Semantic Depth of Field

Robert Kosara Silvia Miksch
Vienna University of Technology
http://www.asgaard.tuwien.ac.at/
{rkosara,silvia}@asgaard.tuwien.ac.at

Helwig Hauser
VRVis Research Center, Austria
http://www.VRVis.at/vis/
hauser@VRVis.at

Abstract

We present a new technique called Semantic Depth of Field *(SDOF) as an alternative approach to focus-and-context displays of information. We utilize a well-known method from photography and cinematography (depth-of-field effect) for information visualization, which is to blur different parts of the depicted scene in dependence of their relevance. Independent of their spatial locations, objects of interest are depicted sharply in SDOF, whereas the context of the visualization is blurred. In this paper, we present a flexible model of SDOF which can be easily adopted to various types of applications. We discuss pros and cons of the new technique, give examples of application, and describe a fast prototype implementation of SDOF.*

Keywords: Depth of Field, Focus and Context, Information Visualization

1. Introduction

Whenever large amounts of data are to be investigated, visualization potentially becomes a useful solution to provide insight into user data. Especially for exploration and analysis of very large data-sets, visualization not only needs to provide an easy-to-read visual metaphor, but also should enable the user to efficiently navigate the display, allowing for flexible investigation of arbitrary details.

Focus and Context (F+C) techniques enable the user to investigate specific details of the data while at the same time also providing an overview over the embedding of the data under investigation within the entire dataset. But F+C encompasses a number of very different techniques that achieve similar goals in very different ways.

1.1. Different Kinds of Focus and Context

The most prominent group of F+C methods are *distortion-oriented* [12] or *spatial methods*. The geometry of the display is distorted to allow a magnification of interesting information without losing the (less magnified) context. It is thus possible to navigate information spaces that are far too large to be displayed on a screen. Examples are fish-eye views [5, 20], hyperbolic trees [9, 10, 18], stretchable rubber sheets [21], etc. Distortion-oriented techniques are usually used in an explicit way, by actively bringing the interesting objects into focus, e.g. by clicking on objects or dragging them around.

For smaller numbers of objects that have a lot of data associated with them, a visualization method is useful that shows just a limited number of data dimensions, and allows the user to select which of the objects are to be shown in more detail – we call these *dimensional methods*. The context in this case are not only the other objects, but also the remaining data dimensions. This type of method also shows more detail, but in terms of data dimensions, not screen size. Examples are magic lenses [22] and tool glasses [2], where the user moves a window over the display, the objects inside which are displayed in more detail.

The third type of focus and context allows the user to select objects in terms of their features, not their spatial relations; usually by assigning a certain visual cue to them – we therefore call these methods *cue methods*. They make it possible to query the data for information which is not immediately visible in the initial visualization, while keeping the original layout, and thus not destroying the user's mental map [17]. An example for such a system is a Geographic Information System (GIS) that makes it possible to display crime data, certain cities, or hospitals [14]. This data is displayed in the same context as before, but the relevant parts of the display have a higher color saturation and opacity than the rest. This leads the viewer's attention to the relevant objects easily without removing context information. In contrast to distortion-oriented techniques and magic lenses, with this type of method, the user first selects the criteria, and then is shown all the objects fulfilling them.

The technique presented in this paper is of the third type, but we use a different visual cue for discriminating focus and context.

Figure 1. A lantern with a bridge as context.

1.2. The Uses of Blur and Depth of Field

Blur is usually considered to be an imperfection: it makes features harder to recognize and can hide information completely. But the difference between sharp and blurred parts of an image is a very effective means of guiding the viewer's attention. In photography, the depth-of-field (DOF) effect leads to some parts of the image being depicted sharply, while others are blurred [1]. The viewer automatically looks at the sharp parts, while the blurred parts provide non-disturbing context for the objects of interest (see Fig. 1 for an example). The same effect is also used in cinematography [8], where focus changes can guide the audience's attention from one character to another, from a character to an object he or she just noticed, etc.

Because the human eye (like every lens system) also has limited DOF [6], an important characteristic of human vision is that whenever we get interested in a specific part of our environment, we 1) bring the the object of interest into the center of our eye (where the area of most acute vision, the *fovea centralis*, is located), and 2) focus on that object. From the above applications of DOF (photography and cinematography), we know that this process is easily inverted: If we display sharp objects in a blurred context, the viewer's attention is automatically guided to the sharp objects. This also gives us reason to believe that DOF is perceived preattentively, i.e. within 50ms of exposure to the stimulus, and without serial search [23]. This means, it very efficiently makes use of the bandwidth of the human visual system to convey a lot of information in very little time.

We have developed an F+C technique we call *Semantic Depth of Field* (SDOF) for information visualization, which renders objects sharp of blurred, depending on their current relevance. It thus makes use of the phenomena described above to guide the viewer's attention.

2 Related Work

There have been surprisingly few attempts to use DOF or blur in visualization at all; the ones relevant to this work are shortly summarized here.

In a system for the display of time-dependent cardiovascular data [25], a stereoscopic 3D display is included that is controlled by the viewer's eyes. Like a microscope, only one thin slice through the data appears sharp, all others are blurred and therefore almost invisible. Eye tracking equipment determines what the user is looking at, and that point is brought into focus. This makes it possible to concentrate on one detail without the surrounding structures confusing the viewer. Later work [26] describes "non-linear depth cues", which means displaying structures that currently are of interest (like single organs) in focus, and other objects out of focus, not based on their distance from the camera, but on their importance.

The Macroscope [13] is a system for displaying several zoom levels of information in the same display space. For this purpose, the images on all levels are drawn over each other, with the more detailed ones drawn "in front", i.e., drawn over the less magnified layers. The layers' transparency can be changed so that the background (context) can be more or less visible. The less detailed layers are blurred so as to not distract the viewer, but serve as context.

The most interesting existing approach for this work is a display of geographical information [3]. In this system, up to 26 layers of information can be displayed at the same time. Each layer has an interest level associated with it that the user can change. The interest level is a combination of blur and transparency, making less interesting layers more blurred and more transparent at the same time. This work does not seem to have been followed up on recently.

In this paper, we describe a general model of SDOF, i.e., of selectively using sharpness vs. blur to emphasize/deemphasize certain parts of the data. We clearly embed SDOF within the scope of information visualization and computer graphics. In addition to the above examples, we provide a flexible solution which easily is adopted to various kinds of applications, as demonstrated later on.

3. Semantic Depth of Field (SDOF)

SDOF allows the user to select relevant parts of a visualization that are then pointed out by deemphasizing all the rest through blur. The discrimination between relevant and irrelevant objects can be binary (an object is either relevant

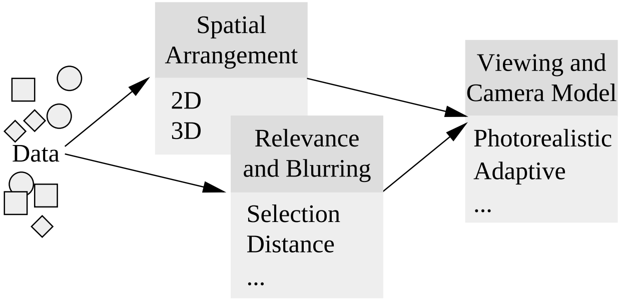


Figure 2. SDOF Building Blocks.

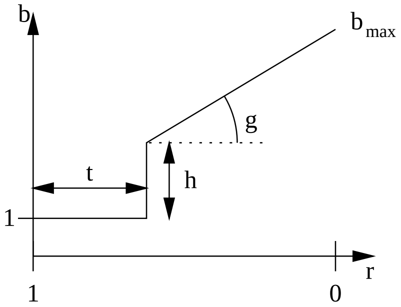


Figure 3. The Blur function.

or irrelevant) or continuous (an object can have a relevance value between the two extremes).

Different relevance metrics for objects have to be offered by the application, that have to deal with the specific information and tasks the application is made for. Examples for binary relevance measures are the set of chessmen that threaten a specific piece in a chess tutoring system (see Fig. 5c and the accompanying video), the layer containing roads in a GIS application (Fig. 5d), or all incidents related to high blood glucose in a graphical patient record. Continuous functions could express the age of files in a file system viewer (Fig. 5a), the recent performance of stocks in a stock market browser, or the distance of cities from a specified city in terms of flight hours.

The building blocks of SDOF are discussed in the following subsections, and are summarized in Fig. 2 and Tab. 1.

3.1. Spatial Arrangement

In information visualization, usually some kind of layout algorithm is used to arrange objects in the visualization space (typically 2D or 3D). The special challenge of information visualization is the fact that data often does not have any inherent structure that naturally translates to a layout. Mapping functions are a very important part of visualization because they determine how well the user can build a mental map that he or she can use to understand and navigate the visualization. Changing the layout often means having to learn a new layout, and thus losing one's ability to navigate easily.

In our model, the spatial mapping function is called $\mathbf{place}$; it translates from the original data space to an intermediate visualization space (2D or 3D).

3.2. Relevance and Blurring

Independently of the spatial arrangement, the blur level of each object is determined. This is done in two steps: First, each object is assigned a relevance value r by the relevance function rel. The value of r is in the interval $[0;1]$, where 1 means the object is maximally relevant, and 0 means the object is completely irrelevant. This relevance value is translated into a blur value b through the blur function blur.

The relevance function is application-specific and thus can be very different between applications (see Sect. 5.2 for examples). The blur function can theoretically also take on any shape, but we have found the function depicted in Fig. 3 to be sufficient for our current uses. The user can specify the threshold value t, the step height h, and the maximum blur diameter $b_{\max}$. The gradient g is then calculated by the application.

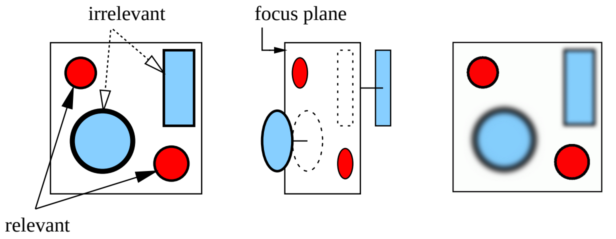
3.3. Viewing and Camera Models

In order to provide a consistent model, and to embed the idea of SDOF in existing work in computer graphics, we discuss camera models for generating images with SDOF. Depending on whether the visualization space is two- or three-dimensional, different camera models can be used to finally achieve the SDOF effect. The camera provides two functions: $\mathbf{camera}$ projects data values from an intermediate space (where the information was laid out by the $\mathbf{place}$ function) to screen space; and dof, which calculates the blur level of each data item depending on its z coordinate and the z_{focus} value the camera is currently focused at.

In the following, we describe two camera models: a regular photo-realistic camera ($\mathbf{camera}_{\mathrm{P}}$) can be used in the 2D case; for 3D, we present the *adaptive camera* ($\mathbf{camera}_{\mathrm{A}}$).

3.3.1 2D SDOF and the Photo-realistic Camera

In the 2D case, objects get a third coordinate in addition to their x and y values. This additional z value depends on the intended blur diameter b of the object: If the camera is focused at z_{focus}, an object with intended blur b has to be moved to a distance of z from the lens of the camera (see Fig. 4):



2D SDOF scene 3D intermediate scene rendered image

Figure 4. The photo-realistic camera and 2D SDOF.

$$b = \mathsf{dof}_{\mathrm{P}}(z, z_{\mathrm{focus}}) = \left| D \frac{f(z_{\mathrm{focus}} - z)}{z_{\mathrm{focus}}(z - f)} \right| \quad (1)$$

$$z = \mathsf{dof}_{\mathrm{P}}^{-1}(b, z_{\mathrm{focus}}) = \frac{D + b}{\frac{D}{z_{\mathrm{focus}}} + \frac{b}{f}} \quad (2)$$

where D is the effective lens diameter as defined in the thin lens model [11], and f is the focal length of the lens in use.

The above equations apply to camera models such as distribution ray tracing [4], linear post-filtering [19], etc.

If the camera uses perspective projection, objects also have to be scaled (and possibly moved) to compensate for depth effects that are not desired in this case.

3.3.2 3D SDOF and the Adaptive Camera

In the 3D case, of course, it is not possible to directly map blur factors to depth values, because the spatial arrangement of data items already contains a third dimension. However, using a simple extension of the photo-realistic camera, it is possible to also handle the 3D case.

The adaptive camera is a modification of a photo-realistic camera that can change its focus for every object point to be rendered. This is easily possible with object-order rendering, but can also be achieved when rendering in image order. In contrast to the photo-realistic camera, the adaptive camera can render SDOF in 2D and 3D scenes. The photo-realistic camera is, in fact, a special case of the adaptive camera (which simply stays focused at the same distance for the whole image).

Function $\mathsf{dof}_{\mathrm{A}}$ is defined like $\mathsf{dof}_{\mathrm{P}}$ in Eq. 1. Different to the 2D case, now the inversion of $\mathsf{dof}_{\mathrm{A}}$ must be resolved for z_{focus}-values:

$$b = \mathsf{dof}_{\mathrm{A}}(z, z_{\mathrm{focus}}) = \mathsf{dof}_{\mathrm{P}}(z, z_{\mathrm{focus}}) \quad (3)$$

$$z_{\mathrm{focus}} = \mathsf{dof}_{\mathrm{A}}^{-1}(b, z) = \frac{D}{\frac{D+b}{z} - \frac{b}{f}} \quad (4)$$

An example for an adaptive camera is splatting [7, 24], which is a volume rendering technique, but which also can be used for information visualization. By changing the size of the splat kernel depending on the b value of a data point, SDOF can be implemented easily.

Another possibility is to use pre-blurred billboards (Sect. 6 and [16]). Objects are rendered into memory, the images are then blurred and mapped onto polygons in the scene.

4. Properties and Applicability

This section discusses some high-level properties of SDOF, how it can be principally applied, and what challenges it brings with it.

4.1. Properties

SDOF, being yet another F+C highlighting technique, has the following properties that make it an addition to the current toolbox:

- SDOF does not distort geometry. It is therefore usable when sizes (of objects or parts of objects (glyphs)) and positions are used as visual parameters. We also believe that it is easier to recognize blurred icons than distorted ones.
- SDOF does not alter colors. If color is used to convey meaning, or the visualization is to be used by color-blind people, SDOF can be used instead of color highlighting. This also means that SDOF is independent of color, and can therefore be used when only gray-scale is available (e.g., printed images).
- SDOF changes the irrelevant objects, rather than the relevant ones. It is therefore useful whenever the relevant objects contain a lot of information whose perception might be impeded by changes.

4.2. Applicability

SDOF requires concrete queries to the data (which can be simple, but have to be formulated nonetheless), and is therefore useful for analyzing and presenting data.

SDOF can serve as an additional aid to guide the viewer's attention, together with brighter colors, etc., or as a completely separate dimension of data display. Because blur is very naturally associated with importance (even more than color), we do not believe that it is suitable for true multi-variate data visualization. It can, however, add another dimension for a short time, when the displayed data is to be queried.

Blurring needs space, so when a lot of very small objects are depicted, it is only of little use. The application can deal with this problem by drawing the objects in such an order that sharp objects are drawn on top of blurred ones. But this

$$\begin{tikzcd} \mathbf{data}[i] \arrow[rr, "\mathbf{place}_{\mathrm{2D}}"] \arrow[r, "\mathsf{rel}"'] & r \arrow[r, "\mathsf{blur}"'] & \begin{pmatrix} \hat{x} \\ \hat{y} \\ b = \mathsf{blur}(r) \end{pmatrix} \arrow[r, "\mathbf{view}_{\mathrm{2D}}"] \arrow[r, "\mathsf{dof}_{\mathrm{P}}^{-1}"'] & \begin{pmatrix} x \\ y \\ z = \mathsf{dof}_{\mathrm{P}}^{-1}(b) \end{pmatrix} \arrow[r, "\mathbf{camera}_{\mathrm{P}}"] & \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \end{tikzcd}$$

$$\begin{tikzcd} \mathbf{data}[i] \arrow[rr, "\mathbf{place}_{\mathrm{3D}}"] \arrow[r, "\mathsf{rel}"'] & r \arrow[r, "\mathsf{blur}"'] & \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ b = \mathsf{blur}(r) \end{pmatrix} \arrow[r, "\mathbf{view}_{\mathrm{3D}}"] \arrow[r, "\mathsf{dof}_{\mathrm{A}}^{-1}"'] & \begin{pmatrix} x \\ y \\ z \\ z_{\mathrm{focus}} = \mathsf{dof}_{\mathrm{A}}^{-1}(b) \end{pmatrix} \arrow[r, "\mathbf{camera}_{\mathrm{A}}"] & \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \end{tikzcd}$$

Table 1. All steps necessary for visualizing data values $\mathbf{data}[i]$ with 2D (top) and 3D SDOF (bottom).

can introduce artifacts, where parts of the display appear sharp only because of the contrast between sharp objects and the background.

4.3. Challenges

SDOF images depend on the output device (similar to tone mapping [15], for example). The reason for this is that blur is not an absolute measure, but depends on the viewing angle that the image covers – this is also the reason why small images look sharper than larger ones: the circles of confusion are not visible in the smaller version, or at least to a smaller extent. We use a calibration step at program startup to account for this problem (see Sect. 5.1).

Images that contain SDOF effects are also problematic when lossy compression is used (like MPEG, JPEG, etc.). In this case, artifacts can be introduced that create a high contrast in a blurred area, and thus distracting the user. But SDOF is most useful in interactive applications, so this problem should play no big role in practice.

5. Parameterization

Parameterization of SDOF consists of two parts: Adaptation to current viewing parameters and user interaction to change the relevance mapping.

5.1. Output Adaptation

We ask the user to select two blur levels on program startup: a) the minimal blur level that can be easily distinguished from a sharp depiction – this value translates to the step height h in Fig. 3; b) the minimal difference in blur that can be distinguished – this value can be used to calculate g, if the smallest difference between any two r values is given. Because this is generally not the case, the blur function is adapted for every image after examining the r values of all objects. These values can vary with the use of the generated image (printing out, projecting onto a wall, etc.), the use of different screens, etc.

5.2. User Interaction

Interaction is a key part of SDOF. Blurred objects are unnatural, and it is therefore important for the user to be able to change the relevance mapping and blur function quickly, and to return to a depiction that shows all objects in focus.

Depending on the application, there are different usage patterns. In many applications, it is useful to be able to point at an object and say “Show me all objects that are older than this”, “Show me all chessmen that cover this piece” (Fig. 5e), or “Show me the cities weighed by their railway distance from this city”.

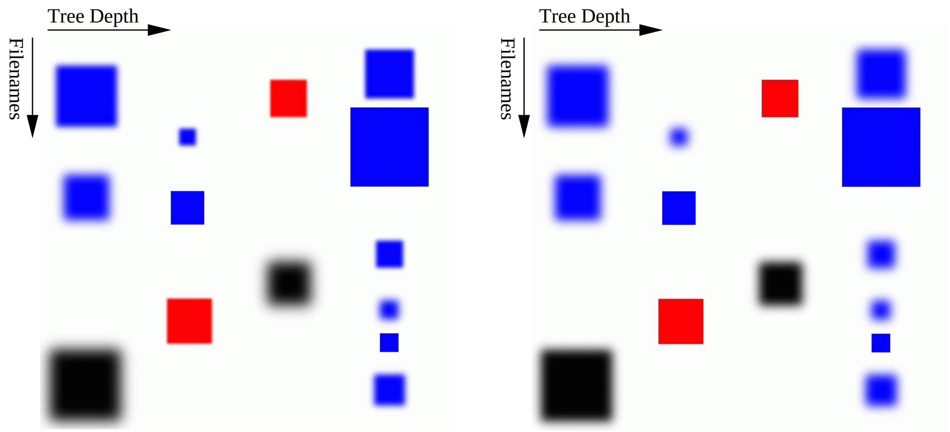
Another way is to select values independently of objects: “Show me all threatened chess pieces of my color”, “Show me all files that were changed today” (Fig. 5b), or “Show me all current patients weighed by their need for drug xyz”.

An additional feature we believe is useful is the *auto focus*. After a pre-specified time, it makes all objects appear sharp again, thus making examination of all objects easier (this function can be turned off).

Transitions between different displays are always animated to enable the user to follow the change and immediately see which objects are relevant in the new display. This is another reason for separating r and b (see section 3.2): The animation is done between the old and the new b values, rather than the r values. This is because the blur function can contain discontinuities that can lead to jumps between blur levels of objects, and are therefore undesirable.

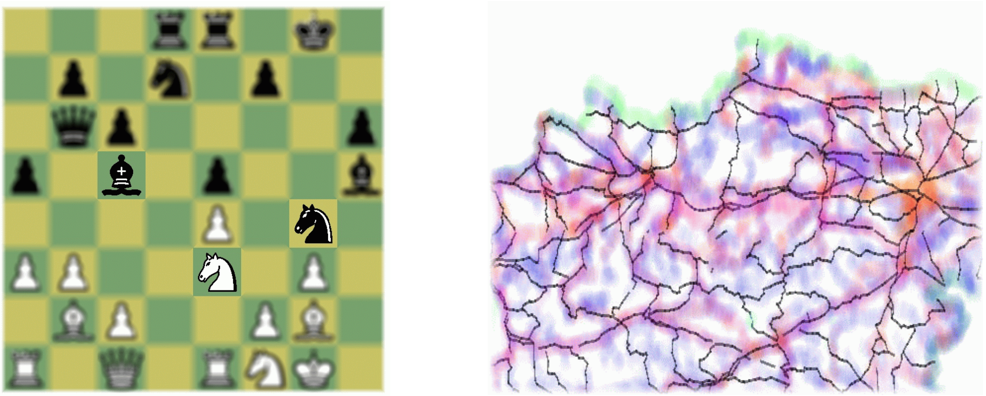
6. Implementation

A method in information visualization should not only be visually effective, but also fast, so that it can be used in-



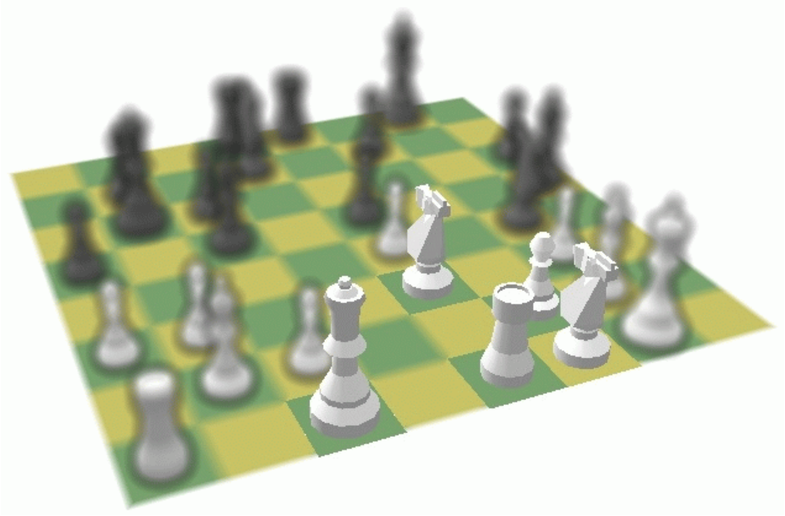
a) A file browser showing the age of files through blur. Continuous relevance function.

b) A file browser showing today's files sharply, older ones blurred. Binary relevance function.



c) A chess tutoring system showing the chessmen that threaten the knight on e3.

d) A Geographic Information System (GIS) showing the roads layer in focus.



e) A chess tutoring system showing the chessmen that cover the knight on e3.

Figure 5. SDOF in action. See Sect. 5.2 and 3 for details.

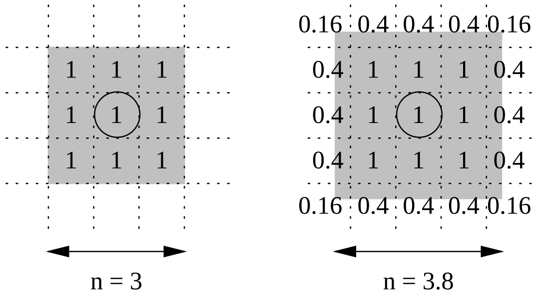


Figure 6. The Box Filter (left), and the generalized box filter for arbitrary sizes (right).

teractively. Blurring used to be a very slow operation because it involves a sum of three color components of many neighboring pixels for every single pixel in the image, and is still not supported by hardware except in high-end graphics workstations. We have implemented SDOF using texture mapping hardware, which makes it fast on currently available consumer PCs. The described method is an implementation of the adaptive camera model (see Sect. 3.3.2).

Blur can be understood as a convolution operation of the image with a blur kernel. In photography, this blur kernel ideally is round, but usually is a regular polygon with six to eight vertices, due to the shape of the aperture.

The more common type of blur kernel in computer science is the box filter (Fig. 6, left). It has the big advantage of being separable [16], which reduces its computational cost from $O(n^2)$ to $O(2n)$, where n is the filter size. It can also be generalized quite easily to arbitrary sizes (Fig. 6, right) other than just odd integers. This implementation directly uses b as its filter size n.

Using graphics hardware is different from a software implementation of a filter in that it does not sum up the color values of surrounding pixels for every single pixel. Rather, it adds the whole image to the frame buffer in one step by drawing it onto a textured polygon (this is done by blending with a special blend function). When the image is drawn in different positions (with one pixel distance between the images), several image pixels are added to the same frame buffer pixel. Because of the limited accuracy of the frame buffer (typically eight bits per color component), this can only be done for small values of n (we have found $n \leq 4$ to yield acceptable images).

For larger blur diameters, we use a two-step approach. First, we sum up four images into the frame buffer, with their color values scaled so that the sum uses the entire eight bits. We then transfer this image to texture memory (this is a fast operation) and use this auxiliary sum as the operand for further calculations. The auxiliary sum already contains the information from four addition steps, so when summing them up further, only one quarter of the addition steps is needed. Because all the values in the box filter (except for the border, which is treated separately) are equal, all auxiliary sums are equal – they are only displaced. This means, that the auxiliary sum only needs to be computed once (as well as another auxiliary for the borders). Summing up auxiliary sums is therefore not only more accurate, it is also faster.

For blur diameters larger than 20 pixels, we first scale the image to one quarter of its size, then blur with half the diameter, and then scale it back ("quarter method").

Using the described method, it is possible to run applications – like the ones shown in the images and the accompanying video – at interactive frame rates (at leat 5 frames per second) on cheap consumer graphics hardware. This number is likely to increase with some further optimizations as well as the use of multi-texturing (which is supported by more and more consumer graphics cards).

7. Evaluation

To show that SDOF is actually perceived preattentively, and to demonstrate its usefulness in applications, we are currently performing a user study with 16 participants. We want to find out a) if SDOF is, indeed, perceived preattentively, which includes the detection and localisation of targets, as well as the estimation of the number of targets on screen (as a number relative to all objects in the image) in the presence of distractors; b) how many blur levels people can distinguish, and how blur is perceived (e.g., linear, exponential, etc.); c) how blur compares to other visual cues which are known to be perceived preattentively (such as color and orientation); and d) how well SDOF can be used to solve simple problems with simple applications (where the emphasis is on the use of SDOF). This study is still in progress at the time of this writing, but we will publish the results as soon as they are available.

8. Conclusions and Future Work

We have presented an extension to the well-known depth-of-field effect that allows objects to be blurred depending on their relevance rather than on their distance from the camera. This technique makes it possible to point the user to relevant objects, without distorting the geometry and other features of the visualization.

Because of the similarity to the familiar depth-of-field effect, and the fact that DOF is an intrinsic part of the human eye, we believe that it is a quite natural metaphor for visualization and can be used quite effortlessly by most users.

SDOF can be used when analyzing and presenting data, and also seems to be effective as a tool for pointing information out in tutoring systems.

We expect to learn a lot about SDOF's properties during our user study, and will use this information to define criteria when and how SDOF can be best used.

As one of the next steps, we want to investigate the applicability of SDOF to other areas of scientific visualization, like volume and flow visualization.

We also want to find out how SDOF can be applied to human computer interaction, to enable the user to grasp important information faster, and to be alerted to important changes without being distracted too much.

9. Acknowledgments

We would like to thank Markus Hadwiger for his help in coming up with a fast method for rendering SDOF images. This work is part of the Asgaard Project, which is supported by *Fonds zur Förderung der wissenschaftlichen Forschung* (Austrian Science Fund), grant P12797-INF. Parts of this work have been carried out in the scope of the basic research on visualization at the VRVis Research Center (`http://www.VRVis.at/vis/`) in Vienna, Austria, which is funded by an Austrian governmental research program called Kplus.

References

[1] A. Adams. *The Camera*. Little Brown & Company, 1991.

[2] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: The see-through interface. *Computer Graphics (Proceedings SIGGRAPH'93)*, 27(Annual Conference Series):73–80, 1993.

[3] G. Colby and L. Scholl. Transparency and blur as selective cues for complex visual information. In *SPIE Vol. 1460, Image Handling and Reproduction Systems Integration*, pages 114–125, 1991.

[4] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. *Computer Graphics (Proceedings SIGGRAPH'84)*, 18(3):137–145, July 1984.

[5] G. W. Furnas. Generalized fisheye views. In M. M. Mantei and P. Orbeton, editors, *Proceedings of the ACM Conference on Human Factors in Computer Systems*, SIGCHI Bulletin, pages 16–23, New York, U.S.A., 1986. Association for Computer Machinery.

[6] E. B. Goldstein. *Sensation and Perception*. Brooks/Cole Publishing Company, 5th edition, June 1998.

[7] J. Huang, K. Mueller, N. Shareef, and R. Crawfis. Fastsplats: Optimized splatting on rectilinear grids. In *Proc. Visualization 2000*, Salt Lake City, USA, Oct. 2000. IEEE.

[8] S. D. Katz. *Film directing shot by shot: Visualizing from concept to screen*. Focal Press, 1991.

[9] M. Kreuseler, N. López, and H. Schumann. A scalable framework for information visualization. In *Proc. Information Vizualization*, Salt Lake City, USA, Oct. 2000. IEEE.

[10] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings CHI'95*. ACM, 1995.

[11] H.-C. Lee. Review of image-blur models in a photographic system using principles of optics. *Optical Engineering*, 29(5):405–421, May 1990.

[12] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. on Computer-Human Interaction*, 1(2):126–160, June 1994.

[13] H. Lieberman. A multi-scale, multi-layer, translucent virtual space. In *IEEE International Conference on Information Visualization*, London, Sept. 1997. IEEE.

[14] I. Lokuge and S. Ishizaki. Geospace: An interactive visualization system for exploring complex information spaces. In *CHI'95 Proceedings*, 1995.

[15] K. Matkovic, L. Neumann, and W. Purgathofer. A survey of tone mapping techniques. In *Proceedings of the Thirteenth Spring Conference on Computer Graphics*, pages 163–170, Budimerce, Slovakia, 1997. Comenius University.

[16] T. McReynolds and D. Blythe. Advanced graphics programming techniques using OpenGL. SIGGRAPH 2000 Course 32, Course Notes, 2000.

[17] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages and Computing*, 6(2):183–210, June 1995.

[18] T. Munzner. Drawing large graphs with H3Viewer and Site Manager. In *Proceedings of Graph Drawing'98*, number 1547 in Lecture Notes in Computer Science, pages 384–393. Springer Verlag, Aug. 1998.

[19] M. Potmesil and I. Chakravarty. A lens and aperture camera model for synthetic image generation. *Computer Graphics (Proceedings SIGGRAPH'81)*, 15(3):297–305, Aug. 1981.

[20] M. Sarkar and M. H. Brown. Graphical fisheye views. *Communications of the ACM*, 37(12):73–83, Dec. 1994.

[21] M. Sarkar, S. S. Snibbe, O. J. Tversky, and S. P. Reiss. Stretching the rubber sheet: A metaphor for visualizing large layouts on small screens. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Visualizing Information, pages 81–91, 1993.

[22] M. C. Stone, K. Fishkin, and E. A. Bier. The movable filter as a user interface tool. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, volume 1 of *Information Visualization*, pages 306–312, 1994.

[23] A. Treisman. Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing*, 31:156–177, 1985.

[24] L. Westover. Footprint evaluation for volume rendering. *Computer Graphics (Proceedings SIGGRAPH'90)*, 24(4):367–376, Aug. 1990.

[25] S. E. Wixson. Four-dimensional processing tools for cardiovascular data. *IEEE Computer Graphics and Applications*, 3(5):53–59, Aug. 1983.

[26] S. E. Wixson. The display of 3d MRI data with non-linear focal depth cues. In *Computers in Cardiology*, pages 379–380. IEEE, Sept. 1990.