Focus+Context Taken Literally

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

Pointing out relevant information to a user is one application of focus+context techniques in information visualization. We present a method for doing this which uses selective blur to direct the user's attention. This method is based on the depth of field (DOF) effect used in photography and cinematography, and is therefore both familiar to users and perceptually effective. Because this method blurs objects based on their relevance rather than their distance, we call it Semantic Depth of Field (SDOF).

We also present four example applications that use SDOF to show its usefulness in practice, and also provide details of a fast implementation that makes it possible to use blur in interactive applications. A short report on the results of a user study we performed is also given.

Keywords: Information Visualization, Focus and Context, Blur, Preattentive Processing

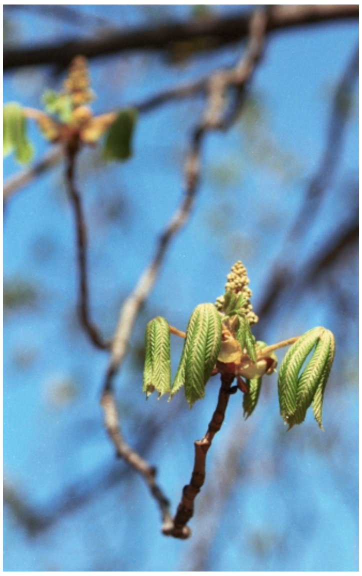


Figure 1. An example of DOF in photography.

1 Introduction

A common feature of applications in information visualization – and also other areas – is to direct the user's attention to certain objects. The reason can be to alert the user to a problem or to show the matching objects in response to a query. The user often also wants to quickly understand the information pointed out in the context of the other information and not just be shown the result items.

This is one type of focus+context (F+C) techniques, which have the goal of providing both detailed information of the currently most relevant objects, as well as giving the user an idea of their context (see the sidebar "Focus+Context").

1.1 Depth of Field

One method for guiding the user's attention is by blurring the less relevant parts of the display, while the relevant information is displayed sharply. This method has been used in photography for a long time, where the *depth of field* (DOF) determines which depth range is depicted sharply (for an example, see figure 1).

It is thus possible to show an object's or person's context by keeping it in the image, but showing it out of focus (the theory behind DOF is briefly described in the sidebar "Optical Basics of DOF").

Using that same idea to blur objects not based on their distance from the camera, but on their current relevance in the application, makes it possible to direct the viewer's attention. We call this method *Semantic Depth of Field* (SDOF).

1.2 Why SDOF?

There are other visual cues that can be used for the tasks that are described in this paper, like color, hue, etc. But we believe SDOF to be a valuable addition to the visualization toolbox. It can be used when all other cues are already used, to either reinforce another cue or provide additional information. Blur is also very intuitive, and therefore allows even untrained users to quickly understand what is pointed out. Blur also has the advantage of working independently of color, thus being available for black and white images as well as being useful for color-blind users.

1.3 Preattentivity

Visualization is so effective and useful because it utilizes one of the channels to our brain that have the highest bandwidths: our eyes. But even this channel can be used more or less efficiently. One special property of our visual system is preattentive processing [16, 17]. Preattentive processes take place within 200ms after exposure to a visual stimulus, and do not require sequential search. They therefore provide a very efficient means of conveying information to the human brain.

Section 5 summarizes some of the results of a user study we conducted to show that SDOF is, in fact, preattentively perceived.

2 Semantic Depth of Field

The central idea of Semantic Depth of Field (SDOF) is to blur objects based on their relevance. Human perception divides our field of view into foreground and background objects (or preferred and non-preferred stimuli [5]). This is a semantical division that does not depend on the physical positions of objects, i.e., closer objects can be considered as background objects, while more distant ones can become foreground objects. Using SDOF helps the eye with this division: blurred objects immediately become the semantic background, while sharp ones stay in the foreground.

SDOF can be used for both 2D and 3D images, but this paper focuses on 2D applications – a 3D application is discussed elsewhere [7].

2.1 Relevance and Blur

Apart from the spatial layout of the visualization, relevance and blurring are the main parts of rendering an SDOF image (figure 2).

Each object or data point is assigned a relevance value r by a relevance function. This function is provided and its results are calculated by the application. The actual function, and how many and which parameters it needs, depends on the application. Some might not allow any interaction at all, while others could be very configurable.

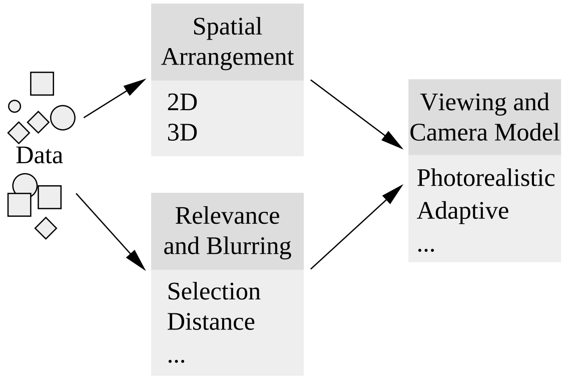


Figure 2. The building blocks of SDOF. Objects need to be arranged in the image, and their relevance has to be assigned, from which their blurring is determined. A model for rendering SDOF images with existing computer graphics techniques is presented in [7].

There are three types of relevance function: binary functions, that only classify objects into two classes; discrete ones, that create a small number of classes; and continuous functions, that use the whole range of values.

Our relevance function is similar to what is called a *degree of interest* (DOI) function by others [3]. But relevance is completely independent of layout (quite in contrast to fisheye views [3], for example), and is not directly used for blurring (but is passed through the blur function first, see below).

The value range of r goes from 0 to 1, inclusively; where 1 stands for an object of maximum relevance to the current query, and 0 for a completely irrelevant object. For example, a file system browser could assign a relevance of 1 to all files that were last changed at most three days ago, and 0 to all objects that are “older”. It could allow the user to change that threshold to different minimal ages, or to switch to a continuous r scale that assigns a value of 0 to the oldest file, a value of one to the newest one, and interpolates the r value between them for all other files. The user can then use the threshold of the blur function to change the appearance of the visualization.

The relevance value is translated into a blur diameter b, which is used for blurring the object – this is done by the blur function. The blur function can in principle take any shape, but we have found the one depicted in figure 3 to be sufficient for most cases (see section 2.4 for a discussion).

2.2 Properties

Blurring an image or object has two effects: It removes the high spatial frequencies and reduces the contrast. Both effects are direct consequences of the fact that neighboring

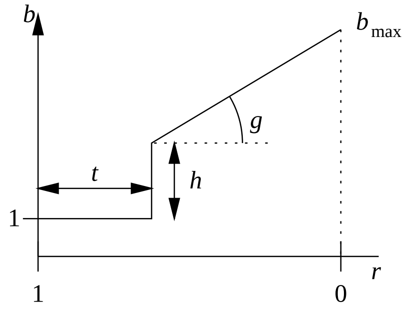


Figure 3. The blur function

pixels are summed up; both effects could be reached independently, but the visual impression would be quite different. This also leads to small details getting lost, so icons may become unreadable, for example. But this only applies to the context objects (see below), and thus should not be a relevant problem.

SDOF is very intuitive. Like a lens in a camera, the lens in the human eye cannot show all objects in focus at the same time. But we are seldomly aware of the fact that parts of our field of view are blurred – our visual system ignores these areas, the same way it ignores blurred parts of photographs. We also showed that it is, in fact, a preattentive feature (see section 5).

SDOF is also independent of color. This has a number of consequences: a) It can be used as an additional cue when color is already used in an application; b) when color is not available (e.g., on black and white printers), SDOF can still be used; c) SDOF can be used by color-blind and color-sighted people alike.

SDOF distorts the irrelevant objects, rather than the relevant ones. We believe this to be more useful, because the user is still able to see the features of relevant objects without having to adapt to a different kind of display. Only the features of irrelevant objects change and can become obscured. This is also an important feature because blurring makes the use of icons or other objects with fine details difficult – these features disappear when blurred. But because this is done to the less relevant objects, it is acceptable (and if not, the user has to "refocus" or find a different relevance funtion).

SDOF can be used to point out objects in any layout. Distortion techniques typically only have one focus, and don't deal very well with several foci distributed over the display (i.e., objects that are not relevant are also emphasized).

2.3 Applicability

Even though SDOF is a very general method, there are applications that it is suited better for, and there are cases when it cannot be applied at all.

SDOF is best suited for an application where objects should be pointed out that are of sufficient size so that they don't have to be magnified to be shown to the user. SDOF does not work well with pixel-based visualizations [6], either.

When there is no knowledge about the output device and no way for the user to interact with the application, SDOF cannot be used. This is due to the dependence of the appearance of blur on the viewing angle (see section 2.4).

But SDOF can be applied in many cases where other visual cues have already been used, and additional ones are needed. It can also be used when the properties of the output device do not allow color, saturation, etc to be used, or when these cues would interfere with the visualization.

SDOF can also be used as a very intuitive cue to point users to information or controls in user interfaces.

2.4 Parameterization

The visual effectiveness of SDOF comes at a price: SDOF depends on the output device and the conditions under which it is viewed. This is similar to other perceptually oriented techniques in computer graphics, like tone mapping.

It is therefore necessary to provide the user with a means to adjust the parameters of the display or at least to use good default values. The user can adjust the values h and $b_{\max}$ in the blur function (Figure 3), which give the program the limits of the usable blur. This can be done at the startup of applications, by showing the user a blurred and an unblurred image of the same object. The user can adjust the blur level of the blurred object to the smallest level that is still distinguishable from the unblurred object. The maximum desired blur is selected through the same interface. The values obtained in this way can be stored per user so that recalibration is only required when the viewing conditions change drastically.

Once the application also knows the threshold t, it is able to calculate g. Now the whole blur function is known, and can be used to render images. The threshold t can be changed quite frequently by the user to show different amounts of objects in focus while examining data.

2.5 Interaction

A key part of SDOF is interaction. Blurring objects is quite useless if the user cannot change the focus, or see what happens after he or she has changed parameters. Changing the focus by simply blurring different objects is also very effective in terms of guiding the user's attention to the now most relevant information. The apparent movement is similar to methods used in cinematography, where the focus is changed from one actor to another, to focus on the currently speaking person, or to make the viewers aware of a person in the background. It is therefore necessary to animate the change in focus, so that the user can follow it. The following list enumerates a few interactions that appear to be typical for SDOF applications:

Selecting the SDOF dimension. The user decides which data dimension is mapped to blurring. This may not be possible in all applications.

Selecting the relevance function. If an application provides more than one relevance function per data dimension, the user has to be able to select one and possibly set some parameters. Often, the relevance function is an implicit feature of the DOI specification.

Changing the threshold. As soon as the display shows SDOF, the user should be able to change the blur function threshold. This can be done directly on the level of the blur function (changing the t value), or by selecting values in the data domain, which are then translated to r values for the threshold by the application.

Autofocus. As soon as the user has seen the relevant information, he or she might want to go back to a sharp display. This is done with the "autofocus" feature, which brings all objects into sharp focus again – after a certain timeout or triggered by the user.

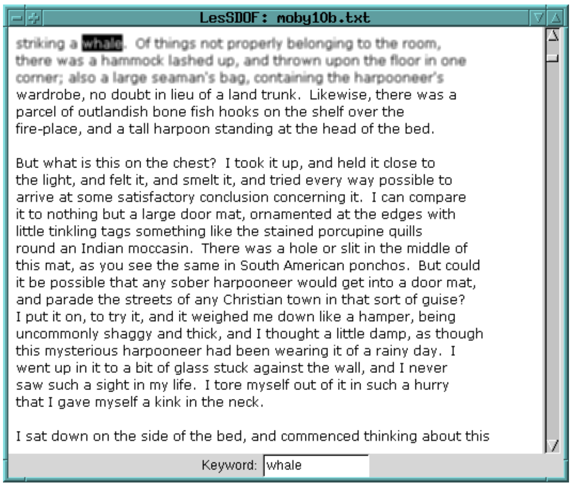
3 Applications

In this section, we describe four applications to demonstrate the usefulness of SDOF.

3.1 LesSDOF: Text Display and Keyword Search

The Problem. Displaying text and being able to search for keywords is a very common application. Most applications only show the found keyword (e.g. using color), but leave it to the user to understand the context. It would be helpful to be shown the whole sentence in order to more quickly make use of the search result.

The Application. *LesSDOF* displays a text file and allows the user to scroll through it, much like the Unix program *less*. When scrolling a whole page, a few lines are displayed on both pages as context. Theses lines are slightly blurred so that user understands that this is context information (Figure 4). When searching for a keyword, the found words are displayed with their fore- and background colors exchanged, and therefore clearly stand out. The sentence in which they appear is displayed sharply, while the rest of the page is blurred. It is possible to jump between hits, and so



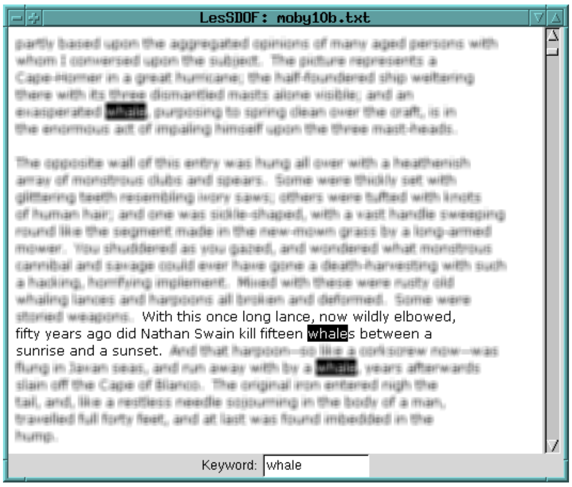
```
striking a whale.  Of things not properly belonging to the room,
there was a hammock lashed up, and thrown upon the floor in one
corner; also a large seaman's bag, containing the harpooneer's
wardrobe, no doubt in lieu of a land trunk.  Likewise, there was a
parcel of outlandish bone fish hooks on the shelf over the
fire-place, and a tall harpoon standing at the head of the bed.

But what is this on the chest?  I took it up, and held it close to
the light, and felt it, and smelt it, and tried every way possible to
arrive at some satisfactory conclusion concerning it.  I can compare
it to nothing but a large door mat, ornamented at the edges with
little tinkling tags something like the stained porcupine quills
round an Indian moccasin.  There was a hole or slit in the middle of
this mat, as you see the same in South American ponchos.  But could
it be possible that any sober harpooneer would get into a door mat,
and parade the streets of any Christian town in that sort of guise?
I put it on, to try it, and it weighed me down like a hamper, being
uncommonly shaggy and thick, and I thought a little damp, as though
this mysterious harpooneer had been wearing it of a rainy day.  I
went up in it to a bit of glass stuck against the wall, and I never
saw such a sight in my life.  I tore myself out of it in such a hurry
that I gave myself a kink in the neck.

I sat down on the side of the bed, and commenced thinking about this
```

Keyword: whale

Figure 4. Scrolling in LesSDOF: The top three lines are context from the last page, and therefore blurred – but still readable.



```
partly based upon the aggregated opinions of many aged persons with
whom I conversed upon the subject.  The picture represents a
Cape-Horner in a great hurricane; the half-foundered ship weltering
there with its three dismantled masts alone visible; and an
exasperated whale, purposing to spring clean over the craft, is in
the enormous act of impaling himself upon the three mast-heads.

The opposite wall of this entry was hung all over with a heathenish
array of monstrous clubs and spears.  Some were thickly set with
glittering teeth resembling ivory saws; others were tufted with knots
of human hair; and one was sickle-shaped, with a vast handle sweeping
round like the segment made in the new-mown grass by a long-armed
mower.  You shuddered as you gazed, and wondered what monstrous
cannibal and savage could ever have gone a death-harvesting with such
a hacking, horrifying implement.  Mixed with these were rusty old
whaling lances and harpoons all broken and deformed.  Some were
storied weapons.  With this once long lance, now wildly elbowed,
fifty years ago did Nathan Swain kill fifteen whales between a
sunrise and a sunset.  And that harpoon--so like a corkscrew now--was
flung in Javan seas, and run away with by a whale, years afterwards
slain off the Cape of Blanco.  The original iron entered nigh the
tail, and, like a restless needle sojourning in the body of a man,
travelled full forty feet, and at last was found imbedded in the
hump.
```

Keyword: whale

Figure 5. Finding a keyword in LesSDOF: There are three hits on this page, with the focus currently on the middle one. The sentence around the keyword is clearly visible, while the rest of the context is blurred.

move the focused sentence, or to show all context sentences in focus (Figure 5).

SDOF Aspects. This application only uses a binary relevance classification. A text string is either a keyword or it is not, a line of text is either new or overlapping from the last page. Blur and other cues (like inverse display of the keywords) are used to reinforce each other in the case of the current keyword, and as orthogonal dimensions for other keywords. This example does not use any color, and is still very effective in guiding the viewer's attention.

Interaction. In LesSDOF, the user cannot directly influence either the relevance or the blur function. When paging through a text, the "overlapping" lines are displayed using the minimum perceivable blur; when showing the results of a search, the irrelevant parts are displayed using the maximum acceptable blur.

3.2 sfsv: File System Viewer

The Problem. File system viewers like the Windows explorer are among the most used applications on today's personal computers. Some aspects of them are quite effective (like the tree view), while others are quite poor. One of the poorer aspects is the ability to quickly look for different information in a directory or directory structure without losing the context. Sorting the data according to a data dimension clearly is not a solution to this problem, because it destroys the user's mental map.

The Application. The *sfsv* application (SDOF-enhanced file system viewer) shows a directory structure in a slight variation of the well-known treeview (Figure 6). It is possible to do different queries on this data and show the results using different visual cues. One of these cues is blur. So if the user selects his or her own files as the focus (Figure 6, bottom right), the files of other users are blurred.

SDOF Aspects. Here, SDOF can be used both as an orthogonal cue and a reinforcement, depending on the user's needs. The combination of cues makes it possible to find files in their context, e.g., the ones that eat up all the hard disk space.

3.3 Sscatter: Scatterplots

The Problem. Scatter plots are a very useful tool to get an overview over data and to test hypotheses. But scatter plots are only really useful for two data dimensions, others must be mapped to visual attributes of the displayed objects. A large number of easily distinguishable cues is therefore needed.

The Application. *sscatter* can read data files in different formats whose structure (column delimiters, sizes, names, how many lines per data point, etc.) can be specified in a configuration file. It displays the data in a scatter plot, where the user can select which data dimensions are mapped to which visual features. When used on data of car models from 1993, for example, one can see that more expensive cars have lower fuel efficiency, and that American and other cars are available over the whole price range (Figure 7). It is also possible to find out that the availability of manual transmission is generally a feature of more expensive cars (Figure 8).

SDOF Aspects. Because the user is free to choose data dimensions, a combination of binary (e.g., availability of manual transmission), discrete (e.g., number of cylinders) or continuous (e.g., price, engine size, etc.) relevance measure. What exactly is needed depends on what the user wants his or her new car to be or do.

3.4 sMapViewer

The Problem. When displaying a large number of information layers in a geographical visualization, the user has to decide whether to be distracted by too much information, or to have less – and possibly too little – information visible at the same time.

The Application. sMapViewer (Figure 9) allows the user to stack layers of geographical information on top of each other. The topmost layer is displayed sharply, while all other layers are increasingly blurred. This creates a sense of depth that makes it easy to see relavant objects clearly in their context.

Interaction. The user can select the layer to be put on top of the stack, thus indirectly changing the relevance of all objects.

SDOF Aspects. The regular version of this program contains the possibility of defining a continuous relevance function. The user study version uses a discrete function in SDOF mode.

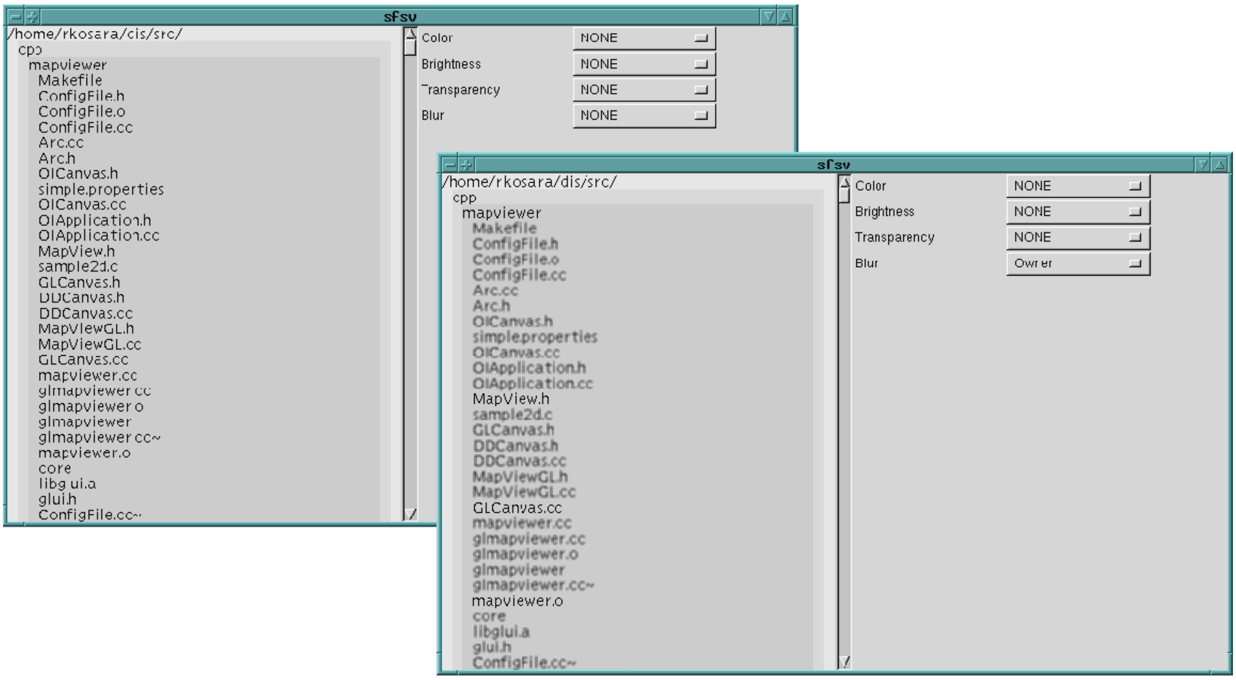
3.5 Other Applications

Other application examples include an SDOF-enhanced 2D and 3D chess tutoring system that can point out pieces and constellations on the board, like which chessman covers which others, or which pieces threaten a particular one [7].

It would also be possible alert the user to new information in a window by displaying it blurred when no new information is there, and displaying it sharply when new data are coming in. This would be especially useful when more than one objects can be the source of an audible alarm, for example.

4 Implementation

One of the reasons blur has been little used in computer graphics is that it is slow when done in software on proces-



```
sfsv
/home/rkosara/cis/src/
  cpp
    mapviewer
      Makefile
      ConfigFile.h
      ConfigFile.o
      ConfigFile.cc
      Arc.cc
      Arc.h
      OlCanvas.h
      simple.properties
      OlCanvas.cc
      OlApplication.h
      OlApplication.cc
      MapView.h
      sample2d.c
      GLCanvas.h
      DDCanvas.h
      DDCanvas.cc
      MapViewGL.h
      MapViewGL.cc
      GLCanvas.cc
      mapviewer.cc
      glmapviewer.cc
      glmapviewer.o
      glmapviewer
      glmapviewer.cc~
      mapviewer.o
      core
      libglui.a
      glui.h
      ConfigFile.cc~

Color          NONE
Brightness     NONE
Transparency   NONE
Blur           NONE
```

```
sfsv
/home/rkosara/dis/src/
  cpp
    mapviewer
      Makefile
      ConfigFile.h
      ConfigFile.o
      ConfigFile.cc
      Arc.cc
      Arc.h
      OlCanvas.h
      simpleproperties
      OlCanvas.cc
      OlApplication.h
      OlApplication.cc
      MapView.h
      sample2d.c
      GLCanvas.h
      DDCanvas.h
      DDCanvas.cc
      MapViewGL.h
      MapViewGL.cc
      GLCanvas.cc
      mapviewer.cc
      glmapviewer.cc
      glmapviewer.o
      glmapviewer
      glmapviewer.cc~
      mapviewer.o
      core
      libglui.a
      glui.h
      ConfigFile.cc~

Color          NONE
Brightness     NONE
Transparency   NONE
Blur           Owner
```

Figure 6. A file system viewer with all files in focus (top left) and one focusing on the files of one user (bottom right).

Program	Size (Pixels)	Percentage Blurred	Framerate	Figure
LesSDOF	500x400	12%	167	Figure 4
LesSDOF	500x400	92%	143	Figure 5
sfsv	380x480	25.5%	31	Figure 6
sscatter	600x600	37.2%	23	Figure 7
sscatter	600x600	46.5%	19	Figure 8

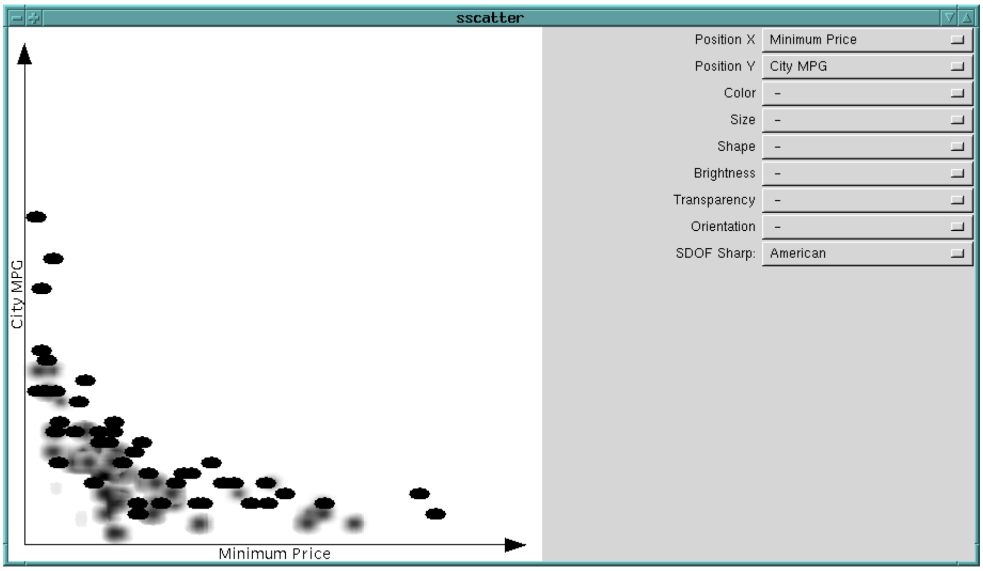
Table 1. Performance figures for the applications shown in this paper.

sor architectures currently used in desktop computers. This is due to the fact that information has to be summed up over an area for every pixel, which quickly adds a load to the application that makes it unusable.

Modern graphics hardware makes it possible to render blurred images very quickly, and thus makes it usable in interactive applications. The key to the implementation described here is texture mapping. Texture mapping is the central operation on low-cost graphics hardware, because it is used extensively by computer games. A number of tasks in visualization have been accelerated tremendously by (ab)using texture mapping and for tasks that at first glance have nothing to do with it.

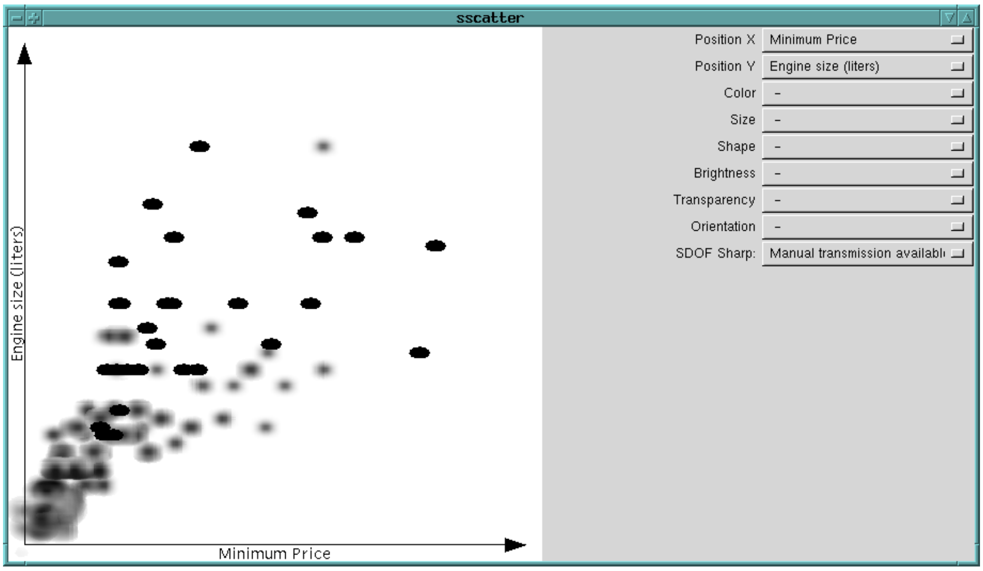
Blurring is simply the summing up the information around a pixel for every pixel in the image. This can be done by drawing an image several times at slightly different positions and having the graphics hardware sum up the color information at every step. We draw the image into the frame buffer once and then copy it into a texture (this is a fast operation). We then use the texture to draw the image several times, by mapping it onto a rectangle. After displacing the image several times in the x direction, we copy the result of this operation into another texture. This texture is then used when displacing the image in the y direction. We thus produce an image filtered through a box filter.

Because of the limited precision of low-cost graphics cards (typically eight bits per color component), we cannot simply add up the image for any blur diameter we want. We therefore have to calculate auxiliary sums that we then sum up in a second step. Because of the filter kernel we use, this actually makes the operation faster. All auxiliary sums (except at the edges) are equal, and only displaced. We therefore only have to calculate this sum once and can then displace it (with larger distances in between) several



Position X: Minimum Price
Position Y: City MPG
Color: –
Size: –
Shape: –
Brightness: –
Transparency: –
Orientation: –
SDOF Sharp: American

Figure 7. A scatterplot of car data showing that more expensive cars have a lower miles per gallon (MPG) number, and that American and other cars are available over the full price and MPG range.



Position X: Minimum Price
Position Y: Engine size (liters)
Color: –
Size: –
Shape: –
Brightness: –
Transparency: –
Orientation: –
SDOF Sharp: Manual transmission availabl

Figure 8. A scatterplot of car data showing that more expensive cars have larger engines, and that the availability of manual transmission is generally a feature of more expensive cars.

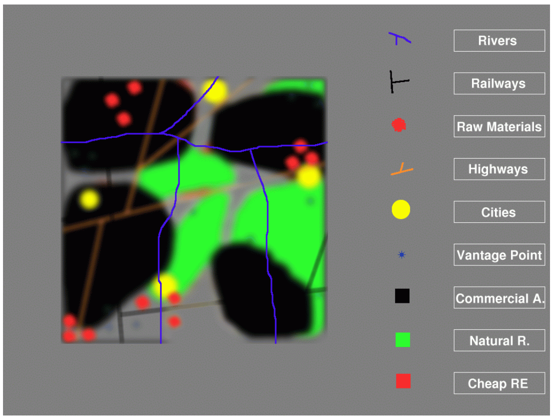


Figure 9. sMapViewer showing rivers, railways, raw materials, highways, cities, vantage points, commercial areas, natural reserves and cheap real estate.

times, thus doing fewer additions overall.

Table 1 gives an overview over frame-rates for some of the applications presented earlier on. These numbers were collected on a standard PC with a 450 MHz CPU and an NVidia GeForce2 MX graphics card.

It is quite clear from that table, that this implementation of SDOF is faster for fewer large objects than many small ones, even if the large objects cover a much bigger part of the display. Most time is obviously lost in the setup and communication between the CPU and the graphics card, and not the actual texture operations on the card. Even though the current implementation uses display lists in many places, there is still room for improvement, so that the amount of communication needed is decreased even further.

5 User Study

We performed a user study with 16 subjects (male, aged 18–25 years, university students, very good vision, basic computer skills) to test the preattentivity of SDOF, and to see how useful it was in applications. This section gives an overview of the results of this study, which are covered in more detail elsewhere [4].

SDOF is preattentive. Subjects were able to detect and locate objects after being shown images containing them (and up to 63 distractors, i.e., blurred objects) for only 200ms (with accuracies of over 90%, depending on blur level and number of distractors). They were also able to estimate the number of sharp objects after the same time. This estimation was shown to be significantly better than chance.

The combination of visual features was also very interesting. For these tests, participants could look at the images as long as they needed to find the answer, but were of course asked to answer as quickly as possible. This test showed that there was no significant difference in search time between blur and color, which was above our expectations. The conjunction of blur and color (“find the red sharp object”) was not significantly slower than a simple search for color or sharpness, either. This is also surprising, because conjunctive search is usually slower than simple search. Sharpness and color were also significantly faster than any of these two features combined with orientation, which was to be expected.

A further test block clearly showed that SDOF cannot be used as a fully-fledged visualization dimension. It was very tiring for participants to try to tell the difference in blur between blurred objects, and they were not able to tell that difference in any meaningful way. SDOF ist still useful for discriminating a small number of classes (three or four, this is subject to further tests), but not more.

6 Conclusions and Future Work

Semantic Depth of Field (SDOF) allows the user to literally focus on the currently relevant information. It is thus possible to display the results of queries in their context and make them easier and faster to comprehend.

The use of blur makes use of a visual feature that is inherent in the human eye and therefore is very perceptually effective. We showed that perception of SDOF was preattentive and not significantly slower than color.

But not only the perception of SDOF, but also its creation is fast with the implementation described in this paper. All this makes SDOF a useful additional visual cue that is at the disposal of visualization application designers.

The next step is to find out how well SDOF works together with other F+C techniques, especially with distortion-oriented ones. We also want to see how SDOF can be applied to areas such as volume and flow visualization; and also its applicability in user interfaces.

7 Acknowledgements

We would like to thank the reviewers and especially the guest editor for their valuable remarks and suggestions that helped to improve this paper.

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.

8 Further Information

More information on SDOF can be found at these two URLs:
`http://www.ifs.tuwien.ac.at/research/sdof/` and `http://www.vrvis.at/vis/research/sdof/`.

Sidebar: Focus and Context

Information Visualization often deals with data that users have no mental image of. A visualization imposes a graphical structure – a mapping from data to screen space – on the data that the user has to learn. It is therefore necessary to change this mapping as little as possible; but there is often not enough space on the screen to display all information with enough detail. Focus+Context (F+C) methods make it possible to show more, more detailed, or more targetted information – and at the same time, giving the user a sense of where in the data the zoomed-in, more detailed, or pointed out information is.

We divide the currently existing F+C methods into three groups:

Spatial Methods. This is the most prominent group of F+C methods. The image created with an existing visualization is distorted to allow more space for the currently more important objects, and less for the context. Examples for this method are fish-eye views [3], hyperbolic trees [8, 12], the document lens [13], stretchable rubber sheets [14], and other distortion-oriented methods [10] (as they are generally called). One draw-back of these methods is that they do not allow control of the degree of interest that is completely independent of the layout of the objects.

Dimensional Methods. The user can move a focus over a visualization to display different data about the same objects. These methods do not make it possible to display more objects, but more or different data dimensions of the already displayed ones. Examples of this type of F+C method are magic lenses [15], tool-glasses [1], etc.

Cue Methods. In an existing visualization, objects that meet certain criteria are stressed by assigning visual cues to them so that they are more prominent to the viewer without hiding the context. An example of such a method is to use color saturation and brightness [11]. Another method that is relevant in this context, is a system that allows up to 26 layers of geographical information to be displayed at the same time [2]. The user can move the focus between these layers by changing their blur level and transparency.

Sidebar: Optical Basics of DOF

In contrast to a pinhole camera (which also is the camera model that is mostly used in computer graphics, especially in visualization), a lens causes more than one light rays to depict every point of an object (Figure 10).

The *lens equation* [9] defines the distance v of the sharp image from the lens, depending on the distance u of the object and the lens' focal length f:

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \tag{1}$$

Because the film has a fixed distance from the lens, only points in a plane parallel to the film plane at a certain distance are perfectly sharp. Other points are not depicted as points, but as small discs, the so-called *circles of confusion* (CoC). Their size depends on the distance of the point from the focus plane, as well as the focal length and the size of the lens. Objects are perceived sharp when they are depicted by CoCs whose diameter is smaller then the resolution of the eye (which depends on their absolute size and viewing distance).

References for Sidebars

This section is only here to show which references are used in the sidebars, to make them easier to associate:

Focus+Context: [3], [8], [12], [14], [10], [15], [1], [11], [2], [13]

Optical Basics: [9]

[3] is also cited in the main text. Is that a problem?

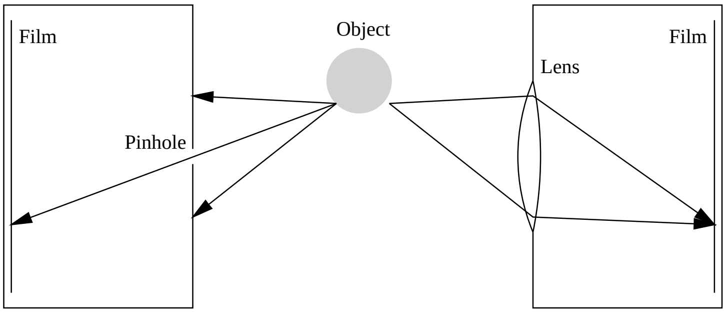


Figure 10. The pinhole camera vs. the lens camera

Author's Biographies

Robert Kosara

Robert Kosara received his Dipl.-Ing. (MSc.) in Computer Science from Vienna University of Technology in 1999. He has just finished his PhD work on Semantic Depth of Field.

Robert currently works as a researcher with the Asgaard Project headed by Silvia Miksch, where he is concerned with visualisation matters. His main research interest is information visualization, with a special focus on the interdisciplinary possibilities of this fascinating field.

Silvia Miksch

Silvia Miksch is Associate Professor at the Institute of Software Technology and Interactive Systems, Vienna University of Technology, Austria. Since 1998, she is head of the Asgaard research group. She was scientific researcher at the Knowledge-Based Systems Group, Austrian Research Institute for Artificial Intelligence (ÖFAI), and was post-graduate research fellow at Knowledge System Laboratory (KSL), Stanford University, CA, USA. From April to September 2001, she was Visiting/Guest Professor at the Department of Simulation and Graphics, Otto-von-Guericke University Magdeburg (Germany) and she will be Visiting/Guest Professor at the Johannes Kepler University Linz (Austria) from October 2001 to January 2002.

Silvia Miksch served in various Conference/Workshop Program Committees and refereed for various conferences and journals. She also organized workshops, served as guest editor, and belongs to the editorial board of the AI Communications (AICOM; IOS Press) and Artificial Intelligence in Medicine (Elsevier).

Her main research interests are Information Visualization, Planning and Scheduling, Data and Process Modeling, Temporal Representation and Reasoning, and Task-oriented Design and Evaluation.

Helwig Hauser

Helwig Hauser graduated in 1995 from Vienna University of Technology (TU Wien), Austria, as a Dipl.-Ing. (≈ MSc.) after studying computer science with special focus on computer graphics. In 1998, he received the degree of Dr.techn. (PhD) from the same university (TU Wien) after a research project in the field of scientific visualization (visualization of complex dynamical systems). From 1995 until mid-2000, Helwig Hauser was Univ.-Ass. (assistant professor) at the Institute of Computer Graphics (TU Wien, also). There he was a founding member of the visualization group around Prof. M. E. Gröller (`http://www.cg.tuwien.ac.at/research/vis/`). Since then, Helwig Hauser is key researcher at the VRVis Research Center, also located in Vienna, leading the basic research in the field of visualization there (`http://www.VRVis.at/vis/`). His main interests include volume, flow, and information visualization, as well as visualization and computer graphics in general. Helwig Hauser is member of IEEE.

References

[1] 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.

[2] 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.

[3] 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.

[4] V. Giller, M. Tscheligi, J. Schrammel, P. Fröhlich, B. Rabl, R. Kosara, S. Miksch, and H. Hauser. Experimental evaluation of semantic depth of field, a preattentive method for focus+context visualization. Technical Report TR-VRVis-2001-021, VRVis Research Center, `http://www.vrvis.at/`, Austria, 2001.

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

[6] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, Jan.-Mar. 2000.

[7] R. Kosara, S. Miksch, and H. Hauser. Semantic depth of field. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (InfoVis2001)*. IEEE, Oct. 22–23 2001.

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

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

[10] 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.

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

[12] 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.

[13] G. G. Robertson and J. D. Mackinlay. The document lens. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Visualizing Information, pages 101–108, 1993.

[14] 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.

[15] 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.

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

[17] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, 2000.