

**METHOD, DEVICE, AND COMPUTER READABLE MEDIUM FOR GENERATING A
DIGITAL PICTURE**

Technical Field

[0001] Embodiments relate to the field of computer graphics and scientific visualization. By way of example, embodiments relate to methods of extraction of curves for three-dimensional (3D) objects, generation of a digital picture illustrating the 3D objects based on the extracted curves, and stylized animation for conveying 3D objects etc.

Background

[0002] Computer graphics has become a popular medium for science, art, and communication. It has long been defined as a quest in producing realistic images. Research in this area has been very successful. Computers can now generate images that are so realistic that you can not tell them apart from photographs. In so many aspects, however, a non-realistic image is more effective and preferable. Experiences show that, so frequently, an artistic drawing or painting of a same scene is proved to be both more effective in communication and more pleasing in visual experience than a photograph. This is mainly because of the abstraction nature of various artistic styles. More examples can be found in mechanical manuals and medical textbooks, in which illustrations instead of photographs are widely used to explore complicate structures and insides of shapes. Under the rule of "as detailed as necessary while as simple as possible", traditional illustrations are especially effective in communication. It is also shown that, comparing with photograph, non-photorealistic images have many advantages, such as, focusing attention on specific features of shapes

while omitting extraneous details, exposing subtle attributes and hidden parts, clarifying complex scenes with distinct stylistic choices, and simplifying shapes to prevent visual overload etc. A lot of researches in the area of non-photorealistic rendering have been carried out in recent years.

[0003] As one of the most important techniques of traditional art and illustration, line drawings can represent a large amount of information in a highly abstract and relatively succinct manner. Computer generated line drawing has been developed from the beginning of computer graphics. It has improved greatly in recent years as the result of the improvement of non-photorealistic rendering techniques.

[0004] Inspired by the effectiveness in communication and the beauty of perceptual experience of traditional line drawings, extensive research has been carried out in computer-generated line drawings for shapes of 3D objects. Appel proposed methods for silhouettes and sharp features extraction in object space and rendering in image space ("The notion of quantitative invisibility and the machine rendering of solids" in *Proceedings of the 1967 22nd national conference*). Gooch et al. introduced an interactive technical illustration system ("Interactive technical illustration" in *Proceedings of the 1999 symposium on Interactive 3D graphics*). Hertzmann and Zorin proposed algorithm to extract silhouettes on smooth surfaces and generate line drawings automatically ("Illustrating smooth surfaces" in *Proceedings of Siggraph 2000*). To convey the structure and complexity of the interior of 3D shapes, Kalnins et al. proposed a method to draw creases in their WYSIWYG NPR drawing system ("WYSIWYG NPR: Drawing strokes directly on 3d models" in *Proceedings of Siggraph 2002*), and Interrante et al. used ridge and valley lines to

enhance transparent skin surfaces ("Enhancing transparent skin surfaces with ridge and valley lines" in *IEEE Visualization 1995*). Wilson and Ma described a method of representing geometric complexity with line drawing in a pen-and-ink style ("Representing complexity in computer-generated pen-and-ink illustrations" in *Proceedings of NPAR 2004*). Ni et al. focused on multi-scale line drawings from 3D meshes and presented a method to view-dependently control the size of shape features depicted in computer-generated line drawings ("Multi-scale line drawings from 3D meshes" in *Proceedings of the 2006 symposium on Interactive 3D graphics*).

[0005] Many categories of feature lines are widely used in state-of-the-art computer generated line drawing systems. Border lines are one of these feature lines which appear in 3D models where the surface is open. Isophotes are lines of constant illumination on a surface which are also the shading boundaries between toon shading regions. Self intersection lines are where the surfaces of the model intersect. Among these, contours, suggestive contours, ridges and valleys are most representative feature lines in recent years. Contours, normally defined as the visible part of silhouettes, are lines where a surface turns away from the viewer and becomes invisible. Contours show strongest cues with model-to-background distinction. Suggestive contours, which is proposed by DeCarlo et al. ("Suggestive contours for conveying shape" in *Proceedings of Siggraph 2003*), are considered as the extension of the actual contours of surfaces. Suggestive contours are lines drawn on clearly visible parts of the surface, where a true contour would first appear with a minimal change in viewpoint Suggestive contours convey shapes elegantly. Suggestive contour is the pioneer of view-dependent feature lines which are especially

important in animation sequences. Ridge and valley lines, also called crest lines, are curves on a surface along which the surface bends sharply. They are ones of the most powerful descriptors for shape variations which is widely used in CAD/CAM (computer-aided design and computer-aided manufacturing) applications.

[0006] Being the state-of-the-art feature lines, however, none of these lines alone can capture all visually important features. Contours alone are quite limited, since they can not capture the structure and complexity of the shape interior. Suggestive contours can not illustrate salient features in convex regions, while ridge/valley lines are view-independent feature lines which fix on surfaces and appear more like surface marks rather than a natural line drawing in animations.

[0007] Therefore, it is desired to generate a digital picture for 3D objects using an improved method of extracting curves of the 3D objects which can make the digital picture more perceptual consistent and can give users more freedom in achieving desirable line drawings.

[0008] Further, it is also desired to extend the generation of a digital picture for 3D objects to generation of a video sequence of digital pictures with a dynamic setting for stylization of the video sequence of the digital pictures. In other words, it is desired to provide a method of generation of animation with digital picture coherence. In this context, animation refers to a video sequence of digital pictures. Stylization may refer to presentation of a digital picture or a sequence of digital pictures according to a style or stylistic pattern rather than according to nature or tradition. For example, stylization may be characteristic of non-photorealistic applications. In this context, digital

picture coherence may refer to consistence of stylization among different digital pictures within the video sequence.

Summary

[0009] In one embodiment, a method is provided for generating a digital picture, the method including receiving a three-dimensional representation for each of a plurality of curves, wherein each curve represents at least partially a shape of a three-dimensional object; grouping the plurality of curves into at least one curve group based on the three-dimensional representations of the curves using three-dimensional information given by the three-dimensional representations; associating each curve group with a picture element based on the curves of the curve group; and forming the digital picture using the picture element.

[0010] According to other embodiments, a device and a computer readable medium according to the method described above are provided.

[0011] It should also be noted that the embodiments described in the dependent claims of the independent method claim are also analogously valid for the corresponding device and computer readable medium where applicable.

Brief Description of the Drawings

[0012] The drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of various embodiments. In the following description, various embodiments are described with reference to the following drawings, in which:

FIG. 1 illustrates the definition of PEL on a surface in one embodiment;

FIGS. 2 (a)-(d) show an illustration of effect of the PELs;

FIGS. 3 (a)-(d) illustrate the effect of a main light and auxiliary lights on the 3D object for obtaining of the PELs;

FIGS. 4 (a)-(f) illustrate the adjustment of auxiliary light locally on a 3D object;

FIG. 5 (a) shows an example of the vertices (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) ; FIG. 5 (b) illustrates the face coordinate system;

FIGS. 6 (a)-(f) show an illustration of the PELs extracted for different 3D objects at different view points;

FIG. 7 illustrates the process of generating a video sequence which comprises a plurality of digital pictures in one embodiment;

FIG. 8 shows a method in one embodiment;

FIG. 9 illustrates a device in one embodiment;

FIG. 10 illustrates the determination of continuation of two curves;

FIG. 11 illustrates the determination of parallelism of two curves;

FIG. 12 illustrates determination of the potential of the two curves to lead to a closed curve;

FIGS. 13 (a)-(c) illustrate the curve grouping among different digital pictures;

FIG. 14 (a) shows a 3D object; FIG. 14 (b) shows the presentation of extracted curves from the 3D object as shown in FIG. 14 (a) in one style; FIG. 14 (c) shows the presentation of extracted curves from the 3D object as shown in FIG. 14 (a) in another style;

FIGs. 15 (a)-(b) illustrate an example of determination of single curve correspondence between different digital pictures; and

FIG. 16 illustrates a computer according to one embodiment.

Description

[0013] Edge detection is a well-studied problem in computer vision. In 2D images, an edge point is defined as a point at which the gradient magnitude assumes a maximum in the gradient direction. Observation in human vision and perception shows that a sudden change in the luminance plays a critical role to represent and recover the 3D information. Observation also shows that for a grey-scale image of an illuminated 3D object under general illumination and reflection conditions, the zero-crossings of the second order directional derivative of the image intensity along the direction of the intensity gradient occur near the ridges and valleys of the 3D object. Inspired by the edge detection in image processing, in one embodiment, a type of curves, which may be named as Photic Extremum Lines (PEL), are defined which characterize the local variation of illumination directly on surfaces of a 3D object.

[0014] In one embodiment, PEL is a set of points on the surface of a 3D object where the variation of illumination in the direction of its gradient reaches the local maximum.

[0015] FIG. 1 illustrates the definition of PEL.

[0016] Given a smooth surface patch $S : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ in FIG. 1, it may be assumed that the illumination function $f: S \rightarrow \mathbb{R}$ is C^3 -continuous. In this context, $\mathbb{R}^2$ refers to two-dimensional real space, i.e. the vector space of pairs of real numbers. $\mathbb{R}^3$ may refer to three-dimensional real space, i.e. the

vector space of triples of real numbers. C^3 means that the third derivation exists and is continuous.

[0017] Given an arbitrary point $p \in S$ as shown in FIG. 1, n is the surface normal at point P on the surface patch S . The tangent vectors S_u and S_v define a local coordinate system on the point p of the surface patch S , where:

$$S_u = \frac{\partial S}{\partial u}, \quad S_v = \frac{\partial S}{\partial v}$$

If S is a regular patch, then S_u and S_v are linearly independent and form the basis of the tangent space.

[0018] The gradient of f may be defined as:

$$\nabla f = \frac{f_u G - f_v F}{EG - F^2} S_u + \frac{f_v E - f_u F}{EG - F^2} S_v \quad (1)$$

where f_u and f_v are the partial derivatives of the scalar function f , i.e., $f_u = \frac{\partial f}{\partial u}$, and $f_v = \frac{\partial f}{\partial v}$; E , F , and G are the coefficients of the first fundamental form of S , i.e., $E = (S_u, S_u)$, $F = (S_u, S_v)$ and $G = (S_v, S_v)$, where $(,)$ is the dot product. Given an arbitrary point $p \in S$ as shown in FIG. 1, n is the surface normal at point P on the surface patch S , and denote w as the unit vector of the gradient of f on the tangent space of point P on the surface patch S , i.e., $w = \frac{\nabla f(p)}{\|\nabla f(p)\|}$.

Then, the PEL may be defined as the set of points satisfying:

$$D_w \|\nabla f\| = 0 \quad \text{and} \quad D_w D_w \|\nabla f\| < 0 \quad (2)$$

D_w is referred to as the directional derivative operator. For a scalar function g , $D_w g = (\nabla g, w)$ is the directional derivative along w . Intuitively speaking, D_w may measure the changes of the function f along the direction w .

[0019] The entire solutions of $D_w \|\nabla f\| = 0$ form closed curves or curves that terminate at the surface boundaries. Thus, the

surface may be divided into two regions, i.e., the region which satisfies $D_w \|\nabla f\| > 0$ and the region which satisfies $D_w \|\nabla f\| < 0$. In one embodiment, PELs correspond to part of the curves where $D_w D_w \|\nabla f\| < 0$, namely, the variation of the illumination reaches its local maximum. PELs effectively convey useful information about the shape of a 3D object.

[0020] FIGs 2 (a) - (d) illustrate the PELs for illustration of the shape of a 3D object.

[0021] FIG. 2 (a) illustrates the surface 200 of a 3D object. The dotted lines in FIG. 2 (b) illustrate curves which satisfy $D_w D_w \|\nabla f\| < 0$ on the surface 200. The dotted lines in FIG. 2 (c) illustrate curves which satisfy $D_w \|\nabla f\| = 0$ and $D_w D_w \|\nabla f\| < 0$ on the surface 200. FIG. 2 (d) illustrates regions 202 of $D_w \|\nabla f\| > 0$ and regions 204 of $D_w \|\nabla f\| < 0$.

[0022] Referring to the surface 200 shown in FIG. 2 (a), the entire solutions of $D_w \|\nabla f\| = 0$ form closed curves or curves that terminate at the surface boundaries. These curves divide the surface 200 into two regions, i.e., $D_w \|\nabla f\| > 0$ (colored in dark grey in FIG. 2 (d)) and $D_w \|\nabla f\| < 0$ (colored in light grey in FIG. 2 (d)). PELs (dotted curves in FIGs. 2 (b)) correspond to part of the curves where $D_w D_w \|\nabla f\| < 0$, i.e., the variation of the illumination reaches the local maxima. The dotted curves in FIGs. 2 (c) correspond to the points satisfying $D_w \|\nabla f\| = 0$ and $D_w D_w \|\nabla f\| < 0$ which do not convey the useful information about the shape of the 3D object. Note that the curves on the hidden surface are not shown in (b) and (c).

[0023] In one embodiment, the illumination on the 3D surface may be totally user-controllable. In other words, in one embodiment, the user may achieve the desired illustration by manipulating light freely.

[0024] The illumination of 3D shapes depends on lighting conditions. Among various light models available in graphics, a commonly-used one is Phong specular-reflection model:

$$I = I_{amb} + I_{diff} + I_{spec} = k_a I_a + k_d I_d (n \cdot l) + k_s I_s (v \cdot r)^{n_s} \quad (3)$$

where n , l , v , r are the surface normal, light direction, view direction, and reflection direction, respectively; n_s is a shininess constant which is larger for surfaces that are smoother, k_a is ambient reflection constant, I_a is ambient light intensity, k_d is diffuse reflection constant, I_d is diffuse light intensity, k_s is specular reflection constant, I_s is specular light intensity. In this context, surface normal at point P of a surface may be referred as a vector perpendicular to the tangent plane to that surface at P . I_{amb} represents ambient light. In this context, ambient light may refer to the illumination surrounding a subject or scene. I_{diff} represents the diffuse light, which is even, directed light coming off a surface. I_{spec} represents specular light. In this context, specular light may refer to light which reflects more intensely along and around the reflection vector from a surface. Since equation (3) is a function of the light and view directions, the corresponding PELs may be view and light dependent in one embodiment.

[0025] In one embodiment, PELs convey shapes of 3D objects by emphasizing significant variations of illumination over 3D surfaces. Aiming at 3D shapes illustration with concerns of human perceptual experiences, in one embodiment, the light

model may emphasize on illumination variations strongly related to shape variations, and may suppress those less related or orthogonal to shape variations. In Phong shading model, the ambient light I_{amb} is constant which does not contribute to the variation of illumination. The diffuse light I_{diff} is determined by both shape and light. The variation of the diffuse light I_{diff} is highly affected by the variation of the surface normal. The specular light I_{spec} is dominated by the viewing and lighting conditions, which may be considered as an orthogonal factor of the 3D surface. In other words, I_{spec} is not relevant to the shape of the 3D objects. Thus, the specular light I_{spec} may introduce feature lines which are less related to the 3D shape and distracting for 3D shape illustration. On the other hand, the power factor of the specular light I_{spec} is more computational expensive. Thus, in one embodiment, the ambient light I_{amb} and the specular light I_{spec} may be ignored, and only the diffuse light is considered. Thus, from equation (3), the following equation is obtained for the lighting model

$$I = k_d I_d (n \cdot l) \quad (4)$$

[0026] Note that the view vector v is not involved in equation (4). Thus, the corresponding PELs in one embodiment are independent of view position.

[0027] In another embodiment, the view dependent property may be desirable in the real-world applications. More importantly, for real-world 3D shapes with complicated geometry and many details, the criteria to locate the feature lines usually vary in different parts and are rather subjective. Thus, it is desirable to provide users more

freedom to control the desired illustration. To satisfy the view dependence as well as the user controllability and performance requirements, in one embodiment, the following light model for PEL is provided.

[0028] The light model that is used in one embodiment comprises a main light. For example, a main light source is set using a directional light whose light rays are parallel to the view vector. In other words, the term "main light" is similar to the "head light", which is widely used in the lighting system of vehicles. This setting is based on human perceptual experience. Thus, the light moves when the view point changes. As a result, the illumination and the corresponding PELs are view dependent.

[0029] In one embodiment, the light model that is used further comprises at least one auxiliary light. In other words, optional spot lights may be used to highlight the user-specified areas. The contribution of auxiliary lights to the illumination is local and independent of the view position. The auxiliary spot lights may be defined with $n \cdot l$, but applied locally to user interested regions, which provide the user freedom to control PEL in different areas.

[0030] FIGs. 3 (a)-(d) illustrate the application of the light model for extraction of PELs in one embodiment. Stanford Bunny is selected to be the 3D object for the illustration of the effect of the light model.

[0031] FIG. 3 (a) illustrates that four lights are applied to the Stanford Bunny for extraction of PELs for illustration of the Stanford Bunny. The main light #1 is a directional light whose direction coincides with the camera direction. By using the main light #1 alone, PELs may convey the rough shape well (FIG. 3 (b)).

[0032] However, some important lines on the eyes and feet may be missing (as shown in (b)). Then auxiliary spot lights #2 and #3 may be used to increase the local contrast on eyes and feet to add more details (as shown in FIG. 3 (c)). Note that the Bunny body contains many short feature lines due to the small relief. These lines, however, may be less favorable by the user. To remove these lines, another auxiliary spot light #4 as shown in FIG. 3 (a) may be added to decrease the local contrast on the body. As a result shown in FIG. 3 (d), most of the small feature lines are removed successfully.

[0033] In more detail, FIG. 3 (a) illustrates an example wherein four lights are used to illustrate the Stanford Bunny. Light #1 is the main light, whose direction coincides with the view direction. The purpose of the main light #1 is to convey the rough geometry of the Stanford Bunny. Lights #2, #3, and #4 are auxiliary lights, which may be used to increase or decrease local contrast. For example, in this case, auxiliary light #2 is applied locally on the eye area. Auxiliary light #3 is applied on the foot area, and auxiliary light #4 is applied locally on the body area of the Stanford Bunny. In one embodiment, auxiliary lights are used to increase or decrease the local contrast of the specified regions. Thus, PELs may be added or removed according to local features of the 3D objects or the requirements of users.

[0034] FIG. 3 (b) illustrates the rough geometry of the Stanford Bunny by PELs when only the main light #1 is applied. As can be seen from FIG. 3 (b), some important lines, such as lines that may illustrate eyes and foot in more detail, are missing. Also, due to some small features of the Stanford Bunny, i.e. features on the body area, there exist many small PELs which cause distractions and are not helpful to convey the shape in the back leg area.

[0035] FIG. 3 (c) illustrates the geometry by PELs of the Stanford Bunny when the main light #1 and the auxiliary lights #2 and #3 are applied. As can be seen from FIG. 3 (c), two auxiliary lights #2 and #3 are used to increase the contrast on the eye and foot.

[0036] FIG. 3 (d) illustrates the geometry by PELs of the Stanford Bunny when a further auxiliary light #4 is used to decrease the contrast on the body. As a result, better extraction of PELs of the Stanford Bunny is achieved.

[0037] For simple shapes, one may usually achieve the desired illustration using the main light alone. However, most of the real-world models have complicated geometry and many details, which may not be illustrated using a single light. Thus, in one embodiment, auxiliary lights are used to improve the illustration. One advantage of PEL is that users may fully control the auxiliary lights, such as the number of lights, their intensities, positions, and directions. It may be a tedious work to manipulate these parameters manually. In one embodiment, to minimize the effort of user interaction, a method is provided to compute the optimal setting of the auxiliary lights automatically.

[0038] In one embodiment, a user may specify several regions of interest to be improved. Each region is small enough that just one auxiliary light I_{aux} may be used.

[0039] Given a user-specified patch $S' \subset S$, $c = \frac{\int_{S'} S dA}{\int_{S'} dA}$. In

this context, A means that the integral is carried out over an area, and c is the barycenter of the region A .

[0040] In one embodiment, a spot light I_{aux} may be applied at position x with light direction $d = \frac{x-c}{\|x-c\|}$. In one embodiment, the contrast of illumination on S' by spot light I_{aux} is maximized if details are preferably to be added by the user. In another embodiment, the contrast of illumination on S' by spot light I_{aux} is minimized if lines are preferably to be removed by the user. For example, the following formulas may be used,

$$\max \iint_{S'} \|\nabla f\|^2 dA \quad \text{to add details, (5)}$$

$$\text{or} \quad \min \iint_{S'} \|\nabla f\|^2 dA \quad \text{to remove details, (6)}$$

where $f = I_{aux}$,

$$I_{aux}(u, v) = \frac{k_{aux} I_{aux}}{k_c + k_l d + k_q d^2} n(u, v) \cdot l,$$

$$l = \frac{x - S(u, v)}{\|x - S(u, v)\|}$$

$$d = \|S(u, v) - x\|$$

and where $S(u, v)$ is the given 3D surface patch; d is the distance between the light source and the surface S ; k_{aux} is the intensity of the auxiliary light; k_c , k_l , k_q are the coefficients of the quadratic function of distance d ; $n(u, v)$ is the unit normal vector, i.e., $n = \frac{S_u \times S_v}{\|S_u \times S_v\|}$; and l is the unit vector of the Light direction.

[0041] Note that the cut-off angle of the spotlight may give rise to the discontinuity of the illumination which in turns may result in unnecessary feature lines. In this context,

cut-off angle may refer to the angle of the cone shape of the spot light, and is associated with the boundary of the spot light applied on the surface S of the 3D object.

[0042] To avoid additional feature lines caused by discontinuity of the illumination, in one embodiment, the cut-off angle of the spot light may be large enough to cover S' . For each point inside the spot light region (which is larger than S'), two illumination functions $f = I_{main}$ and $f = I_{aux}$ are applied. To compute derivatives of illumination, $f = I_{aux}$ is used for any point $p \in S'$, while $f = I_{main}$ is used for any point $p \in S \setminus S'$, which means that p is a point in S and not in S' . Therefore, though the overlap of the two functions results in discontinuities, local continuity may be guaranteed everywhere by switching between the two functions of $f = I_{main}$ and $f = I_{main} + I_{aux}$. Since PELs are defined with local extremum and the illumination is locally continuous for all the points, the auxiliary light may improve the PEL illustration of S' without introducing unnecessary lines on the boundary.

[0043] In one embodiment, the illumination function is set to be $f = I_{aux}$ for the optimization of the auxiliary light and the extraction of PEL in S' . In one embodiment, the optimal auxiliary lights are located in object space. Thus, the optimal setting of the auxiliary lights is not affected by the change of the main light and various viewing conditions. The optimization process is not real time in one embodiment. In other words, the optimization is carried out in the 3D space before extraction of curves of a 3D object for illustration in a 2D image. In one embodiment, once auxiliary lights are determined, the auxiliary lights are fixed in the object

space, meaning that the setting of the auxiliary lights are view-independent. Thus, once the auxiliary light is computed, the auxiliary light may be used in rendering PELs for any view point. Therefore, additional computational overheads may be avoided. More importantly, the auxiliary lights may improve the PEL illustration without causing any incoherence in animation when view point changes.

[0044] Figs. 4 (a)-(f) illustrate an example showing improvement of the PEL illustration on the Bunny eye region using optimal lighting in one embodiment.

[0045] FIG. 4 (a) illustrates the eye region of the bunny model when only main light is applied. FIG. 4 (d) illustrates extracted PELs based on the bunny model shown in FIG. 4 (a). It can be seen that using the main light alone, the eye region is not illustrated well due to the low contrast around the eye.

[0046] FIG. 4 (b) illustrates the eye region of the bunny model wherein a spot light which covers the eye region and maximizes the contrast is applied in addition to the main light. FIG. 4 (e) illustrates the eye region of the bunny model as shown in FIG. 4 (b) using PELs. As can be seen, the features are enhanced and the corresponding PELs illustrate the eye very well.

[0047] FIG. 4 (c) illustrates the eye region of the bunny model wherein a spot light is applied which minimizes the local contrast on the eye region in addition to the main light. FIG. 4 (f) illustrates the eye region of the bunny model as shown in FIG. 4 (c) using PELs. As expected, the feature lines are reduced accordingly.

[0048] Note that the discontinuity of the illumination may occur in the shaded images (FIGs. 4 (b) and (c)). With the guarantee of local continuity for PEL extraction as described

above, user-desirable improved PEL illustration of the bunny eye is achieved without introducing meaningless feature lines on the boundaries (FIGs. 4 (e) and (f)). It also demonstrates that PELs are not extracted from image space. Using edge detection algorithms in image space, the aforementioned discontinuity of illumination may result in an edge, which may not be desirable. This also explains one important difference between PEL and the edge detection in image processing. In this context, object space may refer to a 3D space. Image space may refer to a 2D space.

[0049] In the following, algorithms of extraction of PELs are described in more detail.

[0050] PEL is defined on general surfaces of 3D objects, which can be triangle mesh, subdivision surface or spline based surface etc. As background information, a triangle mesh is a type of polygon mesh in computer graphics. It comprises a set of triangles (typically in three dimensions) that are connected by their common edges or corners. Many graphics software packages and hardware devices can operate more efficiently on triangles that are grouped into meshes than on a similar number of triangles that are presented individually. This is typically because computer graphics do operations on the vertices at the corners of triangles. In this context, to demonstrate the effectiveness of PELs for shapes illustration of 3D objects, an implementation of PELs extraction on triangle mesh is given, because triangle mesh is the most generally used form of 3D object representation in movie, game industry, and scientific visualization. For 3D object represented in other forms, a lot of open-source tools and commercial tools are available to use to convert them into triangle mesh. Thus, it should be noted that PEL is not limited to 3D object represented with vertices.

[0051] Given a triangle mesh S and the illumination f on each vertex, the extraction of PELs on S comprises several steps as described in the following in one embodiment.

[0052] The first step is preprocessing which is optional. In this step, normal n of surface S is smoothed.

[0053] In the second step, the gradient of illumination ∇f for each vertex is computed.

[0054] In the third step, the directional derivative of $D_w \|\nabla f\|$ along the gradient direction w for each vertex is computed.

[0055] In the fourth step, the zero-crossing is detected and the ones which are the local minima are filtered out, i.e. $D_w D_w \|\nabla f\| > 0$.

[0056] In the fifth step, the zero-crossings are traced to get the PEL.

[0057] The preprocessing is described in more detail as follows.

[0058] The purpose of preprocessing is to reduce the noise of illumination. In one embodiment, for the light model for PELs, both the main diffuse light and the auxiliary spot lights are functions of $n \cdot l$. Thus, the noise of illumination may be mainly caused by the noise of geometry, i.e., the surface normal. In one embodiment, the bilateral filter is applied to process the vector field n on S . For each vertex v , its neighborhood may be denoted by $N(v)$. The bilateral filler may be defined as:

$$n'(v) = \frac{\sum_{p \in N(v)} W_c(\|p - v\|) W_s(\rho(v, p)) n(p)}{\sum_{p \in N(v)} W_c(\|p - v\|) W_s(\rho(v, p))}$$

with

$$\rho(v, p) = \frac{\arccos(\|n(v) \cdot n(p)\|)}{\|p - v\|}$$

where W_c is the closeness smoothing filler with parameter δ_c :

$$W_c(x) = e^{\frac{-x^2}{2\delta_c^2}}$$

$\rho(v, p)$ mimics the relative curvature of v and p , and W_s is the feature preserving filler with parameter δ_s :

$$W_s(x) = e^{\frac{-x^2}{2\delta_s^2}}$$

which penalizes large variation of the feature field. In practice, it is found that this step is usually helpful to improve the robustness of the algorithm for the scanned models described herein. In one embodiment, an intuitive method is used to set the parameters δ_c and δ_s . In one embodiment, the user selects a point on the mesh where the surface is expected to be smooth, and then a radius r of the neighborhood of the point is defined. In one embodiment, parameters may be set as: $\delta_c = r/2$ and δ_s is the standard deviation of $\rho(v, p)$ in the selected neighborhood.

[0059] The advantage of smoothing surface normal is that it may be computed only once, which avoids computational overheads and guarantees achieving PELs illustration in real time for meshes of moderate size. In another embodiment, preprocessing may be performed by smoothing the illumination f on the surface, which may cause expensive smoothing operation in each frame once the lighting or viewing condition is changed.

[0060] The computation of derivatives is described in more detail as follows.

[0061] The definition of PEL involves the third order derivatives of the surface illumination, thus the key is to compute the derivatives efficiently and robustly. In one embodiment, the per-vertex derivatives are computed by averaging adjacent per-face derivatives.

[0062] As an illustration, given an arbitrary scalar function g defined on the mesh, i.e., $g: M \rightarrow R$, a triangle $T \in M$ with vertices $V_i \in R^3$ and the associated scalar value $g_i = g(V_i) \in R$, $i = 1, 2, 3$ are considered. The per-face coordinate system is defined in the plane perpendicular to the face normal. Three vertices of T in the per-face coordinate system are denoted by $(x_i, y_i) \in R^2$, $i = 1, 2, 3$. The per-face gradient ∇g may be computed as:

$$\nabla g = \begin{pmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{pmatrix} = \frac{1}{d_T} \begin{pmatrix} y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{pmatrix} \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix}$$

where

$$d_T = (x_1 y_2 - y_1 x_2) + (x_2 y_3 - y_2 x_3) + (x_3 y_1 - y_3 x_1)$$

which is twice the signed area of T .

[0063] The gradient of each vertex is considered as the weighted average of the gradient of its adjacent faces. The vertex coordinate system (U, V) is defined in the plane perpendicular to the vertex normal.

[0064] Note that the vertex and face coordinate systems are usually different. In one embodiment, coordinate system transformation is applied before computing the contribution of the per-face derivatives to the per-vertex derivatives.

[0065] FIG. 5 (a) illustrates the vertices (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) . FIG. 5 (b) illustrates the face coordinate system.

[0066] In one embodiment, the local vertex coordinate system is firstly rotated to be coplanar with the local face coordinate system. Then, given a point $(u_i, v_i) = u_i U_i + v_i V_i$ in the vertex coordinate system, the per-vertex gradient is computed in the local face coordinate system with

$$\nabla g(u_i, v_i) = \begin{pmatrix} U_i \cdot \nabla g(x, y) \\ V_i \cdot \nabla g(x, y) \end{pmatrix}$$

In one embodiment, after the per-vertex gradients for all the adjacent faces of the vertex are computed, they are accumulated with weighted averaging. In one embodiment, the Voronoi area weight w_i is used, which is set to be the portion of the area which lies closest to the vertex. Finally, the per-vertex derivative may be computed as:

$$\nabla g(u, v) = \sum_i w_i \nabla g(u_i, v_i)$$

[0067] The tracing of PELs is described in more detail as follows.

[0068] In one embodiment, once the directional derivatives of variation of illumination at each mesh vertex v is computed, whether the mesh edge $[v_1, v_2]$ contains the zero-crossing may be checked. Let $h(v) = D_w \|\nabla f(v)\|$. If $h(v_1) \cdot h(v_2) < 0$, linear interpolation may be used to approximate a zero-crossing on edge $[v_1, v_2]$ and p may be considered as a PEL vertex. If two PEL vertices are detected on edges of a triangle, a straight line segment may be used to connect the two vertices. Line segments may be connected to form a piecewise-linear PEL curve.

[0069] In one embodiment, the strength of a PEL may be measured by the integral of $\|\nabla f\|$ along the line

$$\int \|\nabla f\| dS \approx \sum_i \frac{\|\nabla f(p_i)\| + \|\nabla f(p_{i+1})\|}{2} \|p_i - p_{i+1}\|$$

In one embodiment, a threshold T may be defined to filter out noisy PELs with strength less than T . In one embodiment, this threshold may be specified by the user.

[0070] Volumetric data are widely used in scientific and medical applications. Volume illustration can be viewed as non-photorealistic rendering applied to volume visualization. In one embodiment, PELs may be applied for volume illustration using isosurfaces. An isosurface is a three-dimensional analog of an isocontour. It is a surface that represents points of a constant value (e.g. pressure, temperature, velocity, density) within a volume of space. In other words, it is a level set of a continuous function whose domain is 3D-space. In medical imaging, isosurfaces may be used to represent regions of a particular density in a three-dimensional CT scan, allowing the visualization of internal organs, bones, or other structures. The visualization of volume datasets with PELs consists of two phases, namely the isosurface extraction phase and the rendering phase. The user interested isosurfaces may be first extracted with the standard marching cubes method in a preprocess phase. In the rendering phase, PELs may be computed for each isosurface, and then combined. To further improve the volume illustration, stylized shading may be also applied to highlight important details and emphasize the inner structure of the volume datasets.

[0071] FIGs. 6 (a)-(f) illustrate the application of PEL for illustration of volumetric datasets. FIGs. 6 (a)-(c) show

that an isosurface is extracted and displayed from different view points. FIGs. 6 (a)-(c) illustrate that the isosurface is conveyed by PELs (solid lines). FIGs. 6 (d)-(f) illustrate another set of isosurface extracted and displayed from different view point. To emphasize the inner structure, the outer isosurface is illustrated with dotted lines only in FIGs. 6 (a)-(f)..

[0072] Comparing with existing feature lines, the definition of PEL with luminance variation guarantees better perceptual consistency. At the same time, since the illumination depends on the view position, light models, material, texture, and geometric properties, PELs are also more general than the existing feature lines. More importantly, PEL provides the user more freedom to control the desired illustration by manipulating the above parameters. Although originated from image processing, PEL is different from image space edge detection in that PEL is computed in object space. The pixel-based representation of feature lines in image space may suffer from low precision due to the loss of 3D scene information during rendering, and is not suitable for further processing, such as stylization. In contrast, PEL does not have these constraints since PEL is defined in object space, thus, the user may totally control the illumination by manipulating the light which in turn helps the user to achieve the desired illustration.

[0073] The process of generating line drawings with PELs according to one embodiment may be summarized as below.

[0074] In the first step, the 3D scene with initial lighting conditions is set up. In other word, a main light is applied. In the second step, line drawing with PELs is computed by the system. In other words, PELs are extracted. In the third step, user manipulates lighting globally or locally. For

example, the user may add one or more auxiliary lights locally on the 3D object. In the fourth step, improved line drawing may be computed with the adjusted light setting. The process then goes back to the third step until desired line drawing is achieved.

[0075] In one embodiment, an effective light setting with main light and auxiliary lights is provided. By using the head light as the main light, intuitive view and light dependent line drawing with PELs may be computed automatically which are already desirable in many cases. More user specified line drawings for more complicated scenes may be achieved with incremental lighting manipulation. User friendly lighting manipulation is provided in this invention. The optimal lighting setting may also be automatically computed with a fast linear optimization algorithm.

[0076] In one embodiment, illustrations of 3D objects using feature lines such as PELs may be further applied into a dynamic setting for stylized animations with frame coherence based on correspondence from clustering curves. In this context, the term frame refers to a digital picture, and animation is a video sequence of a plurality of frames or digital pictures. Frame coherence, or digital picture coherence may refer to consistence of stylization among different digital pictures within the video sequence. Stylization is characteristic of non-photorealistic applications. Based on extracted lines, i.e. PELs, shapes of 3D objects may be represented with stylized curves that wiggle, vary in width and texture, or resemble in other ways than curves drawn by hand with natural media. In these applications, besides achieving perceptual consistent illustrations of shapes, the other goal of illustration is to achieve visually pleasing line drawings. This is widely used

in cartoon animations for TV and movies. A key challenge for curve based stylized animation is to provide frame-to-frame coherence, so that curves adapt smoothly over time to change in the animated scene. Temporal coherence problem is a major challenge of stylized rendering. It is especially challenging for view dependant feature lines, such as PELs, which will merge, split, appear or disappear over time when the viewpoint change.

[0077] In one embodiment, an approach is provided to generate stylized animation with frame coherence. In one embodiment, the frame coherence is based on correspondence of curves obtained on base of curve clustering methods. In particular, the animation may be PEL-based. However, it should be noted that the method described herein is not limited to be applied to PELs, but may be applied to any video sequence of digital pictures with curves extracted for 3D objects from 3D space.

[0078] FIG. 7 illustrates a diagram of stylized animation process with curve coherence. In this example, the process is based on PEL curves grouping. It should be noted that the stylized animation process is not limited to PEL curves, but may be applied to any sequence of digital pictures whose curves are extracted for 3D objects from 3D space.

[0079] In process 701, PELs are extracted for an 3D object for the illustration of the 3D object which will be presented in a video sequence of digital pictures.

[0080] In process 702, a user may influence the animation by setting some rules of curves grouping etc.

[0081] In process 703, PEL curve grouping is performed based on the extracted PELs in process 701 and the input from the user in process 702.

[0082] Optionally, in process 704, a knowledge based line drawing is also involved.

[0083] In process 705, PEL curve correspondence is established based on the curves grouping in process 703 and the knowledge based line drawing in process 704.

[0084] In process 706, stylized animation, namely the video sequence of the digital pictures, is presented.

[0085] FIG. 8 illustrates a method for generating a digital picture which comprises steps 801, 802, 803, and 804 according to one embodiment.

[0086] In one embodiment, the method comprises a step 801 of receiving a three-dimensional representation for each of a plurality of curves, wherein each curve represents at least partially a shape of a three-dimensional object. As an illustration, for an 3D object which is to be presented in a plurality of digital pictures of a video sequence, a plurality of curves, e.g. PELs, may be extracted in order to illustrate the shape of the 3D object. Each curve may have the respective 3D information such as position in the 3D space and in which digital picture to be presented in the video sequence. Each curve illustrates at least a part of the shape of the 3D object. For example, if the 3D object is a human head, each curve may illustrate the shape of part of the human head, e.g. left eye.

[0087] In one embodiment, the method further comprises a step 802 for grouping the plurality of curves into at least one curve group based on the three-dimensional representations of the curves using three-dimensional information given by the three-dimensional representations. As an illustration, for each digital picture, curves may be grouped together if the curves illustrate a same, nearby or similar part of the 3D object. In this context, similar part may refer to the part that is within a pre-defined range on the 3D object, or within a pre-defined range of change of geometries, e.g.

curvatures, on the 3D object. For example, if the 3D object is a human head, curves that are used to illustrate the left eye may be grouped together. As another illustration, for different digital pictures, curves that are used to illustrate a same or similar part of the 3D object may be grouped together. For example, if the 3D object is human head, then curves that illustrate the left eye area to be presented in different digital pictures may be grouped together. In a particular example, for the curve grouping among different digital pictures, the curve group A may comprise one or more curves from each digital picture. It may also happen that for at least one digital picture of the video sequence, no curves are grouped into group A if the at least one digital picture does not contain curves relative to the part of the 3D object that is associated with group A.

[0088] In one embodiment, the three-dimensional information is information about the three-dimensional path of the curve in three-dimensional space. For example, the three-dimensional information may define position or coordinate of each point of the curve in the 3D space.

[0089] In one embodiment, the method further comprises a step 803 for associating each curve group with a picture element based on the curves of the curve group. As an illustration, for a curve group for one digital picture, the curve group may be associated with a part, e.g. left eye region, of the 3D object, e.g. human head. As a further illustration, for a curve group for different pictures, the curve group may be associated with a part, i.e. left eye region, of the 3D object, i.e. human head, to be presented in different digital pictures.

[0090] In one embodiment, the method further comprises a step 804 for forming the digital picture using the picture

element. For example, if the 3D object is a human head, then curve groups may be generated relating to different picture element such as left eye region, right eye region, mouth region etc. The digital picture may be generated based on the curve groups associating with different picture elements. For example, if the curve groups being associated with left eye region are set by the user to be displayed in a bold style, then the curve group associated with left eye may be presented with curves in bold mode.

[0091] In one embodiment, in other words, curves or feature lines are firstly extracted for a 3D object to be represented in different digital pictures of a video sequence. Each curve line reflects at least a part of the shape of the 3D object. Each curve has its own information in the 3D space, such as the position or coordinate in the 3D space. Each curve also has its information of the digital picture of the video sequence that the curve is related to. The curves are grouped according to certain rules based on the information of positions in the 3D space of each curve. Each curve group may be associated with a same or similar part on the 3D object in one digital picture or among different digital pictures. Each curve group may be associated with a picture element, e.g. left eye region, of the 3D object. For each digital picture, at least one curve group is formed that is associated with one picture element. The digital picture is generated based on the picture elements related to the digital picture.

[0092] FIG. 9 illustrates a device 900 for generating a digital picture. The device 900 comprises a receiving unit 901, a grouping unit 902, an associating unit 903, and a generating unit 904.

[0093] In one embodiment, the receiving unit 901 is configured to receive a three-dimensional representation for

each of a plurality of curves, wherein each curve represents at least partially a shape of a three-dimensional object.

[0094] In one embodiment, the grouping unit 902 is configured to group the plurality of curves into at least one curve group based on the three-dimensional representations of the curves using three-dimensional information given by the three-dimensional representations.

[0095] In one embodiment, the associating unit 903 is configured to associate each curve group with a picture element based on the curves of the curve group.

[0096] In one embodiment, the generating unit 904 is configured to form the digital picture using the picture element.

[0097] In one embodiment, a computer readable medium having a program recorded thereon is provided, wherein the program is adapted to make a processor of a computer perform a method for generating a digital picture. In one embodiment, the computer readable medium comprises code of the program making the processor perform the reception of a three-dimensional representation for each of a plurality of curves, wherein each curve represents at least partially a shape of a three-dimensional object.

[0098] In one embodiment, the computer readable medium further comprises code of the program making the processor perform the grouping of the plurality of curves into at least one curve group based on the three-dimensional representations of the curves using three-dimensional information given by the three-dimensional representations.

[0099] In one embodiment, the computer readable medium further comprises code of the program making the processor perform the association of each curve group with a picture element based on the curves of the curve group.

[00100] In one embodiment, the computer readable medium further comprises code of the program making the processor perform the formation of the digital picture using the picture element.

[00101] In one embodiment, the digital picture is a digital picture of a video sequence of digital pictures. In other words, the digital picture is a digital picture of an animation comprising a plurality of digital pictures. In other words, curves may be grouped across frames of a video sequence, e.g. subsequent frames of a video sequence.

[00102] In one embodiment, at least two curves of the plurality of curves are associated with different digital pictures of the video sequence of digital pictures.

[00103] In one embodiment, the three-dimensional representation of a curve is a set of vertices in three-dimensional space. For example, the three-dimensional representation of the curve may be PEL. As described earlier, a PEL may be a set of vertices on the 3D object where the variation of illumination in the direction of its gradient reaches the local maximum.

[00104] In one embodiment, the vertices specify points in three-dimensional space where the illumination of the three-dimensional object changes if the three-dimensional object is viewed from a pre-determined viewpoint and is illuminated by at least one predetermined light source. For example, the at least one predetermined light source may be a main light. For another example, the at least one predetermined light source may include a main light and at least one auxiliary light.

[00105] The PEL stroke clustering is described in more detail in the following.

[00106] In one embodiment, the stylized curves may be generated based on the curve grouping method provided herein

which is a basic low-level operation of perceptual grouping of curves, e.g. PELs. As a common process, both in human and computer vision, the grouping of low-level features give abstract and effective information in understanding a complicated scene. For example, the grouping of PEL curves may give high level information in finding the correspondence of curves between digital pictures. Most previous works in image processing and computer vision focus on the segmentation and grouping of points and straight lines. Comparing with these works, the segmentation and grouping of curved lines is far more complicated since arbitrary curves have more degrees of freedom than points and straight lines.

[00107] In one embodiment, curve grouping follows the principles as described in the following. Comparing with point based image segmentation, curves grouping shows difficulties in some basic aspects, such as curves with noise or gaps, and different structures in different scales. It is difficult to determinate the neighborhood of curves within a curve set and precisely compute the intersection and distance of two arbitrary curves. With the special considerations on grouping curves for stylized line drawing, in one embodiment, a method is provided to find clusters of curves with focus on perceptual grouping rules including:

- Continuation. The evaluation of the curvilinearity of curves takes into account the gap distance relative to the length of the curves, and the amount of bending of the joining curves.
- Parallelism. The evaluation of the potential parallelism considers factors of the separation between curves and the amount of overlapping of the curves.

- Proximity. The minimum distance between all points of two curves.
- Closure. The potential of curves that tend to lead to closed curves.

[00108] In one embodiment, curves may be first divided into short segments at locations that have large curvatures prior to clustering. For example, curves may be divided into short segments at locations that have curvatures larger than a pre-determined value. Then, curve segments are grouped according to a linear combination of the evaluation of continuation, parallelism, proximity and closure separately.

[00109] In one embodiment, the measurement of several parameters may be used to determine whether two curves should be grouped together within one digital picture or among different digital pictures. In one embodiment, such parameters may include:

1. continuity between two curves
2. parallelism of the two curves
3. proximity of the two curves, and
4. potential of the two curves to lead to a closed curve.

[00110] The measurement of the above mentioned parameters are described in detail as follows.

[00111] It should be noted that the measurements of the parameters of continuity, parallelism, proximity, and potential of the two curves to lead to a closed curve are not limited to the particular examples given below, but can be any other measurements that could reflect the parameters of continuity, parallelism, proximity, and potential of the two curves to lead to a closed curve.

[00112] In one embodiment, continuation may be computed based on the length of the joining curve (l_i) relative to the lengths of the two input curves (l_1, l_2), and the bending degree of the joining curve. In one embodiment, the joining curve is generated by sampling points at the ends of the two input curves, and then compute a B-spline curve which interpolates these points and links the two input curves. For example, assuming that curve l_1 has two ends A and B, and curve l_2 has two ends C and D, wherein end A is closer to l_2 relative to the other end B of l_1 , and end C is closer to l_1 relative to the other end D of l_2 . Then a joining curve l_i may be generated by sampling points between end A of l_1 and end C of l_2 . The bending of the joining curve with the maximum curvature of the joining curve ($K_{\max}$) may be determined, which refers to the curvature of a point in the joining curve that has a maximum curvature. Thus, the continuation can be defined as:

$$m_c = l_i / (l_1 + l_2) * K_{\max} \quad (7)$$

[00113] In one example, if the value of the m_c is less than a user defined threshold, the two curves are considered to be continuous.

[00114] FIG. 10 illustrates an example of the determination of continuation of two curves l_1 and l_2 , wherein l_1 has two ending points 1001 and 1002, and l_2 has two ending points 1003 and 1004. In this example, the ending point 1001 of l_1 is nearer to the line l_2 relative to the ending point 1002, and the ending point 1003 of l_2 is nearer to line l_1 relative to the ending point 1004.

[00115] In order to determine whether the curve l_1 and l_2 are continuous, a joining curve l_i may be generated. The joining

curve l_i may be generated by first selecting sampling points in curve l_1 near the ending 1001 and in curve l_2 near the ending 1003. In this example, 5 sampling points are selected in curve l_1 (points 1001 and 1005-1008) near the ending point 1001, and in curve l_2 (points 1003 and 1009-1012) near the ending point 1003, respectively. Based on the selected sampling points, a B-spline curve l_i may be generated which interpolates these sampling points (points 1001, 1003, and 1005-1012) and links the two curves l_1 and l_2 .

[00116] In one embodiment, a value m_c may be calculated based on l_1 , l_2 , and l_i using equation (7). The calculated m_c may be compared with a pre-determined threshold. If m_c is smaller than the pre-determined threshold, the curves l_1 and l_2 may be considered to be continuous. Otherwise, the curves l_1 and l_2 may be considered to be open.

[00117] In one embodiment, the measurement of the parallelism takes into consideration the factors of the separation between the two input curves and the amount of overlapping of the two curves. FIG. 11 illustrates the determination of parallelism of two curves. For example, two curves with respective length of l_1 and l_2 may be represented with B-splines, and a set of points may be sampled for each curve with same parameter values (such as $t = 0.1, 0.2, 0.3 \dots 0.5 \dots 1.0$). Then these two sets of sampling points may be interpolated linearly to generate a third set of points. The third set of points is used to generate a B-spline middle curve between curve l_1 and curve l_2 with length l_i , as shown in FIG. 11. The middle curve may be divided into two parts at the point 1100 with $t = 0.5$. The length of the first part l_{i1} may be represented with m_1 , and the length of the second part l_{i2} may be represented with m_2 . Denoting e_1 , e_2 as the distances between the end point of the two curves and the

middle curve, respectively. In one embodiment, the parallelism may be defined as:

$$m_p = l_i / (l_1 + l_2) * (1 - R_0)$$

where

$$R_0 = (m_1 + m_2) / (m_1 + m_2 + e_1 + e_2)$$

[00118] In one embodiment, if the value of m_p is larger than a user defined threshold, the two curves may be considered to be parallel.

[00119] Proximity for two curves may be defined as the minimum distance between all points of the two curves. For example, denoting p_i , p_j as any points on the two curves, respectively, the proximity of the two curves can be defined as the minimum distance between all the points of the two curves, as:

$$m_D = \min \| p_i - p_j \|$$

[00120] In one embodiment, if the value of m_D is less than a user defined threshold, the two curves are considered to be in proximity.

[00121] Closure m_O may be defined as the potential of curves that tend to lead to closed curves. For example, points are sampled at the ends of the two input curves, and then a B-spline curve is computed which interpolates these points and links the two input curves. For example, assuming that curve l_1 has two ends A and B, and curve l_2 has two ends C and D, wherein end A is closer to end C of l_2 relative to the other end B of l_1 , and end B is closer to end D of l_2 relative to the other end A of l_1 . Then a joining curve l_{i1} may be generated between end A of l_1 and end C of l_2 , and another joining curve l_{i2} may be generated between end B of l_1 and end D of l_2 . Then, the sum of curvature along the joining curve may be computed. In one embodiment, two curves with large sum

of curvature m_0 of joining curve may be considered to be open. For example, if the sum is larger than a user defined threshold, the two curves may be considered to be open. In one embodiment, if the sum of curvature m_0 is less than a user defined threshold, the two curves may be considered as closed.

[00122] FIG. 12 illustrates the determination of whether two curves l_1 and l_2 tend to form a closed curves, i.e. potential of the two curves to lead to a closed curve, wherein curve l_1 has two ending points 1201 and 1202, and curve l_2 has two ending points 1203 and 1204.

[00123] In this example, the ending point 1201 is closer to ending point 1203 of curve l_2 relative to the other ending point 1202 of curve l_1 , and ending point 1202 is closer to the ending point 1204 of curve l_2 relative to the other ending point 1201 of curve l_1 . Then a joining curve l_{i1} may be generated between the ending point 1201 of curve l_1 and the ending point 1203 of curve l_2 , and another joining curve l_{i2} may be generated between the ending point 1202 of curve l_1 and the ending point 1204 of curve l_2 . For example, the joining curve l_{i1} may be generated by first selecting sampling points in curve l_1 near the ending 1201 and in curve l_2 near the ending point 1203. In this example, 5 sampling points are selected in curve l_1 (points 1201 and 1205-1208) near the ending point 1201, and in curve l_2 (points 1203 and 1209-1212) near the ending point 1203, respectively. Based on the selected sampling points, a B-spline curve l_{i1} may be generated which interpolates these sampling points (points 1201, 1203, and 1205-1212) and links the two curves l_1 and l_2 . Similarly, the joining curve l_{i2} may be generated by first selecting sampling points in curve l_1 near the ending 1202 and

in curve l_2 near the ending point 1204. In this example, 5 sampling points are selected in curve l_1 (points 1202 and 1213-1216) near the ending point 1202, and in curve l_2 (points 1204 and 1217-1220) near the ending point 1204, respectively. Based on the selected sampling points, a B-spline curve l_{i2} may be generated which interpolates these sampling points (points 1202, 1204, and 1213-1220) and links the two curves l_1 and l_2 . In one embodiment, the sum of curvature along the joining curves l_{i1} and l_{i2} is computed. In one embodiment, the sum may be compared with a user-defined threshold. In one embodiment, if the sum of curvature m_o is less than a user defined threshold, the two curves may be considered as closed. If the sum is larger than the user defined threshold, the two curves l_1 and l_2 may be considered as open.

[00124] In one embodiment, the grouping of two curves into a same group depends on measurement of all the parameters m_c , m_p , m_D , m_o described above. In one embodiment, a linear combination may be used to compute an overall grouping value M_A . Users may adjust the weights of the linear combination to emphasis specific parameters. In one embodiment, when the overall grouping value is larger than a user defined threshold, the input curves may be grouped into a same group.

[00125] For example, M_A may be expressed as

$$M_A = a * m_c + b * m_p + c * m_D + d * m_o \quad (8)$$

where a , b , c , d are weights of m_c , m_p , m_D , m_o , respectively. A threshold TH may be set to be a default value or may be set by a user. In one example, if M_A is larger than TH , then the two curves are determined to be in one curve group.

[00126] In one example, weights a , b , c , and d may be set to be a default value or may be determined, e.g. by a user. In one example, weights a , b , c , d may be determined according to different emphasis on different parameters m_C , m_P , m_D , and m_O , respectively. For example, if only continuation and parallelism are considered as important parameters, then the respective weights a and b may be set to be certain non-zero values, and weights c and d may be set to be zero. For another example, if only one parameter of proximity is considered to be important to evaluate whether two curves should be grouped together, then the respective weight c may be set to be a certain non-zero value, and weights a , b , and d may be set to be zero. In a further example, if all the parameters of m_C , m_P , m_D , m_O are considered to be important, then weights a , b , c and d may be set to be certain non-zero values respectively according to the respective importance of each parameter.

[00127] In one embodiment, it is decided whether two curves are grouped to a same curve group based on at least one of a measure of the continuity between the two curves; a measure of the parallelism of the two curves; a measure of the proximity of the two curves; and a measure of the potential of the two curves to lead to a closed curve. As an illustration, referring to the equation (8) of M_A as given above, at least one of the weights a , b , c , and d is a non-zero value, and thus M_A is determined based on at least one parameter of m_C , m_P , m_D , and m_O .

[00128] In one embodiment, it is decided whether two curves are grouped to the same curve group based on a comparison of

at least one of a measure of the continuity between the two curves, a measure of the parallelism of the two curves; a measure of the proximity of the two curves, and a measure of the potential of the two curves to lead to a closed curve with a pre-defined threshold. For example, referring to the equation (8) of M_A as given above, at least one weight of a , b , c , d are set to be a non-zero value. Thus, M_A is determined based on at least one parameter of m_C , m_P , m_D , and m_O , and M_A is further compared with the pre-defined threshold TH to determine whether the two curves should be grouped to a same curve group.

[00129] In one embodiment, it is decided whether two curves are grouped to the same curve group based on a comparison of a combination of at least two of a measure of the continuity between the two curves, a measure of the parallelism of the two curves, a measure of the proximity of the two curves, and a measure of the potential of the two curves to lead to a closed curve with a pre-defined threshold. As an illustration, referring to the equation (8) of M_A as given above, at least two weights of a , b , c , d are set to be non-zero values. Thus, M_A is determined based on at least two parameters of m_C , m_P , m_D , and m_O , and M_A is further compared with the pre-defined threshold TH to determine whether the two curves should be grouped to a same curve group.

[00130] In one embodiment, the combination is a weighted sum at least two of a measure of the continuity between the two curves, a measure of the parallelism of the two curves, a measure of the proximity of the two curves, a measure of the potential of the two curves to lead to a closed curve. As an

illustration, parameters of m_c , m_p , m_D , and m_o are weighed by a, b, c, and d, respectively (equation (8)). A user may set the weight of each parameter according to the importance of evaluation of whether the two curves should be grouped to a same curve group. In case the grouping of curves is not ideal, the user may further adjust the weights for different parameters in one embodiment.

[00131] In one embodiment, the curve grouping is performed both in a single digital picture and between different digital pictures of a video sequence of digital pictures. The curve grouping in a single digital picture may assist in finding high level structure of curves while the grouping between digital pictures help in finding correspondence of digital pictures within the video sequence.

[00132] When working in 3D space, the additional depth dimension may give further information for curve grouping.

[00133] In one embodiment, curve grouping between digital pictures is designed for the purpose of finding curve correspondence. In an animation, a corresponded curve in each digital picture may be grouped into a single group. For example, these curves may locate in similar positions and have similar geometrical properties. In this context, similar positions may refer to that the positions of the curves are within a pre-defined range with each other. Similar geometrical properties may refer to the differences between curvatures of the curves may be within a pre-defined range. With these curve groups information, curve styles and inbetween frame interpolation may be applied to different curve groups.

[00134] FIG. 13 (a)-(c) illustrate an example of curve grouping for different digital pictures.

[00135] As can be seen, FIG. 13 (a)-(c) show a rotating human head in three different digital pictures of a video sequence in an animation. In FIG. 13 (a), curves A and B in the first digital picture illustrate the left eye of the face. In the second digital picture FIG. 13 (b), only curve A1 is used for the illustration of the left eye. In the third digital picture FIG. 13 (c), curves A2 and B2 are used to illustrate the left eye. In one embodiment, as a result of the curve grouping, for the left eye, two curve groups may be determined: namely one group with curves A, A1, A2 and the other group with curves B and B2. Based on the curve grouping, in one embodiment, different styles may be applied to the curves within these two groups. This embodiment may facilitate understanding of the shape of a 3D object and help to achieve visually pleasing animations with stylized strokes.

[00136] Since completely automatic curve grouping may not guarantee satisfied grouping in some cases, in one embodiment, user interaction may be incorporated to adjust and modify curve grouping during the grouping process. For example, the user may adjust weights of a , b , c , and d as shown in equation (8). For another example, the user may adjust the threshold TH of M_A .

[00137] In one embodiment, the curve grouping may provide high level information for the structure of a 3D object, which makes it possible to incorporate semantics information to generate more elegant line drawings. Semantics information may give meaning to the curve, for example, it is an eye or a mouth of a person's face. The curve grouping may be applied in a wide range of applications, in the following, the effectiveness of the method of curve grouping is demonstrated

with its application in knowledge-based line drawing, and stylized stroke animations.

[00138] The knowledge based line drawing is described in more detail in the following.

[00139] Observation shows that lines of a line drawing by an artist may not always follow the geometry or lighting on the object surface. For example, exaggeration of lines for special expressional purposes is often seen in hand-made line drawings. This is apparent in some traditional artistic styles, such as cartoon. It is well understood by people because of some common knowledge on the objects represented in the line drawing.

[00140] In one embodiment, a method is provided for animated facial illustration with curves by incorporating templates and semantics information in the drawing process. In one embodiment, the method is based on the grouping of the curves extracted for a 3D object, and a pre-defined domain with strict constraints. In this context, the domain with strict constraints may refer to the requirement that a curve group clearly corresponds to a pre-defined component. For example, a curve group may include curves in the area of left eye region, and the curve group may be matched with templates of the pre-defined component, namely left eye region. In one embodiment, the curves are PELs.

[00141] In one embodiment, a knowledge-based algorithm is provided to incorporate templates and semantics information in the drawing process. In one embodiment, after the curves are grouped, each curve group may be matched into a pre-defined component. For example, a curve group may be matched with a pre-defined component: left eye. A plurality of templates of the pre-defined component may already be stored in the system. In one embodiment, a template of the pre-

defined component may be selected which matches the curve group most. In one embodiment, each curve group may be replaced with the selected template. In one embodiment, the selected template may be further stylized, i.e. in a bold drawing mode. In one embodiment, the selected template may be transformed and composited to fit into the digital picture. In this way, stylistic line drawing may be achieved. Curves in these line drawings might not follow the features of the image, but as a whole, the line drawing may be still "correct" and beautiful. This process may be optional, and it helps in applying stylized curves in animations since pre-defined drawings can have well-defined curve correspondence.

[00142] In one embodiment, curve grouping is performed automatically according to rules of continuation, parallelism etc. as described early.

[00143] In one embodiment, the knowledge-based algorithm works with all the digital pictures within the video sequence.

[00144] In one embodiment, a predefined template database may be built for a variety of areas, such as human faces, animals, buildings etc.

[00145] In one embodiment, PEL curves are first grouped to represent meaningful components including eye, nose and mouth. Then a search in the library may be carried out to find a pre-defined line drawing to match the stroke clusters. Different matching parameters which represent different styles of facial sketching may be used in this process. Finally, PEL stroke clusters may be replaced with pre-defined user drawings.

[00146] FIGs. 14 (a)-(c) illustrate an example with line drawings in two different styles generated from a pre-defined library.

[00147] FIG. 14 (a) illustrates a 3D object. Based on the 3D object shown in FIG. 14 (a), curves such as PELs which represent the shape of the 3D object may be extracted. FIG. 14(b) illustrates an example of extracted curves in one style from a stroke template. FIG. 14(c) illustrates another example of extracted curves in another style from another stroke template.

[00148] The curve correspondence is described in more detail in the following.

[00149] Finding the correspondence and evolving of curves, e.g. PELs, extracted for a 3D object between digital pictures is the key of stylized animation based on the curves. It may be applied in interpolating curves smoothly, rendering coherent artistic curves in different styles, and manipulating animations with operations in curve space etc. As mentioned earlier, this is a challenging problem, especially for PEL curves.

[00150] In one embodiment, correspondences of curves are established based on curve grouping. In an animation sequence, a set of curves for all the digital pictures are extracted, and then separated into small groups with curves from different digital pictures. In one embodiment, point correspondence of curves in a group may be found by projecting curves into screen space and finding nearest points in screen space. In this context, the screen space may refer to the 2D space. The extracted curves of the 3D object in the 3D space are projected into a 2D space to form the 2D illustration of the 3D object. In case of bad grouping in this way, as mentioned earlier, user interaction may be incorporated, and some rules of this user interaction may be learned and applied onto further curve grouping process.

[00151] In one embodiment, a correspondence is established for each group of a digital picture. In other words, for a digital picture, a group of curves may be determined for a picture element, e.g. left eye. Then correspondence is established for all the groups that is associated with the left eye in different digital pictures of the video sequence.

[00152] In one embodiment, with curve grouping, correspondences of curve groups are established first, and then the correspondence of a single curve is determined within a group. Finding curve correspondence in this hierarchical matching improves the effectiveness of the algorithm. As an illustration, referring to FIG. 13, FIGs. 13 (a)-(c) illustrate 3 different digital pictures of the 3D object human head of a video sequence. A curve group F being associated with the left eye of the human head may be determined to contain curves A, B, A1, A2, and B2, wherein curves A, B, A1, A2, and B2 are all used to illustrate the shape of the left eye. In one embodiment, curve group F may be further divided into 3 sub-groups. For example, a curve sub-group F1 being associated with the left eye in the first frame may be determined to contain curves A and B. A curve group F2 being associated with the left eye in the second frame may be determined to contain curve A1. A curve group F3 being associated with the left eye in the third frame may be determined to contain curves A2 and B2. In one embodiment, as curve sub-groups F1, F2, and F3 are all associated with the component: left eye, a correspondence may be established among curve sub-groups F1, F2, and F3.

[00153] In one embodiment, after correspondence is established for curve sub-groups F1, F2, and F3, correspondence of a single curve may be determined. For example, for curve sub-groups F1, F2, and F3, it may be determined that curves A,

A1, and A2 are corresponding curves as curves A, A1, and A2 are in similar positions on the human head and share similar geometrical properties. Similarly, curves B and B2 are corresponding curves. In one embodiment, a same style may be applied to a same set of corresponding curves through different digital pictures of the video sequence.

[00154] In other words, in one embodiment, curves of different digital pictures may be grouped together which are associated with a same picture element, i.e. left eye. Then the curve group may be divided into sub-groups with each sub-group corresponding to a digital picture. Further, single curve correspondence may be established based on the curves in the sub-groups. In a further embodiment, a point correspondence may be determined further based on single curve correspondence. The establishment of the correspondence among different curve sub-groups, single curves in the curve sub-groups, and points in the single curves may be referred as a process of hierarchical matching.

[00155] FIG. 15 illustrates an example of hierarchical matching.

[00156] In an animation, in the first digital picture as shown in FIG. 15 (a), there is a curve S with pixels A and B. In the second digital picture as shown in FIG. 15 (b), there are curves S', S1, and S2. Through hierarchical matching, a curve group F may be determined to comprise curves S, S', S1, and S2 which are all used to illustrate the shape of the left eye. Further, for the curve group F, a curve sub-group F1 may be determined for frame 1 which contains curve S, and another curve sub-group F2 may be determined for frame 2 which contain curves S', S1, and S2. In a further step, single curve correspondence may be determined. For example, curve S in curve sub-group F1 may be determined to correspond to

curve S' in the curve sub-group F2, wherein curves S and S' are located on the same position of the left eye and share similar geometrical properties.

[00157] Curve correspondence may be used in a wide range of applications, including smooth curve interpolation for inbetweening, various styles artistic stroke rendering for stylized animation, and manipulation of animation in curve space etc. Finding curve correspondence with curve grouping give intuitive cues in developing novel advanced algorithms for different applications.

[00158] FIG. 16 illustrates a computer 1600 according to one embodiment.

[00159] In one embodiment, the computer 1600 may include a processor 1601. In one embodiment, the computer 1600 may further comprise a memory 1602. In one embodiment, the computer 1600 may further comprise input 1603 for receiving a three-dimensional representation for each of a plurality of curves, wherein each curve represents at least partially a shape of a three-dimensional object. In one embodiment, the computer 1600 may further comprise a user input 1604. In one embodiment, the computer 1600 may further comprise a display 1605. In one embodiment, the computer 1600 may further comprise a code reading unit 1606 for reading code from another computer readable medium. For example, all the component of the computer 1600 are connected with each other through a computer bus 1607.

[00160] In one embodiment, the memory 1602 may have a program recorded thereon, wherein the program is adapted to make a processor 1601 perform a method for generating a digital picture, the memory comprising code of the program making the processor 1601 perform the reception of a three-dimensional representation for each of a plurality of curves, wherein

each curve represents at least partially a shape of a three-dimensional object; code of the program making the processor 1601 perform the grouping of the plurality of curves into at least one curve group based on the three-dimensional representations of the curves using three-dimensional information given by the three-dimensional representations; code of the program making the processor 1601 perform the association of each curve group with a picture element based on the curves of the curve group; and code of the program making the processor 1601 perform the formation of the digital picture using the picture element.

[00161] In one embodiment, the processor 1601 reads the program on the memory 1602 to perform the program of generating the digital picture.

[00162] In one embodiment, the processor 1601 obtains the 3D representation of the curves through the curve representation input 1603.

[00163] In one embodiment, the processor 1601 obtains input from the user through user input 1604.

[00164] In one embodiment, the program code may be recorded on another computer readable medium (not shown). In this case, the processor 1601 may read the codes from the other computer readable medium through code reading unit 1606, and perform the method of generating a digital picture as described herein.

[00165] In one embodiment, the generated digital picture is shown on the display 1605.

[00166] While the invention has been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the scope of the invention as defined by the

appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

[00167] In this document, the following documents are cited:

1. Appel, "The notion of quantitative invisibility and the machine rendering of solids" in *Proceedings of the 1967 22nd national conference*
2. Gooch et al. "Interactive technical illustration" in *Proceedings of the 1999 symposium on Interactive 3D graphics*
3. Hertzmann and Zorin, "Illustrating smooth surfaces" in *Proceedings of Siggraph 2000*
4. Kalnins et al. "WYSIWYG NPR: Drawing strokes directly on 3d models" in *Proceedings of Siggraph 2002*
5. Interrante et al. "Enhancing transparent skin surfaces with ridge and valley lines" in *IEEE Visualization 1995*
6. Wilson and Ma, "Representing complexity in computer-generated pen-and-ink illustrations" in *Proceedings of NPAR 2004*
7. Ni et al. "Multi-scale line drawings from 3D meshes" in *Proceedings of the 2006 symposium on Interactive 3D graphics*.
8. DeCarlo et al. "Suggestive contours for conveying shape" in *Proceedings of Siggraph 2003*

Claims

What is claimed is:

1. A method for generating a digital picture comprising
receiving a three-dimensional representation for each of
a plurality of curves, wherein each curve represents at least
partially a shape of a three-dimensional object;
grouping the plurality of curves into at least one curve
group based on the three-dimensional representations of the
curves using three-dimensional information given by the
three-dimensional representations;
associating each curve group with a picture element
based on the curves of the curve group;
forming the digital picture using the picture element.
2. The method according to claim 1, wherein the digital
picture is a digital picture of a video sequence of digital
pictures.
3. The method according to claim 2, wherein at least two
curves of the plurality of curves are associated with
different digital pictures of the video sequence of digital
pictures.
4. The method according to any one of claims 1 to 3,
wherein it is decided whether two curves are grouped to a
same curve group based on at least one of
a measure of the continuity between the two curves
a measure of the parallelism of the two curves
a measure of the proximity of the two curves

a measure of the potential of the two curves to lead to a closed curve.

5. The method according to claim 4, wherein it is decided whether two curves are grouped to the same curve group based on a comparison of at least one of
a measure of the continuity between the two curves
a measure of the parallelism of the two curves
a measure of the proximity of the two curves and
a measure of the potential of the two curves to lead to a closed curve
with a pre-defined threshold.

6. The method according to claim 4, wherein it is decided whether two curves are grouped to the same curve group based on a comparison of a combination of at least two of
a measure of the continuity between the two curves
a measure of the parallelism of the two curves
a measure of the proximity of the two curves and
a measure of the potential of the two curves to lead to a closed curve
with a pre-defined threshold.

7. The method according to claim 6, wherein the combination is a weighted sum at least two of
a measure of the continuity between the two curves
a measure of the parallelism of the two curves
a measure of the proximity of the two curves
a measure of the potential of the two curves to lead to a closed curve.

8. The method according to any one of claims 1 to 7, wherein the three-dimensional representation of a curve is a set of vertices in three-dimensional space.

9. The method according to claim 8, wherein the vertices specify points in three-dimensional space where the illumination of the three-dimensional object changes if the three-dimensional object is viewed from a pre-determined viewpoint and is illuminated by at least one predetermined light source.

10. The method according to any one of claims 1 to 9, wherein the three-dimensional information is information about the three-dimensional path of the curve in three-dimensional space.

11. The method according to claim 3, wherein each curve group which is associated with a picture element is further divided into a plurality of curve sub-groups, each curve sub-group being associated with the same picture element for different digital pictures.

12. The method according to claim 11, further comprising setting a stylistic pattern for each picture element to be shown in the digital picture.

13. The method according to claim 12, wherein for a same picture element, a same stylistic pattern is applied for different curve sub-groups through different digital pictures.

14. The method according to claim 11, further comprising

setting a stylistic pattern for at least one curve within a curve sub-group.

15. The method according to claim 14, wherein a same stylistic pattern is applied for the at least one curve within the curve sub-group and at least one corresponding curve within all the other curve sub-groups within the same curve group, wherein the at least one corresponding curve refers to the curve that shares a similar position or similar geometrical properties relative to the three-dimensional object with the at least one curve.

16. A device for generating a digital picture, comprising
a receiving unit being configured to receive a three-dimensional representation for each of a plurality of curves, wherein each curve represents at least partially a shape of a three-dimensional object;

a grouping unit being configured to group the plurality of curves into at least one curve group based on the three-dimensional representations of the curves using three-dimensional information given by the three-dimensional representations;

an associating unit being configured to associate each curve group with a picture element based on the curves of the curve group;

a generating unit being configured to form the digital picture using the picture element.

17. A computer readable medium having a program recorded thereon, wherein the program is adapted to make a processor of a computer perform a method for generating a digital picture, the computer readable medium comprising

code of the program making the processor perform the reception of a three-dimensional representation for each of a plurality of curves, wherein each curve represents at least partially a shape of a three-dimensional object;

code of the program making the processor perform the grouping of the plurality of curves into at least one curve group based on the three-dimensional representations of the curves using three-dimensional information given by the three-dimensional representations;

code of the program making the processor perform the association of each curve group with a picture element based on the curves of the curve group;

code of the program making the processor perform the formation of the digital picture using the picture element.