Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Portrait drawing from corresponding range and intensity images*

Lu WANG[†1], Li-ming LOU[†1], Cheng-lei YANG[1], Yue-zhu HUANG[2], Xiang-xu MENG[1]
([1]*Department of Computer Science and Technology, Shandong University, Jinan 250101, China*)
([2]*Department of Public Security of Fujian Province, Fuzhou 350003, China*)
[†]E-mail: luwang_hcivr@sdu.edu.cn; lou.li.mingllm@gmail.com
Received Dec. 29, 2012; Revision accepted May 9, 2013; Crosschecked June 6, 2013

Abstract: We propose a real-time rendering system for automatically creating a caricature drawing, i.e., an exaggerated portrait, of a human face, based on simultaneous use of a range image (or 3D mesh) and a registered photograph of the same face. Combining these information sources provides complementary information. Significant geometric lines such as occluding contours and suggestive contours are extracted from the range data, while textured areas corresponding to shading features are extracted from the photograph. These are combined, and then distorted to produce the final caricature. The final output may be produced using a choice of non-photorealistic rendering styles. Our system method works well for low resolution range images; for these it is fast enough to allow the viewpoint to be chosen in real time. The final output combines significant lines, textured areas, and optional shading, giving a pleasing result which preserves not only the shape cues of the geometric description, but also other essential visual characteristics of the facial image that cannot be deduced from geometry alone.

Key words: Portrait drawing, Line drawing, Geometric lines, Textured lines, Caricature, Exaggeration
doi:10.1631/jzus.CIDE1306 **Document code:** A **CLC number:** TP391.9

1 Introduction

Human faces are fascinating and diverse. Many styles of portrait drawings exist, and portraiture is one of the main aims of art. Some pictures attempt to achieve high fidelity, while others are greatly simplified and further, deliberately exaggerated to produce a caricature style result that tries to capture the essence of the subject with a feeling of humor or sarcasm (Rosin and Collomosse, 2013). Experienced artists often go far beyond likeness, and try to combine the character and appearance of the subject through very simple, yet powerful sketches which exaggerate facial features as interesting portrait caricatures. However, creating such portrait drawings by hand is time consuming and requires considerable skill. In this study we consider a real-time interactive system for creating distinctive portrait drawings, capable of exaggerating the differences between the subject face and an average face. As illustrated in Fig. 1, we take a range image (or a 3D mesh) and a simultaneous photograph as the input, and the output is a cartoon-style drawing from a user-chosen viewpoint. The benefit of combining these two information sources, which to our knowledge has not been done before for portraiture, is that the information contained in each source is complementary. Important lines, particularly those that are viewpoint dependent, can be extracted only from a 3D range image (or mesh), while other important information can be found only in an intensity image (e.g., to distinguish the white/black parts of the eyeball).

Our system is aimed to produce portrait caricatures in real time when, for example, the user changes the viewpoint of the drawing, or the extent of

* Project supported by the National Natural Science Foundation of China (Nos. 61003149 and U1035004) and the Natural Science Foundation of Shandong Province, China (No. ZR2010FQ011)

exaggeration and the rendering style are altered. In practice, this means that the resolution of the model should be kept within moderate limits; fortunately, this is compatible with our objectives of producing a fairly simple, cartoon-style drawing. Nevertheless, it does mean that we must pay close attention to how to extract information robustly, particularly from the range image.

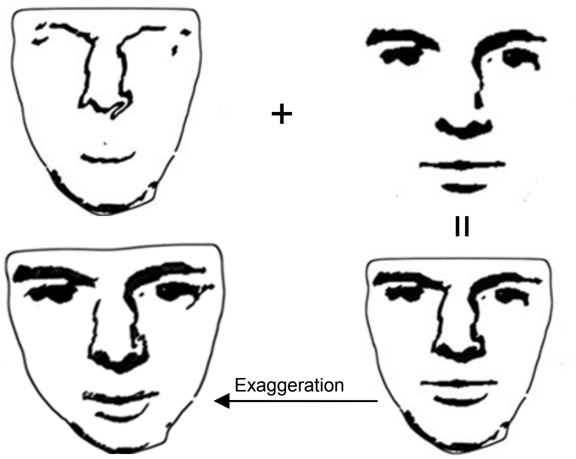


Fig. 1 Portrait caricatures production by adding feature lines from geometric and textured areas from a photograph

In the following, we refer to information acquired from the photograph as textured areas (e.g., the colored part of the eye, and eyebrows), while information extracted from the range image is generally in the form of lines such as suggestive contours and occluding contours, which we refer to as geometric lines.

When rendering the final result, the system provides area strokes and line strokes for stylization. Area strokes are triangle texturing computed by analyzing the geometry and image intensity information around each feature vertex, especially in the areas around the mouth and eyes. In contrast, line strokes are purely based on rendering geometric lines without texture mapping. We incorporate various portrait-drawing principles to provide several simple, yet effective, non-photorealistic portrait renderers such as a pen-and-ink shader, a hatch shader, and a sketch shader. These are able to generate various life-like impressions in different styles from different viewpoints. To obtain more coherent and satisfactory results, we use smoothing changes in line thickness and opacity to refine rendered output.

The main contribution of our research is to show how 3D range image data and 2D intensity image data are complementary, when producing caricature portraits. Combining geometric lines and textured areas gives a better basis for non-photorealistic rendering (NPR) portrait drawing than using a range image or intensity image alone. Using a range image as a basis allows viewpoint dependent lines to be drawn, in a way that cannot be followed by using an intensity image. Using a 2D intensity image allows non-geometric information, or information whose geometric boundaries are hard to determine, to be added to the result. Clearly, such an approach is applicable to drawings other than portraits. We also suggest an approach to facial feature exaggeration for caricaturing a face, based on modifying the original face relative to an average face. While such variations from the average have been suggested before, we use a different approach to combine the exaggeration applied to different facial features. We note that exaggeration can be performed before or after the extraction of the feature lines.

2 Related work

2.1 Line drawings

Line drawings have been an important medium for artistic expression, scientific illustration, and entertainment graphics (Isenberg *et al.*, 2006). They are typical of quick or rough sketches, conveying the most important information about the object being drawn. The object's silhouette is used to emphasize shape and form perhaps with secondary use of shading or texturing. Computer graphics researchers have extensively studied the automatic generation of such lines. In particular, NPR has focused on two main input sources for generating lines for such drawings: 2D images (photographs) (Frowd *et al.*, 2007; Kang *et al.*, 2007; Rosin and Collomosse, 2013), in which case rendering is generally limited to producing a drawing from the same viewpoint as the photograph itself, and 3D range images (or meshes), which allows for the computation of viewpoint-dependent outlines and silhouettes. Cole *et al.* (2008) and DeCarlo (2012) estimated that the feature lines that can be automatically extracted from 3D models, such as occluding contours, ridges and valleys, suggestive contours

(DeCarlo *et al.*, 2003), apparent ridges (Judd *et al.*, 2007), demarcating curves (Kolomenkin *et al.*, 2008), and Laplacian lines (Zhang *et al.*, 2009), account for about 80% of artistic lines in sketches, and play an important role in shape perception, but concluded that none of them work best for all cases. Therefore, this type of research tends to be guided by the specific application area, where certain types of feature lines are preferred.

Here, we are specifically interested in line drawings for portraiture. Although much research has been done on producing NPR line drawings, computer generated portraiture remains a challenging task. Most previous 3D model based NPR line drawing methods do not use prior knowledge of the expected structure of the object (a face has two eyes, one nose, and so on). By taking advantage of our knowledge of facial structure we can, for example, draw different parts with different methods or even go further to exaggerate the distinctive characteristics responsible for an individual's unique appearance in portrait drawing. For example, we use both textured areas and geometric lines to represent the eyes, while geometric lines alone are adequate to depict the shape of the nose.

2.2 EDFM caricature generation

How do humans draw caricatures? Brennan (1985) pointed out that to draw caricatures one should try to visualize a completely average face, and then make those features that differ most from the average more distinguished. Exaggerating the difference of features from the mean (EDFM) (Fujiwara *et al.*, 2001), is widely accepted among caricaturists to be the main approach to caricature generation. This method usually asks for precisely building the correspondence between the current face and the generic model. Unlike the traditional EDFM method which applies a single scaling parameter to the whole face, we separately consider each facial component to be exaggerated, and determine an appropriate exaggeration factor from the mean, and then deform the underlying mesh or feature areas using an ellipsoidal region of influence. During the transformation of the model from the original face to a new characterized face, the user is able to control the extent to which the face is exaggerated. This can range from the original face, to exaggerated caricature, until unrecognizable.

This example-based caricature generation using exaggeration provides an alternative way to create caricatures from the EDFM method by teaching the program certain styles using training data (Liang *et al.*, 2002; Chen *et al.*, 2004; Xu *et al.*, 2008). A caricature is produced by algorithmically calculating the difference of the mean face shape to the subject face shape, and the difference of the caricature face shape (drawn by the artist) to the subject face shape. Features are exaggerated according to the exaggerations produced by the training data. As a result, such methods are perhaps too inflexible, restricting artistic freedom, as they depend too heavily on the training data. They are nearly always restricted to 2D because of the difficulties of acquiring 3D training data.

Our approach takes 3D range data (or a mesh) as input, allowing portraits to be drawn from different views, unlike photograph-based methods, which usually draw a frontal caricature. Chen *et al.* (2006) extended the original 2D model to stereo by fusing caricature images but needed the target object obtained from different views. Other 3D based approaches (Boyer, 2005) produce a face mesh by comparing a face to a canonical model, and consider mainly how to exaggerate the face shape, taking little account of the techniques of traditional portrait drawing. Clarke *et al.* (2011) focused on 3D cartoon face exaggeration by characterizing artistic styles of given caricatures with a pseudo stress-strain model and cloned such styles to an input facial image. But the training data is difficult to obtain from a number of 3D caricatures drawn by professionals. Also, the exaggeration was performed at global levels, so the results lack sufficient characteristics of caricatures.

3 Portrait drawing system

Our portrait drawing system works in three stages (Fig. 2). The input range image is preprocessed to give a triangle mesh (clearly, a triangle mesh can also be used directly as input). The feature areas of the face such as the eyes and the mouth are marked up by the user in the accompanying registered 2D intensity (texture) image, as a basis for making a comparison to the average face. Then, geometric lines representing features are extracted from the mesh, and textured areas are extracted from the intensity image. These

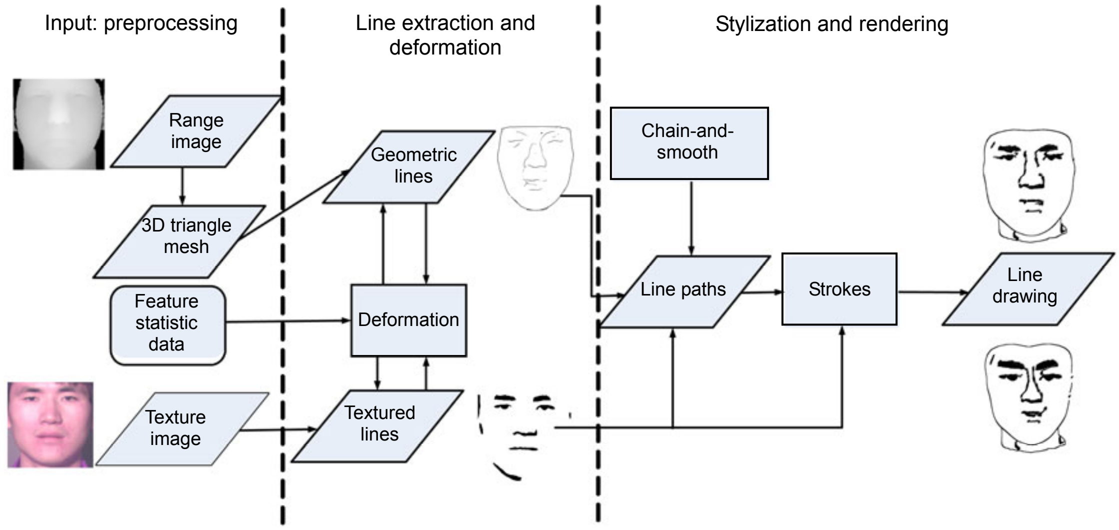


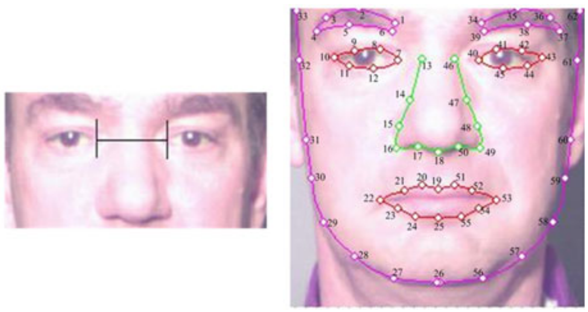
Fig. 2 Workflow of our portrait drawing system

are then deformed to make those facial features significantly different from the average face even more distinctive. Finally the geometric lines and textured areas are rendered, using strokes as the basic rendering primitive, with stylistic control over such parameters as opacity and width.

3.1 Preprocessing

During the preprocessing stage, to register the facial components and standardize the input, if necessary, we rotate the face about the tip of the nose so that the line between the inner corners of the eyes is horizontal, and rescale the face so that the inter-ocular distance is a unit distance (Fig. 3a). All other distances are now measured by normalizing this unit distance. Since we are interested only in the face itself, we can trim irrelevant parts of the range image such as hair and neck (producing suitable renderings of the hair from range and texture images is a separate, and difficult, problem). Blake (2006) suggested that the facial area usually takes five times the unit distance in width and seven times the unit distance in height. Here we first set the nose tip as the center and then use the unit distance to automatically attain the facial part model. Generally speaking, the system takes a corresponding range and intensity image acquired from any range scanner as the input. The range image can be converted into a triangle mesh after triangulation using non-uniform sampling, which takes more samples around the feature area while including all the 62 feature points.

To determine an average face, a series of faces are needed; we have range data and corresponding intensity images from the Biometrics Database of the University of Notre Dame (Flynn *et al.*, 2003). We choose 100 representative symmetry range images (50 males and 50 females), and manually, roughly mark 33 feature points on the texture image, which allows us to determine a total of 62 feature points (or landmarks) for each face by symmetry (Fig. 3b). For asymmetrical faces, we need to mark all the 62 feature points. These feature points can also be marked using the active shape model (ASM) algorithm (Chen *et al.*, 2004). Some feature points, especially around the nose, can be automatically detected from the range image, but areas like the eyes and the mouth are hard to accurately detect for the type of limited-resolution (220×220) depth maps we are using. These feature points delineate the main facial components and character: global face shape, eyebrows, eyes, nose, and mouth.



(a) (b)

Fig. 3 Inter-ocular distance (a) and landmarks (b)

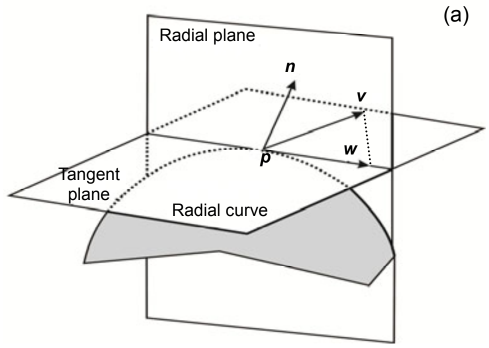
We determine the average location of each landmark, the character of the facial component, and the corresponding standard deviation in position. We now consider any feature character whose difference from the average is larger than the standard deviation to be a distinctive feature, which should be exaggerated in the final output.

3.2 Feature extraction

As noted, to achieve real-time performance, we start with limited-resolution range (and intensity) images, which makes precise extraction of detailed geometry difficult, or impossible for small-scale features such as double eyelids or wrinkles. Further problems are caused by self-occlusion and other issues, resulting in noisy or even missing depth data. We have thus devised methods capable of working under these particular circumstances—we rely partly on making geometric line extraction more robust, and partly on the intensity image to supply additional information. For example, it might be possible to detect the locations and extent of the eyebrows in high-resolution geometric models, but not in low-resolution models. Here we determine them from the 2D image instead.

3.2.1 Geometric lines

We regard occluding contours and suggestive contours as the most useful shape cues for use in portrait drawing. Suggestive contours fall on surface faces where the radial curvature of the surface is zero. They are defined as lines connecting surface points whose radial curvature is zero: the view vector $\boldsymbol{v}$ is projected onto the local surface tangent plane at point $\boldsymbol{p}$ to obtain projected view direction $\boldsymbol{w}$; the radial plane spans $\boldsymbol{w}$ and surface normal $\boldsymbol{n}$, slicing the surface along the radial curve, whose curvature is the radial curvature (Fig. 4a). Computing the radial curvature requires estimation of the principal curvature and principal direction at each point. Current 3D feature line detection methods such as those in DeCarlo *et al.* (2003) require a smooth mesh as input and cannot be reliably applied directly to noisy range images—typically, there are too many noisy lines. One might consider using the mesh smoothing method (Sun *et al.*, 2007) to eliminate the noise, but while mesh smoothing is repeatedly used to eliminate the noise, some features of the face which are




```
Input: range image M                                              (b)
Output: extended suggestive contour lines
Preprocessing: Convert M into triangle mesh M'
For each visible face in M'
    Find reliable suggestive contours
End
For each reliable suggestive contour
    For each ring of neighborhood up to k neighborhood away
        Extend the suggestive contour if possible
    End
End
```

Fig. 4 Definition of radial curvature (a) and the extended suggestive contour algorithm (b)

essential for portrait drawing will also be eliminated. We thus use an improved feature line detection algorithm (Fig. 4b) which we originally proposed in Huang *et al.* (2009), inspired by the idea of hysteresis threshold (Canny, 1986): if a suggestive contour passes through a triangle, it is very likely to continue to pass through adjacent triangles. Therefore, if there is strong evidence for a suggestive contour in one place, the suggestive contour may be extended to successive triangles even if the contour there is weaker. Suggestive contours are defined as locations with zero radial curvature k_r, where its derivative along the projected view direction $\boldsymbol{w}$ is positive:

$$k_r(\boldsymbol{p})=0,\ \ D_{\boldsymbol{w}}(k_r(\boldsymbol{p}))>0. \quad (1)$$

We use the technique described in Rusinkiewicz (2004) to locally estimate the surface curvatures with their derivatives, and then filter and trim the suggestive contour with thresholds θ and t_d to find the zero-crossing points recognized as the initial reliable suggestive contours:

$$0<\theta<\arccos\left(\frac{\boldsymbol{n}(\boldsymbol{p})\cdot\boldsymbol{v}(\boldsymbol{p})}{\|\boldsymbol{v}(\boldsymbol{p})\|}\right),\ \ t_d<\frac{D_{\boldsymbol{w}}(k_r)}{\|\boldsymbol{w}\|}, \quad (2)$$

where $\boldsymbol{v}(\boldsymbol{p})$ and $\boldsymbol{n}(\boldsymbol{p})$ represent the view direction and normal vector of point $\boldsymbol{p}$, respectively. Our extended suggestive contour algorithm first finds an initial reliable suggestive contour, which will be extended to selected surrounding areas, typically specified into 2-ring area, and then chained and smoothed to generate the line path for further stylization. The results of our method are compared with those produced by the method presented in DeCarlo *et al.* (2003) (Fig. 5).

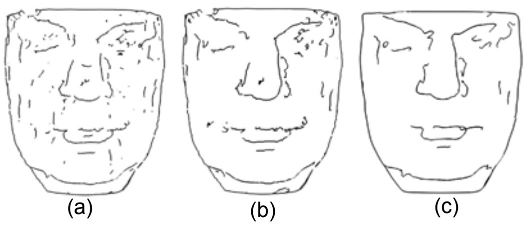


Fig. 5 Suggestive contours

(a) DeCarlo's method; (b) Our extended method; (c) After using the chain-and-smooth method

The system is designed to allow a flexible approach to stylization. After geometric line segments have been extracted, they should be chained to obtain longer, smoother line paths suitable as a basis for rendering. Our approach to chaining is described elsewhere (Huang *et al.*, 2009), based on the idea that a segment should be added only to the end of an existing feature line path if the segment agrees in direction and is near enough. Fig. 5c gives the lines after using the chain-and-smooth method.

3.2.2 Textured areas

Some facial features are hard or impossible to completely detect from a limited-resolution mesh. Nevertheless, certain important textured areas around the eyebrows, eyes, and mouth can be easily detected using the 2D intensity image. Such features are relatively dark compared to skin, and thus a simple approach for their extraction is to set thresholds for the texture image. However, we have found the results of such an approach to be unsatisfactory since they are sensitive to variations in lighting conditions. To overcome this problem, the texture image is first preprocessed to reduce the effects of non-uniform lighting. Some sophisticated illumination correction methods have been proposed in the literature (Rosin and Collomosse, 2013). In this study we use the following simple processing pipeline to extract the complementary textured areas. A morphological opening using a circular structuring element with a diameter of 19 pixels is applied to estimate the background intensity by removing details. This opened image is then subtracted from the original image to generate an illumination-corrected image. A small amount of Gaussian blurring (σ=1 pixel) is further applied since the original texture maps have some striping artifacts. Finally, standard image thresholding with an automatically determined threshold is applied using Tsai (1985)'s moment preserving algorithm.

Fig. 6 shows the elements of the processing pipeline. As can be seen in the top row, direct thresholding of the texture image results in large shadow areas which are connected to facial features such as the eye and eyebrow, whereas more informative results are obtained with our more sophisticated pipeline, as shown in the bottom row. These textured areas are then mapped to the corresponding positions on the surface mesh, which allows them to be rendered from different viewpoints, and to be deformed for caricaturing. While these textured areas are not entirely intrinsic properties of the face (e.g., eyebrow positions, but other textures may be lighting-dependent shadows), they are surprisingly effective in suggesting complicated surface details and appearance without the need for detailed modeling.

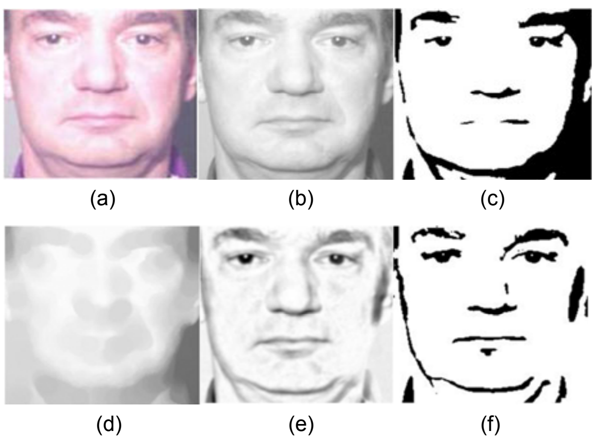


Fig. 6 Textured area pipeline

(a) Face image; (b) Grayscale image; (c) Simple thresholding result; (d) Estimated background intensity; (e) Illumination corrected image; (f) Final thresholded image

3.2.3 Combining geometric lines and textured areas

To produce recognizable and attractive face drawings, we now combine the complementary information provided by the geometric lines and the textured areas, which together give a much more convincing result than using either source of information alone. In particular, we note that the texture information provides extra information around the eyes, eyebrows, base of the nose, and mouth, and under the lower lip, while the geometric lines indicate structure particularly around the nose, eyes, and mouth, as well as the outline of the face and hence its overall shape.

The textured area information is mapped onto the surface using conventional texture coordinates. During the stylization stage, our line generator produces stylized surface strokes based on the projected line paths and textured areas. Figs. 1 and 7 illustrate the procedure of combining geometric lines and textured areas. A drawback of this hybrid approach is that the textured areas are determined from a single image; hence, they cannot be updated when the viewpoint or the lighting environment changes. However, many of the texture features are actually present on the surface (e.g., coloration of the eye), or very close to it (the eyebrows are not very far above the surface of the face), and do not change, or change very little with changes in viewpoint.

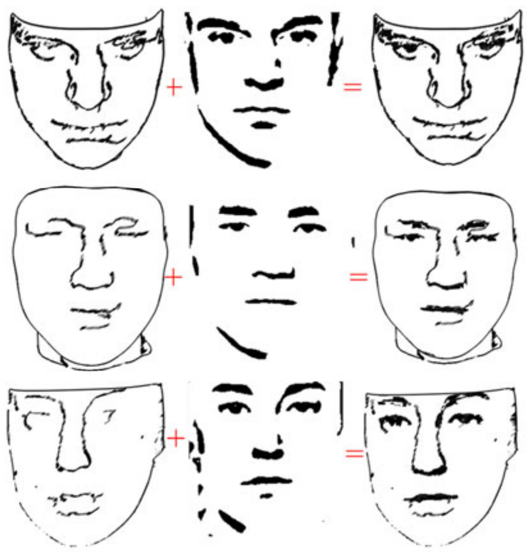


Fig. 7 Combining geometric lines and textured areas

Other less important textured areas may be due to shadows, changing with lighting conditions; these are of secondary importance, and have relatively little effect on perception of the output drawing, unless some rather unusual lighting conditions have been chosen. For example, the size of the shadow under the nose gives some clue as to the geometry of the nose, and this will change somewhat with lighting conditions, perhaps slightly altering the apparent size of the nose itself. However, if the face is illuminated from below, this shadow would be reduced to the region of the nostrils, creating a rather different appearance.

3.3 Exaggeration

Exaggeration is commonly used in art and entertainment to increase recognizability and individuality, and also for such purpose as humour. In computer generated facial caricature, the most common exaggeration method is the EDFM model, which can be expressed as

$$\boldsymbol{v}' = \overline{\boldsymbol{v}} + \lambda(\boldsymbol{v} - \overline{\boldsymbol{v}}), \tag{3}$$

where $\boldsymbol{v}'$ is a vertex of the caricature model, $\overline{\boldsymbol{v}}$ is a vertex of the average model, $\boldsymbol{v}$ is a vertex of the original mesh, and λ is the exaggeration factor. This particular EDFM model works only if each specific input model has the same mesh connectivity as the average model. In practice, mesh models of faces are typically constructed independently, without any such one-to-one correspondence. While producing conforming meshes from a series of models is possible (Soon and Lee, 2006), this is a time consuming process.

3.3.1 Regions to be exaggerated

Here, we take a straightforward approach which has low computational cost. We identify six feature regions based on the landmarks placed on the image (Fig. 3b): two eyes, two eyebrows, a nose, and a mouth; we also identify the global face shape as a seventh region. Deformation takes place in two stages: the six detailed regions are deformed in the first stage, and the seventh region deformation, of global face shape, is applied in the second stage. Since these regions (apart from the overall region for the whole face) are clearly separated by the landmarks, it is easy to prevent undesirable interactions between distortions of the regions, and the regions may be deformed independently. The characteristics of each region (Table 1) are recorded using a set of carefully chosen facial parameters relative to the normalized unit, i.e., the interocular distance, allowing comparison of facial regions to the average face. These 12 selected features have been constructed to capture the feature characteristics recommended in portrait drawing books (Blake, 2006; Freeman, 2006). The height of the nose ridge is converted to the angle η formed by the nose tip and the mid-point of the eyes (Fig. 8).

The feature region boundaries are polygons defined by corresponding landmarks (Table 1). We now perform region-based deformation for each region using the basic idea of EDFM to decide which regions are to be exaggerated and by how much.

To overcome the problem that different faces have different mesh structures for each feature region, we use an ellipsoidal region of influence containing each region to be deformed, which is independent of mesh structure.

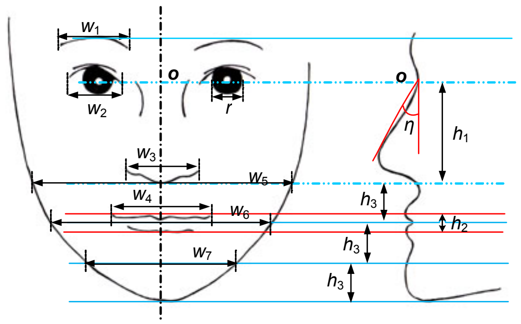


Fig. 8 Face parameters

Table 1 Parameters representing each face region

Region	Landmark points*	Characteristic parameters**
Eyebrows	1–6, 34–39	Location, width w_1
Eyes	7–12, 40–45	Location, width w_2, iris width r
Nose	13–18, 46–50	Location, width w_3, height h_1, nose steepness η
Mouth	19–25, 51–55	Location, width w_4, height h_2
Face shape	26–33, 56–62	Width w_5, w_6, w_7, height h_3

* See Fig. 3; ** See Fig. 8

3.3.2 Ellipsoidal region exaggeration

The basic idea here is to place the point to be exaggerated most at the center of an ellipsoidal region, and to deform that point according to the EDFM model, and other points inside the ellipsoidal region far from the center should be deformed less progressively, until a deformation of zero is produced at the boundary of the ellipsoidal deformation region. This corresponds to λ in the EDFM model dropping to zero, from the center of the ellipsoid, along a radius, to its surface. This ensures that the deformed parts of the face, inside the ellipsoid, still connect smoothly to the undeformed background parts of the face.

By carefully choosing the transition function which determines how λ varies, this ellipsoidal exaggeration model can smoothly deform the mesh within the ellipsoid.

We now describe the approach in detail. The surface of an ellipsoid can be parameterized by θ and φ using

$$\begin{cases} x = x_0 + a \cdot \cos\theta \sin\varphi, \\ y = y_0 + b \cdot \sin\theta \sin\varphi, \\ z = z_0 + c \cdot \cos\varphi, \end{cases} \quad (4)$$

Here a, b, and c are the ellipsoid radii, and $\boldsymbol{o}=(x_0, y_0, z_0)$ is the center of the ellipsoid. Each vertex of the mesh lying inside the ellipsoid is now transformed according to

$$\boldsymbol{v}' = \boldsymbol{o} + \gamma\omega\lambda(r)(\boldsymbol{v} - \boldsymbol{o}), \quad (5)$$

where $\boldsymbol{v}'$ is a vertex position in the deformed model, and $\boldsymbol{v}$ is its original position. ω is the exaggeration factor for this facial feature derived by comparing this feature to the average face and the standard deviation of this feature in a range of faces; facial features are illustrated as the characteristic parameters for each region type in Table 1. γ is a user-controlled parameter used to adjust the amount of exaggeration. Separating γ and ω in this way provides an easier-to-use interface. $\lambda(r)$ is a carefully chosen function which deforms the ellipsoid center by the maximum amount and reduces the deformation to zero at the boundary of the ellipsoid; r measures the fractional distance between the ellipsoid center and its boundary, in any given direction determined by θ and φ. We use a cubic function satisfying

$$\lambda(0) = 1,\ \lambda(1) = 0,\ \lambda'(0) = 1,\ \lambda'(1) = 0. \quad (6)$$

If $\gamma\omega<0$, the feature region is compressed; otherwise, the feature region is magnified.

The key issue in such ellipsoidal region exaggeration is how to determine the center (x_0, y_0, z_0) and sizes (a, b, c) of the ellipsoid. For individual features, the landmarks of each region are used to determine a rectangular bounding box in the image, and diameters $2a$ and $2b$ are set to equal to its width and height, respectively; (x_0, y_0) is the center of this rectangular bounding box. c is typically set to be the z-direction distance range of the region, using $c=h_1\tan\eta$ and $z_0=c/2$. For the global face shape, the ellipsoid is selected to be large enough (typically, $a=w_5$, $b=10h_3$, $c=h_1\tan\eta$) to encompass the whole face, with the center located at the midpoint between landmarks 26 and 27.

3.3.3 Deformation and the algorithm pipeline

Let us consider only the geometric lines and ignore the textured areas. There are two possible ways to produce deformed line drawings (Fig. 9). The first, and correct, approach is to deform the mesh, and then compute the positions of the new suggestive contours and other lines of interest from the deformed mesh. The second approach is to first compute the lines of interest, and then simply deform the lines computed, treating the vertices on these lines as points to be deformed. In general, the second approach is theoretically incorrect; furthermore, some lines extracted from an undeformed mesh may not have correspondence with lines extracted from the deformed mesh, and vice versa.

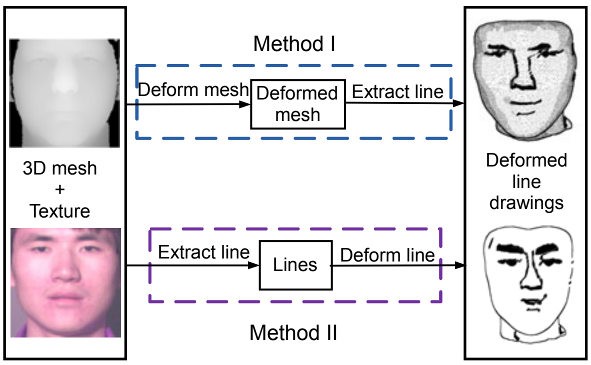


Fig. 9 Alternative placement of deformation in pipeline
Method I deforms the whole mesh while Method II deforms only the feature lines

However, the speed of interaction is greatly improved if we use this latter, incorrect approach, for two reasons. First, the number of vertices on the geometric lines and feature areas is far smaller than that for the whole mesh. Table 2 shows that, for the models used in our experiments, the ratio of the number of feature lines to the total number of triangles is about 10%. Second, we do not have to re-extract geometric lines from the mesh every time the deformation control parameter γ is changed.

Table 2 Numbers of triangles and numbers of geometric lines in the front view and side view for various facial models

Model	Total number of triangles	Number of geometric lines	
		Front view	Side view
02463d452	9028	908 (10%)	726 (8%)
04202d346	8587	698 (8%)	502 (6%)
04202d350	10687	1216 (11%)	975 (9%)
04228d339	7727	851 (11%)	782 (10%)
04267d141	7158	873 (12%)	642 (9%)
04311d182	9170	1501 (16%)	1135 (12%)
04388d197	9935	1021 (10%)	954 (10%)

In practice, to produce faces which are reasonable caricatures of the input face, rather than grossly exaggerated faces, the actual difference in the results produced by these two approaches is not very noticeable (Fig. 10). In fact, the deformed drawing produced by the second approach may even be preferred, since it keeps the same feature lines as the original face.

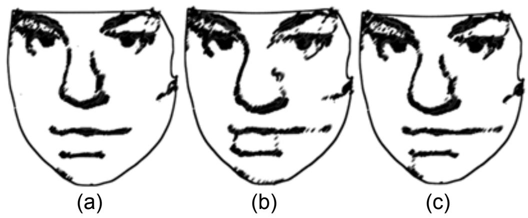


Fig. 10 Caricatures using alternative pipelines
(a) Drawing without deformation; (b) Drawing using deformation at a 'correct' position in pipeline; (c) Drawing using deformation at a suggested position in pipeline

Finally, we note that the only way to deal with the feature areas is by using the second approach, as in some sense they have already been 'extracted' in the process of taking a photograph. We cannot take a new photograph of the deformed mesh as would be required by the 'correct' approach (perhaps this can be simulated, but at a high cost).

4 Rendering performance

Given geometric line segments extracted from the model, and textured areas, now we need to stylize and render them—we need to choose a basic style of rendering, such as hatched lines, and the parameters used to control that style.

In our system, strokes are used as the default rendering primitive, and we provide two kinds of strokes, line strokes for the geometric lines and texture strokes derived from the textured area. Line strokes are used in approximately parallel groups following the underlying line path to delineate the geometric shape, and controlled by varying line width and transparency continuously, as suggested in Huang *et al*. (2009). Texture strokes are rendered as a triangle strip determined by the textured area, providing various details of the object that are not well represented by lines. However, drawing primitives other than strokes may be used to give a variety of NPR styles: for example, a stippling technique using discrete dots as drawing primitives may be used. We provide a pen-and-ink shader, a hatch shader, and a sketch shader—these have been found to be effective for NPR portrait drawing (Fig. 11). We apply the chosen shader to the line paths that have already been computed. The system provides area strokes as well as line strokes for stylization; although area strokes can be optionally drawn using line strokes (Fig. 12), using textured area rendering is much more natural, especially for the eyes. This approach is able to generate various life-like impressions, in different styles, from a user-chosen viewpoint. The results in Figs. 13 and 14 are generated by our system.

Fig. 11 Distinctive exaggerated drawings with alternative styles and viewpoints
(a) Sketch shader; (b) Hatch shader; (c) Hatch shader with more strokes; (d) Pen-and-ink shader

Fig. 12 Rendering the eyes using line strokes (a) or texture strokes (b)

Fig. 13 Original (a) and exaggerated (b) faces

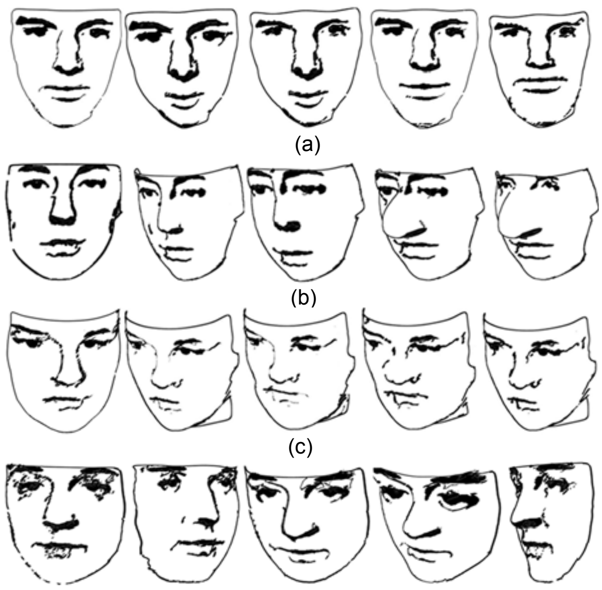


Fig. 14 Various portrait drawings produced by our system
(a, b) Exaggeration of the nose, the mouth, and face shape; (c) Different exaggeration rates and different rendering styles; (d) Different viewpoints and different rendering styles

All experiments are performed on a laptop with a 1.66 GHz processor and 1 GB RAM. The total processing time includes the time for geometric line and textured area extraction, stylistic rendering, and optional deformation. As shown in Table 3, without graphic hardware acceleration or code optimization, our system can generate images for moderate resolution models (of about 10 000 triangles) in less than a second and can thus be used in real-time applications. Model 04311d182 is noisier than model 02463d452, and thus takes more time, although these two models have the same resolution.

Table 3 Time taken to render various facial models

Model	FNum	Time (s)		
		G+R	G+T+R	G+T+D+R
02463d452	9028	0.345	0.386	0.760
04202d346	8587	0.228	0.254	0.422
04202d350	10687	0.567	0.590	0.843
04228d339	7727	0.195	0.236	0.347
04267d141	7158	0.271	0.325	0.372
04311d182	9170	0.779	0.831	0.995
04388d197	9935	0.417	0.458	0.544

FNum: number of triangles. G+R: using geometric lines only; G+T+R: using geometric lines and textured areas; G+T+D+R: using geometric lines and textured areas, and with deformation

We compare our method with the methods presented in DeCarlo *et al.* (2003) and Kang *et al.* (2007) (Fig. 15). Given a noisy 3D mesh or 3D range data (Fig. 15a), DeCarlo *et al.* (2003)'s line detection method obtains poor results (Fig. 15c). Kang *et al.* (2007)'s image-based line drawing method provides clean results, but many features are missed; also, the results are from only one view, and cannot be stylized or exaggerated (Fig. 15d). Results of our method (Fig. 15e) are good for low-resolution range images or mesh, while giving more visual characteristics of the facial image, and the image can be freely stylized and exaggerated (Fig. 14a).

Fig. 15 Comparison of our method with DeCarlo *et al.* (2003)'s line detection method and Kang *et al.* (2007)'s image-based line drawing method
(a) Input 3D mesh; (b) Input texture image; (c) Results obtained using DeCarlo *et al.* (2003)'s method from the 3D mesh; (d) Results obtained using Kang *et al.* (2007)'s method from the texture image; (e) Results obtained using our method

5 Conclusions

We present a system allowing the production of drawings of faces, which simulates the appearance of hand-drawn images and caricatures, starting from a range image and corresponding photograph. The method is a simple, yet effective, procedure for creating stylized 3D models with a high degree of flexibility in selecting artistic flavors. Caricaturing is based on the availability of a library of such inputs to allow exaggeration from the mean. The system allows real-time user control over the production of drawings from models of moderate complexity, after landmarks have been initially marked on the face. In addition to various contours derived from the geometry of the face, significant textured areas are derived from the photograph. These are necessary for non-photorealistic rendering of essentially superficial details such as pupils, eyebrows, and other areas. Compared with the use of either data source alone, this hybrid use of geometric lines and textured areas allows a more comprehensive and comprehensible portrait to be drawn. In the future, we intend to use integral invariants to estimate geometric lines more robustly and consider such features as mustache, beard, eyeglasses, ears, forehead, and hair.

References

Blake, W., 2006. Drawing Portraits: a Step-by-Step Art Instruction Book. Watson-Guptill Publications.

Boyer, V., 2005. An Artistic Portrait Caricature Model. *LNCS*, **3804**:595-600. [doi:10.1007/11595755_72]

Brennan, S.E., 1985. The caricature generator: the dynamic exaggeration of faces by computer. *Leonardo*, **18**(3):170-178. [doi:10.2307/1578048]

Canny, J., 1986. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **8**(6):679-698. [doi:10.1109/TPAMI.1986.4767851]

Chen, H., Liu, Z., Rose, C., Xu, Y., Shum, H.Y., Salesin, D., 2004. Example-Based Composite Sketching of Human Portraits. Proc. 3rd Int. Symp. on Non-photorealistic Animation and Rendering, p.95-153. [doi:10.1145/987657.987673]

Chen, Y.L., Liao, W.H., Chiang, P.Y., 2006. Generation of 3D Caricature by Fusing Caricature Images. IEEE Int. Conf. on Systems, Man and Cybernetics, p.866-871. [doi:10.1109/ICSMC.2006.384498]

Clarke, L., Chen, M., Mora, B., 2011. Automatic generation of 3D caricatures based on artistic deformation styles. *IEEE Trans. Visual. Comput. Graph.*, **17**(6):808-821. [doi:10.1109/TVCG.2010.76]

Cole, F., Golovinskiy, A., Limpaecher, A., Barros, H.S., Finkelstein, A., Funkhouser, T., Rusinkiewicz, S., 2008. Where do people draw lines? *ACM Trans. Graph.*, **27**(3), Article 88, p.1-11. [doi:10.1145/1360612.1360687]

DeCarlo, D., 2012. Depicting 3D shape using lines. *SPIE*, **8291**:829116. [doi:10.1117/12.916463]

DeCarlo, D., Finkelstein, A., Rusinkiewicz, S., Santella, A., 2003. Suggestive contours for conveying shape. *ACM Trans. Graph.*, **22**(3):848-855. [doi:10.1145/882262.882354]

Flynn, P.J., Bowyer, K.W., Phillips, P.J., 2003. Assessment of Time Dependency in Face Recognition: an Initial Study. Audio and Video-Based Biometric Person Authentication, p.44-51. [doi:10.1007/3-540-44887-X_6]

Freeman, J.T., 2006. Drawing Portraits. Crowood Press, UK.

Frowd, C., Bruce, V., Ross, D., McIntyre, A., Hancock, P.J.B., 2007. An application of caricature: how to improve the recognition of facial composites. *Vis. Cogn.*, **15**(8):954-984. [doi:10.1080/13506280601058951]

Fujiwara, T., Koshimizu, H., Fujimura, K., Fujita, G., Noguchi, Y., Ishikawa, N., 2001. 3D Modeling System of Human Face and Full 3D Facial Caricaturing. Proc. 17th Int. Conf. on Virtual Systems and Multimedia, p.625-633. [doi:10.1109/VSMM.2001.969721]

Huang, Y., Martin, R.R., Rosin, P.L., Meng, X., Yang, C., 2009. Expressive line drawings of human faces from range images. *Sci. China Ser. F*, **52**(2):295-307. [doi:10.1007/s11432-009-0039-3]

Isenberg, T., Neumann, P., Carpendale, S., Sousa, M.C., Jorge, J.A., 2006. Non-photorealistic Rendering in Context: an Observational Study. Proc. 4th Int. Symp. on Non-photorealistic Animation and Rendering, p.115-126. [doi:10.1145/1124728.1124747]

Judd, T., Durand, F., Adelson, E., 2007. Apparent ridges for line drawing. *ACM Trans. Graph.*, **26**(3), Article 19. [doi:10.1145/1276377.1276401]

Kang, H., Lee, S., Chui, C.K., 2007. Coherent Line Drawing. Proc. 5th Int. Symp. on Non-photorealistic Animation and Rendering, p.43-49. [doi:10.1145/1274871.1274878]

Kolomenkin, M., Shimshoni, I., Tal, A., 2008. Demarcating curves for shape illustration. *ACM Trans. Graph.*, **27**(5), Article 157. [doi:10.1145/1409060.1409110]

Liang, L., Chen, H., Xu, Y.Q., Shum, H.Y., 2002. Example-Based Caricature Generation with Exaggeration. Proc. 10th Pacific Conf. on Computer Graphic and Application, p.386-393.

Rosin, P., Collomosse, J. (Eds.), 2013. Image and Video-Based Artistic Stylisation. *In*: Computational Imaging and Vision. Springer-Verlag London. [doi:10.1007/978-1-4471-4519-6]

Rusinkiewicz, S., 2004. Estimating Curvatures and Their Derivatives on Triangle Meshes. Proc. 2nd Int. Symp. on 3D Data Processing, Visualization and Transmission, p.486-493. [doi:10.1109/TDPVT.2004.1335277]

Soon, A., Lee, W.S., 2006. Shape-based detail-preserving exaggeration of extremely accurate 3D faces. *Vis. Comput.*, **22**(7):478-492. [doi:10.1007/s00371-006-0023-5]

Sun, X., Rosin, P.L., Martin, R.R., Langbein, F.C., 2007. Fast and effective feature preserving mesh denoising. *IEEE Trans. Visual. Comput. Graph.*, **13**(5):925-938. [doi:10.1109/TVCG.2007.1065]

Tsai, W.H., 1985. Moment-preserving thresholding: a new approach. *Comput. Vis. Graph. Image Process.*, **29**(3):377-393. [doi:10.1016/0734-189X(85)90133-1]

Xu, Z., Chen, H., Zhu, S.C., Luo, J., 2008. A hierarchical compositional model for face representation and sketching. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(6):955-969. [doi:10.1109/TPAMI.2008.50]

Zhang, L., He, Y., Xie, X., Chen, W., 2009. Laplacian Lines for Real-Time Shape Illustration. Proc. Symp. on Interactive 3D Graphics and Games, p.129-136. [doi:10.1145/1507149.1507170]