

Tactile Access to Scalable Vector Graphics for People with Visual Impairment

Keywords: Scalable Vector Graphics, Universal Access, Tactile Graphics for Visual Impaired People, Multi-Modal Interface, Accessibility

Martin Rotard, Ph.D.

Researcher in Computer Graphics and Human Computer Interaction
University of Stuttgart
Stuttgart
Germany

[rotard\(A.T.\)vis.uni-stuttgart.de](mailto:rotard(A.T.)vis.uni-stuttgart.de)

<http://www.vis.uni-stuttgart.de/~rotard>

Biography

Martin Rotard has been staff-member and PhD student of the Visualization and Interactive Systems Institute, University of Stuttgart, since June 2001. His research interests are new graphical standards on the web like Scalable Vector Graphics, MathML, X3D, SMIL, etc., Human Computer Interaction, Usability, and Accessibility. He was engaged in the multimedia-teaching project ITO (Information Technology Online) that was funded by the German Federal Ministry of Education and Research. His development activities are in the field of XML-based graphics, universal access on the web, e-learning, graphical user interfaces and software usability.

Thomas Ertl, Prof. Dr.

Professor of Visualization and Computer Graphics
University of Stuttgart
Stuttgart
Germany

[ertl\(A.T.\)vis.uni-stuttgart.de](mailto:ertl(A.T.)vis.uni-stuttgart.de)

<http://www.vis.uni-stuttgart.de/>

Biography

Thomas Ertl received a masters degree in computer science from the University of Colorado at Boulder and a PhD in theoretical astrophysics from the University of Tuebingen. Currently, Dr. Ertl is a full professor of computer science at the University of Stuttgart, Germany and the head of the Visualization and Interactive Systems Institute (VIS). Prior to that he was a professor of computer graphics and visualization at the University of Erlangen where he lead the scientific visualization group. Besides that, he is a cofounder and a member of the managing board of science+computing ag, a Tuebingen based IT company. His research interests include visualization, computer graphics and human computer interaction in general with a focus on volume rendering, flow visualization, multiresolution analysis, parallel and hardware accelerated graphics, large datasets and interactive steering. Dr. Ertl is coauthor of more than 150 scientific publications and he served on several program and paper committees as well as a reviewer for numerous conferences and journals in the field.

Abstract

For visually impaired people Scalable Vector Graphics (SVG) provide a great opportunity to access graphical information. We propose an interactive transformation schema, which transforms SVG into a tactile representation. This paper describes how this transformation schema is realized and supported by navigation through the SVG shape hierarchy and by applying filters. Additionally meta information is extracted from the SVG and presented on the braille line and by voice output.

Table of Contents

[1. Motivation](#)

[2. Transformation Schema](#)

[3. Navigating through the Shape Hierarchy](#)

[4. Applying Filters to the Graphics](#)

[5. Results and Outlook](#)

[Acknowledgements](#)

[Bibliography](#)

1. Motivation

Images became standard in nowadays information publishing. The information presented in images must be accessible to all people. For users with visual disabilities it is necessary to access this information with non-visual devices. Scalable Vector Graphics provide a great opportunity to explore graphics on tactile devices. Text, shapes, attributes, and meta information of SVG can be extracted easily.

Tactile output devices have a low spatial resolution (e.g. 16x16 or 120x60) and can display just monochrome output (pins up or down). We propose a transformation schema, which makes it possible to display and to explore SVG on these devices. We developed a software environment, which can be used directly by blind users for exploring graphics as well as by developers to simulate several tactile output devices [Rotard et al. 2004]. The user can navigate in the shape hierarchy of the graphics and apply filters. The tactile output is complemented by braille line and voice output. [Figure 1](#) shows a screenshot of our software environment. The scaled original graphics is at the bottom of the user interface and in this example the contour filtered graphics in an incremental buildup is at the top. The parameters for the transformation schema could be set using the panel on the right side. There is also a navigation bar and a simulated braille display at the bottom of the user interface. The tree view on left side could be used by developers for a fast navigation in the SVG shape hierarchy.

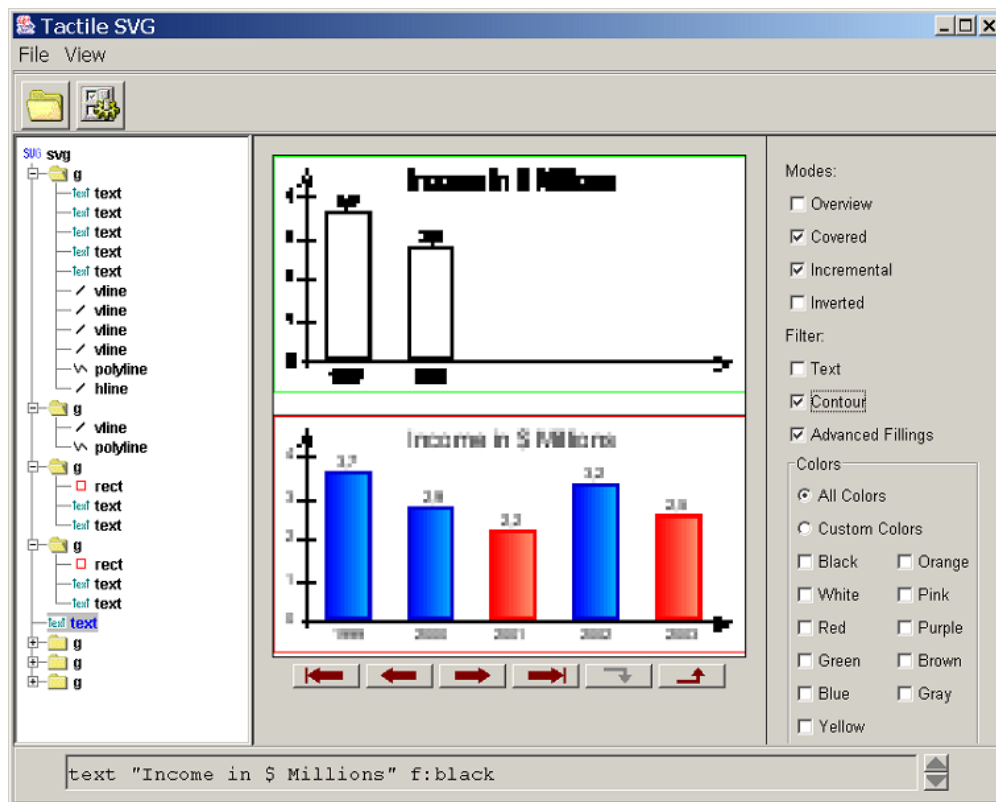


Figure 1: Screenshot of our software environment

Some related work has been done in this field by the Web Accessibility Initiative [WAI]. A guideline on how to create SVG in an accessible way can be found in [Jackson 2003] [McCathieNevile et al. 2002]. Recent publications concerning accessing SVG for visually impaired people focus on specialized areas like extracting meta information [Herman et al. 2002], visualization of tactile maps [Campin et al. 2003] and transformation for tactile embossers [Gardner et al. 2001] [Gardner et al. 2004].

2. Transformation Schema

Our system is developed in Java and uses several open source tools or libraries. The Batik SVG Toolkit is used for rasterizing SVG images, JDOM for accessing the XML files and FreeTTS for the voice output [Batik] [JDOM] [FreeTTS]. Our equipment is a tactile matrix device from Metec, which has 120x60 tactile pins. We developed a Java driver for this device, which is plugged in into the software environment. Because of the flexible interface other tactile devices can be integrated easily. Therefore our results are not restricted to this special hardware. Furthermore our software environment can simulate this transformation schema for all sizes of tactile devices.

The diagram in [Figure 2](#) shows the transformation schema of our system. In a preprocessing step the system analyzes the XML file and builds up an internal structure tree. For each shape a new node is added where the collected meta information is stored. This meta data is among others the shape type, the fill and the stroke color, the content of the title-tag and description-tag, and the id-attribute. For each shape the color of the stroke and filling is stored as SVG color keyword name for information output and as the nearest color of the "basic color terms" for filtering (see [Chapter 4](#) for further details). If the current node is a "use"-tag, it is resolved to the referencing shape. In our transformation we consider all defined CSS (Cascading Style Sheet) attributes. There is also a simplifier mechanism that removes groups, which are empty or have just one shape. If this group has attributes, they are assigned to the child shape. The structure tree is used for navigation and for output of the meta information on the braille line and by voice output.

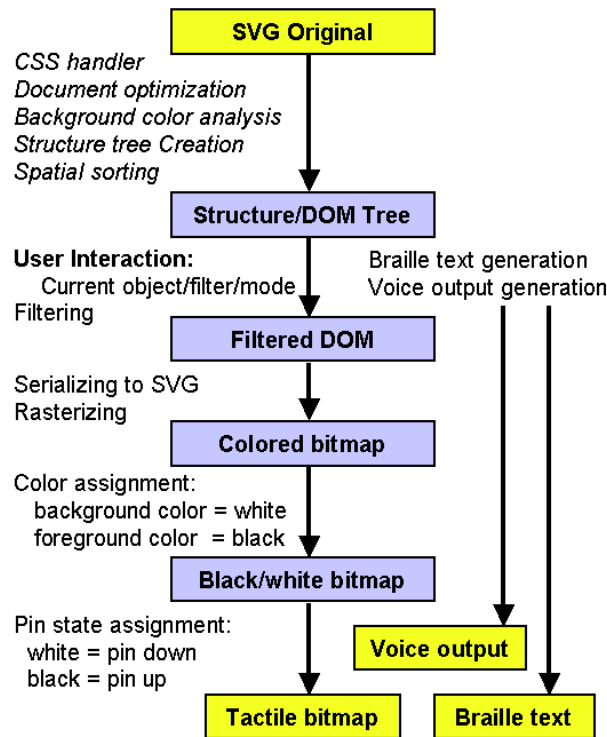


Figure 2: SVG transformation schema

Because the tactile device is restricted to two pin modes, we have to find out the background color. Therefore a heuristic method tries to find one of the first shapes with approximately the size of the entire graphik. The fill color of this shape is identified as background color. If no shape with this attributes could be found, the background color is set to white. All colors similar to the background color are rendered as pins down. All other colors are identified as foreground and where rendered as pins up. This can be inverted interactively. Because of the low resolution and the single foreground color the shapes are rendered without the use of antialiasing methods.

The shapes in SVG are not stored spatially sorted. As a consequence the shapes in an incrementally accumulated buildup will appear unsorted on the output (see [Chapter 3](#)). Therefore our system calculates a bounding box of all shapes and groups and sorts them spatially. In this calculation the transform attribute of the current shape and of all parent nodes have to be considered. The sorting is done on the references to the nodes in the structure tree and not in the SVG-file, so the geometry stays unchanged. In our system the shapes are sorted from left to right and from top to bottom.

3. Navigating through the Shape Hierarchy

In our environment the graphics is scaled to fit the size and resolution of the tactile device. Our concept of exploring SVG images is simple. The user can navigate to the next/previous shape or group and enter or leave groups. At each navigation step the graphics is updated on the tactile device and the user is informed on the braille line and by voice output about some meta information concerning the current shape or group. This could be the shape type, the fill and stroke color, the contents of a text shape, the title- and description-tag and the id-attribute. Additionally the user is informed about the position of the shape or group and about other attributes.

The buildup of the graphics can be either shape-by-shape or incrementally accumulated with respect to the spatial sorting. In the shape-by-shape mode the "visibility"-attribute of all shapes are set to *hidden*, except the current shape or group is set to *visible*. In the mode, where the buildup is incrementally accumulated, all previous shapes of the spatial sorting with respect to the current shape or group are mapped to the foreground. All other shapes are mapped to the background (see [Figure 3](#)).

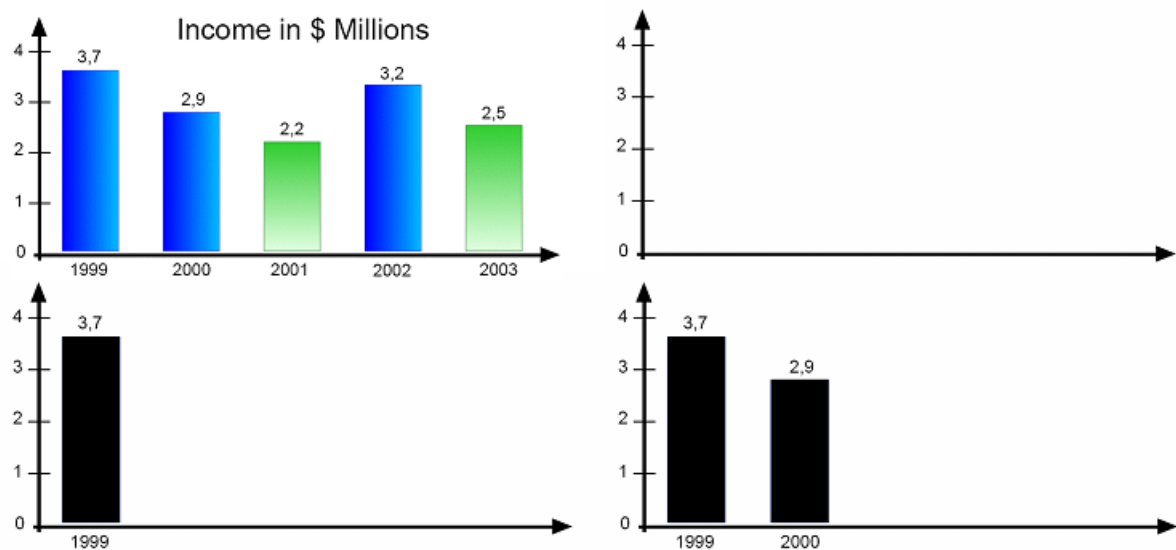


Figure 3: Original graphics and three incrementally accumulated buildup steps

For orientation there is a command to show the whole graphics at once. In this mode it is even possible to use filter effects and to navigate through the shape hierarchy. The user can turn on a mode where the shapes are drawn without minding the occlusion. This makes it possible to see a shape or group in one piece, which is partly hidden by another shape or group. The user can also zoom in and out. In this case the output size can be larger than the size of the tactile device. Using the keys or the mouse the point of view can be moved.

4. Applying Filters to the Graphics

To support the exploration of SVG, the visually disabled users can interactively apply and combine special filters. There is a text filter, which sequentially shows the position of each text string on the tactile output device by toggling the text shape between foreground and background. The textual contents is printed on braille line and output by voice. Another filter removes gradients and patterns in fillings and SVG filter effects. For a blind user this fillings could be confusing. For example the result of the transformation schema for a color gradient could be an area, where some parts are rendered in the foreground color and some parts in the background color. [Figure 4](#) shows some bars of the original graphics on the left, the result of the standard transformation scheme in the middle, and on the right the result with the filter, which removes advanced fillings. A contour filter removes the filling and shows just the outline of shapes (see [Figure 1](#)). This helps the blind users to feel the shape or group more clearly.

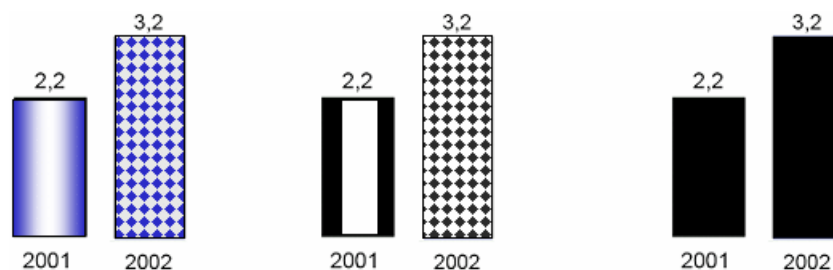


Figure 4: Original graphics, result of the standard transformation scheme, and result of the filter, which removes advanced fillings

Because the pins of tactile output devices have just two modes (pins up or down), it is necessary to have color filters. In our approach we use the "basic color terms" established by Brent Berlin and Paul Kay in 1969 [[Berlin et al. 1969](#)] as default set of colors. The "basic color terms" build a basis of distinguishable colors which are white, black, gray, red, pink, yellow, green, blue, purple, brown, and orange. Of course the number of colors in the set and the color values could be adapted to the needs of the user. In our software environment the RGB colors and SVG color keyword names of shapes are assigned to the

nearest color of the "basic color terms" in the L*a*b color space [McLaren et al. 1976]. E.g. if the user turns on the green and black color filter then just the shapes are shown, which have a similar color to green and blue (see Figure 5). If the assignment is ambiguous, a color can be assigned to more than one color of the "basic color terms". In these filters the color of the stroke and filling is considered separately. Additionally the fill and the stroke color of the shape is output as the (nearest) SVG color keyword name on the braille line and by voice output.

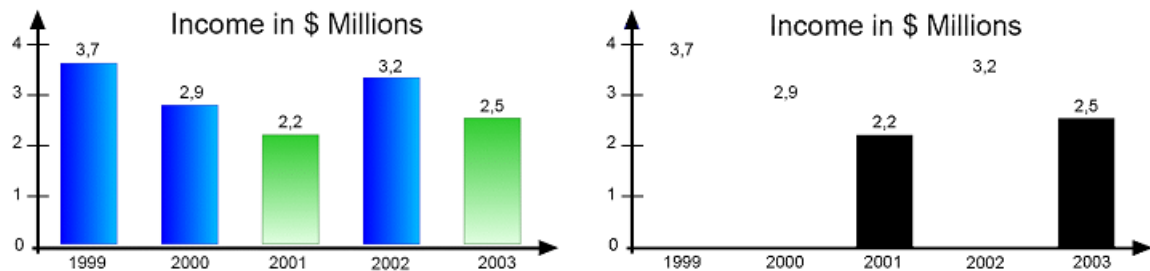


Figure 5: Original and result of the transformation schema with the color filter for green and black

5. Results and Outlook

A first evaluation of our software environment by a blind colleague in our group has shown that our approach is very promising. The metadata output, the filters, and the easy handling of the navigation where helpful for the exploration of the graphical content. The graphical quality of SVG rendered with Batik in the reduced size on our tactile device is higher than the quality of the down scaled bitmap graphics. As result of our transformation schema Figure 6 shows a bar chart graphics on our tactile device.

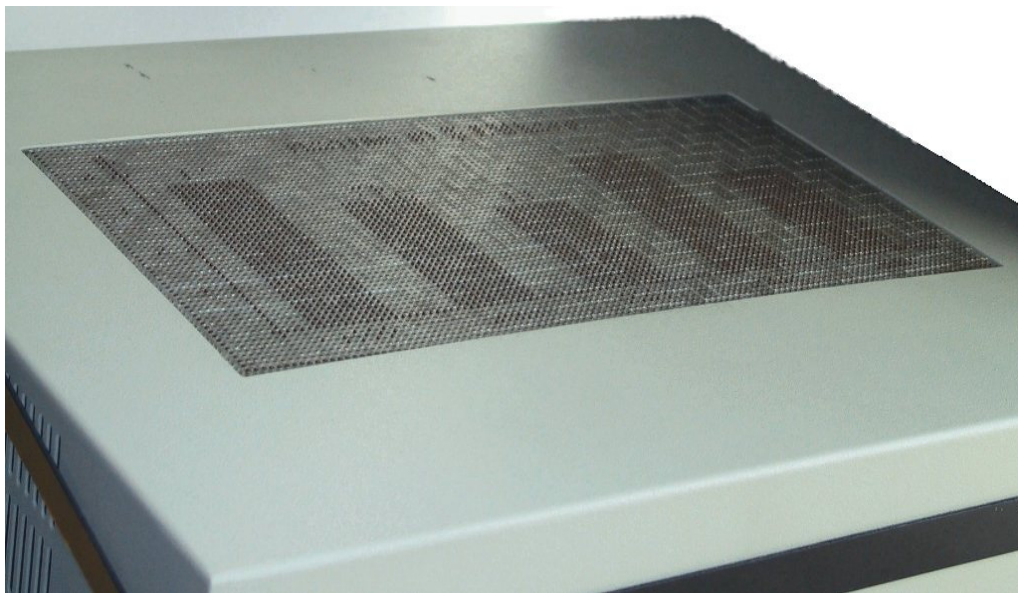


Figure 6: Result of the transformation schema on our tactile device

Our transformation schema could also be used by people who are color blind to handle false colors and for visually impaired people to zoom the graphics. Currently we do not support animations and scripting of SVG, but it would be possible to realize this in the future. The pins of our tactile matrix device are slow and can be set just sequentially, because they are set electromagnetically. Therefore the amperage is too high to set 7200 pins simultaneously. Modern piezo technology would be much faster and the pins can be set in parallel. Animation and scripting effects would be much easier to explore for blind people on piezo than on electromagnetically devices.

Acknowledgements

We would like to thank Waltraud Schweikhardt, Alfred Werner, Andreas Hub, Daniel Weiskopf, and Gerhard Weber for their help to finish this paper and for the fruitful discussions.

Bibliography

[Batik]

Apache XML Project: *Batik*, <http://xml.apache.org/batik/>, July 2004

[Berlin et al. 1969]

Berlin, Brent; Kay, Paul: *Basic Color Terms: Their Universality and Evolution*, University of California Press, Berkeley, Los Angeles, 1969

[Campin et al. 2003]

Campin, M. Benjamin; Brunet, Louis L.; McCurdy, William A.; Siekierska, Eva M.: *SVG Mapping for People with Visual Impairment*, SVG Open 2003, <http://www.svgopen.net/2003/papers/svgmappingforpeoplewithvisualimpairments/>, July 2004

[FreeTTS]

Kwok, Philip; Lamere, Paul; Schröder, Marc; Vos, David; Walker, Willie: *FreeTTS - A speech synthesizer written entirely in the Java programming language*, <http://freetts.sourceforge.net/>, July 2004

[Gardner et al. 2001]

Gardner, John A.; Bulatov, Vladimir: *Smart Figures, SVG, and Accessible Web Graphics*, Proceedings of the Technology And Persons With Disabilities Conference, 2001

[Gardner et al. 2004]

Gardner, John A.; Bulatov, Vladimir: *Directly Accessible Mainstream Graphical Information*, In: Klaus, Joachim; Miesenberger, Klaus; Burger, Dominique; Zagler, Wolfgang (Eds.): *Proceedings of the 9th International Conference on Computers Helping People with Special Needs (ICCHP)*, Springer, Berlin, 2004

[Herman et al. 2002]

Herman, Ivan; Dardailler, Daniel: *SVG Linearization and Accessibility*, Proceedings of the SVG Open Conference 2002, http://www.svgopen.org/2002/papers/herman_dardailler_svg_linearization_and_accessibility/, 2002

[Jackson 2003]

Jackson, Dean: *Making Accessible SVG*, SVG Open 2003, http://www.svgopen.net/2003/paperAbstracts/making_accessible_svg.html, July 2004

[JDOM]

JDOM Project: *JDOM*, <http://www.jdom.org/>, July 2004

[McCathieNevile et al. 2002]

McCathieNevile, Charles; Koivunen, Marja-Riitta: *Accessibility Features of SVG*, W3C Note, 2002, <http://www.w3.org/TR/SVG-access/>, July 2004

[McLaren et al. 1976]

McLaren, K.: *The development of the CIE 1976 (L*a*b*) uniform colour space and colour-difference formula*, Journal of the Society of Dyers and Colourists 92, 1976

[Rotard et al. 2004]

Rotard, Martin; Otte, Kerstin; Ertl, Thomas: *Exploring Scalable Vector Graphics for Visually Impaired Users*, In: Klaus, Joachim; Miesenberger, Klaus; Burger, Dominique; Zagler, Wolfgang (Eds.): *Proceedings of the 9th International Conference on Computers Helping People with Special Needs (ICCHP)*, Springer, Berlin, 2004

[WAI]

W3C: *Web Accessibility Initiative*, <http://www.w3.org/WAI/>, July 2004