Design Gallery Browsers Based on 2D and 3D Graph Drawing (Demo)

Brad Andalman[1], Kathy Ryall[2], Wheeler Ruml[3],
Joe Marks[1], and Stuart Shieber[3]

[1] MERL — A Mitsubishi Electric Research Laboratory
201 Broadway
Cambridge, MA 02139, U.S.A.
E-mail: {andalman,marks}@merl.com
[2] Department of Computer Science
University of Virginia
Charlottesville, VA 22903, U.S.A.
E-mail: ryall@cs.virginia.edu
[3] Division of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138, U.S.A.
E-mail: {ruml,shieber}@eecs.harvard.edu

Abstract. Many problems in computer-aided design and graphics involve the process of setting and adjusting input parameters to obtain desirable output values. Exploring different parameter settings can be a difficult and tedious task in most such systems. In the Design Gallery™ (DG) approach, parameter setting is made easier by dividing the task more equitably between user and computer. DG interfaces present the user with the broadest selection, automatically generated and organized, of perceptually different designs that can be produced by varying a given set of input parameters. The DG approach has been applied to several difficult parameter-setting tasks from the field of computer graphics: light selection and placement for image rendering; opacity and color transfer-function specification for volume rendering; and motion control for articulated-figure and particle-system animation. The principal technical challenges posed by the DG approach are *dispersion* (finding a set of input-parameter vectors that optimally disperses the resulting output values) and *arrangement* (arranging the resulting designs for easy browsing by the user). We show how effective arrangement can be achieved with 2D and 3D graph drawing. While navigation is easier in the 2D interface, the 3D interface has proven to be surprisingly usable, and the 3D drawings sometimes provide insights that are not so obvious in the 2D drawings.

1 Introduction

Many problems in computer-aided design and graphics involve the process of setting and adjusting input parameters to obtain desirable output values. Exploring different parameter settings can be a difficult and tedious task in most such systems. Managing and organizing the exploration of input-parameter space is usually the responsibility of the user; the computer is used as a passive instrument, not unlike a brush or pencil. Design Galleries were introduced [4] as a general paradigm for designing computer graphics and animations in which the parameter-setting task is divided more equitably and more appropriately between user and computer. In the Design-Gallery (DG) approach, the computer generates a representative set of graphics or animations for perusal by the user; the representative set is dispersed uniformly in the space of possible graphics. Achieving uniform dispersion is a novel and challenging technical problem, which is discussed in detail in the original paper. Once satisfactory dispersion has been achieved, the other technical problem in the DG paradigm is arrangement: the automatically generated graphics or animations — typically there will be several hundred of them — must be presented in a logical and accessible way to facilitate browsing by the user.

One solution to the arrangement problem uses graph drawing.[1] If we map each generated image or animation to a graph node, and use a perceptual distance metric (see [4] for the details of several such metrics) to generate edge weights for each edge in a complete graph, a graph drawing that correlates the perceptual distances between nodes with distances in a 2D or 3D embedding should provide an intuitive visualization of the set of graphics or animations.

This idea is illustrated in Figure 1. The figure contains a 2D Design Gallery of 584 ray-traced images of the same geometric scene model, each illuminated by a single light source. The differences in the images result from different light parameters, e.g., ones relating to light type and position. These input parameters are dispersed automatically by the system to yield a diverse selection of light sources. A user might browse this Design Gallery to assemble a set of lights that combine to communicate a particular mood or to focus attention on certain scene elements. 2D graph drawing is used to arrange low-resolution thumbnails of the images in the center display area. A full-size image appears when a thumbnail image is selected; full-size images can be moved to the surrounding gallery for convenience. In the figure, the gallery has been populated with a representative set of images, and the association between thumbnail and gallery images is indicated by overlaid lines.[2] The user can pan and zoom the center display area to examine subsets of the images in more detail. In Figure 2, the user has concentrated attention on the leftmost images of the display in Figure 1.

In Figure 3, 3D graph drawing is used to arrange the same set of images. The images are placed on self-orienting *billboard* polygons that float in space, and among which the user can navigate. The polygons are represented in the Virtual Reality Modeling Language (VRML), so navigation can be done using

[1] We have also investigated another solution based on uniform-depth hierarchical arrangement, which requires solving a graph-partitioning problem to form the hierarchy [4].

[2] In the interactive system, the association between thumbnail and full-size images is indicated dynamically: placing the mouse over an image in the gallery highlights its associated thumbnail, and vice versa.

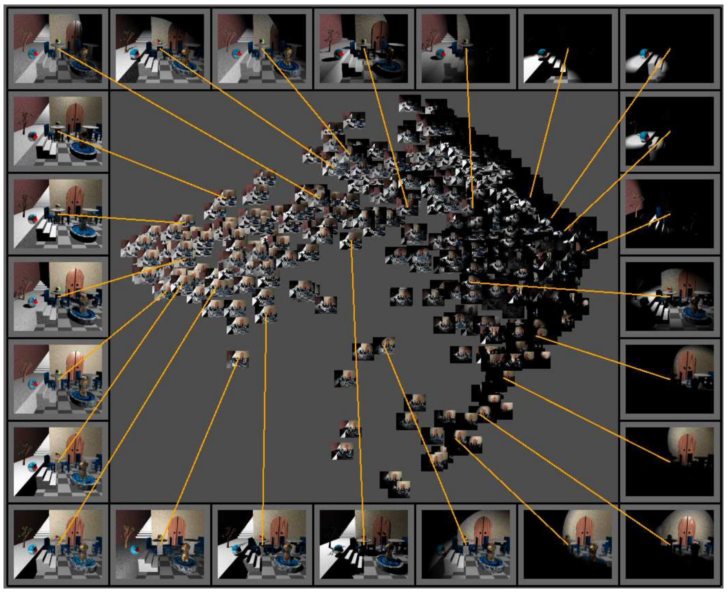

Fig. 1. A 2D Design Gallery for light selection and placement.

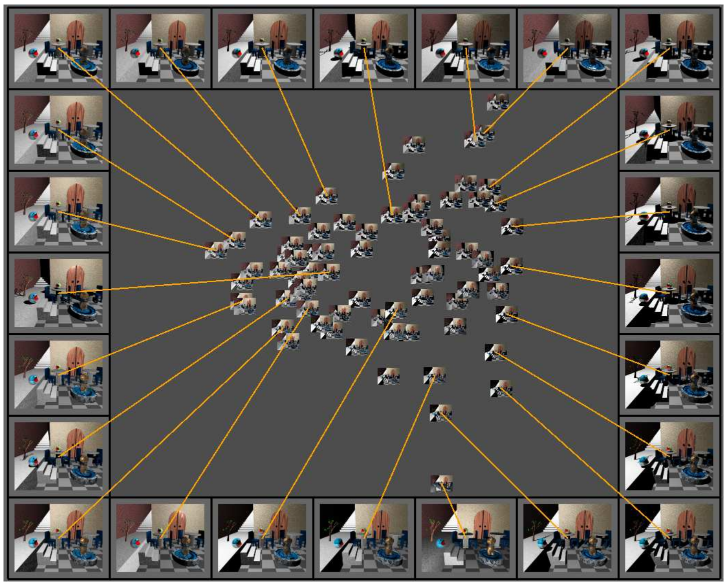

Fig. 2. The same Design Gallery after some panning and zooming.

any VRML browser. These browsers offer a variety of navigation modes, including *walk* (free movement subject to gravity), *fly* (free movement without gravity), and *examine* (free movement with additional capabilities for examining and manipulating individual objects) [1]. We have augmented the standard VRML-browsing functionality by providing a semi-transparent plane orthogonal to the user's viewing direction that can be used to focus attention on foreground images. Figure 4 concentrates on approximately the same images featured in Figure 2; this view was achieved by rotating the 3D view, moving forward, and using our semi-transparent "curtain" to hide background images. We also allow the user to tag certain images as "lighthouses": a flashing beacon is attached to images so designated, making them easier to find again later.

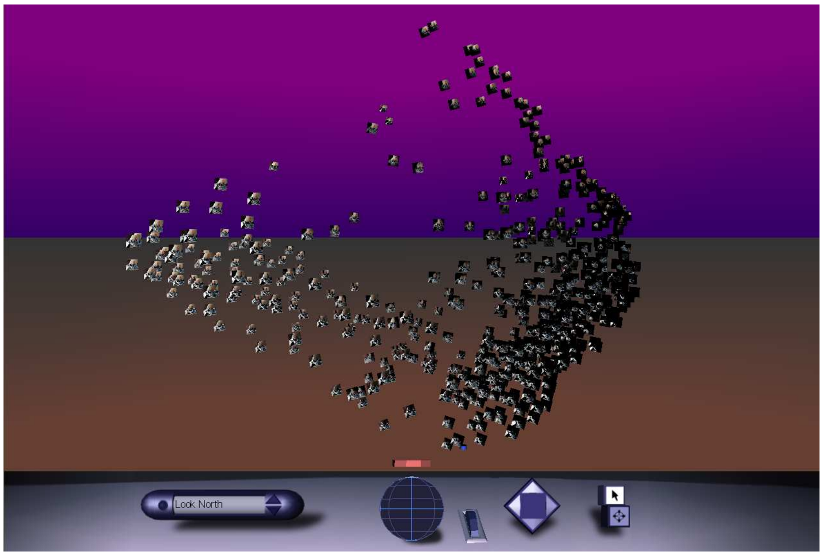

Fig. 3. A 3D Design Gallery for light selection and placement.

The 2D and 3D locations of the image nodes in the respective graph layouts were computed using Torgerson's classical multidimensional scaling (MDS) method [6].[3] From a matrix of perceptual distances between each pair of images, MDS computes the appropriate 2D or 3D layout coordinates for each image such that the distances in the drawing closely match those in the input matrix. Classical scaling is one of the simplest and fastest algorithms for MDS, but it is less general than iterative methods. When the input distances come from measure-

[3] Rubner et al. have independently investigated the use of MDS techniques for arranging a collection of images [5]. The use of MDS for graph drawing was pioneered by Kruskal and Seery [3].

Fig. 4. The same Design Gallery after some 3D navigation and manipulation.

ments in a high-dimensional Euclidean space (which is true for the applications discussed here, although it need not be the case in other DG applications), the algorithm can be viewed as an efficient technique for principal-component analysis [2].

2 Comparisons

We expected the 2D Design Gallery interface to be far superior to the 3D interface: panning and zooming in 2D is much easier than navigating in 3D, and perceiving graph structure at a glance seems to be easier in 2D. However, we found the 3D interface to be surprisingly useful and usable. In some instances, the 3D interface supplied insights that were not apparent in 2D. For example, Figures 5 and 6 show Design Galleries for 256 volume-rendered images of a computed-tomography (CT) data set representing a human pelvis. The differences in the images result from the use of different opacity transfer functions for the various issue types. The graph drawings in both figures appear essentially the same, modulo a reflection. However, a slight rotation of the 3D graph drawing (see Figure 7) reveals some additional structure: images that depict predominantly bone and muscle lie approximately in a plane (left), while images that depict varying amounts of fatty tissue lie off this plane (right). The additional structure evident in 3D can make it easier to find and locate images with specific characteristics.

Several factors make a 3D interface more successful in our application than one might expect:

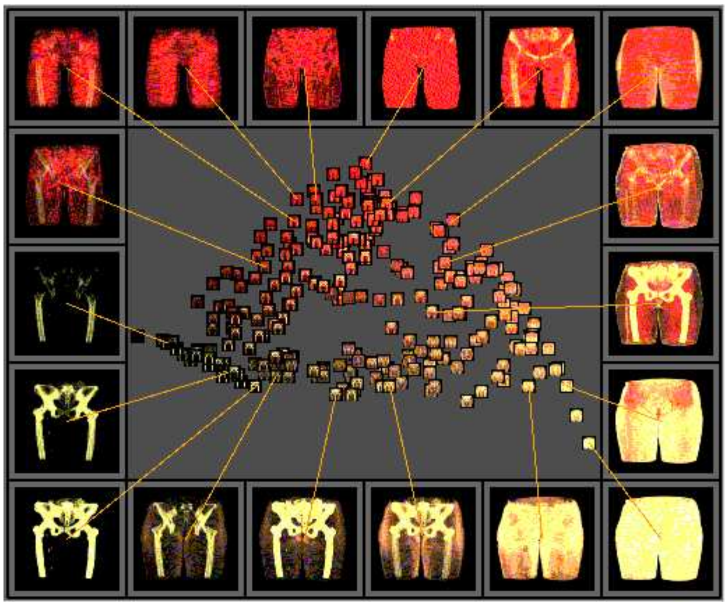

Fig. 5. A 2D Design Gallery for volume-rendered images.

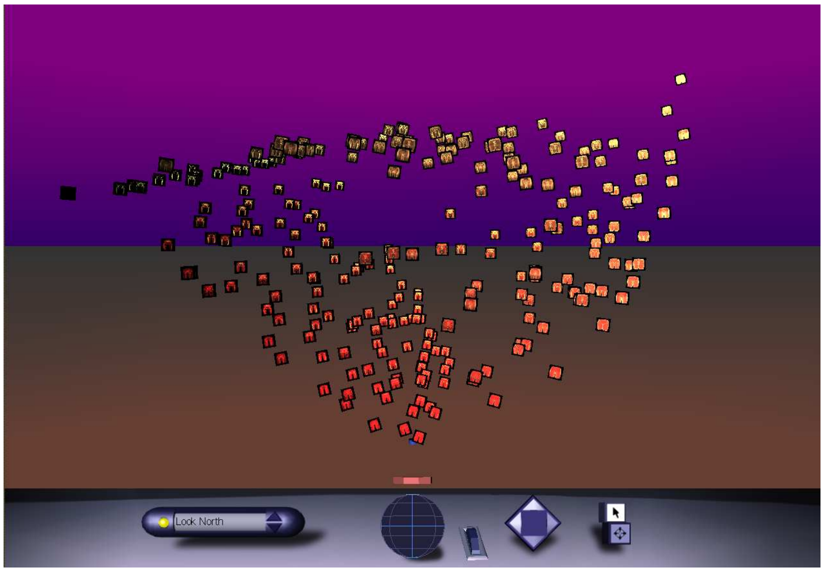

Fig. 6. A 3D Design Gallery for the same volume-rendered images.

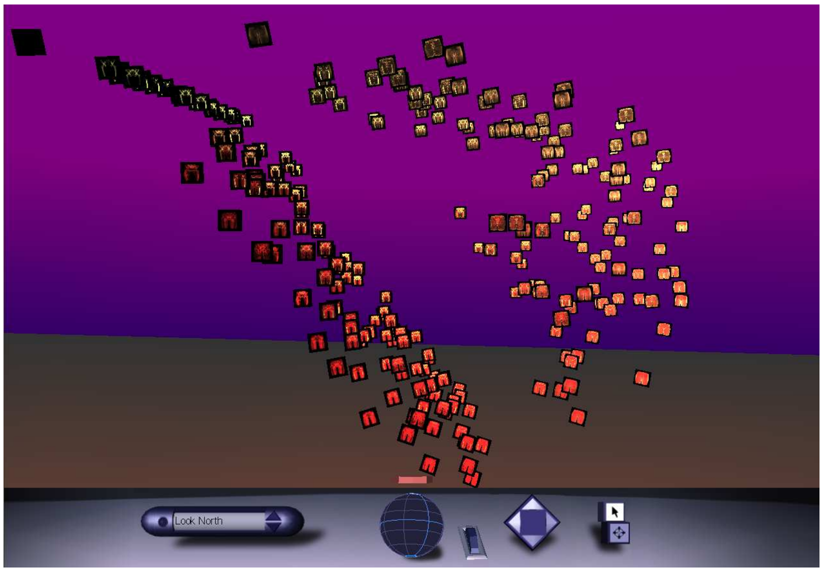

Fig. 7. The 3D Design Gallery after a small rotation.

– Edges and text labels are not needed to convey any useful information; their absence makes the drawings much clearer.
– Because each node is an easily perceived image, and because similar images cluster together, it is easier for the user to discern and identify regions of the graph, which aids navigation.
– Because we are trying to visualize distances in a high-dimensional space (the perceptual similarities for the volume-rendered images were expressed in 24 dimensions, for example), having three dimensions instead of two for our graph drawings means that less information is lost when projecting from the higher-dimensional perceptual-similarity space onto the lower-dimensional drawing space. Even though the first two embedding dimensions captured around 40% of the variance in the CT images, compared with about 15% for the third dimension, the extra information is more useful in combination with the other dimensions than one would expect from considering the variance alone. We speculate that this advantage of the third dimension may grow in importance as the number and variety of graph nodes increase.
– Because each graph node is a single texture-mapped rectangle, we can take advantage of graphics workstations that have been optimized specifically to display texture-mapped polygons efficiently.[4] This permits smooth motion through 3D drawings with hundreds of graph nodes, which is crucial for the

[4] Most of our experimentation was done on a high-end Silicon Graphics workstation with multiple processors and raster managers, and lots of memory.

usability of the interface.

- The VRML interaction metaphors suffice for most of our needs (this might not be true for other graph-drawing applications, e.g., ones in which graph manipulation is required), so we can take advantage of mature, standard methods for navigating 3D virtual environments.

3 Conclusion

Browsing automatically generated Design Galleries is a new application for graph drawing. We have developed browsing interfaces based on both 2D and 3D graph drawing. The 2D interface has proven to be usable and useful; counter to our initial expectations, so has the 3D interface. However, we have identified several factors peculiar to our application that makes 3D graph drawing easier and more appropriate. While the future of 3D graph drawing as a general visualization paradigm may be problematic, we speculate that it may prove useful for niche applications such as ours.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Numbers IRI-9350192 and IRI-9618848 and by a grant from MERL — A Mitsubishi Electric Research Laboratory to SMS. This work was done while KR was at the Division of Engineering and Applied Sciences, Harvard University.

References

1. R. Carey and G. Bell. *The Annotated VRML 2.0 Reference Manual.* Addison Wesley Developers Press, Reading, Massachusetts, 1997.
2. J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53:325–338, 1966.
3. J. B. Kruskal and J. B. Seery. Designing network diagrams. In *Proceedings of the First General Conference on Social Graphics*, pages 22–50, Leesburg, Virginia, October 1978.
4. J. Marks, B. Andalman, P. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: A general approach to setting parameters for computer graphics and animation. In *SIGGRAPH 97 Conf. Proc.*, pages 389–400, Los Angeles, California, August 1997.
5. Y. Rubner, L. J. Guibas, and C. Tomasi. The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In *Proc. of the DARPA Image Understanding Workshop*, New Orleans, Louisiana, May 1997.
6. W. S. Torgerson. *Theory and Methods of Scaling.* Wiley, New York, 1958. See especially pages 254–259.