SVG Rendering of Digital Images: an Overview

Sebastiano Battiato, Gianpiero Di Blasi,
Giovanni Gallo, Salvo Nicotra
Dipartimento di Matematica ed Informatica
Università di Catania - Viale A. Doria 6 – 95125
Catania Italy
{battiato, gdiblasi, gallo, snicotra} @dmi.unict.it

Giuseppe Messina
STMicroelectronics – AST Catania Lab Imaging
Group - FAB. M6, Contrada Blocco Torrazze,
Casella Postale 421, 95121 Catania, Italy
giuseppe.messina@st.com

ABSTRACT

The SVG (Scalable Vector Graphics) standard allows representing complex graphical scenes by a collection of graphic vectorial-based primitives, offering several advantages with respect to classical raster images such as: scalability, resolution independence, etc. In this paper we present a full comparison between some advanced raster to SVG algorithms: SWaterG, SVGenie, SVGWave and Vector Eye. SWaterG works by a watershed decomposition coupled with some ad-hoc heuristics, SVGenie and SVGWave use a polygonalization based respectively on Data Dependent and Wavelet triangulation, while Vector Eye is a commercial tool. Final quality has been assessed over a large dataset of images both in terms of PSNR and compression ratio.

Keywords
SVG, Triangulation, Watershed, Wavelet, Vectorialization.

1. INTRODUCTION

SVG is a language for describing two-dimensional graphics and graphical applications in XML ([6], [10]). In this work we are interested in reviewing existing techniques devoted to cover the gap between the graphical vectorial world and the raster world typical of digital photography. SVG format could find useful application in the world of mobile imaging devices, where typical camera capabilities should match with limited color/size resolutions displays. The core of the current SVG developments is the version 1.1. Actually the SVG 1.2 specification is under development and available in draft form. Major details can be found directly at the W3C site [12]. An exhaustive overview of the recent SVG development and related applications can be found in the proceedings of SVG Open Conference [12].

Two advanced techniques SVGenie [3] and SVGWave [5] have been applied to approximate local pixel neighborhood by triangles: the Data Dependent Triangulation (DDT) ([7]), the Wavelet Based Triangulation (WBT) ([9]).

The DDT replaces the input image with a set of triangles according to a specific cost function able to implicitly detect the edge details. The overall perceptual error is then minimized choosing a suitable cost function able to simplify triangulation. Recently further optimization of such function has been introduced for Colour Filtering Array demosaicing ([11]) and for image interpolation. The WBT uses the wavelet multilevel transformation to properly extract the details from the input images; a reverse process of triangulation, starting from the lowest level, is applied to derive the final WBT. A triangulation is, by first, achieved at the lowest level, introducing large triangles; then the process is refined by iterating for each level, the level details of each single triangle, according to the wavelet transformation. Both these decomposition could be directly managed by SVG primitives. Although the quality achieved in this way is rather good the size of the resulting files may be very large. The triangulation of both DDT and WBT are then processed by the polygonalization. The main purpose of this step is to minimize the dimensions of the resulting files by merging triangles together, according to specific similarity metrics, reducing the amount of perceptual redundancies. Another recent approach uses a raster-to-vector technique SWaterG ([4]) by advanced watershed decomposition ([15]) coupled with some ad-hoc heuristics devoted to obtain high quality rendering.

POSTERS proceedings ISBN 80-903100-8-7
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.
Copyright UNION Agency – Science Press

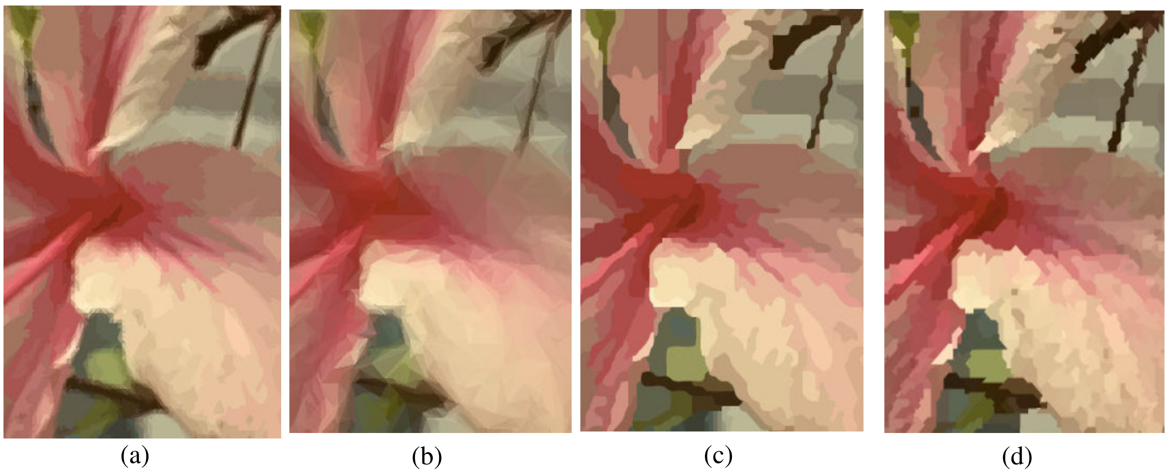
Figure 1**: A comparison between some advanced SVG rendering (scale 4:1):** (a) **SVGenie,** (b) **SVGWave,** (c) **Vector Eye,** (d) **SWaterG.**

The system is composed by two main steps: first, the image is partitioned into homogeneous and contiguous regions using watershed decomposition. Secondly, an SVG representation of such areas is achieved by ad-hoc chain code building.

In [1] a conversion tool able to obtain SVG representation of Geomatics data has been presented but the overall results are limited by the small set of SVG primitives used for the final rendering (e.g. only <rect>). Recently, some commercial and free softwares have been developed using some ad-hoc solution to the "raster to SVG" problem. Among other: Autotrace ([2]), Kvec ([8]), Vector Eye ([14]). Almost all software are devoted to SVG rendering of graphic images (e.g. clip art), showing in such case good performances but also several perceptual drawbacks when applied to digital pictures acquired by consumer devices.

2. EXPERIMENTS & CONCLUSIONS

A large set of experiments have been performed in order to evaluate the capabilities of the various techniques and comparing results between them. According to our results Vector Eye [14] has the best performances among commercial software, although only advanced solution ([4], [5]) are able to obtain high quality rendering. This is particularly true also considering for comparison both PSNR and compression ratio between original raster images and final SVG file (also gzipped). Figure 1 shows an image that has been vectorized with three advanced methods and with Vector Eye (i.e. magnified at 400%). Major details, on line demo and results can be found at the following web address: http://svg.dmi.unict.it/.

3. REFERENCES

[1] B. Antoniou, L. Tsoulos, *Converting Raster Images to XML and SVG*. In Proc. of SVGOpen, 2004

[2] Autotrace, Convert Bitmaps to Vector Graphics http://autotrace.sourceforge.net/, 2004

[3] S. Battiato, G. Gallo, G. Messina, *SVG Rendering of Real Images Using Data Dependent Triangulation*. In Proc. of ACM/SCCG2004, 2004

[4] S. Battiato, A. Costanzo, G. Di Blasi, G. Gallo, S. Nicotra, *SVG Rendering by Watershed Decomposition*, In Proc. of SPIE EI - Internet Imaging VI - Vol.5670.3, 2005

[5] S. Battiato, G. Barbera, G. Di Blasi, G. Gallo, G. Messina, *Advanced SVG Triangulation Polygonalization of Digital Images*, In Proc. of SPIE EI - Internet Imaging VI, Vol.5670.1, 2005

[6] D. Duce, I. Herman, B. Hopgood, *Web 2D Graphics File Format*. Computer Graphics *forum* - Vol.21(1), pp.43-64, 2002

[7] N. Dyn, D. Levin, S. Rippa, *Data Dependent Triangulation for Piecewise Linear Interpolation*. IMAJ. Numerical Analysis, Vol.10, 1990

[8] KVec, Raster Vector Conversion: http://www.kvec.de, 2004

[9] S. Lee, *Wavelet-Based Multiresolution Surface Approximation from Height Fields*, Ph.D. Thesis, Virginia State University, 2002

[10] A. Quint, *Scalable Vector Graphics*. IEEE Multimedia - Vol. 3, pp. 99-101, 2003

[11] D. Su, P. Willis, *Demosaicing of Color Images Using Pixel Level DDT*. In Proc. of IEEE Theory and Practice of Comp. Graph., pp. 16-23, 2003

[12] Scalable Vector Graphics – XML Graphics for the Web – http://www.w3c.org/Graphics/SVG, 2004

[13] SVG Open Conf. and Exhibition, Conf. on Scalable Vector Graphics, http://www.svgopen.com, 2004

[14] Vector Eye – Raster to Vector Converter - http://www.siame.com/index.html, 2004

[15] L. Vincent, O. Soille, *Watersheds in Digital Spaces: an Efficient Algorithm Based on Immersion Simulations*. IEEE Trans. on PAMI, Vol.13(6), pp. 583-598, 1991