Advanced SVG Triangulation/Polygonalization of Digital Images

S. Battiato[1], G. Barbera[1], G. Di Blasi[1], G. Gallo[1], G. Messina[2]
[1]Università di Catania, D.M.I. Viale A. Doria 6, 95125 Catania (Italy)
[2]STMicroelectronics – AST Catania Lab - FAB. M6, Contrada Blocco Torrazze, Casella Postale 421, 95121 Catania (Italy)

ABSTRACT

The paper presents two techniques to convert raster images in a Scalable Vector Graphics format. The first method is a technique that generates a Data Dependent Triangulation just by swapping the edges generated by a regular triangulation; the second technique is a Wavelet Based Triangulation and generates a multi-level triangulation based on the data collected from the wavelet multi-level transformation. After triangulation step a further reduction is introduced by a suitable polygonalization step. The proposed techniques have been compared with other raster to vector conversions in terms of both perceptual and measured quality.

Keywords: SVG, Triangulation, Watershed, Wavelet, Polygonalization.

I. INTRODUCTION

The SVG (Scalable Vector Graphics) ([4], [10], [14]) standard allows to represent complex graphical scenes by a collection of vectorial-based primitives, offering several advantages compared to classical raster images such as: scalability, resolution independence, etc. In this work we are interested in finding some heuristic techniques to cover the gap between the graphical vectorial world and the raster real world typical of digital photography in a specific application; SVG format could find useful applications in the world of mobile imaging devices, where typical camera capabilities should match with limited color/size resolutions displays.

Two different techniques have been applied to approximate local pixel neighborhood by triangles: the Data Dependent Triangulation (DDT) ([5]) and the Wavelet Based Triangulation (WBT) ([8]). 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 triangulation. Recently further optimization of the cost function has been introduced for Color Filtering Array demosaicing ([13]) and for image interpolation ([18]). On the other hand the DDT is strictly connected to the original pixel positions; therefore the number of actual triangles is larger than the number of pixels.

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 achieve the WBT. In other words, a triangulation is achieved, by first, at the lowest level introducing large triangles; then the process is refined by iterating for each level, the details of each triangle, according to the wavelet transformation. In this way it is possible to increase the quantity of small triangles into the texturized areas, fixing the amount of large triangles into the smooth areas.

These triangulations 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 resulting triangulation of both DDT and WBT are then processed by the most important step of the described approach: the polygonalization. The main purpose of this function is to minimize the dimensions of the resulting files by merging triangles together, using specific similarity metrics and reducing the amount of redundancies present into the SVG files.

SVG has been used for the same purpose as here described with software like Vector Eye [17] , Autotrace [1], Ras2Vec [12], Potrace [11], and SWaterG [3].

The paper is structured as follows. The section II presents the SVG standard whereas sections III and IV introduce the details of triangulation by using respectively the DDT and the WBT techniques. The successive section introduces the polygonalization. A final section closes the paper showing experimental results and comparisons with other techniques tracking directions for future works.

II. SVG - SCALABLE VECTOR GRAPHICS

SVG standard is a World Wide Web Consortium (W3C) recommendation designed for many purposes (e.g. advertising, clip art, business presentations, etc.) also using advanced features like animation and filter effects. The language describes 2D-graphics and graphical applications in XML, and is mainly devoted to a series of application where scalability in terms of output resolution, color spaces, and available bandwidth is required.
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. SVG Mobile Profiles (Basic and Tiny) targeted to resource-limited devices, are part of the 3GPP platform for third generation mobile phones. Finally SVG Print is a set of guidelines to produce final-form documents in XML suitable for archiving and printing. SVG is a drawing tool featuring all the basic vector graphics primitives. These include lines, polylines, polygons, rectangles, circles, and ellipses. Beside these basic shapes, SVG also supports paths, including cubic and quadratic Bezier curves and elliptical arcs. It is also possible to specify transformations on objects either as an additive instruction (scale, translate, rotate, or skew) or as a global transformation matrix. Major details can be found directly at the W3C site [14] whereas in ([4], [10][9]) the overall SVG capabilities are summarized.
It is also possible to find an exhaustive overview of the recent SVG development and related applications in proceedings of SVG Open Conference [15].

III. DDT – DATA DEPENDENT TRIANGULATION

A triangulation is a partition of a two-dimensional plane into a triangles set. The triangulation techniques which try to optimize the approximation of a particular function, also taking advantage from the information obtained from the co-domain, are named Data Dependent Triangulation (DDT).
The optimum triangulation research, by considering a fixed criterion, is a not well posed problem; however good results are obtained by considering a locally optimal criterion. A triangulation is called "Optimal" if it is impossible to obtain a further improvement by swapping any edges.
The algorithm introduced by Lawson [7] is an iterative technique able to find a locally optimal triangulation. This approach inspects all the inner edges and exchanges those that reduce the total cost of the triangulation; the algorithm iterates the process until the total cost is still unchanged.
Another approach, based on Lawson's method and developed by Yu, Morse and Sederberg [18], has been used as starting point to develop the proposed approach:

1. Every pixel of the image is split into two triangles by using the diagonal with the lowest cost;
2. The obtained triangulation is furthermore adapted to the image data by executing *step 3* in an iterative process, which is stopped when a locally optimal triangulation is reached;
3. If the polygon formed by adjacent triangles is convex, the following look-ahead step is achieved:
   a. If the exchange introduces a reduction of the edges cost sum, then the diagonals are swapped;
   b. Otherwise, for every edges E_i of the polygon, with $i=2..5$, if the exchange of E_1 with its opposite diagonal E_1', and the simultaneous swapping of the edge E_i with its opposite diagonal E_i' decrease the costs of the edges of the polygon and the costs of the edges $E_6,\dots, E_{13}$ of the adjacent triangles, then both exchanges are achieved (See Figure 1).

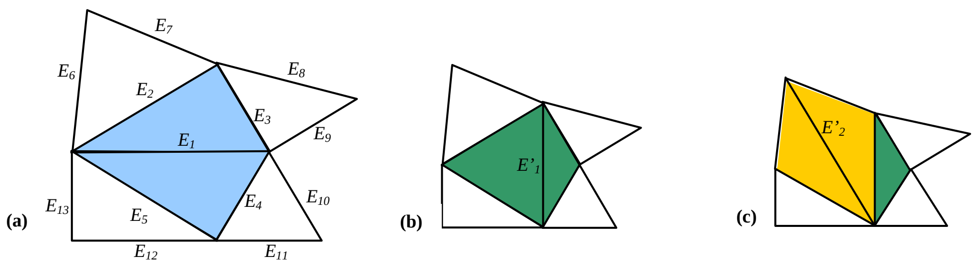


Figure 1 - Example of look-ahead **(a)** Initial configuration, **(b)** Exchange of E_1 with the opposite diagonal E'_1. **(c)** Simultaneous exchange of E_1 with E'_1 and E_2 with E'_2.

To estimate the triangulation, the optimality criterion is fixed using the following cost function:

$$Cost(E) = |\nabla P_1| \cdot |\nabla P_2| - \nabla P_1 \cdot \nabla P_2 \quad (1)$$

where E is a common edge of two triangles and $\nabla P_i = (a_i, b_i)$ are the gradient of the planes P_i, that are the interpolating linear functions of the triangles. In our case, to further speed-up the overall process we used the following approximation:

$$cost(e) = (|a_1| + |b_1|) \cdot (|a_2| + |b_2|) - (a_1 \cdot a_2 + b_1 \cdot b_2) \quad (2)$$

that is cost effective. Such cost function is similar to the simplified cost function used in [5] to describe the roughness of triangulated terrain (Sobolev seminorm). Further investigation will be devoted to compare the cost function (2) with alternative approaches as described in ([5], [13] and [18]).

The global triangulation cost is summarized in:

$$cost(T) = \sum_{\forall t_1, t_2 \text{ contiguous in } T} \left[(|a_1| + |b_1|) \cdot (|a_2| + |b_2|) - (a_1 \cdot a_2 + b_1 \cdot b_2) \right] \quad (3)$$

IV. WBT – WAVELET BASED TRIANGULATION

The Wavelet Based Triangulation algorithm (WBT) generates a triangulation which approximates an image, taking advantage from the characteristics given by the multi-resolution analysis carried out by the discrete wavelet transformation. The wavelet transformation permits to extract the frequencies coefficients from a signal, supplying a decomposition of the signal with a hierarchical set of approximations and proper details. Due to the wavelet locality property, the decomposition coefficients provide information about the inner details of restricted regions of the image.
Once the wavelet decomposition coefficients until the level l_i, where $i=1..l$ and l is the total number of iterations, have been obtained (e.g. using a standard bi-orthogonal model) the successive steps to generate the triangulation are:

- Create a initial triangulation using predefined structures and the wavelet coefficient of each level l_i;
- For each level in the range from the first l_i to the final level l_f
  - Iterate *n* time the following steps:
    - Select the triangles with a higher energy detail,
    - Refine the selected triangles;
    - Adapt all the triangles, finding the good trade-off between the error and the Delaunay's criterion;
    - Adapt all the vertices with a neighborhood low energy detail;
    - Decimate the shorter edges;
    - Decimate the unnecessary vertices;
    - Repeat the triangle adaptation step.

The image is then subdivided into a regular region grid tacking into account the current resolution and the initial level. Each predefined model represents the better way to achieve a triangulation of a simple region, due to the presence of horizontal, vertical, diagonal or no predominant discontinuities. By using the wavelet coefficients it is possible to discriminate among the aforementioned cases and building, region by region, the initial triangulation of the image. To improve the reconstruction quality of the image, maintaining the complexity of the triangulation, local operators are then applied. At each level the resolution of the grid is increased and each triangle belongs to several regions (the triangle covers a region if the center of the region belongs to the triangle).

The refining operation increases the number of triangles into the highly texturized regions and reduces the number of triangles into the less texturized regions. In order to achieve the refinement, the triangles containing regions with a energy detail higher than a fixed threshold,

$$max_{energy}(T) > \delta_e \quad (4)$$

are subdivided. The subdivision of the triangle T is achieved considering the direction of the possible discontinuity: the region with the greatest energy is considered if its center belongs to the area of T, that means it do not belongs to the boundaries; T is then subdivided into three smaller triangles, as shown in Figure 2a.

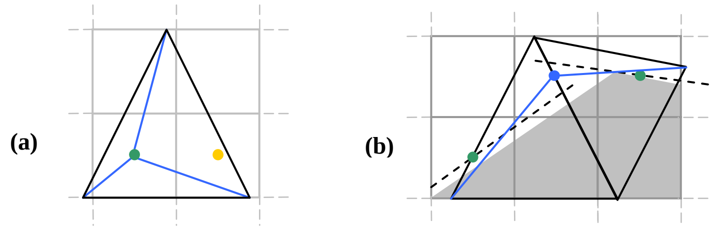


Figure 2 Examples of Triangle subdivisions (a) Subdivision into three triangles, (b) Joined Subdivision for center belonging on boundaries.

If the center of the region belongs to the boundary of the triangle T, adjacent to the triangles T', the subdivision is accomplished jointly to T', in function of the common edge to which the center belongs (Figure 2b). Two triangles are joinable if the direction of the discontinuity, for both the triangles, forms a convex angle with the common edge, and this angle is higher than $\left(\frac{1}{2}\pi - arccos(\delta_p)\right) \in \left[\frac{1}{6}\pi, \frac{1}{2}\pi\right]$, where δ_p is a fixed coupling parameter. If both the single triangle subdivision and the coupled triangles subdivision could not be achieved, the simple subdivision using the barycenter of the triangle is accomplished.
The first adjustment operator performs exchange among the edges by using the Delaunay's criterion of optimality and a suitable error measure. The Delaunay's criterion estimates as locally optimal those triangles wherein the minimum angle present is also the maximum angle possible. As measure of the inner triangle error the Mean Square Error (MSE) is used. The MSE is calculated on each pixel of the image whose center belongs to the considered triangle. These two classes of goodness are then averaged through a weighted mean by using the parameter α.
For each triangle T, considering the polygons formed with every adjacent triangles of T, all the possible exchanges of the diagonals are evaluated and the combination with the lower cost is performed (see Figure 1).
The second adjustment operator aims to move the vertices by using the region centers as attraction points. If a vertex belongs to an area without details, it will be moved, in a fixed neighborhood, to the region center with the maximum rate between the energy detail and the square distance between the vertex itself and the center. Using this technique the vertices will be condensed into the area of the image with larger discontinuities, thus enhancing the characterization of the edges (Figure 3).

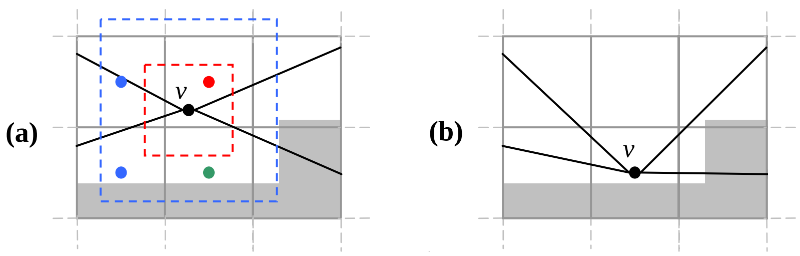


Figure 3 - Displacement example of a vertex v. (a) The only region center belonging to the red area has not enough detail energy to satisfy the displacement constraints; also in the blue area the only center with such constraints is the green one. (b) Vertex v after the displacement.

Other used local operators are the edge removal operator and the vertex removal operator.
An edge can be removed if it is shorter than a fixed threshold value. If an edge must be removed its vertices are removed and replaced by the middle point (Figure 4); if it is impossible to use the middle point, one of the vertices of the removed edge is used.

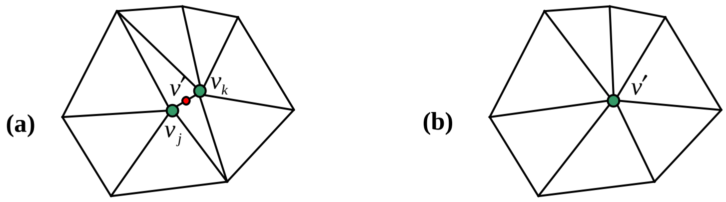


Figure 4 - Example of edge removal. (a) The edge $E=(v_j,v_k)$ is shorter than a fixed minimum threshold; then the vertex v' is fixed as a new vertex for *E*. (b) Structure after the removal of *E* and substitution of the old vertices v_j and v_k with v'.

To achieve a further reduction of the triangulation complexity, it is possible to find and remove vertices with redundancies. By considering the intensity values of the vertices as the third spatial coordinate, each triangle is set up with a particular orientation given by the normal to its surface. If the triangulation is consistent each normal to the triangles will have outgoing directions. A vertex belonging to a set of triangles with similar orientation, in function of a fixed criterion, is redundant, therefore it can be removed. The used criterion is:
Given a vertex *v* which is associated to the triplet (c^1,c^2,c^3) of intensity values, it can be removed if

$$\forall v_i \mid \exists E=(v_i,v) \text{ then } \forall j=1,...,3, \; \left|c_i^j - c^j\right| \le \tau_j \tag{5}$$

where τ_1, τ_2 and τ_3 are the thresholds for each component of the triplet.
Once the vertex has been removed a new triangulation of the polygon, formed by the union of the triangles connected to the removed vertex, must be generated (Figure 5).

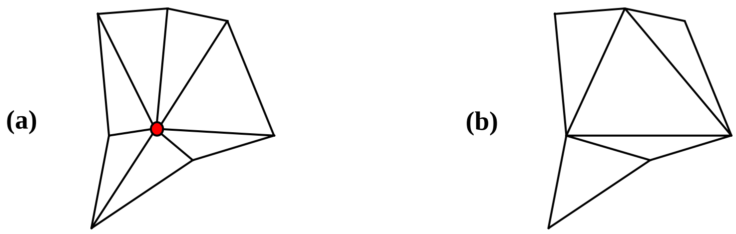


Figure 5 - Example of vertex removal. (a) The red vertex is a candidate to removal because of its similitude to the connected vertices. (b) Structure after vertex rejection and new triangulation.

Each adjustment and removal operation must be controlled before the execution due to possibility of lacks into the final triangulation.

V. POLYGONALIZATION

The polygonalization algorithm, starting from the optimal triangulation, inspects every triangle looking for similarity with one of the adjacent polygon or triangle to achieve the fusion (Figure 7).
The measure of similarity is achieved in terms of intensity values into a specific color space (i.e. the YC_bC_r space) because it contains a better liaison to the Human Visual System [9].
The polygonalization methods can be classified into two categories:

- Triangles-based merging;
- Polygons-based merging.

The proposed algorithm belongs to the Polygons-based merging class:

Given two distinct and adjacent triangles T_1 and T_2,

- if T_1 and T_2 do not belong to any polygon then the new polygon P, union of the two triangles, is added to the polygonalization;
- if T_1 does not belong to any polygon but T_2 belongs to the polygon P and T_1 is similar to P then T_1 is combined to P;
- if T_1 and T_2 belong to two different polygons P_1 and P_2 and those polygons are similar, then P_1 and P_2 are combined together and P_2 is removed from the polygonalization.

The Absolute Difference Separated Ratio (ADSR) is used to estimate the similarity between two elements. The ADSR is satisfied if

$$\forall j = 1,2,3 \quad \left| I_j(a) - I_j(b) \right| \leq \tau_j \cdot (max(I_j) - min(I_j)) \tag{6}$$

that is for each color intensity I_j the distance between the values a and b must be less or equal to the multiplication of the fixed threshold τ_j and the maximum difference.

It is also possible to reduce the dimension of the final SVG output file by considering further characteristics of the color space and of the polygonalization itself:

1. By using the color space YC_bC_r it is possible to give more relevance to the luminance channel Y ($j=1$) rather than to the chrominance channels C_b and C_r ($j=2$ and 3 respectively): then the ADSR is modified considering the values τ_j such that $\tau_1 \leq \tau_2 = \tau_3$;
2. The polygonalization can also be obtained by incremental fusion steps using different increasing thresholds, generating different levels of approximation: starting from zero, the values of τ_j are increased at each fusion step until the threshold value τ_1 reaches a fixed maximum approximation target τ; the difference between the values of τ_1, into two successive iterations, is fixed by the parameter *step*. At the same time the parameter $step_c$ fixes the respective increments of the thresholds τ_2 and τ_3; this parameter is estimated using the following, empirically estimated, function:

$$step_c = \left\lfloor \frac{\tau}{step} \right\rfloor + 1 \; ; \tag{7}$$

3. The boundaries of the single polygons can be simplified to minimize the number of vertices: if the vertices belong to the same straight line all these vertices, but the first and the last points, are removed (Figure 6);
4. The color of each element of the polygonalization is assigned in function of the colors of the respective vertices: in a simplest way the color of the single polygon or triangle is the mean value of the vertices colors. However in the case of polygon it is better to give a different weight to the internal vertices colors instead of the boundary vertices, in function of the following:

   Given n_i and n_e the number of inner and outer vertices respectively, $m_i^j = \frac{\sum_{v_i} c_i^j}{n_i}$, $m_e^j = \frac{\sum_{v_e} c_e^j}{n_e}$ the averages of inner and outer color components, and fixed the coefficient $\alpha \in [0,1]$, the color value of the element j of the polygon will be:

$$c_P^j = \frac{[n_i + \alpha(n_i - n_e)] \cdot m_i^j + [n_e + \alpha(n_e - n_i)] \cdot m_e^j}{n_i + n_e} \tag{8}$$

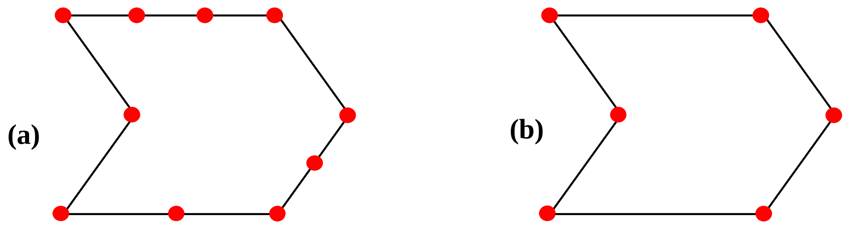


Figure 6 - (a) A Polygon with a not simplified perimeter (b) the same simplified Polygon. The red dots represent the outer edges.

VI. EXPERIMENTS AND CONCLUSIONS

Each triangle/polygon can be easily remapped into a SVG representation in the following way:

```
<path d="M x1,y1 L x2,y2 L x3,y3 Z" stroke="#FFFFFF" fill="#FFFFFF"/>
```

where `x1,y1, x2,y2, x3,y3` are the vertex coordinates (`xi,yi` for further vertices), and `#FFFFFF` is the filling color, obtained using the aforementioned equation (4). The generalization to RGB images is simply obtained, replacing `#FFFFFF` with `#RRGGBB`, where `RR, GG, BB` are respectively the hex representation of the red, green and blue mean value inside the considered triangle/polygon.

As objective quality metrics we use PSNR (peak signal-to-noise ratio); PSNR is very common in image processing (e.g. comparison between an original image and a coded/decoded image). Given a source image $f(i,j)$ that contains N by N pixels and a reconstructed image $F(i,j)$ where F is reconstructed by decoding the encoded version of $f(i,j)$, PSNR measured in decibels (dB) is computed by using:

$$PSNR = 20\log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \tag{9}$$

where MSE is the Mean Square Error between f and F, computed on pixel basis. Typical PSNR values range between 20 and 40.

The final quality of SVG rendering coming from raster images has been assessed through an extensive experimental phase. The thresholds, introduced into the polygonalization step, are fundamental to obtain a pleasant raster-to-vectorial conversion. These thresholds determine the percentage of similarity exploited to merge the polygon elements: a value of 0% means that the elements of each polygon are perfectly equal; a value of 100% of similarity means that the whole elements are merged together. Such percentage values fix the maximum and minimum ratio factor for quality over compression. The final target must reach a good trade-off between high quality and small dimensions. Each image is characterized by its own features, thus the perfect ratio between quality and compression should be fixed step by step, or the possibility to obtain a pleasant result could be compromise. The polygonalization created from a triangulation fixes the upper bounds in terms of quality and size: the polygonalization quality will never reach the quality given by its initial triangulation; likewise the dimension of the polygonalization will be always lower than the triangulation size. Due to the restricted space and to the large amount of measures achieved to fix the optimal values of the various thresholds, we will briefly analyse, for instance, the threshold τ. In Figure 10 the different output obtained, from Lena image, by varying the threshold τ, are illustrated. It is clearly visible the direct proportionality between τ and the compression factor and the inverse proportionality between τ and the image quality. Also in Figure 10.d the reduction of the size without compromising the quality, introduced by growing the chrominance threshold to τ_c=5%, is shown. Furthermore the PSNR measures (Figure 11), by reducing progressively τ, is slightly increased, but the visual difference is actually evident by looking to the images in Figure 10.a, c and d.

Experiments have been conducted both on a typical standard dataset of images [6] and on a database of images acquired by real handset imaging devices to which the proposed techniques are mainly oriented (Figure 12, 13 and 14). For sake of comparison, SVG images obtained using our techniques (DDT-Polygonalization, named SVGenie and WBT-Polygonalization, named SVGWave) have been compared to the technique in [3] (SwaterG) and to the results provided by the commercial software [17] (Vector EYE). Further experiments can be found in the SVG UniCT Group Page at http://svg.dmi.unict.it/.

Future works will investigate the conversion of the generated polygons, which are represented by unions of segments, into Bezier and/or B-splines curves.

REFERENCES

[1] Autotrace – Convert bitmaps to vector graphics http://autotrace.sourceforge.net/

[2] S. Battiato, G. Gallo, G. Messina, *SVG Rendering of Real Images using Data Dependent Triangulation*, In Proc. of ACM/SCCG2004, Spring Conference on Computer Graphic, 2004, Slovakia.

[3] S. Battiato, A. Costanzo, G. Di Blasi, G. Gallo, S. Nicotra, *SVG Rendering by Watershed Decomposition*, To appear in Proceedings of SPIE Electronic Imaging 2005 - Internet Imaging VI - Vol.5670.3 - San Josè, USA, January 2005;

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

[5] N. Dyn, D. Levin, S. Rippa, *Data Dependent Triangulation for Piecewise Linear Interpolation,* IMAJ. Numerical Analysis, vol.10 (1990), pages 137-154.

[6] Kodak's Photo CD system: ftp://www.cipr.rpi.edu/pub/image/still/KodakImages/

[7] C.L. Lawson, *Software for C1 Surface Interpolation,* Mathematical Software III, J.R. Rice, Academic Press, pages 161-194, New York 1977.

[8] S. Lee, *Wavelet-Based Multiresolution Surface Approximation from Height Fields*, Ph.D. Thesis, Virginia Polytechnic Insitute and State University, Blacksburg, February 2002.

[9] M. J. Nadenau, S. Winkler, D. Alleysson, M. Kunt, *Human vision models for perceptually optimized image processing - a review*, submitted to Proceeding of the IEEE, (http://dewww.epfl.ch/~nadenau/), September 2000;

[10] A. Quint. *Scalable Vector Graphics*. IEEE Multimedia vol.3 (2003) pages 99-101

[11] Potrace - Transforming bitmaps into vector graphics: http://potrace.sourceforge.net/

[12] Ras2Vec - Raster to vector conversion program http://xmailserver.org/davide.html

[13] D. Su, P. Willis. *Demosaicing of Color Images Using Pixel Level Data-Dependent Triangulation*. In Proceedings of IEEE Theory and Practice of Computer Graphics (2003) pages 16-23.

[14] Scalable Vector Graphics (SVG) – XML Graphics for the Web – http://www.w3c.org/Graphics/SVG (2003).

[15] SVG Open 2005 Conference and Exhibition, *4th Annual Conference on Scalable Vector Graphics Enschede,* The Netherlands - August 15-19, 2005, www.svgopen.com.

[16] P. Trisiripisal, *"Image Approximation Using Triangulation",* Virginia Polytechnic Institute and State University, Blacksburg, May 2003.

[17] Vector Eye – Raster to Vector Converter - http://www.siame.com/index.html (2003).

[18] X. Yu, B.S. Morse, T.W. Sederberg. *Image Reconstruction Using Data Dependent Triangulatio*n, Journal IEEE Computer Graphics and Applications, vol.21 (3) (2001) pages 62-68.

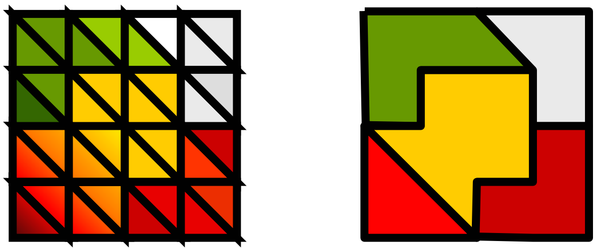
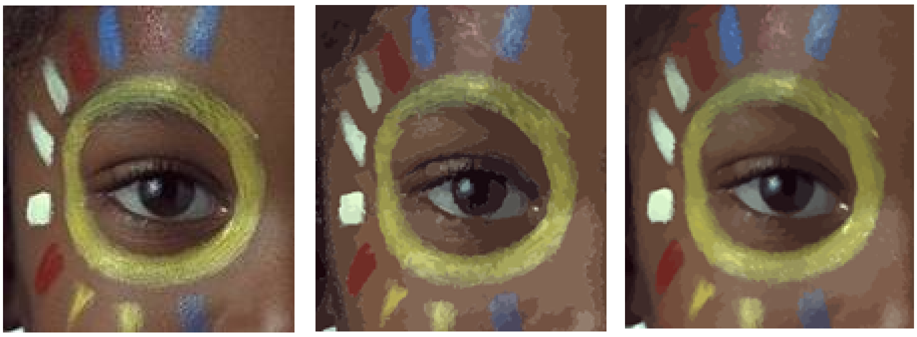


Figure 7 – Triangle Merging obtained using an ad-hoc similarity metric.



Original
Size: 126,3 Kb [24 bpp]
Compressed (jpeg):
39,7 Kb [7,5 bpp]

Vector Eye
Size: 215,8 Kb [41 bpp]
Compressed (Gzip):
69,3 Kb [13,2 bpp]

DDT- Polygonalization
Size: 188,3 Kb [35,8 bpp]
Compressed (Gzip):
51,4 Kb [9,8 bpp]

Figure 8 – A magnified detail of picture shown on the left, rendered respectively with VectorEye, using high quality settings (on the center) and our approach (on the right). Perceptive quality of DDT-Polygonalization (SVGenie) approach is sensibly improved. Also the comparison in terms of compression size is competitive.

(a)

(b)

(c)

(d)

Figure 9: SVG rendering of two standard images Kodim01 a), Kodim08 b) using the Wavelet based approach (SVGWave) together with a magnified details (4:1) c), d).

(a)

(b)

(c)

(d)

Figure 10. Output examples by varying the τ threshold . (a) $\tau=\tau_c$ =10%;. Bpp=6,1; PSNR=22,0. (b) $\tau=\tau_c$ =5%; Bpp=22,6; PSNR=24,2. (c) $\tau=\tau_c$ =2,5%; Bpp=72,6; PSNR=25,1. (d) τ=2,5%; τ_c=5%; Bpp=58,9; PSNR=25,0.

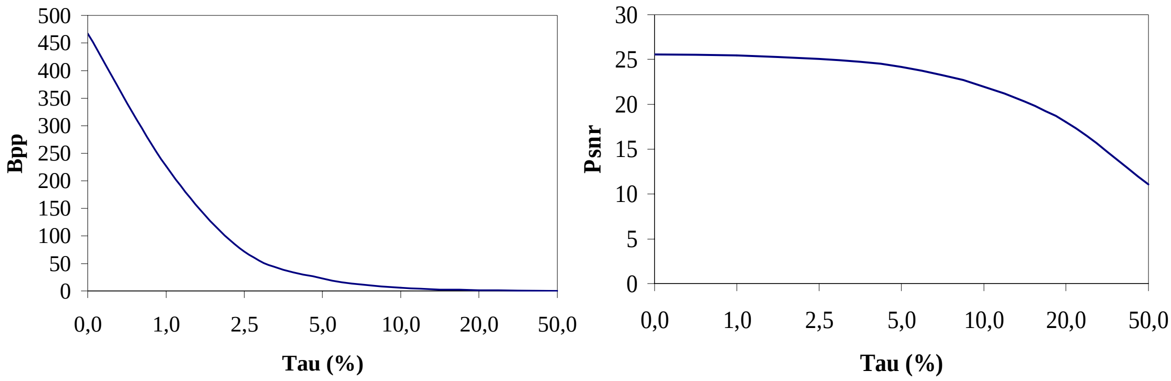
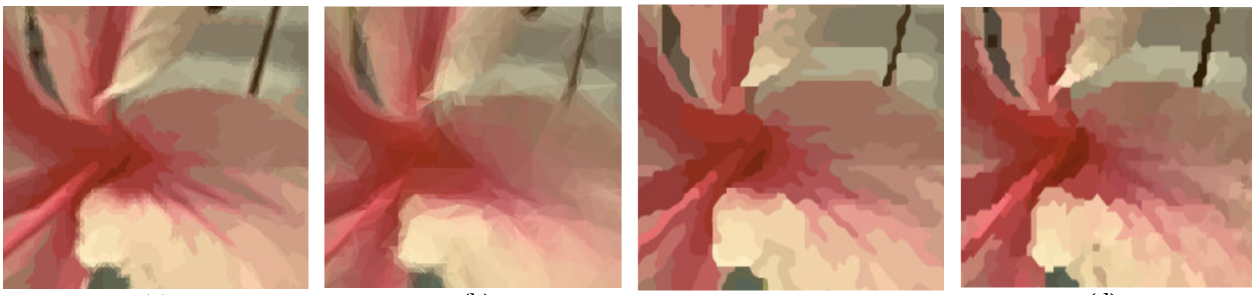
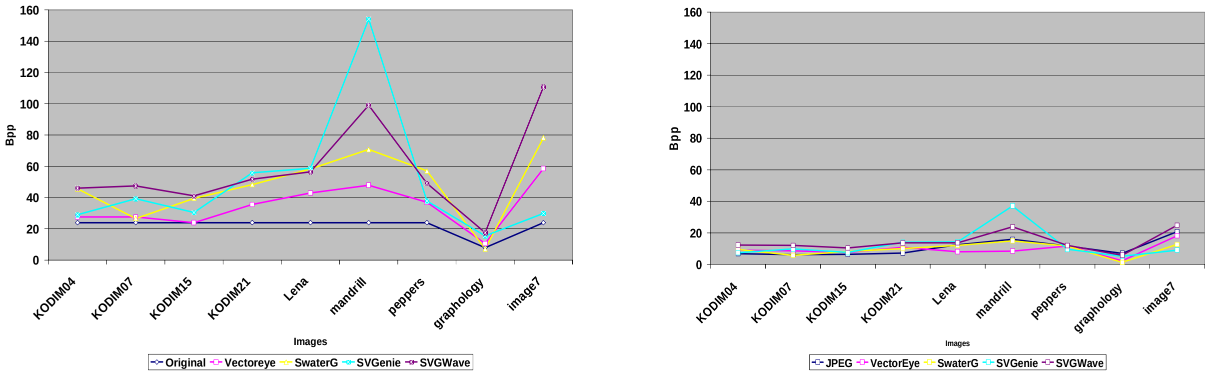


Figure 11. Variation of the SVG file size, expressed in Bits per pixels (Bpp), and variation of the PSNR by changing the value of τ ($\tau_c = \tau$).



(a) (b) (c) (d)

Figure 12: A comparison between the proposed method and other methods (scale 4:1) **(a)** SVGenie, **(b)** SVGWave, **(c)**Vector Eye [17], **(d)** SWaterG [3].



(a) (b)

Figure 13: A comparison of the bpp (bit per pixels) values obtained by the proposed method and other methods using (a) the uncompressed SVG file format and (b) the compressed SVGZ file format. The Jpeg format is also inserted into the graph (b) to compare the compression rate.

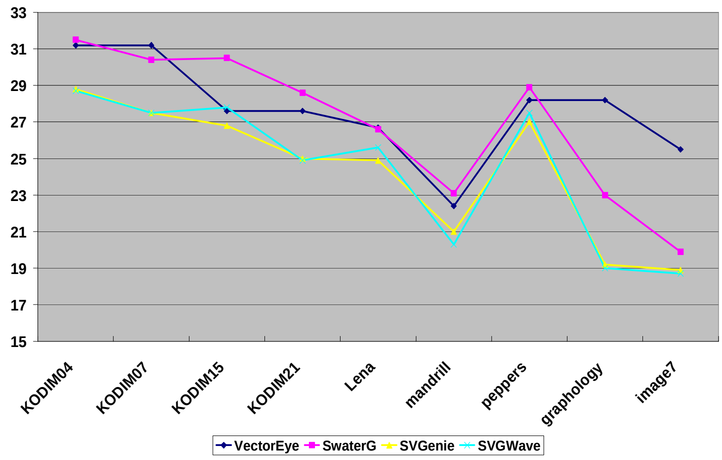


Figure 14: A comparison of the PSNR (Peak Signal Noise Ratio) values obtained by the proposed methods and the other methods.