

## Computational caricaturization of surfaces



Matan Sela\*, Yonathan Aflalo, Ron Kimmel

Geometric Image Processing Lab, Technion-Israel Institute of Technology, Haifa 32000, Israel

### ARTICLE INFO

#### Article history:

Received 17 August 2014

Accepted 27 May 2015

#### Keywords:

Poisson equation

Isometry/pose invariance

Gaussian curvature

Laplace–Beltrami operator

### ABSTRACT

The question whether a caricature of a 2D sketch, or an object in 3D can be generated automatically is probably as old as the attempt to answer the question of what defines art. In an attempt to provide a partial answer, we propose a computational approach for automatic caricaturization. The idea is to rely on intrinsic geometric properties of a given model that are invariant to poses, articulations, and gestures. A property of a surface that is preserved while it undergoes such deformations is self-isometry. In other words, while smiling, running, and posing, we do not change much the intrinsic geometry of our facial surface, the area of our body, or the size of our hands. The proposed method locally amplifies the area of a given surface based on its Gaussian curvature. It is shown to produce a natural comic exaggeration effect which can be efficiently computed as a solution of a Poisson equation. We demonstrate the power of the proposed method by applying it to a variety of meshes such as human faces, statues, and animals. The results demonstrate enhancement and exaggeration of the shape's features into an artistic caricature. As most poses and postures are almost isometries, the use of the Gaussian curvature as the scaling factor allows the proposed method to handle animated sequences while preserving the identity of the animated creature.

© 2015 Elsevier Inc. All rights reserved.

### 1. Introduction

A caricature is an illustration of an object in which some features are exaggerated. The terms *feature exaggeration* usually refers to an operation that relates measures of a given object to those of a reference one. For example, a small mouth, relates the size of a given mouth to its average in a given population. The procedure we have in mind is one that emphasizes and extends such discrepancies. An interesting question is what happens when there is no reference population or an average to relate to. At the other end, it would be intriguing to design a caricaturization procedure that would have the same effect on an articulated object which is not affected by its pose, or posture (Fig. 1).

The oldest caricatures known today were drawn in ancient Egypt at about 3000 BC [2]. Later, the Greek and the Roman Empires used caricatures to intimidate people and spread their propaganda. The characters in these ancient caricatures were often monstrous hybridizations of humans and animals, see for example Fig. 2. In the fifteenth century, there was a change in the attitude towards caricatures. It has probably started with Leonardo da-Vinci, who in his study and exploration for shapes and forms searched for people with extremely deformed facial characteristics, that he used for his scien-

tific art. Followers of da-Vinci started to emphasize properties of faces by exaggerating features in their drawings.

Drawing professional caricatures is a skill that requires long practice and innate talent. Skilled caricaturists are intimately familiar with human faces and their saddle variations. They can perceptually differentiate key features in one's face from an average one. Before drawing, a caricaturist finds the discrepancy of common parts and memorizes them. Then, she draws the subject while amplifying the more distinct characteristics. The process is described in Fig. 3. The choice of features and the ability to exaggerate while maintaining the identity is where art comes into play. Obviously, personal touch and drawing skills would lead to different projections onto different artists perceptions of reality.

Several papers attempt to automate a process of caricature drawing. The main efforts in this area can be divided into model-based and geometry-based approaches. The model based procedures are usually designed specifically for human faces that are represented by some parametric model. The discrepancies between the representations of a given subject face and an average one are exaggerated. In the geometry-based approach, intrinsic features of a shape are exaggerated. Here, we show how the proposed framework bridges between these two models.

While designing an automatic exaggeration procedure a fundamentally important property is robustness to isometric deformations, such as rigid body transformations, as well as changes of expressions and poses. Based on the above arguments, the two axioms we would

\* Corresponding author.

E-mail address: [matansel@tx.technion.ac.il](mailto:matansel@tx.technion.ac.il) (M. Sela).



Fig. 1. Caricaturization of a synthetic model [1]. Left to right: original model, caricature with exaggeration parameter  $\gamma = 0.25$ , and  $\gamma = 0.5$ .



Fig. 2. The Egyptian God Typhoon (left) probably goes back to 3000 BC and a Greek Gordon (right) 600 BC as reproduced in [2].

like to have for an automatic shape caricaturization procedure are thereby

- The ability to extend discrepancies with respect to a reference model.
- Robustness or invariance to isometric deformations.

Indeed, when exaggerating a shape in motion, a key principle is to ensure that the processed sequence of exaggerated shapes are all related in a similar fashion to the original one. This property would enable applying the algorithm to animated figures and 3D scans while preserving the identity of the modified creature. Here, we propose a caricaturization framework that obeys these guidelines. Our axiomatic construction is based on the analysis provided in [4] by which facial expressions and articulations are almost isometries. Next, the simplest differential isometry invariant measure is the Gaussian curvature. We show how to efficiently modify the geometry

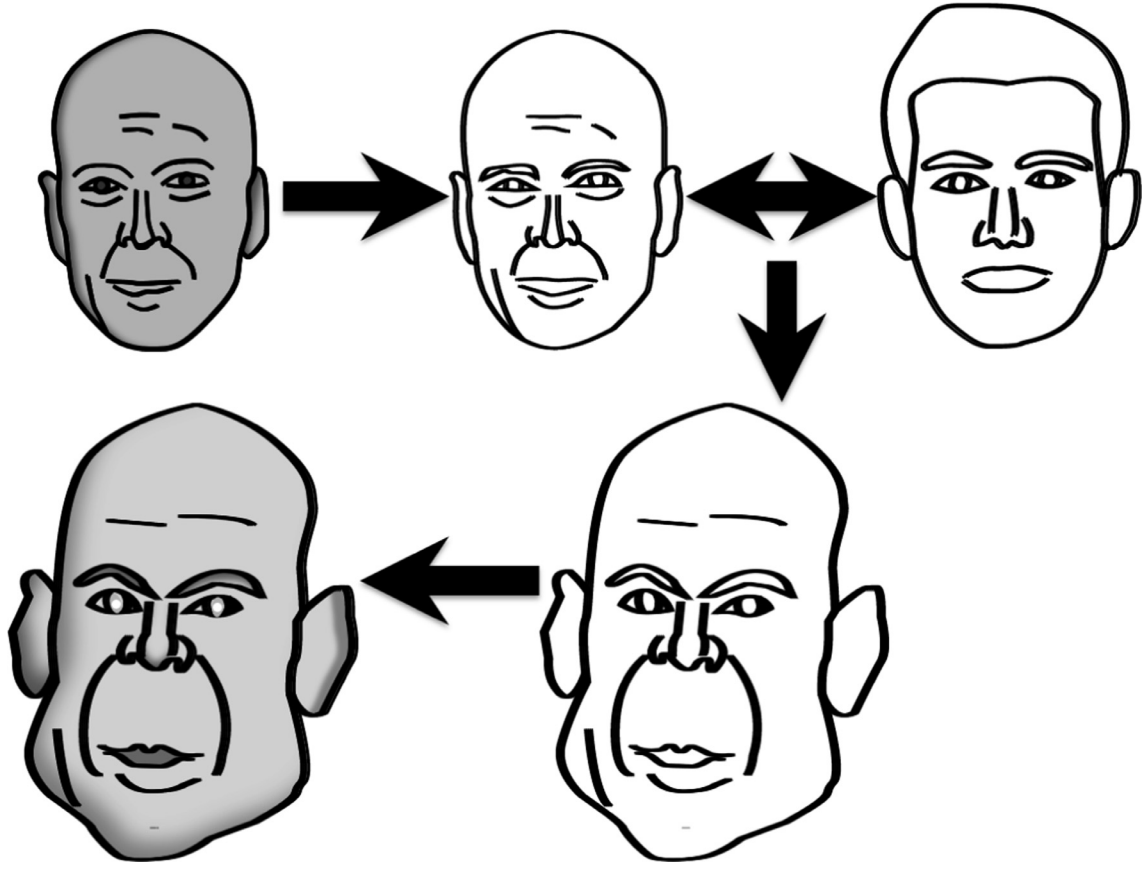
of a given surface such that its area is locally amplified by some scalar function of the Gaussian curvature. The proposed model links between the two schools of thoughts of computerized caricaturization. Given the correspondence between the subject shape and a reference average shape, the relation between the curvatures at corresponding points determines the local stretching of the shape. The results of the algorithm are demonstrated to be robust to almost isometric deformations. It is simple to incorporate the proposed framework as part of a real-time system using the implementation consideration we provide.

The organization of this paper is as follows. We start by reviewing caricature generation related papers in Section 2. We next introduce the proposed approach by briefly reviewing a classical surface modification technique to which we apply our axiomatically designed amplification factor in Section 3. Details about discretization and boundary conditions are given in Section 5. Section 6 discusses experimental results of the model applied to synthetic and real surfaces, as well as potential applications. We conclude with future directions in Section 7.

## 2. Related work

The challenge of automatically generating caricatures occupied researchers since the 1980's. Brennan [3] introduced the idea of exaggerating the discrepancies between a sketch of a subject and an average face. She developed an interactive system in which the user picks feature points on a facial image. Simple interpolation of these points provides a sketch of the faces. Then, the sketch is aligned and registered to an average one, and the distances between matched feature points are exaggerated to construct a caricature.

Blanz and Vetter [5] proposed an extension of Brennan's concept to facial surfaces embedded in  $\mathbb{R}^3$ . By finding bijective correspondences between approximately 200 faces, they constructed a linear model for facial geometries and facial textures. The caricature



**Fig. 3.** Abstraction of caricature drawing process, according to [3]. The artist extracts key characteristic curves of the subject from a facial image, and compares them against those of an average face. The caricature emphasizes the discrepancies.

construction procedure was implemented by increasing the distance of the sample from the statistical mean. Their algorithm was trained on many faces between which dense and accurate correspondences were manually provided.

Alkeman et al. [6] designed an interactive caricaturization framework. They provide a tool for manually tiling facial images with simplices (points, lines, and triangles) and translating some of them. Using blending functions, the deformation of each triangle is interpolated. They extended their system to 3D faces in [7]. In [8], Lim et al. developed a mobile system for real-time caricature generation out of facial images. A survey of these and other caricaturization techniques can be found in [9].

At the other end, Steiner et al. [10] developed a method for enhancing and exaggerating planar shapes based on intrinsic (geometric) properties. They used the *level set method* [11] to iteratively stabilize inverse geometric diffusion operators of a given curve. This exaggeration method was intrinsic and did not require a reference average model.

Eigensatz et al. [12] proposed a shape deformation technique that enables manipulating the principal curvatures at arbitrary points on the shape. The non-linear relation between the principal curvatures and the shape vertex positions lead the authors to a costly non-linear least-squares optimization process.

Kim et al. [13] proposed an interactive method for exaggerating shape features by manipulating parts of the mesh in the frequency domain of the Laplace–Beltrami operator. Lewiner et al. [14] developed a real-time interactive caricaturization framework that exaggerates spectral representation of the difference between a given face and a template one. Cimen et al. [15] developed a method for caricaturizing 3D meshes based on its saliency measure using Free Form Deformation techniques.

In [16], Yu et al. introduced a surface editing framework in which the surface embedding gradients are manipulated smoothly through a solution of a Poisson equation. Chuang et al. in [17] extended that approach to feature enhancement and smoothing. Their technique scales the same gradient fields and solves a modified version of the Poisson equation known as the *screened Poisson equation*. That equation enables balancing between the fidelity and enhancement regularization of the manipulated surface.

### 3. Surface enhancement

#### 3.1. Notations

Consider the boundary of a 3D shape as a surface or a two dimensional manifold  $M \subset \mathbb{R}^3$  given in a parametric form as  $S : \Omega \subset \mathbb{R}^2 \rightarrow M$ . Throughout this exposition, the surface  $M$  will be described by its coordinates  $S(u, v) = \{x(u, v), y(u, v), z(u, v)\}$ , where  $x$ ,  $y$ , and  $z$  are the coordinate-functions of the manifold.

The first partial derivatives of the parametric form,  $S_u$  and  $S_v$ , are two vectors that are tangent to the surface and linearly independent. Thus, one can set the basis for the tangent plane  $T_p M$ , about a point  $p \in S$  on the surface, as  $\{S_u, S_v\}$ .

The Jacobian of the parametrization  $J$  is a  $3 \times 2$  matrix whose columns are  $S_u$  and  $S_v$ , respectively. We define the matrix  $G = J^T J$  as the *regular metric* of the surface. Clearly,  $G$  is invertible, positive definite, and can be written explicitly in terms of  $S_u$  and  $S_v$  as

$$G = \begin{pmatrix} \langle S_u, S_u \rangle & \langle S_u, S_v \rangle \\ \langle S_v, S_u \rangle & \langle S_v, S_v \rangle \end{pmatrix}. \quad (1)$$

Since the basis  $\{S_u, S_v\}$  is not necessarily orthogonal, the inner product between two vectors  $\underline{a}, \underline{b} \in T_p M$  represented in this basis is



**Fig. 4.** The proposed method is invariant to poses and articulations that are mathematically captured by almost isometric transformations. Left to right: original shape, first pose exaggerated with  $\gamma = 0.25$ , first pose with  $\gamma = 0.5$ , second pose with  $\gamma = 0.5$ , second pose with  $\gamma = 0.25$ , and the original shape in its second pose.

given by

$$\langle \underline{a}, \underline{b} \rangle_G = \underline{a}^T G \underline{b}. \quad (2)$$

Next, consider two surfaces  $M_1$  and  $M_2$ , and suppose we are given a parametrization for the first surface  $S_1(u, v)$ . Suppose we are given a bijective map between the surfaces  $\phi: M_1 \rightarrow M_2$ . We can then build a parametrization for the second surface  $M_2$  by composing  $\phi$  on  $S_1$ . That is, a parametrization for the second surface is given by

$$S_2(u, v) = \phi \circ S_1(u, v). \quad (3)$$

Given the parametrizations for both surfaces, we can calculate the metric matrices,  $G_1$  and  $G_2$ , at each point  $(u, v)$  in the parametrization domain, see Eq. (1). If at each point  $(u, v)$  in the parametrization domain,  $G_1(u, v) = G_2(u, v)$ , then  $\phi$  is said to be an *isometry*. If there exists an isometry between two surfaces  $M_1$  and  $M_2$ , they are said to be *isometric*.

A curve on one surface has the same length as that of its isometrically mapped version. An area of a patch on one surface is mapped by  $\phi$  to a patch that has the same area. In fact, any geometric property that can be measured in terms of the metric  $G$  and its derivatives and integration with respect to  $u$  and  $v$  are preserved by the isometry  $\phi$ . We call such properties *isometric invariant* or *intrinsic*.

For each function  $f: S \rightarrow \mathbb{R}$  defined on the surface there exists a corresponding function in the parameter domain  $\tilde{f}: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ , such that

$$\tilde{f}(u, v) = f \circ S(u, v). \quad (4)$$

Similarly, we define a *tangent vector field* on the surface,  $\vec{F}: M \rightarrow T_p M$ , as an assignment of a tangent vector to each point on the surface. Since these fields are in the tangent plane, one can represent them in terms of the basis  $\{S_u, S_v\}$ . For each tangent vector field  $\vec{F}$  on the surface, there exists a corresponding vector field with identical coordinates in the parametrization domain  $\tilde{F}$  represented by the natural basis of  $\mathbb{R}^2$ .

As defined in [18], the gradient of a function on the surface, which we term here as *gradient with respect to the metric  $G$*  is a differential operator that satisfies

$$\langle \nabla_G f(p), \vec{w} \rangle_G = d_p f = \lim_{h \rightarrow 0} \frac{f(p + h\vec{w}) - f(p)}{h}, \quad (5)$$

for each  $\vec{w} \in T_p M$ . It follows that

$$f(p + h\vec{w}) = f(p) + h \langle \nabla_G f(p), \vec{w} \rangle_G + O(h^2). \quad (6)$$

For the corresponding function in the parametrization domain  $\tilde{f}(u, v)$ ,

$$\begin{aligned} \tilde{f}(u_0 + hw_1, v_0 + hw_2) \\ = \tilde{f}(u_0, v_0) + h \langle \nabla_{u,v} \tilde{f}(u_0, v_0), \vec{w} \rangle + O(h^2), \end{aligned} \quad (7)$$

where  $p = S(u_0, v_0)$ , and  $\vec{w} = (w_1, w_2)$ . Since for some small enough  $h > 0$ , the left hand sides of (6) and (7) are equal, it follows that

$$\langle \nabla_G f(p), \vec{w} \rangle_G = \langle \nabla_{u,v} \tilde{f}(u_0, v_0), \vec{w} \rangle. \quad (8)$$

And, therefore

$$\vec{w}^T G \nabla_G f(p) = \vec{w}^T \nabla_{u,v} \tilde{f}(u_0, v_0). \quad (9)$$



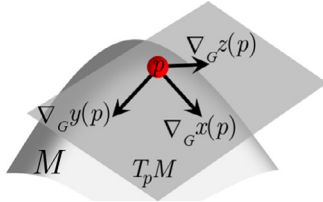


Fig. 5. The gradient fields of the coordinate functions at each point on the surface reside in the tangent plane.

for each  $\vec{w} \in T_p M$ . This provides a formula for the gradient in terms of the gradient of its corresponding function in the parametrization domain  $\tilde{f}$  and the parametrization  $S$  as

$$\nabla_G f(p) = G^{-1} \nabla_{u,v} \tilde{f}(u_0, v_0). \quad (10)$$

Treating the space of functions and the space of fields on the surface as an Hilbert space, one can define inner product between two functions,  $f$  and  $g$ , as

$$\langle f, g \rangle_{L^2} = \int_{\Omega} f(x)g(x)da \quad (11)$$

and the inner product between two vector fields,  $\vec{F}$  and  $\vec{H}$ , as

$$\langle \vec{F}, \vec{H} \rangle_{F^2} = \int_{\Omega} \vec{F}(x)^T G \vec{H}(x) da. \quad (12)$$

Here,  $da = \sqrt{\det G} du dv$ .

The divergence is a differential operator that produces a function from a tangent field. It is defined as minus the adjoint operator of the gradient, with respect to the above inner products. That is, the divergence of fields on a manifold, denoted as  $\nabla_G \cdot$ , satisfies

$$\langle \nabla_G \cdot \vec{F}, g \rangle_{L^2} = -\langle \vec{F}, \nabla_G g \rangle_{F^2}, \quad (13)$$

for any function  $g$  and a field  $\vec{F}$  on the surface. By integration by parts and assuming that  $g$  vanishes on the boundaries, one can show that the divergence with respect to the metric of a tangent field can be calculated by

$$\nabla_G \cdot \vec{F} = \frac{1}{\sqrt{\det G}} \nabla_{u,v} \cdot (\sqrt{\det G} \vec{F}), \quad (14)$$

where  $\vec{F}$  is the corresponding field of  $\vec{F}$  in the parameterization domain.

The Laplacian is a differential operator that maps functions to functions. The Laplacian of functions on the surface is called the Laplace–Beltrami operator and is denoted here as  $\Delta_G$ . It is defined as minus the divergence of the gradient. Thereby, it can be written as

$$\Delta_G \cdot f = \frac{1}{\sqrt{\det G}} \nabla_{u,v} \cdot (\sqrt{\det G} G^{-1} \nabla_{u,v} \tilde{f}) \quad (15)$$

### 3.2. Gradient-based deformation

The gradient fields of the coordinate functions with respect to a given metric,  $\nabla_G x$ ,  $\nabla_G y$  and  $\nabla_G z$ , define three vector fields that reside in the tangent plane  $T_p M$ , defined by  $S_u$  and  $S_v$ , about each point  $s$  on the surface  $p \in M$ , see Fig. 5.

The gradient-based deformation method, proposed in [16], first transforms each field locally and arbitrarily,  $\nabla_G x(p) \rightarrow \mathcal{T}_p(\nabla_G x(p))$ . In order to derive the deformed surface,  $\tilde{S} = \{\tilde{x}, \tilde{y}, \tilde{z}\}$ , whose coordinate-functions gradient fields mostly agree with the manipulated ones in a least-squares sense, the method minimizes the following term for each coordinate function separately

$$\int_S \|\nabla_G \tilde{x}(p) - \mathcal{T}_p(\nabla_G x(p))\|^2 da(p), \quad (16)$$

with similar functionals for  $y$  and  $z$ . Here,  $da(p)$  is an area element about the point  $p \in S$ . The Euler–Lagrange equation of the above functional is the Poisson equation

$$\Delta_G \tilde{x} = \nabla_G \cdot (\mathcal{T}_p(\nabla_G x)),$$

or in surface notation,

$$\Delta_G \tilde{S} = \nabla_G \cdot (\mathcal{T}_p(\nabla_G S)). \quad (17)$$

This simple method gave birth to a rich family of surface deformation techniques. For example, restricting the deformations to isometries, the set of local transformations  $\{\mathcal{T}_p\}$  is limited to rotation matrices. In the proposed exaggeration method, we restrict ourselves to simple scalar scaling.

Other linear mesh deformation techniques includes thin-shell [19] and Laplacian-based [20], to name just a few. The paper [21] reviews some of these techniques and compares between them. For a broader overview of these and other related deformation methods, we refer to [22].

Next, we define a model-based framework for *relative exaggeration*. Let  $S$  be the surface of a given object. In addition, assume we are given a bijective map  $\phi: S \rightarrow \hat{S}$ , between  $S$  and the surface,  $\hat{S}$ , of a given representative average object. Such maps can be obtained by surface registration algorithms like non-rigid ICP with a sparse correspondence [23].

Based on our assumptions, there are two requirements from a caricaturization process: First, it needs to exaggerate the discrepancies from a given model. Second, it needs to be invariant to pose or independent of the embedding. To satisfy the latter, we propose to scale the given surface gradient fields by some function of the absolute Gaussian curvature  $|K|$  of the surface  $S$ . The choice of the Gaussian curvature, the simplest isometry invariant differential quantity, as an amplification factor makes our method robust to isometries of the surface.

In order to exaggerate the discrepancies from the average model, we choose the following function as a local scaling factor

$$f(p) = |K(p)|^{\beta \log \frac{da(p)}{d\hat{a}(\phi(p))}}, \quad (18)$$

where  $\beta$  is some positive constant,  $da(p)$  is an area element about  $p \in S$  and  $d\hat{a}(\phi(p))$  is the corresponding area element about  $\phi(p) \in \hat{S}$  – the average representative surface. Note that given  $\phi$  the corresponding area elements can be easily computed by defining a corresponding parametrization.

In warped surface elements where there is no local scaling, that is  $da(p) = d\hat{a}(\phi(p))$ , no exaggeration is executed. Regions that were shrank with respect to the reference model will shrink even more, while those that were stretched will stretch more in the proposed method. In the case where no prior average model exists, we could assume, for example, that  $\log \frac{da(p)}{d\hat{a}(\phi(p))} = \epsilon$  and thereby scale the gradient fields by  $|K|^\gamma$ , where  $\gamma = \epsilon\beta$  is some non-negative constant.

In order to find a regular surface whose gradients with respect to the parametrization best fits the scaled ones, we minimize the squared difference between the tangent fields and end up with the following Poisson equation

$$\begin{aligned} \Delta_G \tilde{S} &= \nabla_G \cdot (f(p) \nabla_G S) \\ &= \nabla_G \cdot \left( |K(p)|^{\beta \log \frac{da(p)}{d\hat{a}(\phi(p))}} \nabla_G S \right). \end{aligned} \quad (19)$$

Here,  $\tilde{S}$  is the exaggerated surface we are looking for, and the differential operators are taken with respect to the metric of the given surface  $S$  (Fig. 6). When there is no reference surface  $\hat{S}$ , the enhancement simplifies to the solution of the following Poisson equation

$$\Delta_G \tilde{S} = \nabla_G \cdot (|K|^\gamma \nabla_G S). \quad (20)$$

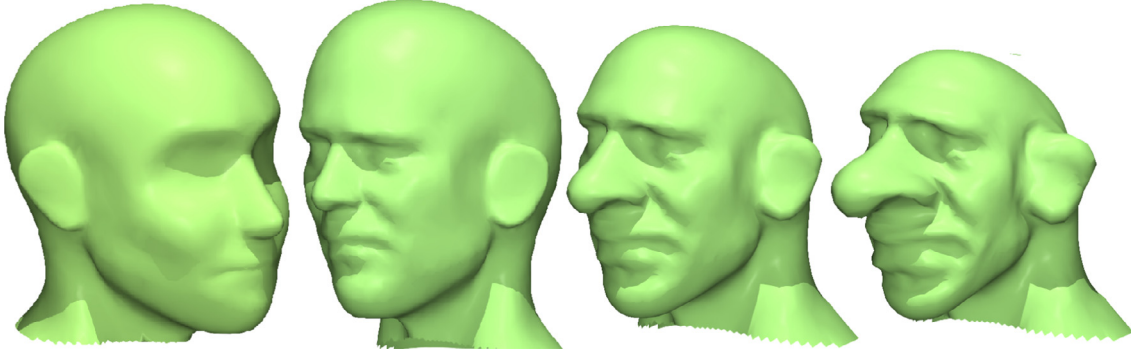


Fig. 6. Model-based caricaturization of a faces. From left to right: average face model, warped model, exaggerated with  $\beta = 0.3$ , and  $\beta = 0.5$ .

#### 4. Isometry preservation

We define two isometric surfaces as parametrized surfaces between which an isometric map exists (see Section 3.1). By construction, the proposed surface enhancement model is robust to isometric transformations. That is, given two isometric surfaces  $S_1 \approx S_2$ , if we can find corresponding surfaces  $\tilde{S}_1$  and  $\tilde{S}_2$  that satisfy Eq. (20) we also have that  $\tilde{S}_1 \approx \tilde{S}_2$ .

**Theorem 4.1.** *If the surfaces  $S_1$  and  $S_2$  are isometric, and there exist two surfaces  $\tilde{S}_1$  and  $\tilde{S}_2$  such that*

$$\nabla_{G_1} \tilde{S}_1 = |K_1| \nabla_{G_1} S_1, \quad (21)$$

and

$$\nabla_{G_2} \tilde{S}_2 = |K_2| \nabla_{G_2} S_2, \quad (22)$$

then,  $\tilde{S}_1$  and  $\tilde{S}_2$  are isometric.

**Proof.** By definition, the metric of  $\tilde{S}_1$  can be written in terms of the Jacobian of the parametrization as

$$\tilde{G}_1 = \tilde{J}_1^T \tilde{J}_1, \quad (23)$$

where  $\tilde{J}_1$  is a matrix whose columns are  $\tilde{S}_{1u}$  and  $\tilde{S}_{1v}$ .

By writing the gradient of the coordinate functions of  $\tilde{S}_1$  as functions on the surface  $S_1$  explicitly, we get

$$\begin{aligned} \nabla_{G_1} \tilde{S}_1 &= G_1^{-1} \nabla \tilde{S}_1 \\ &= G_1^{-1} \tilde{J}_1. \end{aligned} \quad (24)$$

Therefore

$$\tilde{G}_1 = \nabla_{G_1} \tilde{S}_1^T G_1 G_1 \nabla_{G_1} \tilde{S}_1. \quad (25)$$

By plugging relation (21) into the above equation, we get

$$\begin{aligned} \tilde{G}_1 &= |K_1|^2 \nabla_{G_1} S_1^T G_1 G_1 \nabla_{G_1} S_1 \\ &= |K_1|^2 \nabla S_1^T \nabla S_1 \\ &= |K_1|^2 G_1. \end{aligned} \quad (26)$$

By repeating the same derivation for  $\tilde{S}_2$ , we similarly obtain

$$\tilde{G}_2 = |K_2|^2 G_2. \quad (27)$$

Since we assumed that the surfaces  $S_1$  and  $S_2$  are isometric, the metric and the Gaussian curvatures are equal for corresponding points on the these surfaces. Hence

$$|K_1|^2 G_1 = |K_2|^2 G_2. \quad (28)$$

We can conclude that

$$\tilde{G}_1 = \tilde{G}_2 \quad (29)$$

which proves that indeed the surfaces  $\tilde{S}_1$  and  $\tilde{S}_2$  are isometric.  $\square$

Since we minimize the difference between the fields on isometric surfaces, and the divergence is intrinsic, the exaggeration is robust to isometries in a least squares sense. In practice, surfaces which are isometric to one another are rigidly equivalent in the discrete case. However, experimentally, for almost isometric shapes, the isometric deviation of the resulting caricatures, i.e. the Gromov–Hausdorff distance, is negligible.

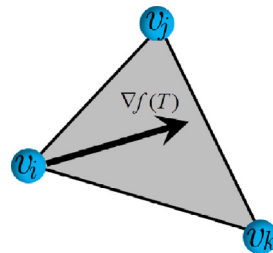
#### 5. Implementation considerations

One possible implementation of the proposed framework is when approximate continuous surfaces by triangulated meshes. In what follows, we briefly review discrete approximations of a continuous function defined on the surface, its gradient, the surface Gaussian curvature, a discrete divergence operator [24], and a discrete Laplace–Beltrami operator [25]. We conclude by combining these elements together for solving the system of equations (20), with some additional constraints that would fix, for example, some surface points as boundary conditions.

For the discretization we use the finite element method (FEM), in which every function  $f$  defined on the surface is assumed to be piecewise linear. As such, it can be represented with the piecewise linear basis functions  $\{B_i\}_{i=1}^n$  as

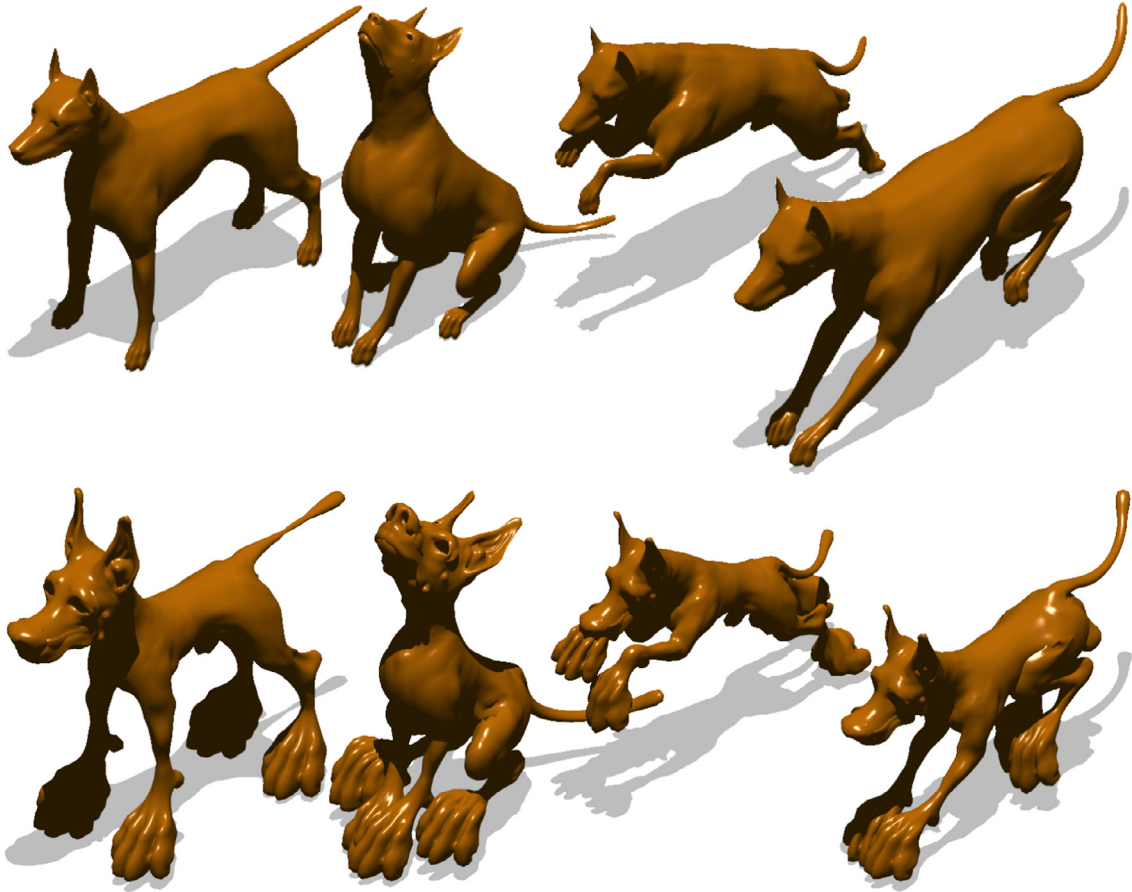
$$f(x) = \sum_{i=1}^n f_i B_i(x),$$

where  $n$  is the number of vertices,  $f_i$  is the value of the function  $f$  at the vertex  $v_i$ , and  $B_i(x)$  is a linear B-spline function that is equal to 1 at the vertex  $v_i$  and 0 at the rest of the vertices. Within each triangle  $T = (v_i, v_j, v_k)$ ,  $f$  is linearly interpolated only by  $f_i, f_j$ , and  $f_k$ . It can be shown that the gradient of  $f$  within each triangle is constant [25], and given by



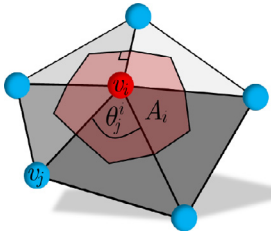
$$\nabla f(T) = (f_j - f_i) \frac{(s_i - s_k)^\perp}{2A_T} + (f_k - f_i) \frac{(s_j - s_i)^\perp}{2A_T},$$

where  $s_i, s_j$ , and  $s_k$  are the coordinates of the vertices  $v_i, v_j$ , and  $v_k$ , respectively,  $\perp$  denotes the counterclockwise  $90^\circ$  rotation operator in the triangle plane, and  $A_T$  is the area of the triangle  $T$  (Fig. 7).



**Fig. 7.** Exaggerated shapes of dogs. Top: different poses (approximately isometric) shapes, and bottom: exaggeration with a factor  $\gamma = 0.3$ .

There are different computational schemes for locally estimating the Gaussian curvature of a triangulated mesh. See [26] for a review and evaluation. Introduced in [25] is the so called Gauss–Bonnet scheme given by



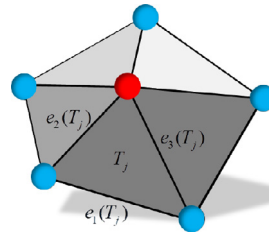
$$K(v_i) = \frac{1}{A_i} \left( 2\pi - \sum_{j: v_j \in \mathcal{N}_1(v_i)} \theta_j^i \right),$$

where  $A_i$  is the Voronoi area about the vertex  $v_i$ , and  $\theta_j^i$ s are the incident angles to the vertex  $v_i$ . Another scheme, presented in [27], approximates the second fundamental form at each vertex  $v_i$ . The eigenvalues of the second fundamental form are the principal curvatures,  $k_1$  and  $k_2$ , by which we derive the Gaussian curvature to be  $K(v_i) = k_1(v_i) \cdot k_2(v_i)$ . An implementation of this scheme in MATLAB is given in Gabriel Peyre's Toolbox Graph [28]. We tested both schemes in our experiments.

A discrete formulation of the divergence operator can be derived from the divergence theorem

$$\iint_A \nabla \cdot \vec{F} da = \oint_{\partial A} \vec{F} \cdot \hat{n} dl,$$

where  $\vec{F}$  is a given vector field, and  $\hat{n}$  is an outward pointing unit vector normal to the boundary. We assume that  $\vec{F}$  is constant within each triangle of the mesh, and approximate its divergence at a vertex  $v_i$  by averaging the divergence over a 1-ring neighborhood



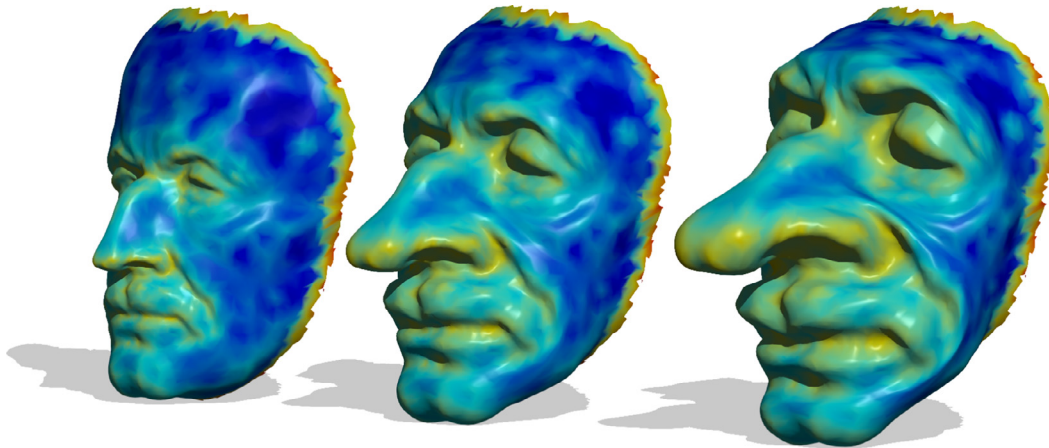
$$\begin{aligned} \text{Div}(\vec{F})(v_i) &\cong \frac{\iint_{\mathcal{N}_1(v_i)} \nabla \cdot \vec{F} da}{\iint_{\mathcal{N}_1(v_i)} da} \\ &= \frac{\sum_{T_j \in \mathcal{N}_1(v_i)} \iint_{A(T_j)} \nabla \cdot \vec{F} da}{\sum_{T_j \in \mathcal{N}_1(v_i)} A(T_j)} \\ &= \frac{\sum_{T_j \in \mathcal{N}_1(v_i)} \oint_{\partial A(T_j)} \vec{F} \cdot \hat{n} dl}{\sum_{T_j \in \mathcal{N}_1(v_i)} A(T_j)} \\ &= \frac{\sum_{T_j \in \mathcal{N}_1(v_i)} \sum_{e_k \in T_j} \vec{F}(T_j) \cdot e_k(T_j)^\perp}{\sum_{T_j \in \mathcal{N}_1(v_i)} A(T_j)}, \end{aligned}$$

where  $\mathcal{N}_1(v_i)$  is the 1-ring neighborhood comprised of the triangles incident to  $v_i$ ,  $A(T_j)$  is the area of the triangle  $T_j$ , and  $e_k(T_j)^\perp$  is an outwardly directed vector normal to the  $k^{\text{th}}$  edge of the triangle  $T_j$  with similar length to the edge  $e_k(T_j)$ .

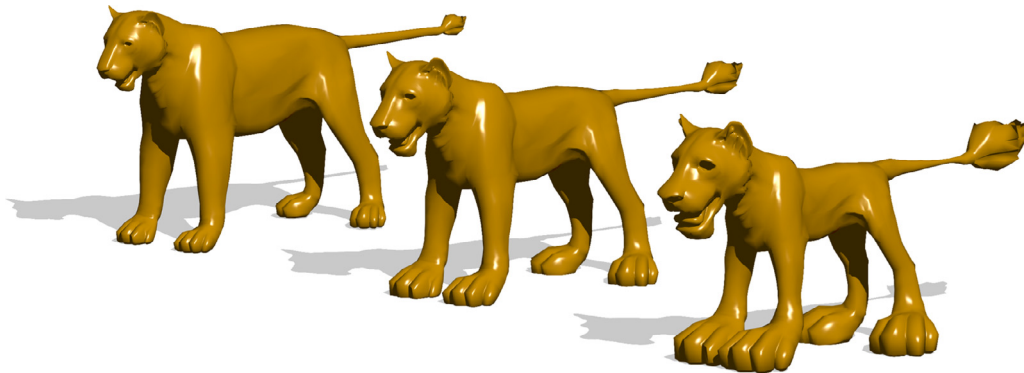




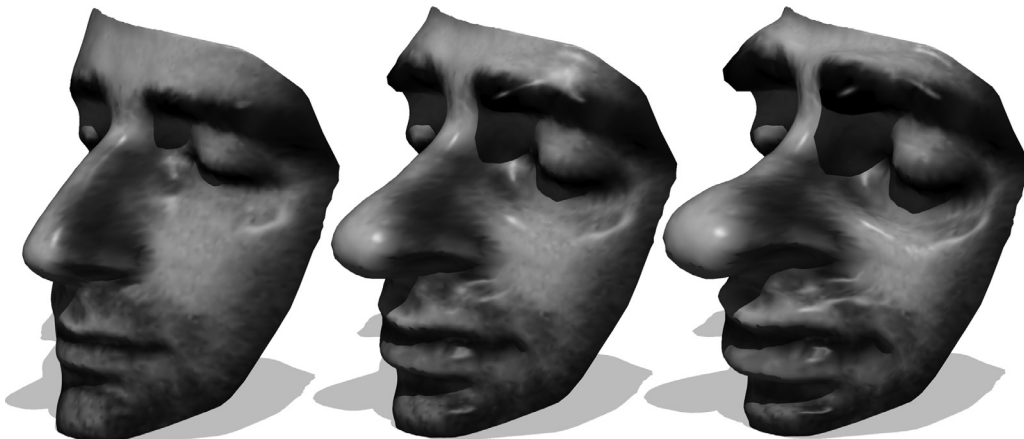
**Fig. 8.** The color of the meshes represents the approximated Gaussian curvature of the original mesh, where green denotes small values and purple large ones. From left to right: original model,  $\gamma = 0.5$ , and  $\gamma = 1$ .



**Fig. 9.** The color of the meshes represents the approximated Gaussian curvature of the original mesh, where blue denotes small values and red large ones. From left to right: original model, exaggerated with  $\gamma = 0.5$ , and  $\gamma = 0.75$ .

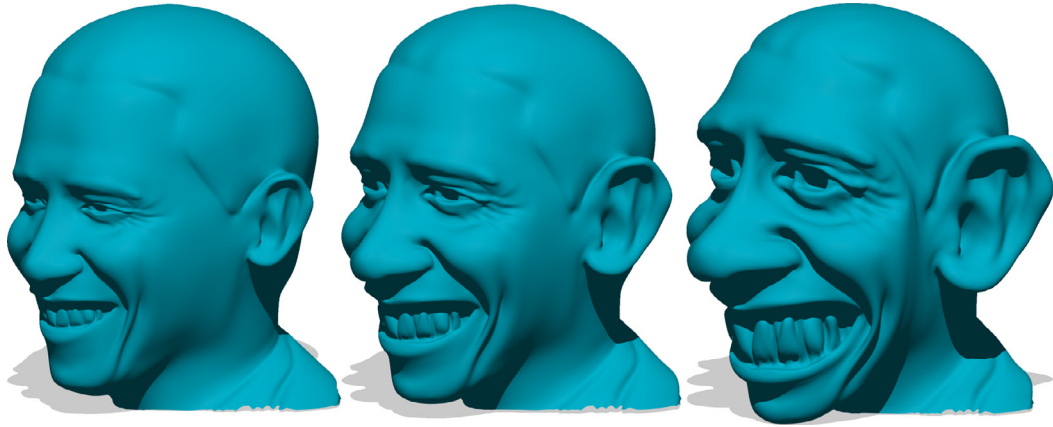


**Fig. 10.** Caricature generation for a lion. From left to right: the original model, exaggeration with  $\gamma = 0.25$ , and  $\gamma = 0.5$ .

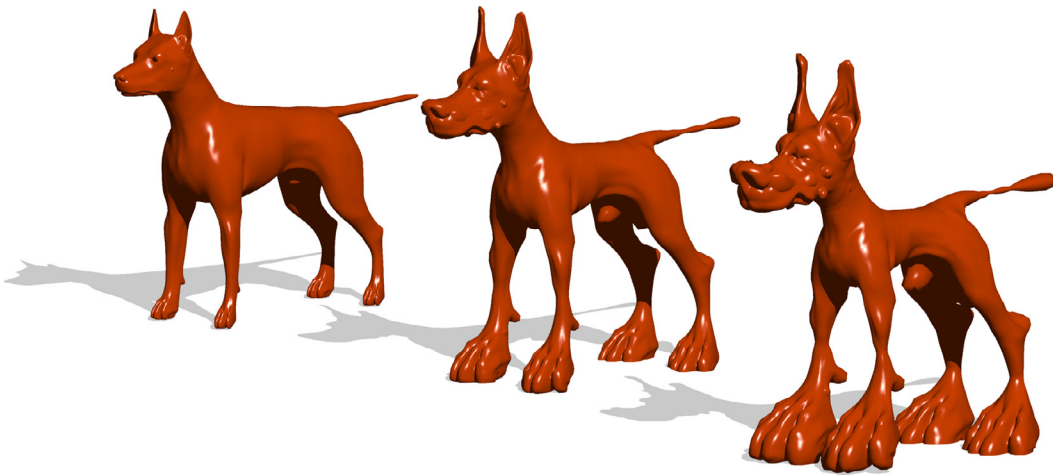


**Fig. 11.** Caricature generation for a textured face. From left to right: the original model, exaggerated shape with  $\gamma = 0.5$ , and  $\gamma = 0.75$ .





**Fig. 12.** Caricature generation of a faces. From left to right: original model, exaggerated with  $\gamma = 0.5$ , and  $\gamma = 0.75$ .



**Fig. 13.** Schnauzerization: caricature generation of a dog model. From left to right: original model, Schnauzer with  $\gamma = 0.5$ , and  $\gamma = 0.75$ .



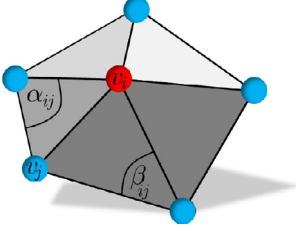
**Fig. 14.** A printed face model and its caricature with  $\gamma = 0.5$ .

**Table 1**

Pre-processing and core run-times (in ms) of the proposed algorithm for exaggerating features of shapes from TOSCA dataset.

Model name	# of vertices	# of faces	Pre-processing runtime	Core runtime
Wolf	4344	8684	349.5	7.4
Centaur	15768	31532	1538.2	29.3
Dog	25290	50528	2944.7	47.0
Human	52565	105032	7245.6	105.8

For discretizing the Laplace–Beltrami operator, we use the cotangent weight scheme for triangulated meshes [25]. We plug the values of a given function at each vertex,  $f_i = f(v_i)$ , to a vector, and multiply it by  $L = A^{-1}W$ , where  $A$  is the diagonal matrix of the Voronoi cell areas about each vertex, as introduced in [29].  $W$  is the classic cotangent weight matrix



$$W_{ij} = \begin{cases} \sum_{j: v_j \in \mathcal{N}_1(v_i)} (\cot \alpha_{ij} + \cot \beta_{ij}) & \text{if } i = j \\ -\cot \alpha_{ij} - \cot \beta_{ij} & \text{if } i \neq j, v_j \in \mathcal{N}_1(v_i) \\ 0 & \text{otherwise,} \end{cases}$$

where  $\alpha_{ij}$  and  $\beta_{ij}$  are the angles opposite to the edge  $(v_i, v_j)$ .

**Boundary conditions:** one limitation of the proposed framework is the fact that the Gaussian curvature is undefined along the boundary of a surface. Still, many real world surfaces, like those acquired by depth sensors, have a boundary. In practice, applying our framework to meshes with boundaries results in shapes with distorted bound-

aries. We need to specify the boundary of the shapes and to set proper boundary conditions.

In order to find vertices on the boundary of the shape, we query the neighbor triangles of each edge. If an edge is neighbor to a single triangle, it is labeled as a boundary edge. The vertices that comprise this edge are labeled as boundary vertices.

There are various boundary conditions that could apply to the proposed model. Here, we describe three options, other alternatives could be used depending on the application. The boundary can be either hardly or softly constrained.

For hard constraints, we formulate the linear system as the constrained least-squares optimization problem

$$\begin{aligned} \min_{\tilde{x}} \quad & \|L\tilde{x} - b\|_A^2 \\ \text{s.t.} \quad & \tilde{x}_j = x^*, v_j \in \partial S, \end{aligned} \quad (30)$$

where  $b = \text{Div}(|K|^\gamma \text{Grad}(x))$ . Note that the norm is weighted by the area of the Voronoi cells about any vertex. Therefore, the norm can be written as  $\|F\|_A^2 = \text{trace}(F^T A F)$ . The linear equality constraints can be written as  $B\tilde{x} = x^*$ , where  $B$  is a sampling matrix, in which each row is zero, besides those that correspond to the constrained vertices.  $x^*$  is a vector of the final positions of the constrained vertices. Using the Lagrange multipliers approach, the solution to the above problem is given by

$$\begin{pmatrix} L & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \lambda_x \end{pmatrix} = \begin{pmatrix} \text{Div}(|K|^\gamma \text{Grad}(x)) \\ x^* \end{pmatrix}. \quad (31)$$

Similar systems are solved for  $\tilde{y}$  and  $\tilde{z}$ .

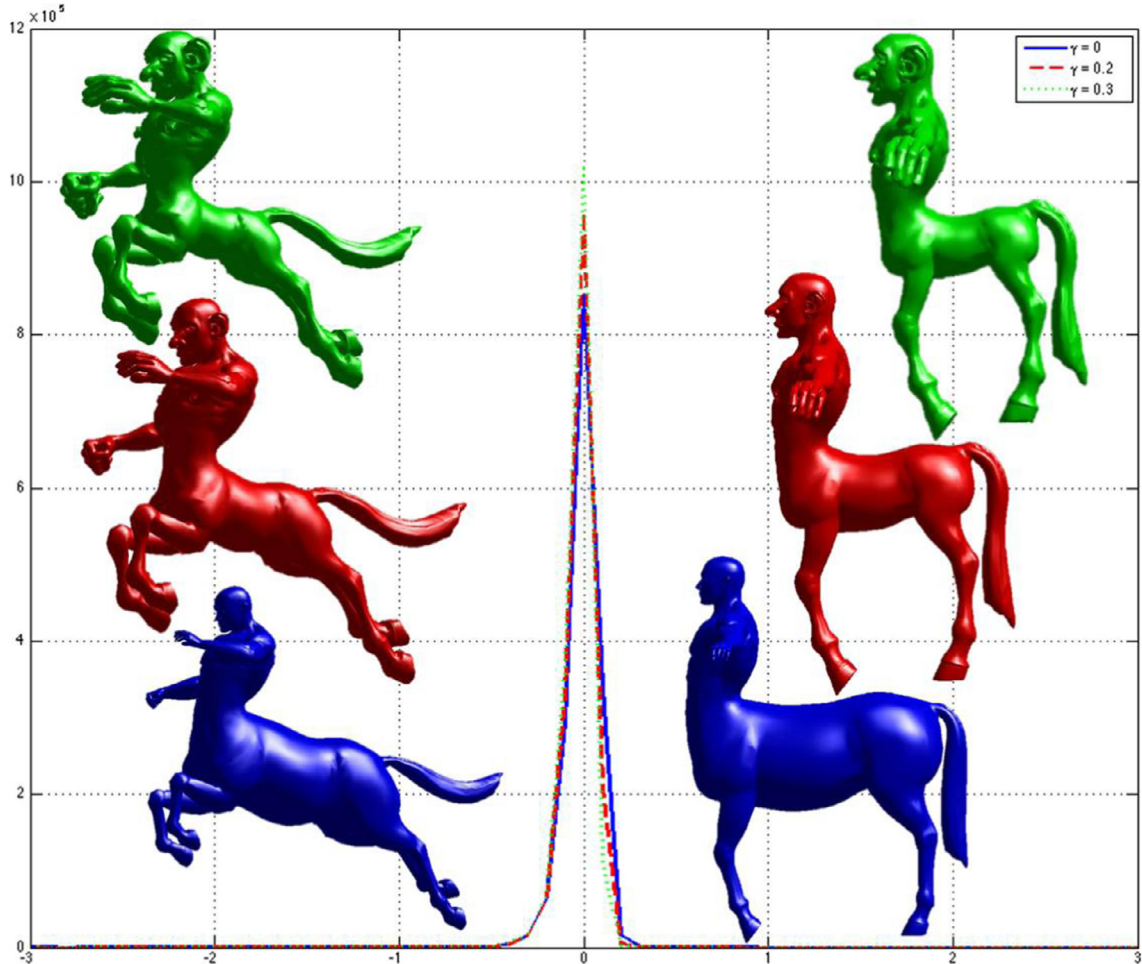


Fig. 15. Histogram of relative geodesic distortion (in %) between caricatures of a Centaur. From bottom to top: almost isometric centaurs, exaggeration factor  $\gamma = 0.2$ , and  $\gamma = 0.3$ .

For soft constraints, one can add a controllable quadratic term for manipulating the boundary vertices and minimize the following unconstrained optimization problem

$$\min_{\tilde{x}} \|L\tilde{x} - b\|_A^2 + \mu \|B\tilde{x} - x^*\|_A^2. \quad (32)$$

The solution to Problem (32) is as follows:

$$\tilde{x} = (LAL + \mu B^T AB)^{-1} (ALb + \mu AB^T x^*). \quad (33)$$

The third type of boundary constraints are set by fixing  $\gamma = 0$  on selected triangles. This technique enables keeping some regions fixed throughout the exaggeration.

## 6. Results and discussion

We evaluate the proposed caricature generation framework by conducting and discussing several example use cases. First, we demonstrate its applicability by caricaturizing a number of synthetic surfaces and captured geometries. Then, we discuss the performance analysis of a real-time implementation of the proposed algorithm. Next, we examine the robustness of the method to almost isometric deformations. We conclude the section by proposing two possible applications.

### 6.1. Qualitative evaluation

We demonstrate the power of the self-caricaturization method by applying it to triangulated meshes. The color of the meshes in Figs. 8 and 9, represents the approximated Gaussian curvature at each point. The caricatures of the armadillo model in Fig. 8 and the face in Fig. 9 demonstrate how regions with high Gaussian curvature scale up, whereas, regions with small Gaussian curvature shrank. Regions with approximately constant Gaussian Curvature were scaled uniformly. These caricatures were created using the geometric driven approach. That is, without a reference model (Fig. 10).

We tested our method on faces and shapes with boundaries, see Figs. 9, 11, and 12. A printed three dimensional replica of the shapes in Fig. 12 are presented in Fig. 14. The resulting caricatures demonstrate an exaggeration of unique features and expressions of the subject's face in a natural comical fashion. The nose was stretched while flat regions like the cheeks, shrank, suggesting the applicability of our method to face recognition (Fig. 13).

As with most mesh deformation techniques, the proposed method may introduce self-intersections, particularly for large exaggeration factors. However, since we solve a Poisson equation, this intersection occurs only globally. Using post-processing self-collision handling methods, such as [30], one can remove these intersections (Table 1).

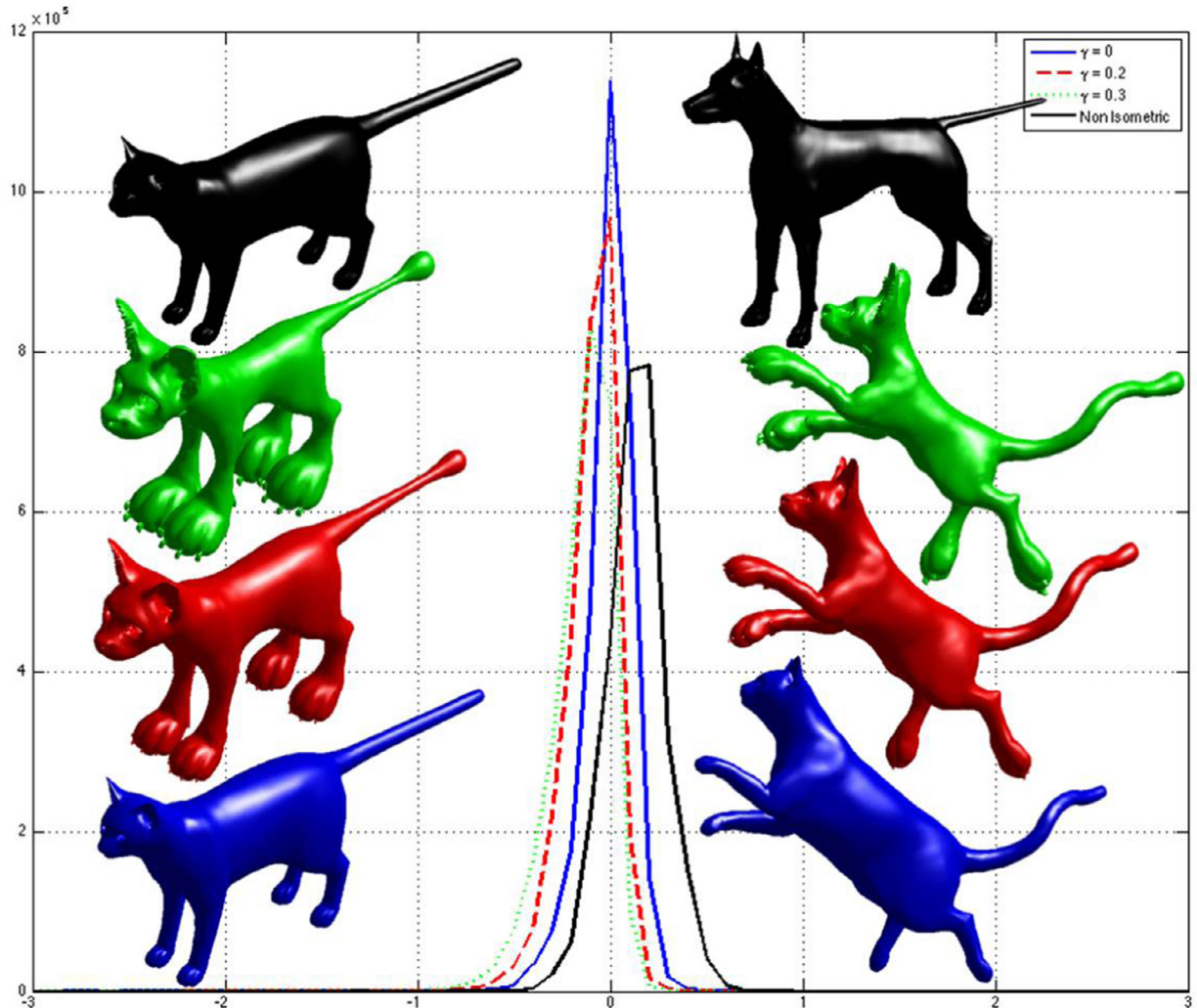
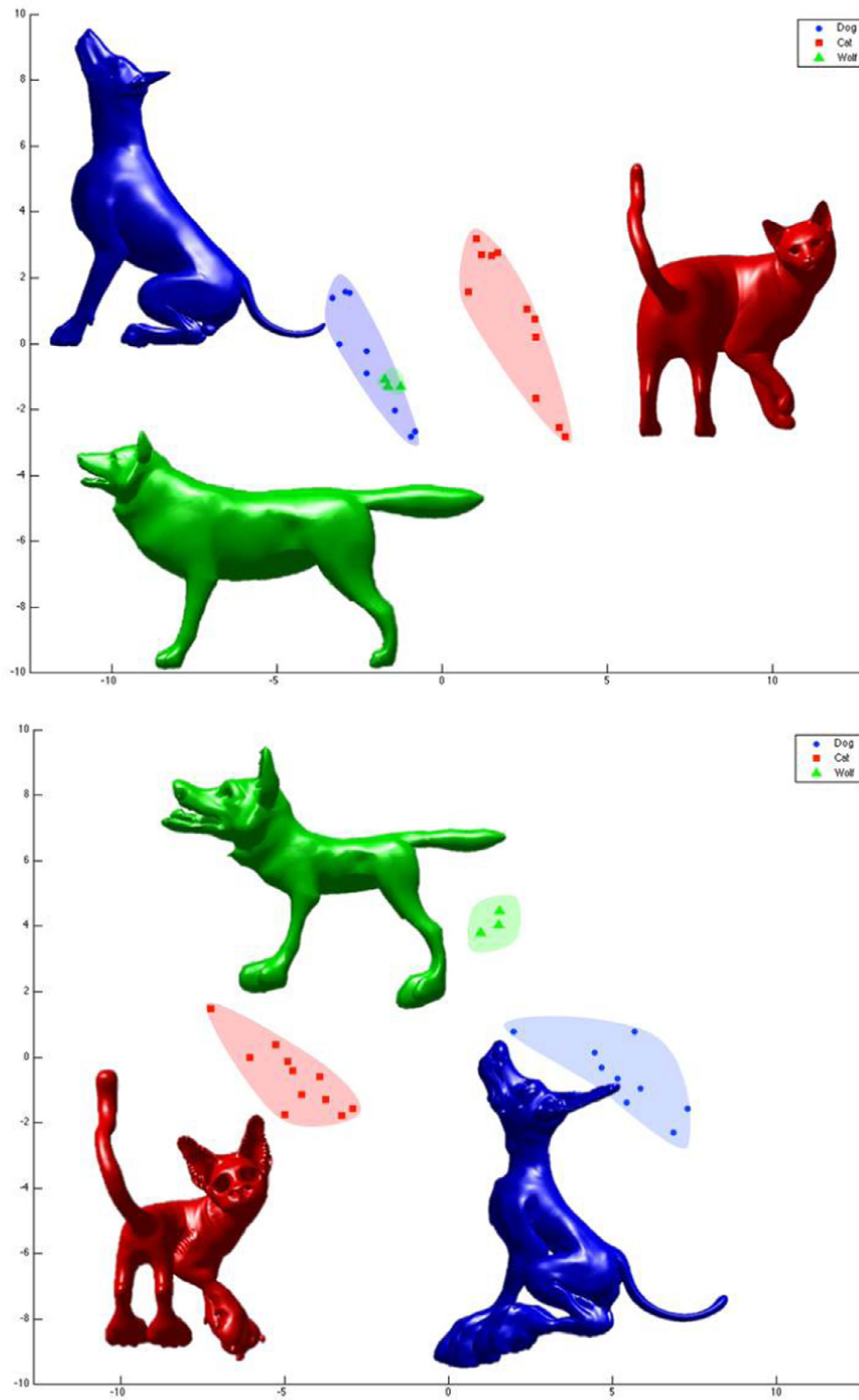


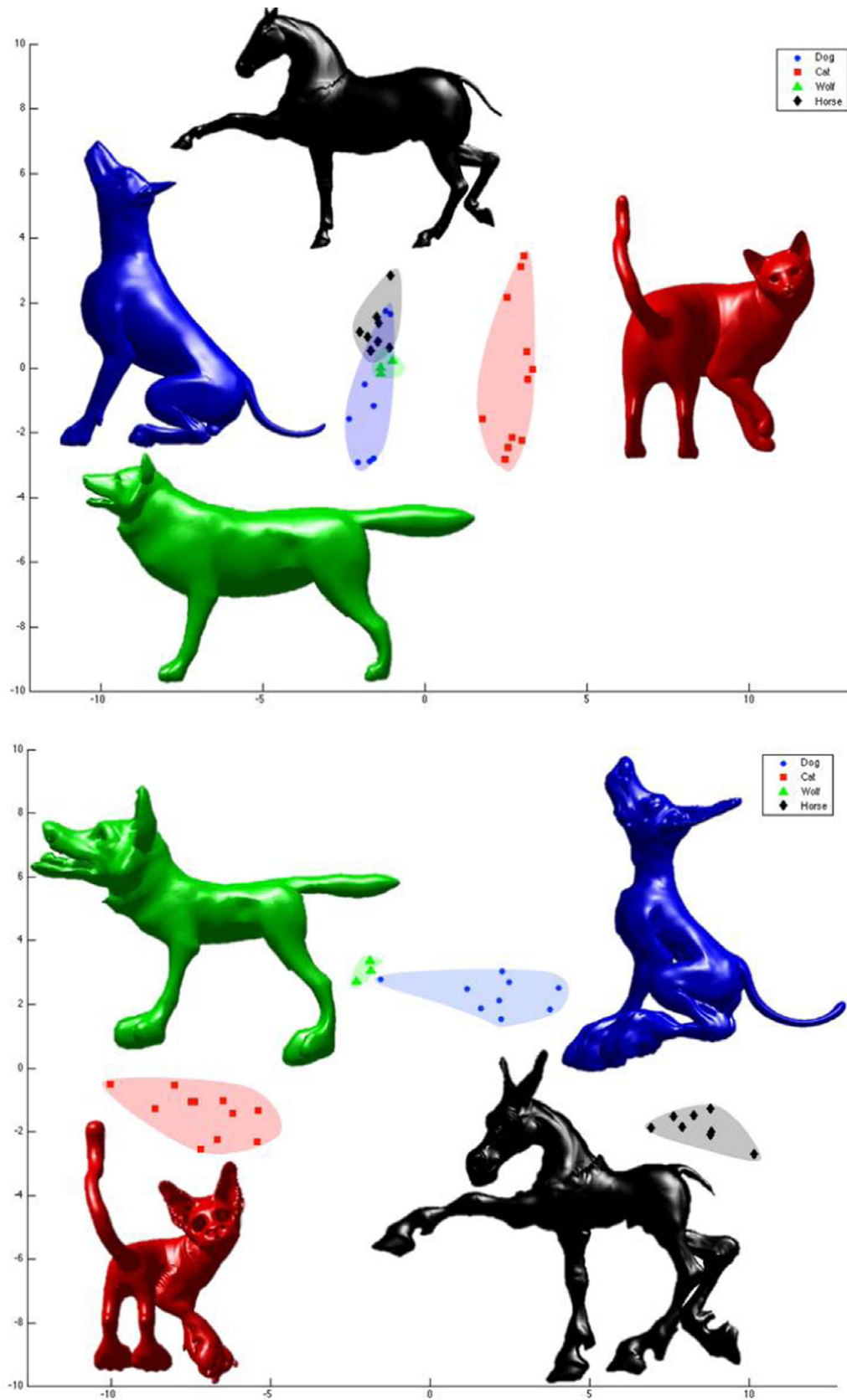
Fig. 16. Histogram of relative geodesic distortion (in %) between caricatures of a cat. From bottom to top: almost isometric cats, exaggeration factor  $\gamma = 0.2$ , and  $\gamma = 0.3$ , as isometric as possible map between a cat and a dog.





**Fig. 17.** 2D Multidimensional scaling of the inter-geodesic distance matrices of dogs, cats and wolves. Top: without caricaturisation. Bottom: caricaturized shapes with exaggeration factor  $\gamma = 0.3$ .





**Fig. 18.** 2D Multidimensional scaling of the inter-geodesic distance matrices of dogs, cats, wolves, and horses. Top: without caricaturisation. Bottom: caricaturized shapes with exaggeration factor  $\gamma = 0.3$ .



Fig. 19. Exaggerated shapes of dogs. Top: different poses (approximately isometric) shapes, and bottom: exaggeration with a factor  $\gamma = 0.3$ .

## 6.2. Run-time analysis

The algorithm was implemented both in Matlab and in C++. For the C++ version, we used the open source version of Eigen Library [31]. All the experiments were executed on a 2.6 GHz Intel core i7 machine with 8 GB RAM. Run-times for various meshes from TOSCA dataset are shown in Table 1. The pre-processing stage includes calculating the Laplacian matrix, the Gaussian curvature, the gradient of the original shape, and matrices for calculating the discrete divergence efficiently. This stage could be implemented more efficiently and in parallel. The core runtime includes taking the power of the curvature in each triangle, multiplying the gradients fields by  $|K|^\gamma$ , and solving the resulting sparse linear system.

## 6.3. Robustness to almost isometric deformation analysis

To evaluate the robustness of the proposed method to almost isometric deformations of shapes, we applied it to shapes in two different poses. We calculated the histograms of normalized distortion of geodesic distances between corresponding points on the two shapes, before and after exaggeration was applied (see Figs. 15 and 16). The results demonstrate that caricaturizing two almost isometric shapes indeed produces two almost isometric caricatures.

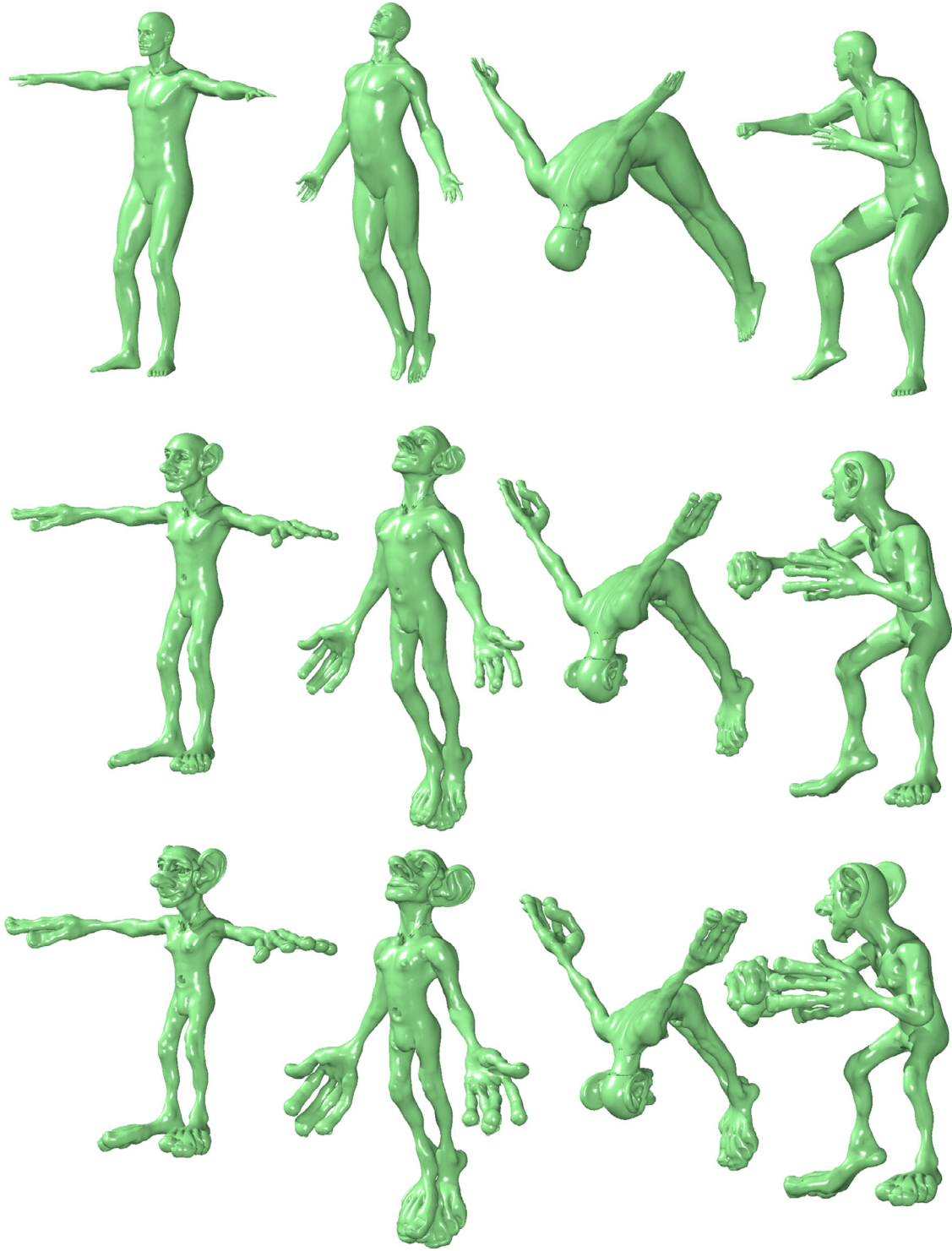
To demonstrate the isometry preservation property, we exaggerated the centaur shape Fig. 4 in two different poses. Figs. 19–21 show examples of exaggerated shapes in various poses for different values for  $\gamma$ .

In order to explore the power of the proposed method on sequences of (naturally nearly isometric) shapes in motion, we generated an artificial motion by linearly interpolating between different poses of the same object. This interpolation was then projected onto the space of isometries by solving a Poisson equation, as described in [32,33]. A video demonstrating the applicability of the proposed framework to animated sequences was produced and is provided in the supplementary material.

## 6.4. Shape classification by exaggeration

A prominent task in shape analysis is the ability of a computer to classify shapes. To facilitate this task we propose to exaggerate the features of the query shapes as a preprocessing step before classification. This idea is justified by an experiment which shows that humans perform better in memorizing and recognizing caricaturized faces than regular ones, [34]. In his book about the human brain, Kurzweil [35] mentions the ability of the brain to recognize the subject behind a caricature. That is, the human pattern recognition ability is robust to some classes of exaggerations.

We now demonstrate the power of caricaturisation in facilitating recognition of shapes undergone various almost isometric deformations. First, we manually sampled corresponding points on quadruped mammals in various poses. We then normalize the shapes and calculate the geodesic distances between the corresponding points. In order to represent each shape as a point in the plane, we applied the Multidimensional Scaling to the geodesic distance



**Fig. 20.** Exaggerated human shape. From top to bottom: almost isometric shapes, exaggeration factor  $\gamma = 0.3$ , and  $\gamma = 0.4$ .

matrices. We repeated the process after exaggerating the shapes. The embeddings of the shapes to the plane are presented in Figs. 17 and 18. The exaggeration is shown to separate the mammal classes, making the classification task simpler.

#### 6.5. Application : exaggeration volume

In order to demonstrate the smoothness of the exaggeration factor, we used the proposed caricaturization method for generating

some musical video clips. We first construct a look-up table of caricatures for equally sampled values of the exaggeration parameter  $\gamma \in [0, 1]$ . For each video frame, we calculate the mean absolute value volume of the audio (sound) within the corresponding time-frame of the song. Then, we find its the closest exaggeration parameter  $\gamma$  in the look-up table and set the video frame to its corresponding rendered caricature. The resulting video clip shows a 3D shape caricaturized in synchronization with the played song that can be animated in real-time as an entertainment feature.



**Fig. 21.** Exaggerated human shape. From top to bottom: almost isometric shapes, exaggeration factor  $\gamma = 0.3$ , and  $\gamma = 0.4$ .

As an alternative to the volume, one can pick the self-exaggeration parameter at each time frame based, for example, on the dominant or median frequency of the sound track. In order for the shape to change temporally in a smooth fashion, we averaged the sound signal by filtering it using a causal low pass filter. The camera position in the shape rendering process can also change in correlation with the music. A video clip demonstrating the proposed application is provided as a supplemental material of this paper.

## 7. Conclusions

We introduced a novel caricature generation framework for surfaces. The method scales the gradient fields of the surface coordinates

by a function of the Gaussian curvature of the surface, and solves a corresponding Poisson equation for finding the exaggerated shape. When a reference shape is provided, local discrepancies are used to amplify the scaling effect, while in the absence of a reference shape, the reference is assumed to be a scaled down version of the given one thereby letting the Gaussian curvature define the relative stretch. We justified analytically and demonstrated empirically that the proposed method is invariant to isometries and thus robust to poses, articulations and postures. This important property allows us to apply the same procedure to a shape in motion (articulation, expression, or pose) while keeping the same perceptual identity of the resulting exaggerated figure. The method was applied to faces and shown to emphasize unique features while reducing flat areas with few-flat



details. In the future, we plan to explore alternative ways to scale the embedding gradients for providing a richer user-interface for computerized caricature generation.

## Acknowledgement

Model sources include: [37] Barack Obama : by Claudio Gasparini and Autodesk [36]; Athena : by Alex Baranov; Max Planck : by Christian Rssl and Jens Vorsatz of Max-Planck-Institut für Informatik; Armadillo: Stanford Computer Graphics Laboratory; This work has been supported by grant agreement no. 267414 of the European Community's FP7-ERC program.

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at [10.1016/j.cviu.2015.05.013](http://dx.doi.org/10.1016/j.cviu.2015.05.013).

## References

- [1] A. Baranov, <<http://www.3dcadbrowser.com/download.aspx?3dmodel=76578>>, (2013).
- [2] T. Wright, A history of caricature and grotesque in literature and art, 1865.
- [3] S.E. Brennan, Caricature generator: the dynamic exaggeration of faces by computer, *Leonardo* 18 (3) (1985) 170–178.
- [4] A.M. Bronstein, M.M. Bronstein, R. Kimmel, Expression-invariant representations of faces, *Image Process. IEEE Trans.* 16 (1) (2007) 188–197.
- [5] V. Blanz, T. Vetter, A morphable model for the synthesis of 3d faces (1999) 187–194.
- [6] E. Akleman, J. Palmer, R. Logan, Making extreme caricatures with a new interactive 2d deformation technique with simplicial complexes, in: *Proceedings of Visual, 2000*, pp. 165–170.
- [7] E. Akleman, J. Reisch, Modeling expressive 3d caricatures, in: *ACM SIGGRAPH 2004 Sketches*, ACM, 2004, p. 61.
- [8] Y.-K. Lim, A. Fedorov, S.-D. Kim, 3d caricature generation system on the mobile handset using a single photograph, in: *Proceedings of the International Conference on Parallel Processing Workshops, 2007. ICPPW 2007*, IEEE, 2007, p. 37.
- [9] L. Clarke, M. Chen, B. Mora, Automatic generation of 3d caricatures based on artistic deformation styles, *Vis. Comput. Graph. IEEE Trans.* 17 (6) (2011) 808–821.
- [10] A. Steiner, R. Kimmel, A. Bruckstein, Planar shape enhancement and exaggeration 1 (1996) 523–527.
- [11] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations, *J. Comput. Phys.* 79 (1) (1988) 12–49.
- [12] M. Eigensatz, R.W. Sumner, M. Pauly, Curvature-domain shape processing, in: *Computer Graphics Forum, 27, Wiley Online Library, 2008*, pp. 241–250.
- [13] B. Kim, J. Rossignac, Geofilter: Geometric selection of mesh filter parameters, in: *Computer Graphics Forum, 24, Wiley Online Library, 2005*, pp. 295–302.
- [14] T. Lewiner, T. Vieira, D. Martínez, A. Peixoto, V. Mello, L. Velho, Interactive 3d caricature from harmonic exaggeration, *Comput. Graph.* 35 (3) (2011) 586–595.
- [15] G. Cimen, A. Bulbul, B. Ozguc, T. Capin, Perceptual caricaturization of 3d models, in: *Computer and Information Sciences III*, Springer, 2013, pp. 201–207.
- [16] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, H.-Y. Shum, Mesh editing with poisson-based gradient field manipulation 23 (3) (2004) 644–651.
- [17] M. Chuang, M. Kazhdan, Interactive and anisotropic geometry processing using the screened poisson equation 30 (4) (2011) 57.
- [18] M.P. Do Carmo, M.P. Do Carmo, *Differential Geometry of Curves and Surfaces*, 2, Prentice-hall Englewood Cliffs, 1976.
- [19] M. Botsch, L. Kobbelt, An intuitive framework for real-time freeform modeling, *ACM Trans. Graph. (TOG)* 23 (3) (2004) 630–634.
- [20] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, H.-P. Seidel, Laplacian surface editing (2004) 175–184.
- [21] M. Botsch, O. Sorkine, On linear variational surface deformation methods, *Vis. Comp. Graph. IEEE Trans.* 14 (1) (2008) 213–230.
- [22] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, B. Lévy, et al., *Polygon Mesh Processing*, 2010.
- [23] T. Weise, H. Li, L. Van Gool, M. Pauly, Face/off: live facial puppetry, in: *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, ACM, 2009, pp. 7–16.
- [24] Y. Tong, S. Lomboyda, A.N. Hirani, M. Desbrun, Discrete multiscale vector field decomposition 22 (3) (2003) 445–452.
- [25] M. Meyer, M. Desbrun, P. Schröder, A.H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, in: *Visualization and Mathematics III*, Springer, 2003, pp. 35–57.
- [26] T. Surazhsky, E. Magid, O. Soldea, G. Elber, E. Rivlin, A comparison of gaussian and mean curvatures estimation methods on triangular meshes 1 (2003) 1021–1026.
- [27] D. Cohen-Steiner, J.-M. Morvan, Restricted delaunay triangulations and normal cycle (2003) 312–321.
- [28] G. Peyre, *Toolbox graph, MATLAB Cent. File Exch. Sel.* 2 (2009).
- [29] U. Pinkall, K. Polthier, Computing discrete minimal surfaces and their conjugates, *Exp. Math.* 2 (1) (1993) 15–36.
- [30] D. Harmon, D. Panozzo, O. Sorkine, D. Zorin, Interference-aware geometric modeling, in: *ACM Transactions on Graphics (TOG)*, 30, ACM, 2011, p. 137.
- [31] G. Guennebaud, B. Jacob, et al., *Eigen v3*, 2010, <<http://eigen.tuxfamily.org>>.
- [32] M. Botsch, R. Sumner, M. Pauly, M. Gross, Deformation transfer for detail-preserving surface editing, in: *Vision, Modeling & Visualization, Citeseer*, 2006, pp. 357–364.
- [33] R.W. Sumner, J. Popović, Deformation transfer for triangle meshes, in: *ACM Transactions on Graphics (TOG)*, 23, ACM, 2004, pp. 399–405.
- [34] R. Mauro, M. Kubovy, Caricature and face recognition, *Mem. Cognit.* 20 (4) (1992) 433–440.
- [35] R. Kurzweil, *How to Create a Mind: The Secret of Human Thought Revealed*, Penguin, 2012.
- [36] C. Gasparini, Autodesk, <<http://www.123dapp.com/123c-3d-model/barack-obama/884967>>, (2012).
- [37] A.M. Bronstein, M.M. Bronstein, R. Kimmel, *Numerical geometry of non-rigid shapes*, Springer Science & Business Media, 2008.