Expressive Distortion of Strokes and 3D Meshes

Petra Neumann, Tobias Isenberg, Sheelagh Carpendale
Department of Computer Science
University of Calgary, Canada
{pneumann|isenberg|sheelagh}@cpsc.ucalgary.ca

Thomas Strothotte
Department of Simulation and Graphics
Otto-von-Guericke University of Magdeburg, Germany
tstr@isg.cs.uni-magdeburg.de

Abstract

We introduce methods for applying two-dimensional distortions in a manner that allows us to create three-dimensional variations of stroke-based models and 3D polygonal meshes. These variations can be achieved interactively, supporting the creation of expressive variations of a given 3D model or of strokes generated from it. In order to allow for a seamless integration into existing systems and to enable additional processing of both strokes and meshes, a pipeline approach is taken in both cases. In addition, we discuss the interaction possibilities arising from the interplay between the model orientation and the strokes or the geometry. Finally, to demonstrate the usability of the proposed methods we suggest several application domains.

Keywords: distortion, non-photorealistic rendering, artisic strokes, caricatures, cartoons, 3D mesh distortion, interaction techniques.

1 Introduction

Distortion approaches have often been used for modifying and manipulating data visualizations [Leung and Apperley 1994]. We extend this idea and apply distortion techniques to stroke and 3D mesh models, creating expressive renditions. We start with a common line rendering system (OPENNPAR, see [Halper et al. 2003]) to create a set of strokes or silhouettes and extend the ways by which one can manipulate this data. Distortion techniques are applied to the strokes early enough in the pipeline to still allow for either further stroke stylization or other stroke processing. In addition, we examine the extension of these methods to the distortion of 3D meshes for use in interactive adjustments and modeling.

Previously, only simple transformations (e. g., scaling, translations, or rotations) could be applied to line drawings using common vector graphics tools. The advantage of our methods is that new expressive effects can be easily created without having to otherwise modify the source data. For example, a single set of strokes can be used to create very different stroke renditions and a single mesh can be manipulated without having to remodel parts of it. In addition, two interaction techniques are possible at the same time: one of these is 3D object transformations and distortion of the resulting silhouette and the other is 3D object transformations and distortion of the mesh parallel to the view plane. The proposed methods provide additional freedom for a designer of models and/or stroke renditions.

Our main contributions lie in the seamless integration of distortion techniques into line rendering using a stroke pipeline approach, and the extension of this concept to the distortion of 3D polygonal meshes by similarly using a mesh pipeline. We also propose several new methods of interaction enabled by these approaches.

The remainder of this paper is structured as follows. First, we discuss some related work in the area of data distortion in Section 2. Next, Section 3 describes the underlying computational approach, Elastic Presentation Framework (EPF) [Carpendale and Montagnese 2001], upon which our distortion methods are based. Then, in Section 4, we present methods to distort stroke data. Next, we show how to extend this concept to mesh data in Section 5. This is followed by a discussion of interaction issues for both stroke and mesh distortion and a number of applications are suggested in Section 6. Finally, in Section 7 we conclude our paper and mention some aspects of future work.

2 Related Work

Distortion is usually introduced to solve the screen real estate problem, namely, how to display large information spaces on a small screen. LEUNG and APPERLEY give a good introduction to these early distortion-oriented presentation techniques [Leung and Apperley 1994]. In addition to the function of enabling the user to comprehend and view a large information space with sufficient context and detail, distortion is an often used tool for emphasis in illustrations (e. g., [Thomas and Johnston 1995; Hodges 2003]).

The distortion methods discussed in our paper are based on the methodology presented by CARPENDALE and MONTAGNESE [Carpendale and Montagnese 2001]. They combine many previously existing 'point' solutions for detail-in-context presentations and integrate them into a single geometric framework. By providing these methods through the application of only one geometric setup and one calculation method, an extrapolation between them is easily possible. Many different distortion effects can be created in a given interaction at the same time. A brief explanation of the details of this framework will be given in Section 3.

Distortion has been applied to the viewing of planar graphs using a graphical fish-eye lens [Sarkar and Brown 1992]. In this system, vertices of graphs are distorted according to a chosen distortion function. Edges are mapped at their endpoints or occasionally at intermediate bend points. In a fundamental work on rendering stroke-based drawings, HSU and LEE present their concept of *skeletal strokes* [Hsu and Lee 1994]. In this context, they discuss methods for distorting stroke definitions in order to fit them to paths.

Distortion has also been applied to line drawings. In particular, when creating caricatures distortion is often employed. Early examples were shown by BRENNAN who automatically distorts a manually defined polyline that resembles an average, front-viewed face based on exaggerations of the variations of specific faces [Brennan 1982; Brennan 1985]. This method has been improved by [Mo et al. 2004] who also take the variance of face features into account. RUTTKAY and NOOT presented their CHARTOON System that enables the design and animation of 2D cartoon faces [Ruttkay and Noot 2000]. Cartoon faces are composited from geometric primitives by the user. Reuse of facial components is permitted and a feature and expression repertoire allows the use of pre-defined deformations for the facial components. LIANG et al. [Liang et al. 2002] take an image-based approach by extracting face features from an input image of a person. These features are then distorted in image space by applying caricature prototypes. Another image-based approach [Hsu and Jain 2003] creates a face graph for input images. Caricatures are created by distorting this face graph-based on measures calculated for the difference of the current face graph from the average facial topology.

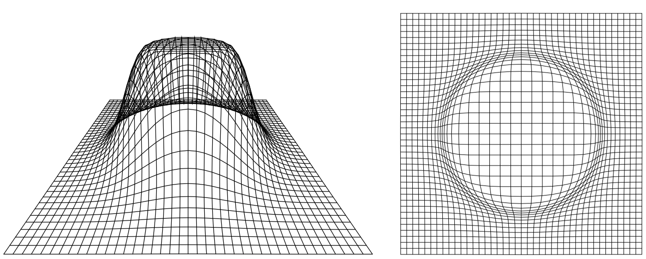
While these systems depend solely on an initial 2D information, MARTIN et al. [Martín et al. 2000] take another approach to introduce distortion in illustrations. Their approach uses *hierarchical extended non-linear transformations* to distort 3D polygonal meshes. Observer-dependent and orientation-dependent control functions are combined to extend the expressive capabilities of the finally rendered illustration. A similar approach was presented in 1999 by RADEMACHER [Rademacher 1999]. In his approach, a 3D model changes shape according to the view direction. Deformations of the object are created by interpolating key deformations that were defined for the model.

In contrast to these approaches that deal with distortion of either 2D strokes or 3D meshes, our work includes methods for both. In the following, we introduce techniques for the interactive distortion of line drawings as created from 3D polygonal meshes and the possibility of applying distortion to the underlying mesh itself.

3 Distortion in 2D With EPF Lenses

Our work makes use of CARPENDALE and MONTAGNESE's *Elastic Presentation Framework* (EPF) [Carpendale and Montagnese 2001]. Within this framework, a two-dimensional data representation is placed onto a base plane that itself is located in the x-y-plane in three-dimensional space. This base plane is viewed from above using perspective projection. By introducing manipulations to the base plane, different presentation styles can be created. These manipulations are called *EPF lenses* and are functions that can be parameterized to produce the desired distortion effects.

Moving parts of the plane in z-direction scales it in size (see Figure 1). An increase in z produces a magnification effect while a decrease produces a zoom-out effect. Therefore, magnification is simulated by moving parts of the base-plane closer to the viewpoint (see example in Figure 1(a)).



(a) Lens principle in 3D. (b) 2D lens effect.

Figure 1: The principle of EPF lenses visualized using a circular lens on a regular grid.

The specific magnification for an EPF lens is calculated using perspective projection. Specifying a magnification value for a lens changes the height of its focal area. Depending on its location in the focal area or in the drop-off region, a data point is raised to a specific height. This causes the distortion effect because the raised point is perceived to be located at a different position on the base plane (see Figure 1(a)).

Based on this procedure, the elastic presentation framework presents a very powerful, flexible, and fast means to specify and compute 2D distortion effects. For specifying a lens, a user only has to place the focus of the lens onto the two-dimensional data representation (as shown in Figure 1(b)) and assign parameters to it (e. g., focus size, magnification factor, etc.). According to this parameterization, the lens internally performs the relocation of data points in 3D (as in Figure 1(b)) and the back-projection to the 2D plane. Although the computations are performed in 3D, an EPF lens only needs a 2D coordinate P as input to compute the distorted 2D coordinate P'. Therefore, EPF lenses are used as "black boxes" in our work in order to compute two-dimensional distortion effects.

4 Distortion of Strokes

Many techniques in non-photorealistic rendering use strokes as the main primitive for depiction. For example, strokes are generated in silhouette rendering (e. g., [Isenberg et al. 2003a]), in hatching (e. g., [Hertzmann and Zorin 2000]), and in stippling (e. g., [Secord 2002]) using a 3D geometry model. In general, these techniques can be classified according to how they present the result of the stroke computation. Either the strokes are generated as pixels that are drawn differently than the background or they are presented analytically. The latter offers a lot of room for further processing. Therefore, in addition to applying textures or common path manipulations, 2D distortion can be applied in order to achieve a much greater variety of effects.

4.1 Repositioning Vertices

The usual process of stroke rendering is modeled in a stroke pipeline that consists of vertex coordinates and edge data that connects these vertices. This pipeline includes at least the following steps:

1. edge detection (e. g., silhouette edge detection or hatching lines computation),
2. edge concatenation that forms strokes from adjacent edges, e. g., to allow further stylization,
3. a hidden line removal (HLR) that removes invisible portions of the strokes, and
4. stroke rendering that creates the actual rendition.

In addition, there can be more steps that, e. g., improve the quality by removing artifacts or adding parameterization data or that modify the appearance of the strokes. In order to apply distortion to the strokes, a distortion step is inserted between steps 3 and 4, after the HLR is completed but before the stroke is rendered. The reason for this sequence is that if the distortion would be applied prior to the HLR, otherwise hidden parts would be visible as well because the HLR typically is based on non-distorted information (e. g., the mesh or a z-buffer).

In order to allow the user to apply EPF distortion as if a lens would be moved over the line rendition, the distortion has to be applied in camera coordinates. However, the stroke vertices of the pipeline not only live in world coordinates before distortion but have to be represented in world coordinates afterwards as well because further processing may rely on this. Therefore, we suggest the following procedure for the computation:

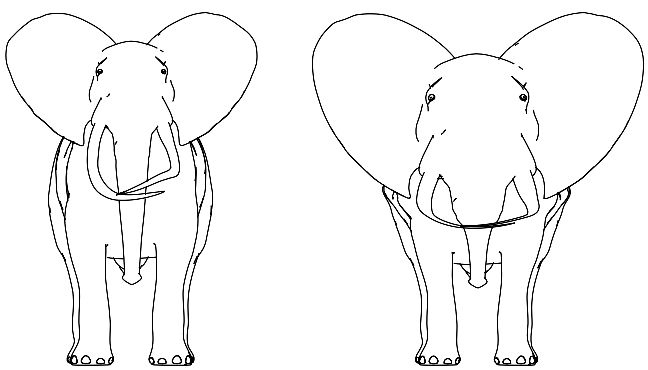
a) projection of vertices from world coordinates (x_w, y_w, z_w) into camera coordinates (x_c, y_c, z_c),

b) two-dimensional EPF distortion using only the (x_c, y_c)-coordinates of the projected vertices yielding the position (x'_c, y'_c), and

c) back-projection of the modified vertex in 3D (x'_c, y'_c, z_c) into world coordinates (x'_w, y'_w, z'_w) in order to be processed in the stroke pipeline.

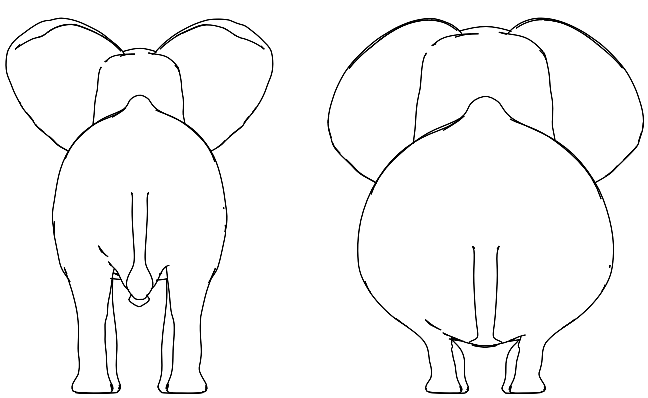
The projected z-depth in camera coordinates is not touched during the distortion step. Therefore, the back-projected strokes do not end up all in one plane in world-space. Instead, due to the distortion they are moved in world coordinates perpendicular to the line of sight. Hence, the vertices move in a plane in world coordinates. This way, further steps in the pipeline can still work with strokes that are represented in 3D.

Two examples for such distortions are depicted in Figures 2 and 3. Here, a 2D EPF lens is used to distort a silhouette rendition of an elephant. In the figures, the advantage of using analytic stroke representations instead of pixel images when applying distortion is clearly visible: although the lens significantly distorts the images, no sampling artifacts are visible because the lens only changes the position of the vertices. Therefore, there is no sampling grid that may cause visual artifacts. Instead, the sampling only happens when the distorted set of strokes is finally rendered.



(a) Undistorted rendition. (b) Dumbo (inspired by the Disney character).

Figure 2: A magnifying lens with a Gaussian dropoff function has been applied to the head of the elephant with the lens centered at the forehead in (a). This way, the ears are enlarged to create an image resembling Dumbo in (b).

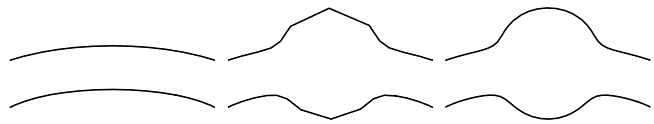


(a) Undistorted rendition. (b) Big bottom elephant.

Figure 3: A similar lens as in Figure 2 has been applied in this case with the lens being centered at the top of the elephant's bottom in (a) and, thus, creating the exaggerating caricature effect in (b).

4.2 Avoiding Artifacts from Discrete Distortion

In the discussion and examples so far only discrete distortion has been applied.[1] This means that the distortion is applied to the vertices of the strokes only. This, unfortunately, can cause unwanted artifacts due to the linear interpolation between the vertices of the strokes[2] because lenses may introduce a significant amount of magnification (see Figure 4).



(a) Undistorted. (b) Discrete. (c) Continuous.

Figure 4: Problem with discrete distortion and big magnification factors. Note the polyline artifacts in (b).

One way to overcome this problem is to use continuous distortion. This, however, requires the support for continuous distortion of strokes and curves in vector graphic file formats. Unfortunately, this is not present at the moment in the common formats such as PDF or Postscript. In order to still be able to create pleasing images with very few visible polygon artifacts, continuous distortion can be emulated by using a fairly dense set of vertices in the original strokes. However, this unnecessarily increases the required storage capacity, and the processing of the graphic is significantly slower. Since such a dense sequence of vertices is not required in all parts of the rendition, we suggest adaptively subdividing the strokes that are magnified and applying discrete distortion to the newly created vertices (see Figure 5). The degree of subdivision needed (i. e., the number of newly introduced vertices) for a specific stroke segment is derived from its local magnification factor. This way the density of vertices in a distorted stroke stays approximately the same as in those not distorted and the number of additional vertices is kept at the necessary minimum.

5 Distortion of Meshes

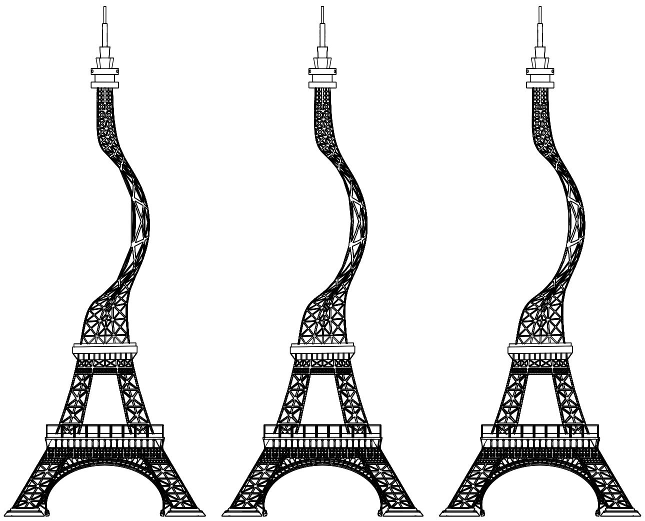
The process of distorting strokes can easily be extended to three-dimensional polygonal meshes. Similar to the strokes discussed previously, meshes consist of vertex coordinates and edges that provide a topology of the mesh by connecting the vertices. Therefore, the concept of stroke distortion can similarly be applied to meshes by also creating a basic mesh processing pipeline that allows manipulations of the mesh:

1. place mesh vertex coordinates and edges into the pipeline (e. g., by loading it from a file),
2. apply distortion to the vertex coordinates, and
3. output or render the mesh.

The use of a mesh pipeline has several advantages. First of all, in addition to these basic steps, further mesh processing such as subdivision or mesh simplification may be applied before and/or after

[1] For more information about discrete and continuous distortion see [Neumann and Carpendale 2003].

[2] Working with parametric curves would reduce this problem. However, even in this case the results may be awkward if only the positions of the control points are modified.



(a) No line subdivision, 32.102 edges. (b) Subdivision on all edges, 62.349 edges. (c) Adaptive subdivision, 47.382 edges.

Figure 5: Use of adaptive subdivision of strokes to imitate continuous distortion and limit the number of necessary edges. Note how the artifacts from discrete distortion are visible in (a) where the Eiffel tower has been distorted but disappear in (b) and (c). In addition, the rendition in (c) has even a slightly better quality than that in (b) because it uses more subdivision where necessary. Even so, (c) has significantly less edges than (b).[3]

the distortion step. Furthermore, the mesh data can be saved to a file which makes several successive steps of manipulation possible.

Similar to the stroke pipeline, only the vertices are manipulated during the distortion and moved perpendicular to the line of sight in world coordinates. Therefore, the same procedure that was used for stroke distortion can be used for mesh manipulation. The vertices are first projected into camera coordinates, then distorted parallel to the x-y-plane of the camera coordinate system, and finally back-projected into world coordinates (see example in Figure 6).

One major application of such a mesh distortion procedure lies in using it as an interactive modeling tool. This makes it possible for users to intuitively manipulate existing meshes while immediately seeing the result. This enables, in particular, inexperienced users of modeling tools to adapt meshes to their own needs without having to know details of how to create geometric models.

However, the two-dimensional nature of the distortion applied results in the model only being distorted in two directions. Therefore, to distort the model in all three dimensions, several successive steps of mesh distortion can be applied while rearranging the viewport for every distortion step. In fact, to achieve certain results, more than one step is necessary as has been shown in the example in Figure 7. The iterative application of distortion steps adds the results of individual lens effects if the model is “frozen” in between the steps.

Similar to stroke distortion, the discrete character of the distortion may lead to visible artifacts in form of large triangles when big magnification values are used. In order to avoid this, adaptive subdivision may be applied. Based on the LOOP subdivision scheme [Loop 1987], we used an adaptive method that subdivides triangles based on an *interest value* [Isenberg et al. 2003b]. The decision of whether to subdivide or not is based on this interest value parameter. This concept is similar to a degree-of-interest function [Furnas 1986] sometimes used in distortion approaches. The interest value is computed from the magnification that is introduced at a particular triangle (see Figure 8). Interestingly enough, the subdivision does not influence the interaction with the model because it only has to be applied when the final lens configuration for a particular distortion step has been found and the model is output.

6 Interaction and Application

In the two domains of two-dimensional strokes and three-dimensional meshes discussed above, two-dimensional distortion using EPF lenses was used as a 2D interaction metaphor. Using this metaphor, expressive manipulations of the underlying data can easily be achieved. In both cases, the use of a stroke or mesh pipeline allows for multiple lenses to be applied in successive steps.

The two-dimensional nature of the methods introduced raises some questions as well as new possibilities for interaction while applying distortion. In the following, some individual interaction issues and a number of application domains will be discussed for both stroke distortion and mesh distortion.

6.1 Stroke Manipulation

As discussed in Section 3, the EPF distortion used in this paper works only on 2D coordinates. In the case of stroke manipulation this means that the lens only affects the 2D component of the strokes. The third dimension that may also be computed (e. g., in case of silhouette or hatching lines) stays constant. Therefore, it is not destroyed in the process and can still be used after the distortion.

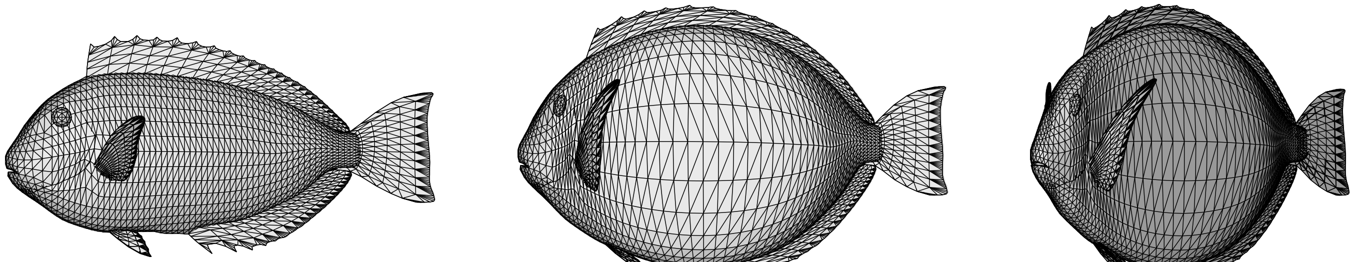
The major advantage of the proposed interaction method is that a user is able to work with the model in 3D (in particular, adapting the view: positioning, resizing, and rotating). At the same time, strokes (e. g., silhouettes or hatching lines) are generated from this 3D model. Therefore, instead of only working with a pre-calculated set of strokes, the user is able to generate 2D stroke data and modify the view on a scene at the same time. This enables the user to create line drawings with a greatly improved freedom of expression.

A common property of the stroke and mesh distortion methods discussed above is that the interaction with the original model occurs in 3D, while the 2D distortion occurs parallel to the viewplane at the same time. This means that if the orientation of the 3D model is modified (e. g., by rotating it) the lens still stays attached to the viewplane. Therefore, after the model manipulation, the lens may affect a different part of the generated set of strokes.

This two-dimensional nature of the distortion has to be considered during interaction with the model. However, this also is one of the major advantages of this interaction technique. Users that are used to working with line renditions are usually comfortable working in 2D. Hence, they do not have to learn a new interaction metaphor. Instead, they can work with 2D lenses that use the same interaction metaphors as traditional magnification glasses.

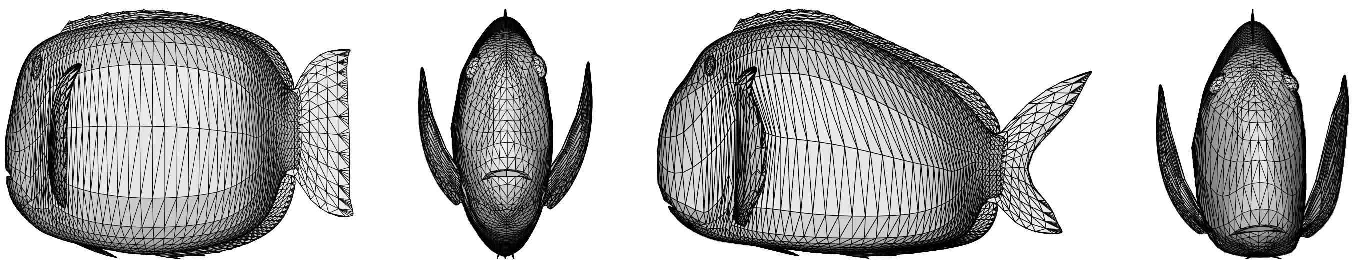
An important application domain is the generation of technical illustrations. In this area, the possibility of partial magnification may be used to show more detail at parts that are relevant in the illustration. Figure 9 shows an example for this application domain where some parts of a technical illustration have been magnified. In addition,

[3]Line subdivision in Figure 5(b) was simulated by applying one iteration of Loop subdivision (any other edge splitting scheme could have been used as well) to the mesh and re-computing the silhouette.



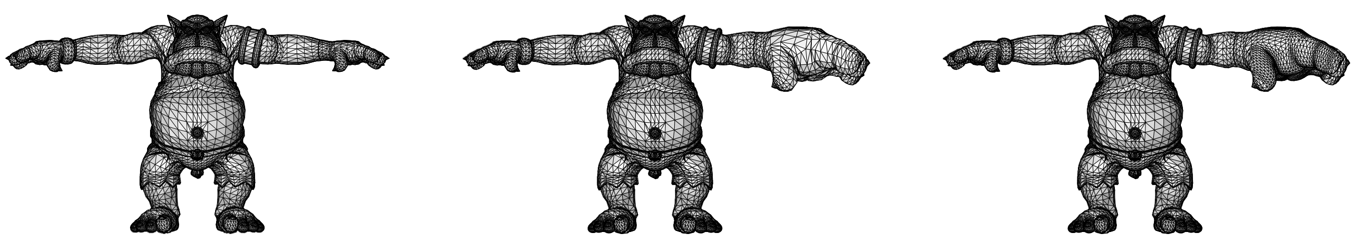
(a) Undistorted mesh. (b) Distorted mesh. (c) Side view of the mesh.

Figure 6: Mesh distortion perpendicular to the viewing direction.



(a) (b) (c) (d)

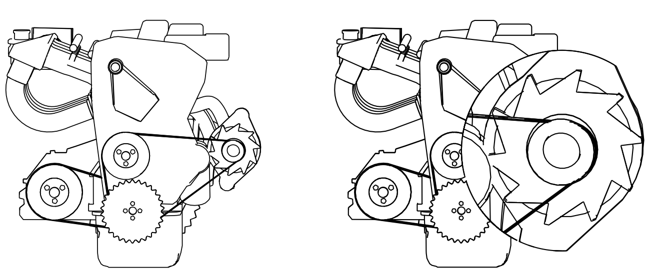
Figure 7: By applying several steps of mesh distortion, a user is able to create various different models from one single mesh employing the method as a modeling tool. The models shown here were created starting with the mesh in Figures 6(b) and 6(c). For each of the two meshes, a side view ((a) and (c)) and a front view ((b) and (d)) are shown.



(a) Undistorted mesh. (b) Distorted mesh. (c) Distorted mesh with adaptive subdivision.

Figure 8: Use of adaptive mesh subdivision to reduce artifacts from discrete magnification.

an interactive exploration of a line drawing illustration can provide more detail on demand. A related application that is noteworthy is the examination of generated or pre-defined stroke sets for errors by using magnification lenses to look at the details.



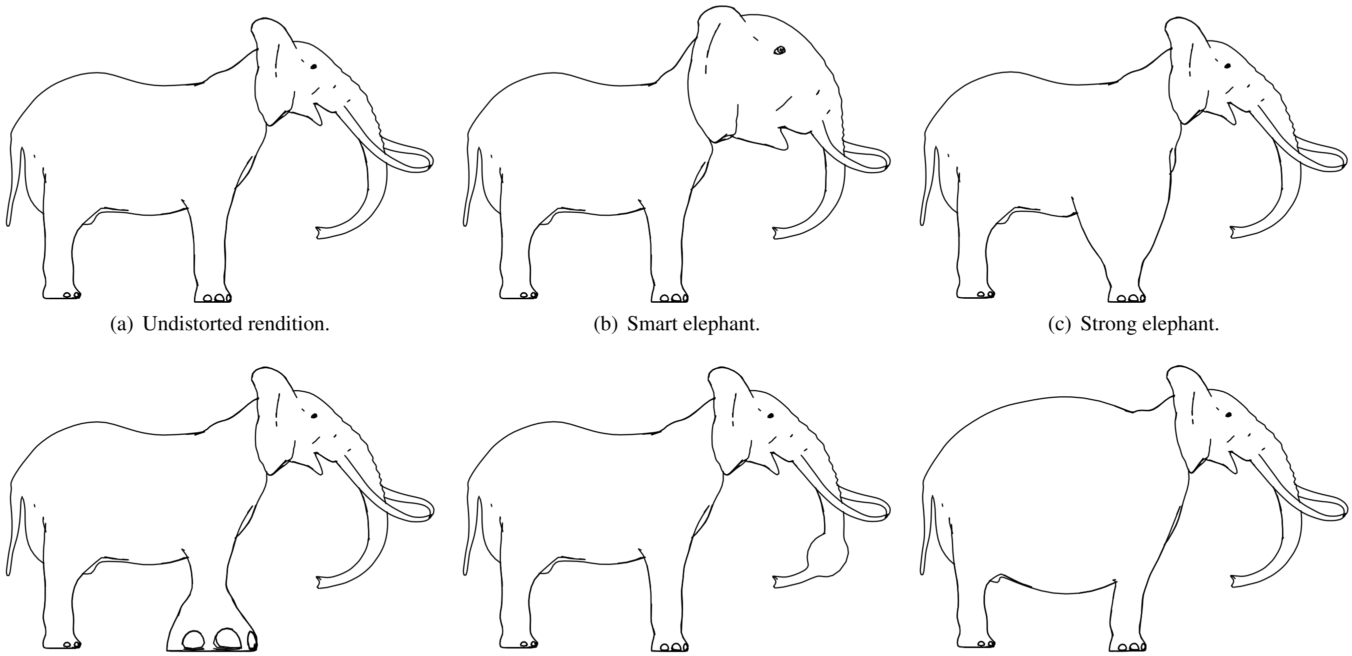
(a) Undistorted rendition. (b) Details magnified.

Figure 9: Application in technical illustration where areas with a lot of detail are magnified. Here, line subdivision is important so that polyline artifacts are avoided.

A second application domain for the stroke distortion is the generation of caricatures and comics. With the use of EPF lenses on silhouette strokes, certain features of a character can easily be exaggerated (e. g., see Figure 3). Users of the system do not have to own the special talent of a caricaturist or a cartoonist. They just 'play' with the given rendition until they receive pleasing results. In addition, many manipulations of the same set of stokes make it possible to create a diverse variety of expressions since the lenses are very powerful. Figure 10 shows such a set of drawings that were all generated from the same silhouette drawing (shown in Figure 10(a)). In addition, we created an example comic with our techniques as shown in Figure 11.

6.2 Mesh Manipulation

For the application of distortion to manipulating 3D meshes, it is also advantageous to be able to use powerful magnification lenses with only a small number of parameters. Users can apply modifications to a mesh without the need to touch or mark individual vertices. Although a similar manipulation is also possible in 3D mod-



(a) Undistorted rendition. (b) Smart elephant. (c) Strong elephant.

(d) Bigfoot elephant. (e) Drinking elephant. (f) Elephant with big belly.

Figure 10: Examples illustrating that a wide variety of expressions can be created from only one set of strokes just by using different lens parameters and positions.

eling tools such as 3D STUDIO MAX, the interaction there usually relies on the user's ability to navigate in 3D space using a 2D interaction device. Users are much more comfortable with and better trained for working with common 2D interaction devices (i. e., 2D mouse) than to navigating in 3D with 3D interaction devices (e. g., SpaceMouse, PHANToM, etc.) which is by far more difficult. In particular, (cartoon) artists are familiar with interacting with their drawings that essentially are 2D projections of 3D objects. Therefore, 2D mesh interaction may be more useful for such a group of users.

In addition, both modes of interaction with the object—interaction with size, position, and orientation as well as interaction with the mesh itself—are also possible in mesh distortion. With respect to using both types of interaction at the same time, the same considerations apply that were already discussed in Section 6.1.

Applications for mesh distortion are found wherever meshes need to be manipulated. One example application that is related to stroke distortion is the modeling of comic characters. For example, the "blow-up" and "sudden grow" effects known from comic animations can be easily modeled using the discussed method (see example in Figure 8 where the creature's (Olaf) hand has been enlarged). In addition, mesh distortion may also be used to detect errors in meshes similarly to line distortion. By magnifying parts of meshes while interactively working with 3D model and lens to view the details it is possible to locate errors.

7 Conclusion

In this paper we have introduced methods to apply two-dimensional distortion to strokes and 3D meshes to achieve artistic effects. In both cases, the use of a data pipeline has allowed us to combine the distortion of vertices with other manipulations. The two-dimensional distortion that we use occurs in camera coordinates and moves the vertices parallel to the viewing plane. We have shown that these techniques allow users to apply manipulations to strokes and meshes very easily and powerfully because of the many lens effects that are possible. In addition, we have discussed interaction issues that are raised by the new methods and how the techniques benefit users. In particular, the interaction with the view on the 3D model on the one side and the two-dimensional stroke or mesh distortion on the other side within the same environment offers exciting new possibilities to users. We also suggested a number of example applications such as comic rendering where these methods are easily applicable.

The methods were implemented in OPENNPAR, a non-photorealistic rendering system [Halper et al. 2003] from which we mainly used its pipeline-based line rendering and mesh manipulation capabilities. In the future, we plan to further explore the methods by extending the system to allow more freedom of expression to the users. In particular, we want to include the free definition of lenses by, e. g., using lens center primitives other than common primitives such as points, lines, circles, and rectangles.

Acknowledgements

Many thanks to Eric Pattison for providing support for the EPF library and to Anthony Tang and Stacey Scott for help with editing. This research is supported by the National Science and Engineering Research Council (NSERC), Canadian Foundation for Innovation (CFI), and Intel Inc.

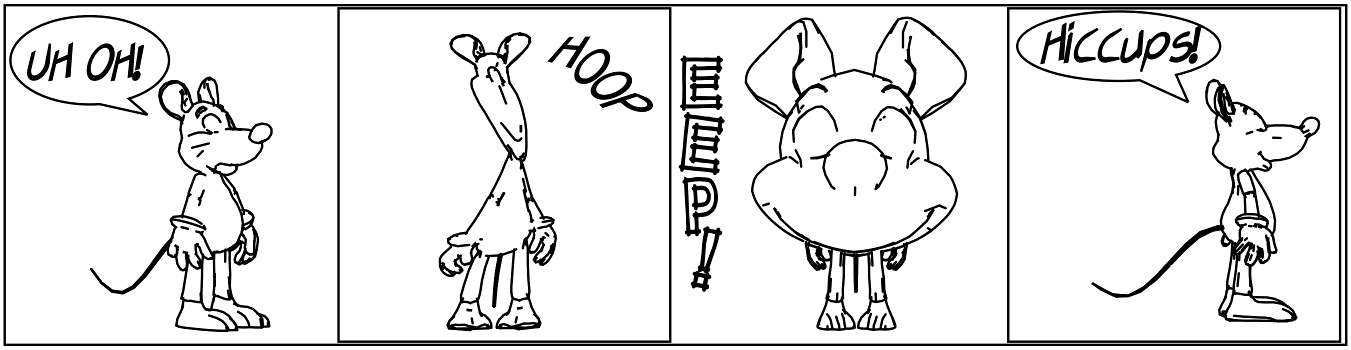


Figure 11: Cartoon inspired by a Calvin and Hobbes comic [Watterson 1990, page 121]. The images were created with the stroke distortion technique while in all cases the same 3D mesh was used.

References

BRENNAN, S. E. 1982. *Caricature Generator*. Master's thesis, Massachusetts Institute of Technology.

BRENNAN, S. E. 1985. Caricature Generator: The Dynamic Exaggeration of Faces by Computer. *Leonardo 18*, 3, 170–178.

CARPENDALE, S., AND MONTAGNESE, C. 2001. A Framework for Unifying Presentation Space. In *Proc. of UIST 2001*, ACM Press, New York, 61–70.

FURNAS, G. W. 1986. Generalized Fisheye Views. In *Proc. of CHI 86*, ACM Press, New York, 16–23.

HALPER, N., ISENBERG, T., RITTER, F., FREUDENBERG, B., MERUVIA, O., SCHLECHTWEG, S., AND STROTHOTTE, T. 2003. OpenNPAR: A System for Developing, Programming, and Designing Non-Photorealistic Animation and Rendering. In *Proceedings of Pacific Graphics 2003*, IEEE, Los Alamitos, CA, 424–428.

HERTZMANN, A., AND ZORIN, D. 2000. Illustrating Smooth Surfaces. In *Proc. of SIGGRAPH 2000*, ACM Press, New York, 517–526.

HODGES, E. R. S., Ed. 2003. *The Guild Handbook of Scientific Illustration*, 2nd ed. John Wiley & Sons, Hoboken, NJ.

HSU, R.-L., AND JAIN, A. K. 2003. Generating Discriminating Cartoon Faces Using Interacting Snakes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 11 (Nov.), 1388–1398.

HSU, S. C., AND LEE, I. H. H. 1994. Drawing and Animation Using Skeletal Strokes. In *Proc. of SIGGRAPH 94*, ACM Press, New York, 109–118.

ISENBERG, T., FREUDENBERG, B., HALPER, N., SCHLECHTWEG, S., AND STROTHOTTE, T. 2003. A Developer's Guide to Silhouette Algorithms for Polygonal Models. *IEEE Computer Graphics and Applications 23*, 4 (July/Aug.), 28–37.

ISENBERG, T., HARTMANN, K., AND KÖNIG, H. 2003. Interest Value Driven Adaptive Subdivision. In *Simulation und Visualisierung 2003*, SCS European Publishing House, Erlangen, San Diego, 139–149.

LEUNG, Y. K., AND APPERLEY, M. D. 1994. A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Transactions on Computer-Human Interaction 1*, 2 (June), 126–160.

LIANG, L., CHEN, H., XU, Y.-Q., AND SHUM, H.-Y. 2002. Example-Based Caricature Generation with Exaggeration. In *Proc. of Pacific Graphics 2002*, IEEE Computer Society Press, Los Alamitos, CA, 386–393.

LOOP, C. T. 1987. *Smooth Subdivision Surfaces Based on Triangles*. Master's thesis, University of Utah, Department of Mathematics.

MARTÍN, D., GARCÍA, S., AND TORRES, J. C. 2000. Observer Dependent Deformations in Illustration. In *Proc. of NPAR 2000*, ACM Press, New York, 75–82.

MO, Z., LEWIS, J., AND NEUMANN, U. 2004. Improved Automatic Caricature by Feature Normalization and Exaggeration. In *ACM SIGGRAPH 2004 Conference Abstracts and Applications*, ACM Press, New York.

NEUMANN, P., AND CARPENDALE, S. 2003. Taxonomy for Discrete Lenses. Tech. Rep. 2003-734-37, University of Calgary, Canada.

RADEMACHER, P. 1999. View-dependent Geometry. In *Proc. of SIGGRAPH 99*, Addison Wesley Publishing Company, New York, 439–446.

RUTTKAY, Z., AND NOOT, H. 2000. Animated CharToon Faces. In *Proc. of NPAR 2000*, ACM Press, New York, 91–100.

SARKAR, M., AND BROWN, M. H. 1992. Graphical Fisheye Views of Graphs. In *Proc. of CHI'92*, ACM Press, New York, 83–91.

SECORD, A. 2002. Weighted Voronoi Stippling. In *Proc. of NPAR 2002*, ACM Press, New York, 37–43.

THOMAS, F., AND JOHNSTON, O. 1995. *The Illusion of Life: Disney Animation*. Hyperion, New York.

WATTERSON, B. 1990. *Calvin and Hobbes – Weirdos from Another Planet*. Warner Books, London, UK.