Jellylens

Content-Aware Adaptive Lenses

Cyprien Pindat, Emmanuel Pietriga, Olivier Chapuis et Claude Puech

English Abstract—Focus+context lens-based techniques smoothly integrate two levels of detail using spatial distortion to connect the magnified region and the context. Distortion guarantees visual continuity, but causes problems of interpretation and focus targeting, partly due to the fact that most techniques are based on statically-defined, regular lens shapes, that result in far-from-optimal magnification and distortion. JellyLenses dynamically adapt to the shape of the objects of interest, providing detail-in-context visualizations of higher relevance by optimizing what regions fall into the focus, context and spatially-distorted transition regions. This both improves the visibility of content in the focus region and preserves a larger part of the context region. In this article we summarize the approach and its implementation.

1 Introduction

Our computer screens are very small compare to the size of some of the dataset we want to visualize. Besides, we keep producing data faster exaggerating the disproportion. This includes maps produced by geographical information systems ; satellite or astronomical imagery ; as well as numerous information visualizations such as complex networks represented as node-link diagrams. None of these datasets fit on computer screens, not even high-resolution wall displays. This call for more powerful visualization techniques, enabling to transition from high-level, low resolution overviews of the data, to zoomed-in, highly-detailed representations of a region of interest.

The three schemes we typically use to navigate large information spaces are: Overview+detail, zooming & panning and Focus+context (F+C) [3]. Techniques based on the latter integrate a detailed representation of a region of interest directly in the surrounding context [4] (fisheye view), which essentially corresponds to the original, unmagnified visualization showing more data at a lower scale. This has both advantages of offering a better understanding of the relationship between the more precise and the contextual views, and permitting the system to easily integrate several lens sharing the same display.

Dispite their potential advantages, F+C visualization systems are far from beeing popular in ubiquitous desktops environment. One significant problem that remains to be addressed is the mismatch between the shape of the lens and the shape of the object(s) of interest. Indeed, most magnification techniques are based on regular lens shapes statically defined by distance functions easily obtained through $L(P)$-metrics [2].

In this article I will introduce Jellylenses [6], which consists in two complementary visualization techniques. I will first present PathLens and then AreaLens.

- *Cyprien Pindat: Univ Paris-Sud & CNRS (LRI) 91405 Orsay, France E-mail: pindat@lri.fr.*
- *Emmanuel Pietriga: INRIA Chile (CIRIC) 7561211 Santiago, Chile E-mail: emmanuel.pietriga@inria.fr.*
- *Olivier Chapuis: Univ Paris-Sud & CNrS (LRI) 91405 Orsay, France E-mail: chapuis@lri.fr*
- *Claude Puech: INRIA Chile (CIRIC) 7561211 Santiago, Chile E-mail: claude.puech@inria.cl*

2 PathLens

PathLens, consists of a lens attached to the mouse cursor, that adapts its shape, circular by default, to the graphical objects considered of interest based on the cursor's location. Intuitively, PathLenses behave approximately like water drops on a spider net, or more generally speaking like drops on an irregular surface featuring elements of varying affinity [7].

Behavior: PathLenses assume a default circular shape that gets combined with implicit descriptions of the geometry of some objects in the scene based on distance fields. The actual shape of a PathLens at a given focal point **P** in the visualization depends on what objects of potential interest are in the vicinity. As illustrated in Figure 1, when far away from any object, a PathLens adopts the default circular shape, and progressively morphs, conserving the same area, as it approaches an object of interest. Not all objects in the scene are necessarily taken into account, and each application can define what objects the lens will adapt to. For instance, on Google Maps^{TM}, a PathLens could be made to adapt to interstate roads only, ignoring any highway, service road or other geographical landmark such as parks and water bodies.

Adaptation We presented a method based on soft-objects from Wyvill [8], to achieve the desired morphing effect (Figure: 1). We define the shape by an implicit function, $f(\mathbf{x})$, as the set of all points $\mathbf{x}$ such that $f(\mathbf{x}) = s$, s being an iso-level. Tessellation of implicit shapes is achieved using the marching square algorithm [5].

The cursor position and the geometry of nearby objects of interest both contribute to the definition of the lens' final shape. These influences are represented in terms of contribution functions: $lens(\mathbf{x})$ represents the contribution of the default lens shape to the final shape, $data(\mathbf{x})$ represents the contribution of proximal objects to the final shape. The shape is thus implicitly represented by:

$$f(\mathbf{x}) = lens(\mathbf{x} - \mathbf{P}) + data(\mathbf{x}) \quad (1)$$

where **P** is the cursor's coordinates.

By carefully choosing $lens(\ \mathbf{x})$ and $data(\mathbf{x})$, we achieve the desired effect shown in figure 1. Moving away from the object, the contribution of $data(\mathbf{x})$ drops smoothly to zero, the lens reverts to its default circular shape. The above method is used to compute both the focus and context regions' contours.

Rendering: We presented an algorithm based on Beier and Neely image morphing technique [1], that transforms a source image into a distorted image. The source image is embedded with non overlapping simple polygons that act as handle for the deformation technique. Enlarging a handle enlarges the region inside it; translating a polygon translates the region inside it; shearing a polygon shears the region inside it. Given a set of handles, each one associated with a transformation, the system creates a mapping that maps the image encompassed in each handle (flat-tops and context) to its destination and, and adjusts the rest of the layout (distortion area) to integrate it smoothly.

To render the AreaLens adaptation techniques, two handles are necessary. One is defined by the context shape, the second by the focus shape. We then act on the focus handle by the magnification transformation define by the focus point and the magnification factor, the identity is associated with the context handle.

3 AREALENS

The second technique, *AreaLens*, consists in the dynamic relayout and resizing of all objects that fall in the lens' scope. The technique tries to preserve the original aspect of those objects as much as possible, magnifying those closest to the cursor while reducing those closer to the lens' periphery, and distorting the regions in-between. The two techniques are complementary, providing solutions to a wide range of tasks and visual configurations. Both work for arbitrary 2D datasets, ranging from networks and maps displayed as vector graphics (Figure 2) to documents and Web pages.

Behavior: While PathLenses only consider the closest object of interest during the adaptation process, AreaLenses consider all objects of interest within a given area of influence. This makes the second technique better suited to the magnification of filled shapes in dense scenes, while PathLenses are better suited to magnification of paths or filled shapes in sparse scenes (in terms of regions of interest).

As the user moves the mouse cursor, objects of interest in a certain area of influence are either smoothly pulled towards the cursor as it approaches them, or pushed away as it gets away from them, eventually reverting to their original location and size when getting out of the area of influence.

Adaptation: This is achieved by two concurrent mapping algorithms a *dispersion* mapping, and a *magnification* mapping. The purpose of the *dispersion* mapping is to push objects away and shrink them to accommodate the objects that will get magnified. The purpose of the *magnification* mapping is to pull objects towards the cursor and magnify them. The *magnification* mapping takes as input the cursor position and identifies the closest objects, to be magnified. The *dispersion* mapping takes as input this set of magnified objects and spreads out the remaining objects. The overall mapping algorithm result in the definition of a transformation for each objects involved that would be used in the rendering process.

Rendering: The same rendering method as for PathLens is used for rendering AreaLenses. We need as many handle as regions of interests. We act on each handle with the transformation resulting from the mapping process.

Evaluation: We conducted a controlled experiment to evaluate the benefits of the AreaLens technique. The purpose of this experiment was both to evaluate the actual performance gain under different conditions, if any, and to assess the potential negative impact of the dynamically changing geometry, that might cause confusion and visual discomfort. We compared regular fisheyes (circular shape, Gaussian drop-off) to AreaLenses, that have a stronger impact in terms of visual changes than PathLenses, as they affect more objects and are thus more likely to suffer from this, especially in dense configurations.

The experiment showed both quantitative and qualitative encouraging results. AreaLens outperformed a classical *Fisheye*, and participants gave positive feedback, they did not report being distracted and found very convenient the way the AreaLens magnify the region of interest all at once.

4 CONCLUSION

Jellylenses break the usual behavior of fisheye lenses by proposing to dynamically change their geometry while preserving visual continuity between the focus and context regions. Our empirical evaluation shows that this approach has strong potential, though this is of course only a first step. More evaluation and more case studies are needed to better understand the advantages and weakness of the approach and of each technique.

References

[1] T. Beier and S. Neely. Feature-based image metamorphosis. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '92, pp. 35–42. ACM, 1992.

[2] M. S. T. Carpendale and C. Montagnese. A framework for unifying presentation space. In *UIST '01: Proceedings of the ACM Symposium on User Interface Software and Technology*, UIST '01, pp. 61–70. ACM, 2001.

[3] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41:2:1–2:31, 2009.

[4] G. W. Furnas. Generalized fisheye views. In *ACM SIGCHI Bulletin*, volume 17, pp. 16–23. ACM, 1986.

[5] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pp. 163–169. ACM, 1987.

[6] C. Pindat, E. Pietriga, O. Chapuis, and C. Puech. Jellylens: Content-aware adaptive lenses. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, p. to appear, New York, NY, USA, 2011. ACM.

[7] H. Wang, P. J. Mucha, and G. Turk. Water drops on surfaces. *ACM Trans. Graph.*, 24:921–929, July 2005.

[8] G. Wyvill. Data structure for soft objects. *The visual computer*, 2(4):227–234, 1986.

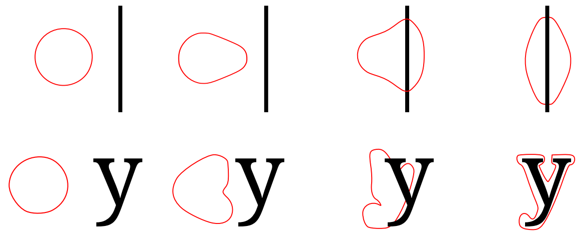

Fig. 1. PathLens morphing effect.

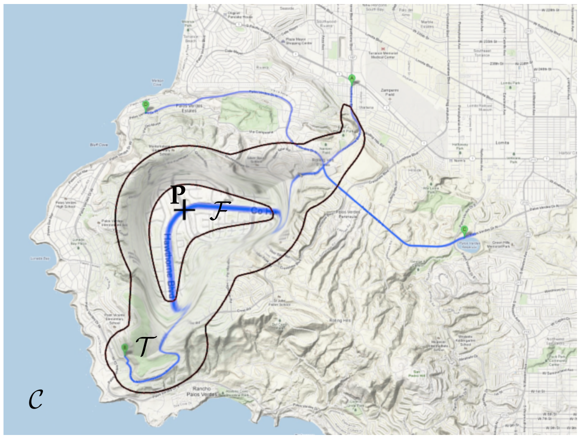

Fig. 2. Regions defined by a constrained lens: focus region $\mathcal{F}$ magnifying the region of interest, context region $\mathcal{C}$, and smooth transition $\mathcal{T}$ between $\mathcal{F}$ and $\mathcal{C}$ achieved through distortion.

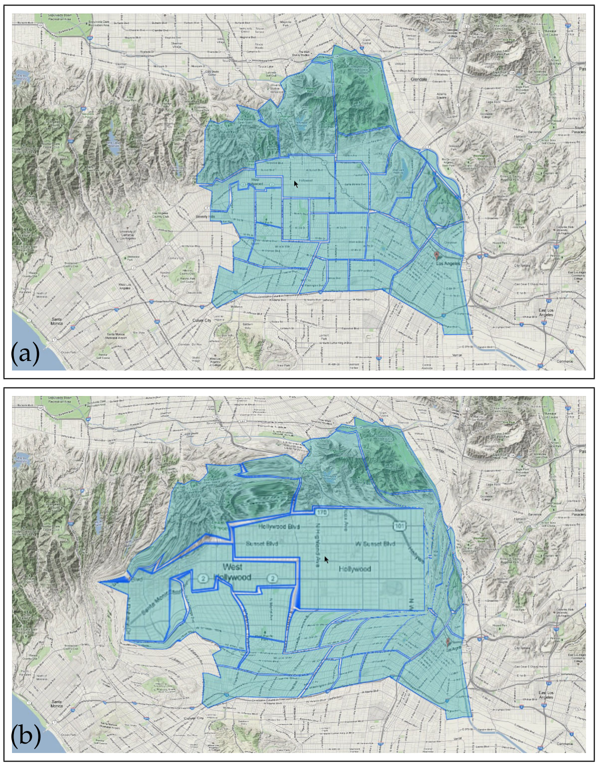

Fig. 3. (a) Map of Los Angeles neighborhoods (no lens applied). (b) AreaLens adapting to Hollywood and part of West Hollywood.