

SpringLens – Distributed Nonlinear Magnifications

T. Germer¹, T. Götzelmann¹, M. Spindler¹, and Th. Strothotte¹

¹Department of Simulation and Graphics, Otto-von-Guericke University of Magdeburg, Germany

Abstract

We present a flexible, distributed and effective technique to model custom distortions of images. The main idea is to use a mass-spring model to create a flexible surface and to create distortions by changing the rest-lengths. A physical simulation works out the displacements of this particle grid. We provide intuitive tools to interactively design such nonlinear magnifications. In addition, our system enables data-driven distortions which allows us to use it for automatic nonlinear magnifications. We demonstrate this with an application for labeling of 3D scenes.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Display algorithms, I.3.5 [Computer Graphics]: Physically based modeling, I.3.6 [Computer Graphics]: Interaction techniques

1. Introduction

The terms “nonlinear magnification”, “fisheye views” and “focus and context visualization” describe various techniques to create distortions of two-dimensional data like images. Although very different aspects have been explored and emphasized in the past [LA94], the main goal has always been to create distorted images or illustrations, such that certain elements were highlighted (or magnified) while other elements provide a form of context or background. Applications of nonlinear magnifications reach from information-visualization to image manipulation (see figure 2).

One can distinguish two main problems in creating nonlinear magnifications. First, a model for the distortion has to be designed, describing the way in which the image is transformed. Second, the distorted image has to be rendered. In this paper, we focus on the first problem and propose a new technique to model nonlinear magnifications. One can further distinguish distortion modeling into analytical approaches with predefined mathematical functions being applied to distort images (e.g. [CM01]), and procedural approaches. Here the system distorts the image given a user-defined, procedurally created “magnification-map” (e.g. [KR97]). We follow the second paradigm and present a flexible, distributed technique for the procedural modeling of nonlinear magnifications. The chosen approach results in image-distortions which exhibit a form of self-organization and which are flexible and efficient enough for being qualified for interactive 3D applications like labeling.

2. Related Work

The exploration of nonlinear magnification reaches back until the beginning of computer graphics. Particularly important techniques are polyfocal displays [KS78], rubber sheets [SSTR93] and graphical fisheye views [SB92]. They all define analytical distortions being applied to images or planar graphs. A different approach is to work in 3D space and to use perspective projections to create nonlinear magnifications. While the perspective wall [MRC91] uses three planes to define a scene to be viewed with a perspective camera, the unified presentation space [CM01] uses one flexible surface with varying height which is viewed from the top. Even though these techniques provide a very intuitive metaphor to nonlinear magnification, they also can be seen as a misuse of perspective projection. While we want to achieve very flexible distortions, perspective projection and homogenous division (which actually distorts the image) represent a very rigid form of deforming images. This causes problems, e.g., with multiple foci.

An important concept introduced by Leung and Apperly [LA94] and deeply examined by Keahey and Robertson [KR97] is the distinction between a transformation- and a magnification-function. The former describes the actual image deformation. Being the derivative of the transformation-function, the magnification-function describes the *degree* of magnification. Keahey and Robertson also formed the notion of “magnification fields”, which are custom, user-defined magnification-functions. By integrating such a mag-

nification field into a transformation-function, very diverse forms of distortions can be accomplished. Unfortunately, this integration is not trivial. Keahey and Robertson suggest an iterative, non-interactive algorithm to approximate the transformation-function [KR97], resulting in a powerful technique to create general distortions of images. They also present the idea of data-driven magnification, where attributes from the data are used to control the distortion.

Yang et al. demonstrate how to apply nonlinear magnifications to 3D scenes [YCB05]. They derive analytical distortion functions and use them to deform a grid-mesh. The 3D scene is rendered into an offscreen-buffer which is used to texture the grid resulting in a distorted image of the 3D scene. We use a similar approach to implement our labeling application (see section 5). However we use a more flexible way to *model* the nonlinear magnification.

3. Spring Model

The main idea of our approach consists of using a mesh of *springs* to distort a two-dimensional surface. Therefore, we introduce the spring mesh metaphor to model a flexible lens, the so called “SpringLens”. It consists of a set of discrete particles (i.e., mass-points) which are arranged in a grid with fixed resolution. Each particle (except for particles at the border of the grid) has 4 neighbors – above, below, right and left. Neighboring particles are connected through springs and therefore exert forces onto each other. These forces affect the motion of the particles which is calculated by a simple iterative physical simulation over time. Figure 1 shows a spring mesh with complex nonlinear magnifications. Having set up the grid of connected particles, an easy way of moving the particles away from their initial regular grid position is to change the rest-length of the springs. If we dynamically change the “size” of some selected springs, the particles will move due to the underlying simulation, creating a *distorted* grid.

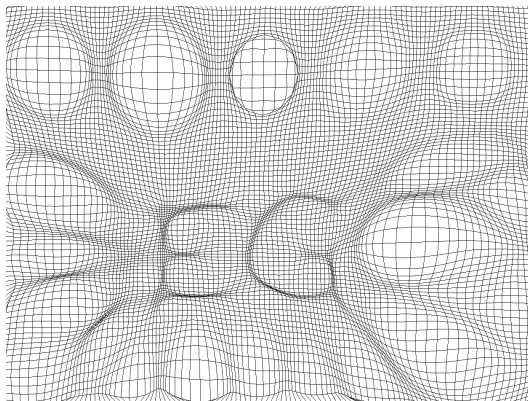


Figure 1: Distortion of a grid with various cursor shapes in the upper part and free-hand distortion in the lower part. The “EG”-letters were created by free-hand shrinking.

Physical Model: The state of particle i consists of three physical attributes: position p_i , velocity v_i and rest-length r_i . When a force f acts on a particle with mass m , it applies the acceleration $a = f/m$ on it. This has to be integrated to get the velocity $v_i = \int f/m dt$ and the new position $p_i = \int v_i dt$. Although we have tried different methods, a simple Euler-integration (i.e., $v_i = \frac{f}{m} \Delta t$ and $p_i = v_i \Delta t$) works well for us.

Between two neighboring particles act spring forces. In general, spring forces are described by Hooke’s law $f_h = -kx$, where f_h is the resulting force, k is the spring constant (i.e., its stiffness) and x is the amount of displacement from the rest-position of the spring. In our system, the rest-position is initialized according to the distances between the particles in the regular grid. Since we actually need a local magnification factor for the *particles*, we store the rest-length for each particle (r_i). The rest-length r for a spring is the sum of its two adjacent particles. In general, a spring force always acts in the opposite direction of the spring-displacement. That means its direction always coincides with the direction of the spring. In the context of SpringLens, this leads to unwanted behavior because the spring mesh could easily “fold back” onto itself. In this case, the spatial relations between the particles (above, below, left, right) are not preserved. To prevent this, we propose a simple modified spring-model. Consider a particle i with position $p_i = (p_{i,x}, p_{i,y})$ in an orthogonal grid. The rest-position p'_n of its upper neighbor can be determined by $p'_n = (p_{i,x}, p_{i,y} + (r_i + r_n))$, where r_i and r_n are the rest-lengths for the particles i and n . Similarly, the rest-positions p'_s, p'_e and p'_w of its lower, right and left neighbors can be calculated by $p'_s = (p_{i,x}, p_{i,y} - (r_i + r_s))$, $p'_e = (p_{i,x} + (r_i + r_e), p_{i,y})$ and $p'_w = (p_{i,x} - (r_i + r_w), p_{i,y})$. Having defined these rest-positions $p'_j, j \in \{n, s, e, w\}$ of the four neighbors, we can calculate a directed displacement $d_j = p'_j - p_j$ from the actual position of each neighbor to its rest-position. The directed spring force is then simply $f_{s_j} = kd_j$. This directed spring force aims to preserve the spatial relations between particles (i.e., above, below, left, right). At the same time, it tries to establish the given rest-distance between them.

If we exclusively apply the spring force to the particles, the system tends to oscillate and to propagate wavelike displacements over the whole grid. To prevent the system from oscillating we add a damping force $f_{d_j} = b(v_j - v_i)$ to each particle i for each neighbor j . Finally, to keep the displacements local, we add a global damping term $f_g = -cv_i$ to each particle. The forces add up to the total force $f_i = f_g + \sum_{j \in \{n, s, e, w\}} (f_{s_j} + f_{d_j})$ acting on each particle i .

Constraints: Due to the distributed approach of SpringLens, we can very easily incorporate constraints to our system. To constrain the distortion at a particle i we simply fix its position and velocity. Then, distortions neither affect it, nor do they propagate beyond it. We use this technique to fix the particles at the edge of the spring mesh to the border of the image.



Figure 2: Distortion of a face. The partial grid in the upper left visualizes the cursor.

Rendering: To actually render the distorted image, we apply a similar technique as in [YCB05] and use a simple textured mesh. We create a surface consisting of quads between the particles and textured by the source image which is rendered on a screen-parallel plane. In addition, we use a new combined visualization which conveys both existing distortion and future distortion (i.e., cursor position and cursor shape). Near the cursor position, we overlay a semitransparent grid which coincides with the springs. The transparency correspond with the “strength” of the cursor. The partially displayed grid shows the cursor position and shape as well as the current distortion at the cursor (see figure 2).

4. Interaction

The interactive design of nonlinear magnifications is a main feature of SpringLens. The performance mainly depends on the number of particles used. A moderate tessellation of the spring mesh (around 150x150 particles) allows both expressive distortions and interactive speeds (about 50 fps on a 2 GHz PC). To manipulate the distortion, we change the rest-lengths according to the cursor described in the previous section using a drop-off function. The user can change the size and the shape of the cursor (i.e., the type of drop-off function). A temporary magnification can be applied by enlarging the rest-lengths of the particles by a predefined increment according to cursor position and shape. The effect can be described as a context-preserving magnifying glass moved over the image. The user can also “draw” a custom persistent magnification. By dragging the cursor, the rest-lengths under the cursor can be permanently increased or decreased. This allows the user to magnify multiple areas of custom shape. To remove the distortion, we simply restore the default rest-length for each particle.



Figure 3: Data drive distortion: Italy and Great Britain are enlarged, while Spain and Scandinavia are shrunk. The region-cursor is currently located at Scandinavia.

5. Data-driven distortion

Due to the distributed nature of SpringLens, it is very easy to incorporate data-driven distortion. We implemented an example using an additional object-ID buffer (created by manual segmentation) to identify homogeneous regions in the source image. The system provides a special “region” cursor which is used to magnify selected regions. To apply such a distortion, the system first detects the object-ID of the particle at the mouse position to find the selected object. Then, the rest-lengths of all particles with the same ID are manipulated according to the current interaction mode. To prevent distortions at the border of the selected region, the magnification is smoothed by averaging the rest-length of neighboring particles to some extent. This way, SpringLens enables automatic, smooth magnifications of arbitrary regions in the image. Note that we don’t use any analytically defined drop-off function here and rather rely on intrinsic features of the image to define the magnification. In figure 3 we have applied this technique to magnify some regions of Europe.

A main advantage of our approach is that it allows data-driven distortion in real-time, allowing for interactive 3D applications. We propose a similar approach to [YCB05] and first render the 3D scene to an offscreen buffer. In addition, we render the color-coded scene into an object-ID-buffer. The particle mesh is then rendered as described in section 3, enabling us to magnify arbitrarily shaped objects in the 3D scene. As the view of the 3D scene is moved, the magnification implicitly follows the selected object. This can be used as an emphasis technique, like scientific illustrators do to guide to viewer attention. Another aim is to preserve enough space to accommodate annotations on important objects. While the user interactively zooms and rotates the 3D object, the important objects can be coherently magnified. We adapted this with an approach [GHS06] for interactively labeling of 3D objects (see figure 4).

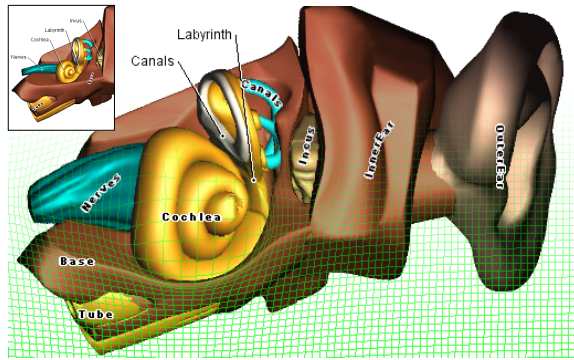


Figure 4: Illustration of the human ear with nonlinear magnification of selected organs. The grid illustrates the distortion. Top left: undistorted model.

6. Discussion

The proposed techniques are easy to implement, yet effective for creating complex distortions. Due to its flexibility, the SpringLens model is perfectly suited for complex applications like automatic data-driven magnification and labeling of 3D scenes. Another benefit of SpringLens is its comprehensibility of interactive magnification design. As the user controls the magnification cursor he immediately can see the effect. However, the magnification not just “pops up”, but evolves very smoothly. The underlying simulation resembles a physical look-and-feel, which contributes to the tangibility of the system. This way, the user can develop a deeper understanding of the distortion and is guided as he designs custom nonlinear magnifications. The accompanying video shows the appeal of SpringLens.

The distributed nature of particles acting on each other gives SpringLens a form of self-organization. If a region of interest is enlarged, spring forces act upon neighboring regions which cause them to be shrunk. Note that the context-particles are not explicitly shrunk but rather affected from the region in focus. Two regions in focus next to each other implicitly exert forces onto each other, which causes them to diverge. This results in a smoother magnification and in an effective utilization of available space. This effect is implicitly worked out by the simulation model and can be seen as a form of self-organization between the particles.

Some parameters of the physical simulation like stiffness (k) and damping (b) can be tweaked to adjust its behavior to either high responsiveness (high k) or smooth motion (high b). Higher values for global damping (c) keeps the effect more local while lower values convey a higher effect of plasticity. Unfortunately the time-step Δt also affects the simulation. With larger values for Δt , fewer iterations are needed for the physical simulation to converge. The drawback is the loss of accuracy. Moreover, very big time-steps as well as very high stiffness and low damping values make the sys-

tem unstable. In our implementation, the time-step can be adjusted to match the values for stiffness and damping.

A drawback of the grid-based approach of SpringLens is its limitation of details that can be magnified. In the application of data-driven distortions, this implies that very small objects cannot be magnified. A third problem arises from the bitmap approach implemented to render the scene. The higher the magnification, the higher texture resolution is needed. Future work will search for solutions to enlarge the possible resolution of the grid and to overcome limitations from texturing with fixed resolution.

References

- [CM01] CARPENDALE M. S. T., MONTAGNESE C.: A framework for unifying presentation space. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology* (New York, NY, USA, 2001), ACM Press, pp. 61–70.
- [GHS06] GOETZELMANN T., HARTMANN K., STROTHOTTE T.: Agent-Based Annotation of Interactive 3D Visualizations. In *6th International Symposium on Smart Graphics* (2006), Springer Verlag.
- [KR97] KEAHEY T. A., ROBERTSON E. L.: Nonlinear magnification fields. In *INFOVIS '97: Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)* (Washington, DC, USA, 1997), IEEE Computer Society, p. 51.
- [KS78] KADMON N., SHLOMI E.: A polyfocal projection for statistical surfaces. *The Cartographic Journal* 15, 1 (June 1978), 36–41.
- [LA94] LEUNG Y. K., APPERLEY M. D.: A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comp.-Hum. Interact.* 1, 2 (1994), 126–160.
- [MRC91] MACKINLAY J. D., ROBERTSON G. G., CARD S. K.: The perspective wall: detail and context smoothly integrated. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1991), ACM Press, pp. 173–176.
- [SB92] SARKAR M., BROWN M. H.: Graphical fisheye views of graphs. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1992), ACM Press, pp. 83–91.
- [SSTR93] SARKAR M., SNIBBE S. S., TVERSKY O. J., REISS S. P.: Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology* (New York, NY, USA, 1993), ACM Press, pp. 81–91.
- [YCB05] YANG Y., CHEN J. X., BEHESHTI M.: Non-linear perspective projections and magic lenses: 3d view deformation. *IEEE CG&A* 25, 1 (2005), 76–84.