

Perceptual Caricaturization of 3D Models

Gokcen Cimen, Abdullah Bulbul, Bulent Ozguc and Tolga Capin

Abstract Caricature is an illustration of a person or a subject that uses a way of exaggerating the most distinguishable characteristic traits and simplifying the common features in order to magnify the unique features of the subject. Recently, automatic caricature generation has become a research area due to the advantageous features of amusement in the fields such as network, communications, online games, and the animation industry. The aim of this study is to present a perceptual caricaturization approach practicing the concept of exaggeration, which is very common in traditional art and caricature, on 3D mesh models synthesizing the idea of mesh saliency.

1 Introduction

Caricature is a visual art that greatly exaggerates certain features of a subject to create a comic effect. The aim of the caricature is to find individual differences and more attractive features than other parts of the subject and exaggerate these features to create humorous look.

To automatically generate a caricature, a system should attempt to imitate the traditional artistic skills. Identification of the unique features is essential for creating caricatures since these features will eventually be exaggerated. To accomplish this challenge, we employ the idea of mesh saliency, proposed by Lee et al. 2005 [7], which is a technique to measure regional importance of 3D meshes. After extracting salient parts of the mesh feature points, the caricature is generated by warping the original 3D model from original feature points to saliency based exaggerated feature points by using a deformation technique. The deformation of 3D meshes is maintained by Free Form Deformation (FFD) method which is a common method

G. Cimen (✉)
Bilkent University, 06800 Ankara, Turkey
e-mail: gokcen.cimen@cs.bilkent.edu.tr

used for mesh deformation [11]. With automatically calculated saliency values and deformation through these values, the method is fully automatic, but the user can determine the scale of feature exaggeration.

Among existing methods, most of them are based on manipulating 2D facial photographs or images. However, one major drawback of 2D facial caricaturization is that they are limited to make exaggeration with one missing dimension so that every feature deformed on a face is on the same plane. This creates pixel-based deformation, so results undesired distortion artifacts by pulling or pushing adjacent pixels. By performing the caricaturization in 3D space, we have successfully overcome these problems. With respect to this, in this study, we present a perceptual approach applying the exaggeration by using automatically perceptual based feature extraction.

The rest of this paper is organized as follows. In the next section, related work is discussed. In Sect. 3, the overview of the proposed framework is given and the steps of the technique are explained. The results and the conclusions are presented in Sect. 4.

2 Related Works

Susan Brennan's work is perhaps the milestone in formally defining the first caricature generation system in 1982 [3]. After Brennan's work, several cartoon caricature generation methods have been proposed with or without a training process [10]. The basic idea of these approaches is exaggerating the difference from the mean face, but forming a mean face requires large databases of random faces with hundreds of features identified in everyone.

Another proposed approach is training the computer to imitate a specific drawing style from examples. Based on a set of training images and their related hand drawn sketches by an artist, Chen et al. [4]'s method automatically generates an artist's stylitic caricature from an input image. In another study, Liang et al. [8]'s work attempts to learn the style of a caricature artist from example caricatures drawn by the artist using partial least squares. Besides, a recent study managed to generate digital caricatures from facial photographs that capture artistic deformation styles from hand-drawn caricatures by Clarke et al. [5]. Even if this approach observes and learns from the artist's products, they need artist's hand drawn style and can only get the copy of the style and results with limited success.

There have been a number of studies to interactively generate facial caricatures. Akleman [1, 2]. However, the limitation of these techniques is that they require prior knowledge of how a face would be deformed to generate caricature and they are designed for artists or art students. In addition, recently Fu et al. [6] proposed an interactive system that decomposes various facial components of a 3D head model and replace these components with other head and facial components stored in a database. Since these method does not make use of a ref-

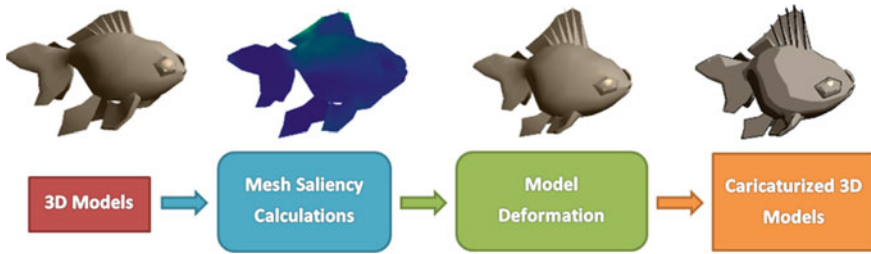


Fig. 1 Overview of the system



Fig. 2 Saliency map for three different 3D models. *Green regions* are more salient than *blues regions*, so they are far more deformed

erence head, the resulting caricature are lack of the likeliness to the original model.

3 Automatic Caricature Generation

Our framework, shown in Fig. 1, includes two main components—mesh saliency calculation and free form deformation. In the saliency calculation step, visually important regions are determined and in the free-form deformation step these regions are enhanced to obtain the caricaturized style.

3.1 Mesh Saliency Calculation

Saliency is the property of objects that attracts attention based on the difference of the object from its surroundings. In Perceptual Caricaturization of 3D Models, for the mesh saliency calculation, a center-surround mechanism proposed by Lee [7], which is a perceptual approach, is applied to determine salient parts of an object. Figure 2 shows results with colorized saliency map of three different models.

Mesh saliency calculation process as follows:

- The first step of saliency calculation involves computing surface curvatures. To compute the curvature of a mesh at a vertex v , the approach of Meyer [9] is applied.
- The Gaussian-weighted average of the mean curvature of a vertex is calculated with the Eq. (1).

$$G(C(v), s) = \frac{\sum_{x \in N(v, 2s)} C(x) \exp[-\|x-v\|^2/(2s^2)]}{\sum_{x \in N(v, 2s)} \exp[-\|x-v\|^2/(2s^2)]} \quad (1)$$

$C(v)$ denotes the mean curvature of vertex v and $N(v, 2s)$ is the neighborhood function which gives the set of points within the distance s for a vertex v .

- Then the computation of the saliency $S(v)$ of a vertex v is calculated by taking the absolute difference between the Gaussian-weighted average computed at fine (small) and coarse (large) scales in Eq. (2)

$$S_i(v) = |G(C(v), s_i) - G(C(v), 2s_i)| \quad (2)$$

3.2 Free Form Deformation

Fundamentally, FFD is a kind of deformation of an object that changes its originally modeled appearance which is known as the object's rest state to other state which is known as deformed state.

To define the lattice space which can be called bounding box in free form deformation technique, bezier volumes are used. Bezier volume defines a volume with the control points within it and can be freely altered to deform the object. Equation (3) shows the calculation for the Bezier volume.

$$Q(u, v, w) = \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 P_{ijk} B_{i,3}(u) B_{j,3}(v) B_{k,3}(w) \quad (3)$$

3.3 Integration

At this stage, the pre-calculated saliency values of the 3D object are mapped to the control points of the bounding box surrounding the object. Then the control points are extended from an estimated pivot points according to new calculated saliency values. To accomplish this, the algorithm works in three steps:

1. The Eq. (4) calculate the saliency value for one control point by computing the weighted average of the saliency values ($s_v(i)$) of the vertices of the object within an estimated distance threshold.

$$S_c(j) = \frac{\sum_{i=0}^m \frac{S'_v(i)}{d(P'_v(i)P_c(j))^2} * (\max(d) - \min(d))^2}{m} \quad (4)$$

In the equation, $P_c(j)$ gives the position of the control points. $S'_v(i)$ is the saliency values and $P'_v(i)$ are the positions of the objects vertices under the estimated distance threshold. If the distance between the control point and vertex is smaller than the threshold, the saliency value of this vertex is taken into account for calculation of the saliency value of the control point.

$$P'_v(i) = \{P_v(i) | d(P_v(i); P_c(j)) < \text{distThres}\}$$

$$S'_v(i) = \{S_v(i) | d(P_v(i); P_c(j)) < \text{distThres}\}$$

2. Pivot points are calculated to determine the exaggeration direction of the control points. Equation (5) shows the calculation of the pivot point position for one control point.

$$V_c(j) = \sum_{i=0}^n \frac{P'_v(i)}{n} \quad (5)$$

3. After saliency values of all control points are calculated with the Eq. (4) and virtual pivot points of them are computed with the Eq. (5), Eq. (6) gives the calculation for the new positions of the one control point.

$$P'_c(j) = \frac{F_d * (P_c(j) - V_c(j)) * S_c(j)}{\max(S_c) - \min(S_c)} \quad (6)$$

In the equation, F_d is the deformation factor which is the only parameter that can be changed by user is used to determine the scale of the exaggeration of the caricaturized 3D model.

4 Results and Conclusion

Figures 3 and 4 show the results of our technique on different 3D models. Each figure starts with the saliency colorization of the models and through (b) to (d), the deformation factor F_d is incremented respectively. As the deformation factor is increased, the salient parts of the models are far more deformed and results in more exaggerated caricaturized models.

In the Perceptual Caricaturization of 3D Models, we have implemented the exaggeration concept by using free form deformation technique with respect to the mesh saliency by trying to extract perceptually more attractive features of the 3D objects to imitate an artist thinking when drawing caricature. This technique can be extended to generate multiple caricatures which can be employed in caricature animation and non-photorealistic animation in the future.

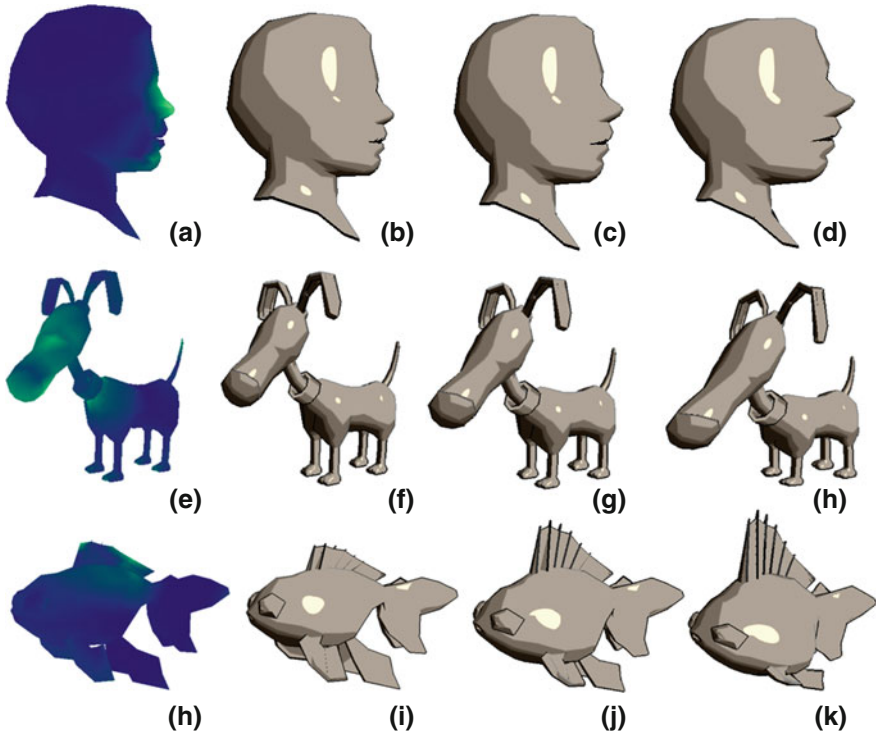


Fig. 3 Image **a** shows saliency. Images from **a** to **d** shows deformation at different deformation factors

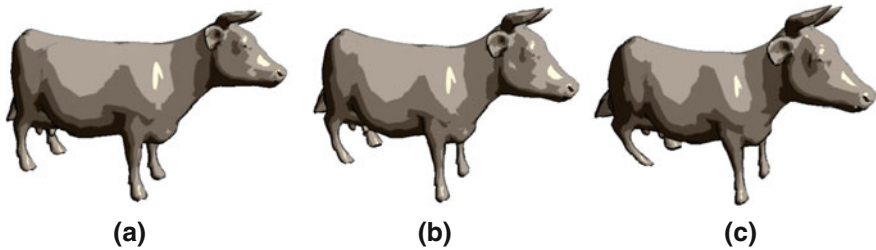


Fig. 4 Images from (a) to (c) shows deformation at different deformation factors

Acknowledgments We thank Ufuk Celikkan and Bengu Kevinc for their support during our study. This work is supported by the Scientific and Technical Research Council of Turkey (TUBITAK, Project number: 110E029).

References

1. Akleman, E.: Making caricatures with morphing. In: ACM SIGGRAPH 97 Visual Proceedings: The Art and Interdisciplinary Programs of SIGGRAPH'97. SIGGRAPH'97, p. 145 (1997)
2. Akleman, E., Reisch, J.: Modeling expressive 3D caricatures. In: ACM SIGGRAPH 2004 Sketches. SIGGRAPH '04, p. 61 (2004)
3. Brennan, S.E.: Caricature the generator: faces dynamic exaggeration of by computer. *Leonardo* **18**, 170–178 (1985)
4. Chen, H., Xu, Y.Q., Shum, H.Y., Zhu, S.C., Zheng, N.N.: Example-based facial sketch generation with non-parametric sampling. In: ICCV '01, pp. 433–438 (2001)
5. Clarke, L., Chen, M., Mora, B.: Automatic generation of 3D caricatures based on artistic deformation styles. *Vis. Comput. Graph. IEEE Trans.* **17**(6), 808–821 (2011). doi: 10.1109/TVCG.2010.76
6. Fu, G., Chen, Y., Liu, J., Zhou, J., Li, P.: Interactive expressive 3D caricatures design. In: Proceedings of Conference IEEE Multimedia and Expo, pp. 965–968 (2008)
7. Lee, C.H., Varshney, A., Jacobs, D.: Mesh saliency. *ACM transactions on graphics. Proc. SIGGRAPH* **24**(3), 659–666 (2005)
8. Liang, L., Chen, H., Xu, Y.Q., Shum, H.Y.: Example-based caricature generation with exaggeration. In: 10th Pacific Conference on Computer Graphics and Applications PG'02, p. 386 (2002)
9. Meyer, M., Desbrun, M., Schroder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. *Vis. Math. III* **3**(7), 35–57 (2003)
10. Mo, Z., Lewis, J.P., Neumann, U.: Improved automatic caricature by feature normalization and exaggeration. In: Proceedings ACM SIGGRAPH 2004 Sketches, p. 57 (2004)
11. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. *Proc. ACM SIGGRAPH* **20**, 151–160 (1986)