Artistic resizing: a technique for rich scale-sensitive vector graphics
Pierre Dragicevic, Stéphane Chatty, David Thevenin, Jean-Luc Vinot

To cite this version:
Artistic Resizing: A Technique for Rich Scale-Sensitive Vector Graphics

Pierre Dragicevic  
University of Toronto  
dragice@dgp.toronto.edu

Stéphane Chatthy  
IntuiLab  
chatty@intuilab.com

David Thevenin  
IntuiLab  
thevenin@intuilab.com

Jean-Luc Vinot  
DSNA / DTI / R&D  
vinot@cena.fr

Artistic Resizing’s inference algorithm is based on a simple bivariate geometry interpolation technique we call orthogonal interpolation. It requires a set of graphic groups (variants) that share the same structure and have a bounding box. Each local affine transformation is extracted and interpolated independently from the others. The first line of the matrix is linearly interpolated along the width of the bounding boxes, whereas the second line is interpolated along their height. The assumption is that horizontal (resp. vertical) resizing only results in horizontal (resp. vertical) motions, i.e., translations, scales and shears. On more than two examples, two monovariate piecewise linear interpolations can thus be applied, eliminating the need for multivariate techniques.

Visual quality is given more and more importance in graphical user interfaces. This points out the need for new methods and tools to effectively involve graphic designers into GUI design and development teams. As a first step in this direction, IntuiKit [Chatty et al. 2004] showed how rich vector graphics could be exploited to combine the visual expressivity of authoring tools such as Adobe Illustrator with the behavioral expressivity of programming toolkits.

Converting static visuals into fully dynamic user interfaces still raises a number of issues. One of them is resizing. The only known technique for resizing vector graphics is uniform scaling. Such a method does not guarantee visual readability for all sizes and aspect ratios. Besides, user interfaces commonly show non-uniform scaling behaviors. When a button is resized for example, invariants are maintained such as label size and centering. Such behavior is traditionally modeled as a set of constraints which can be either coded by a programmer (as it is the case in most toolkits) or inputted to a constraint-solving system. But specifying constraints is hard, if not impossible, on rich and arbitrarily-structured vector graphics, even with declarative or visual approaches.

In spite of their familiarity with media more stable than interactive displays, graphic designers are nevertheless acquainted with the notion of adaptation. For example, a logotype can be used on different media, at different sizes, or on different color backgrounds. Designers often provide several variants of their artwork. On the left, a dock using Artistic Resizing. The graphic document can be loaded into the IntuiKit interpreter and tested at any time, allowing it to be iteratively enriched until the designer is satisfied by the visual appearance at intermediate and extreme sizes and aspect ratios. Animated examples can be found at http://www.intuilab.com/artresize

Figure 1: On the left, vector graphics with non-uniform resizing: the 2nd and 4th variants have been drawn in Illustrator, the other ones are interpolated. On the bottom right, a dock using Artistic Resizing.

Acknowledgements

Thanks to Michel Beaudouin-Lafon, Sylvie Athènes, Yves Rinato, Stéphane Conversy, Sandra Basnyat, Frédéric Jourdan, Céline Schlüeter and Alexandre Lemort for their help.

References


