, S. Pesco, H. Lopes, V. Mello, A. Peixoto and D. Martinez. Tuning manifold harmonics filters. In *Sibgrapi*, pages 110–117, 2010.

[17] P. Li, Y. Chen, J. Liu and G. Fu. 3d caricature generation by manifold learning. In *Multimedia*, pages 941–944, 2008.

[18] L. Liang, H. Chen, Y.-Q. Xu and H.-Y. Shum. Example-based caricature generation with exaggeration. In *Pacific Graphics*, pages 386–393, 2002.

[19] J. Liu, Y. Chen, J. Xie, X. Gao and W. Gao. Semi-supervised learning of caricature pattern from manifold regularization. In *Advances in Multimedia Modeling*, pages 413–424, 2009.

[20] A. Mattos, R. Cesar Jr, J. Mena-Chalco and L. Velho. 3D linear facial animation based on real data. In *Sibgrapi*, pages 271–278, 2010.

[21] Z. Mo, J. P. Lewis and U. Neumann. Improved automatic caricature by feature normalization and exaggeration. In *Siggraph Sketches*, page 57, 2004.

[22] L. Redman. *How to draw caricatures*. Contemporary Books, 1984.

[23] M. Reuter, S. Biasotti, D. Giorgi, G. Patané and M. Spagnuolo. Discrete Laplace-Beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3):381–390, 2009.

[24] G. Rong, Y. Cao and X. Guo. Spectral mesh deformation. *The Visual Computer*, 24(7):787–796, 2008.

[25] R. N. Shet, K. H. Lai, E. A. Edirisinghe and P. W. Chung. Use of neural networks in automatic caricature generation: An approach based on drawing style capture. *Pattern Recognition and Image Analysis*, pages 343–351, 2005.

[26] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl and H. Seidel. Laplacian surface editing. In *Symposium on Geometry Processing*, pages 175–184, 2004.

[27] C.-C. Tseng and J.-J. J. Lien. Synthesis of exaggerative caricature with inter and intra correlations. In *Asian Conference on Computer Vision*, pages 314–323, 2007.

[28] B. Vallet and B. Lévy. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pages 251–260, 2008.

[29] T. Vieira, A. Bordignon, A. Peixoto, G. Tavares, H. Lopes, L. Velho and T. Lewiner. Learning good views through intelligent galleries. *Computer Graphics Forum (Eurographics Proceedings)*, 28(2):717–726, 2009.

[30] H. Yu and J. J. Zhang. Caricature synthesis based on mean value coordinates. In *Computer Animation and Social Agents*, pages 1–10, 2010.

The corresponding work was published in the proceedings of Shape Modeling International, Computers & Graphics 35(3), 2011..

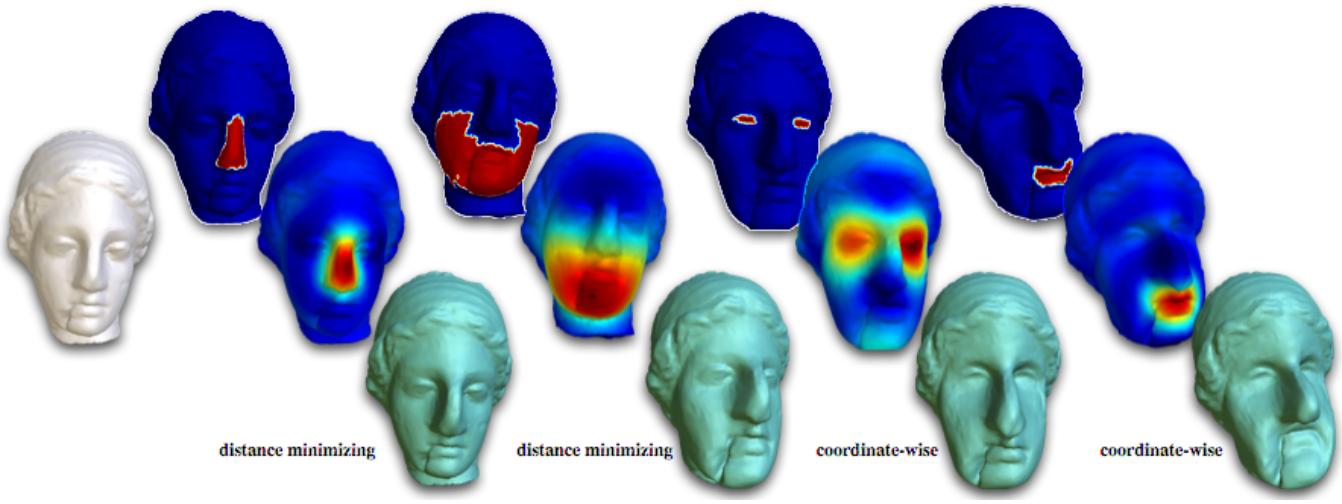

Figure 13: *Progressive caricature of the Egea model, with reference from the Max Planck (Figure 14).*

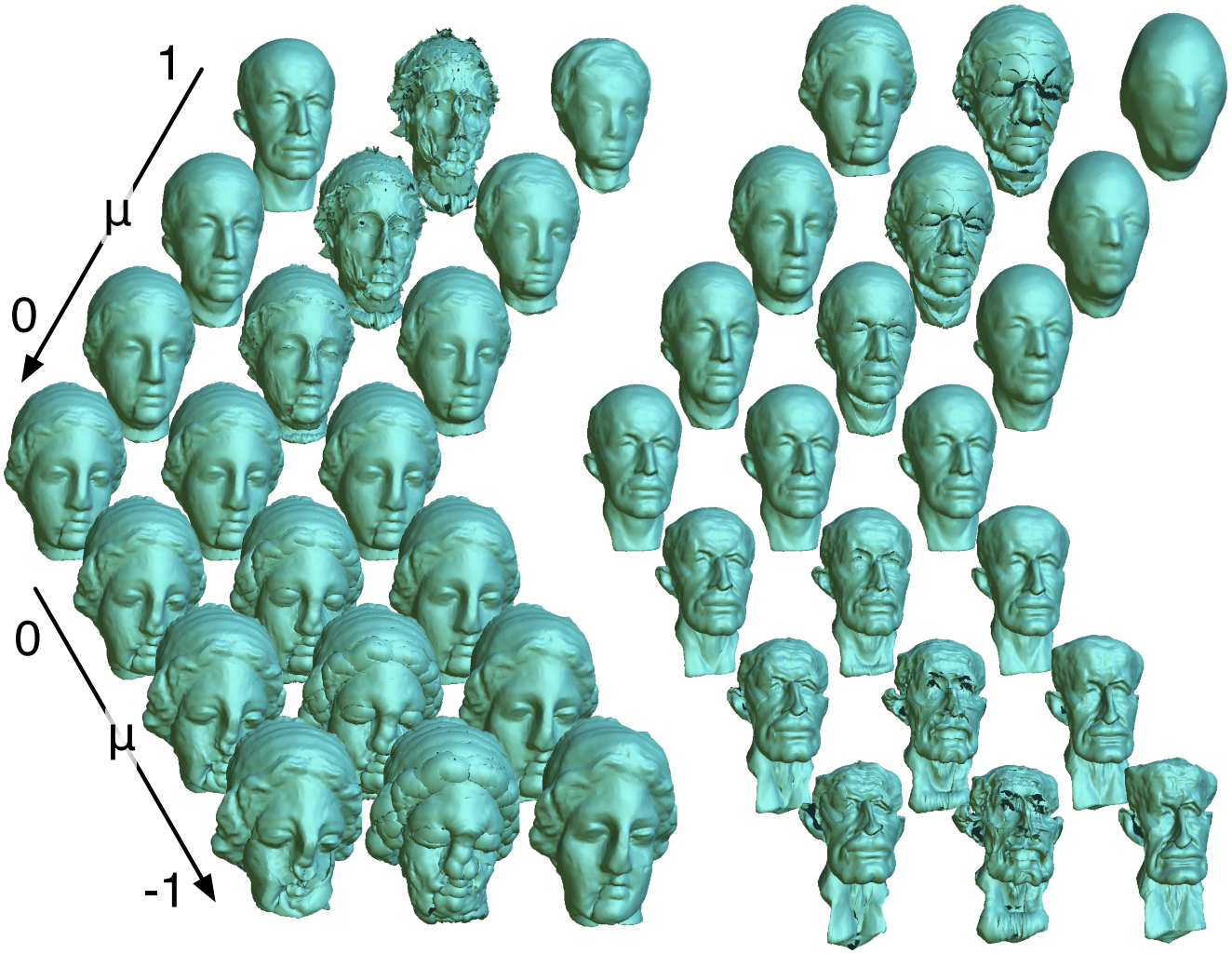

Figure 14: *Varying morphing parameter μ from $+1$ to -1 from Egea to Max Planck (left) and reversely (right), using the coordinate-wise, normal and distance minimizing filter.*

Preprint MAT. 13/10, communicated on December 10^{th}, 2010 to the Department of Mathematics, Pontifícia Universidade Católica — Rio de Janeiro, Brazil.

A Distance optimization details

The distance minimizing filter ${}^o\phi_k$,${}^o\phi_d$ is defined as the global minimum of the quadratic functional $F(\varphi,\varphi_d) = \frac{1}{2}\sum_{v\in V}\|{}^\varphi\mathbf{x}(v) - \mathbf{x}^r(v)\|^2$. First, substituting φ per $\varphi - 1$ leads to a simpler form:

$$\begin{aligned}
{}^\varphi\mathbf{x}(v) - \mathbf{x}^r(v) &= \mathbf{x}_{(\varphi-1)+1}(v) - \big(\mathbf{x}^\delta(v) - \mathbf{x}(v)\big)\\
&= \sum_{k=0}^{n-1} H_k(v)\cdot(\varphi_k - 1)\cdot\widetilde{\mathbf{x}}_k + (\varphi_d - 1)\cdot\mathbf{dx}\\
&\quad - \left(\sum_{k=0}^{n-1} H_k(v)\cdot\widetilde{\mathbf{x}}_k + \mathbf{dx} - \mathbf{x}(v)\right) - \mathbf{x}^\delta(v)\\
&= \mathbf{x}_{(\varphi-1)}(v) - \mathbf{0} - \mathbf{x}^\delta(v)
\end{aligned}$$

Using this substitution and the orthogonality properties of the harmonic basis allows to solve this optimization problem explicitly. Without grouping the high frequencies yet, i.e. considering $\varphi_k = \varphi_d$ for $k \geq n$, we can write F in the harmonic basis.

$$\begin{aligned}
F(\varphi) &= \frac{1}{2}\sum_{v\in V}\|{}^\varphi\mathbf{x} - \mathbf{x}^\delta\|^2\\
&= \frac{1}{2}\sum_{v\in V}\Big\|\sum_{k=0}^{N-1} H_k(v)\cdot(\varphi_k\cdot\widetilde{\mathbf{x}}_k) - \sum_{k=0}^{N-1} H_k(v)\cdot\widetilde{\mathbf{x}^\delta}_k\Big\|^2\\
&= \frac{1}{2}\sum_{v\in V}\Big\|\sum_{k=0}^{N-1} H_k(v)\cdot\Big(\varphi_k\cdot\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{x}^\delta}_k\Big)\ \Big\|^2\\
&= \frac{1}{2}\sum_{v\in V}\Big\langle \sum_{k=0}^{N-1} H_k(v)\cdot\Big(\varphi_k\cdot\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{x}^\delta}_k\Big)\\
&\qquad \Big|\ \sum_{l=0}^{N-1} H_l(v)\cdot\Big(\varphi_l\cdot\widetilde{\mathbf{x}}_l - \widetilde{\mathbf{x}^\delta}_l\Big)\ \Big\rangle
\end{aligned}$$

Grouping the terms by v and using Equation (1):

$$\begin{aligned}
F(\varphi) &= \frac{1}{2}\sum_{k,l=0}^{N-1}\left(\sum_{v\in V} H_k(v)H_l(v)\right)\cdot\\
&\qquad \Big\langle\varphi_k\cdot\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{x}^\delta}_k\Big|\varphi_l\cdot\widetilde{\mathbf{x}}_l - \widetilde{\mathbf{x}^\delta}_l\Big\rangle\\
&= \frac{1}{2}\sum_{k,l=0}^{N-1}\delta_{k,l}\Big\langle\varphi_k\cdot\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{x}^\delta}_k\Big|\varphi_l\cdot\widetilde{\mathbf{x}}_l - \widetilde{\mathbf{x}^\delta}_l\Big\rangle\\
&= \frac{1}{2}\sum_{k=0}^{N-1}\Big\|\varphi_k\cdot\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{x}^\delta}_k\Big\|^2\\
&= \frac{1}{2}\sum_{k=0}^{n-1}\Big\|\varphi_k\,\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{x}^\delta}_k\Big\|^2 + \frac{1}{2}\sum_{k=n}^{N-1}\Big\|\varphi_d\,\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{x}^\delta}_k\Big\|^2
\end{aligned}$$

The last line is deduced by grouping the high frequencies.

The minimum of functional F is achieved when its gradient $\left[\frac{\partial F}{\partial\varphi(0)}, \frac{\partial F}{\partial\varphi(2)}, \ldots, \frac{\partial F}{\partial\varphi(n-1)}, \frac{\partial F}{\partial\varphi_d}\right]$ vanishes:

$$\begin{aligned}
\frac{\partial F}{\partial\varphi_k} = 0 &= \Big\langle\ \widetilde{\mathbf{x}}_k\ \Big|\ \varphi_k\cdot\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{x}^\delta}_k\Big\rangle\\
&= \varphi_k\cdot\|\widetilde{\mathbf{x}}_k\|^2 - \Big\langle\ \widetilde{\mathbf{x}}_k\ \Big|\ \widetilde{\mathbf{x}^\delta}_k\Big\rangle.
\end{aligned}$$

The high frequency part goes the reverse way of the previous derivation of F:

$$\begin{aligned}
\frac{\partial F}{\partial\varphi_d} = 0 &= \sum_{k=n}^{N-1}\Big\langle\ \widetilde{\mathbf{x}}_k\ \Big|\ \varphi_d\cdot\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{x}^\delta}_k\Big\rangle\\
&= \sum_{k,l=n}^{N-1}\delta_{k,l}\cdot\Big\langle\ \widetilde{\mathbf{x}}_k\ \Big|\ \varphi_d\cdot\widetilde{\mathbf{x}}_l - \widetilde{\mathbf{x}^\delta}_l\Big\rangle\\
&= \sum_{k,l=n}^{N-1}\sum_{v\in V} H_k(v)H_l(v)\cdot\Big\langle\ \widetilde{\mathbf{x}}_k\ \Big|\ \varphi_d\cdot\widetilde{\mathbf{x}}_l - \widetilde{\mathbf{x}^\delta}_l\Big\rangle\\
&= \sum_{v\in V}\Big\langle\ \sum_{k=n}^{N-1} H_k(v)\cdot\widetilde{\mathbf{x}}_k\\
&\qquad \Big|\ \sum_{l=n}^{N-1}\varphi_d\cdot H_l(v)\cdot\widetilde{\mathbf{x}}_l - H_l(v)\cdot\widetilde{\mathbf{x}^\delta}_l\Big\rangle\\
&= \sum_{v\in V}\Big\langle\ \mathbf{dx}(v)\ \Big|\ \varphi_d\cdot\mathbf{dx}(v) - \mathbf{dx}^\delta(v)\Big\rangle\\
&= \varphi_d\cdot\sum_{v\in V}\|\mathbf{dx}(v)\|^2 - \sum_{v\in V}\Big\langle\ \mathbf{dx}(v)\ \Big|\ \mathbf{dx}^\delta(v)\Big\rangle.
\end{aligned}$$

Each equation is linear and independent of the other. The minimum argument ${}^o\phi$,${}^o\phi_d$ of F is then explicitly given by:

$${}^o\phi_k = \frac{\Big\langle\ \widetilde{\mathbf{x}}_k\ \Big|\ \widetilde{\mathbf{x}^\delta}_k\Big\rangle}{\|\widetilde{\mathbf{x}}_k\|^2}, \quad {}^o\phi_d = \frac{\sum_{v\in V}\Big\langle\ \mathbf{dx}(v)\ \Big|\ \mathbf{dx}^\delta(v)\Big\rangle}{\sum_{v\in V}\|\mathbf{dx}(v)\|^2}\ .$$

The corresponding work was published in the proceedings of Shape Modeling International, Computers & Graphics 35(3), 2011..