

# A new way of Rich Image Representation (VectorPixels)

Alain Simons, Edmond Prakash and James Wood

University of Bedfordshire

University Square

UK LU1 3JU, Luton,

United Kingdom

Alain.Simons@beds.ac.uk | Edmond.Prakash@beds.ac.uk | Jim.Wood@beds.ac.uk

## ABSTRACT

Generating images on computer systems happens for decades on the basis of the same technologies. On the one hand you have bitmap and on the other hand vector technology. Bitmap or pixel technology is mostly used for the representation of rich color images. The other available imaging technology, vectors is mainly used to present logos and drawings without photo realistic details.

As known both have their disadvantages and advantages. In recent years a lot of research has been done to combine the advantages of both techniques in a comprehensive solution. Much research focuses on giving a vector illustration a realistic picture look. These studies usually use bitmap technology on top of vector technology or vice versa.

There was no development from scratch and that is exactly what this paper wants to propose, an algorithm to invent the pixel again. By always applying new developments on existing technologies generating images became a mix of various techniques and a rather complex matter.

With the use of VectorPixels (VP) the concept of a pixel will be redefined. In contrast with a pixel (Picture Element) a VectorPixel will be defined by a mathematical description and resolution independent.

## Keywords

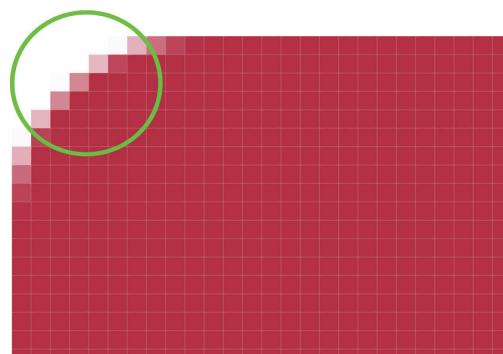
Pixel, VectorPixel, image description, resolution independent, enlargement.

## 1. INTRODUCTION

Every day people are looking at images on a personal computer, tablet or mobile device. Programs used on these computer systems are becoming more complex and the use of images most specific on websites has increased significantly in recent years. Instead of working on a wired network, computer system users are using more often wireless networks. Wireless networks are becoming faster but a reduction of data transmission enable a more fluent browsing. VectorPixels would also minimize this problem by

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

using a reduced file size. This can be done by using Scalable Vector Graphics (.svg) to store traditionally files. The .svg file format is supported by a lot of vector based software but delivers a large file when saving rich images.



**Figure 1. Jagged corner when using a bitmap solution when enlarging.**

A second issue is that making enlargements on bitmap images will always end pixelated (Figure 1). Lines for example will be jagged and a mixture of pixels with different values of shades will be generated. This technique works fine when an image



**Figure 2. Smooth vector based corner when enlarged.**

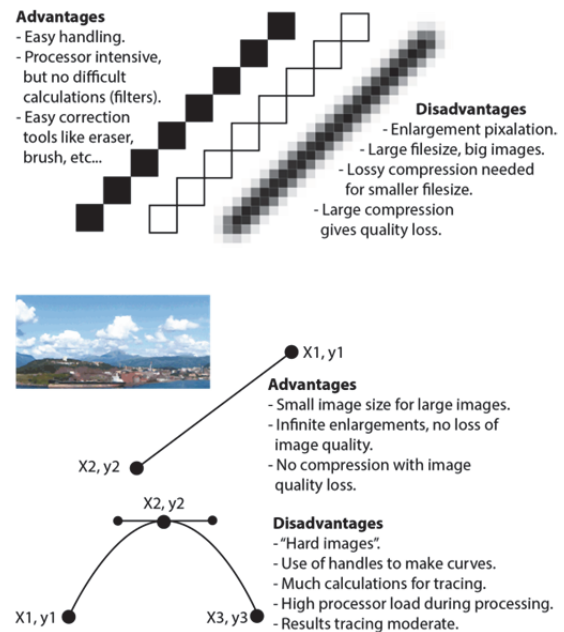
is viewed at 1:1 ratio and not enlarged. VectorPixels aims to provide a solution for this problem and achieve the same image quality as obtained with vectors (Figure 2) with a “Richer Image”. Vectors present a sharply defined shape but there is no “natural look” or smooth color transition. The new proposed approach wants to keep the advantages of both imaging technologies but get rid of their disadvantages in particular, lack of smoothness for vector-based solutions and pixelated enlargements for bitmaps (Figure 3).

## 2. RECENT SOLUTIONS

Research done by Orzan, Bousseau et al. [1] describe a new vector based tracing method for rendering smooth gradients from diffusion curves. This technique improves the smoothness of transition between shapes but not the overall quality of the image. Also in this case, image information is added that wasn't there before. This research is partly based on a paper published by Elder and Goldberg [3] in March 2001 which used edge maps for certain image adjustments.

Tracing is an important issue in creating VectorPixels. At a first stage there is a simple line tracing, secondly an Edge tracing is carried out. Edge tracing has been the subject of a lot of research. There are different kinds of Edge detection like Ant-Based Detection by Aydin [9] which uses a state transition function based on 5x5 edge structures. Sobel is another possibility a technique used by Kunal [10] which uses an edge mask generated by a fast edge detector. Generating VectorPixels is complex, given that different techniques have to be combined to get a result. These techniques are line tracing, edge tracing and generating the VectorPixel itself. Also a VectorPixel has to be positioned with

shape definition, fill, etc... Currently there is no solution that combines all these techniques with each other.



**Figure 3. Advantages and disadvantages of existing image technologies: pixels above, vectors below..**

## 3. PROPOSED SOLUTION

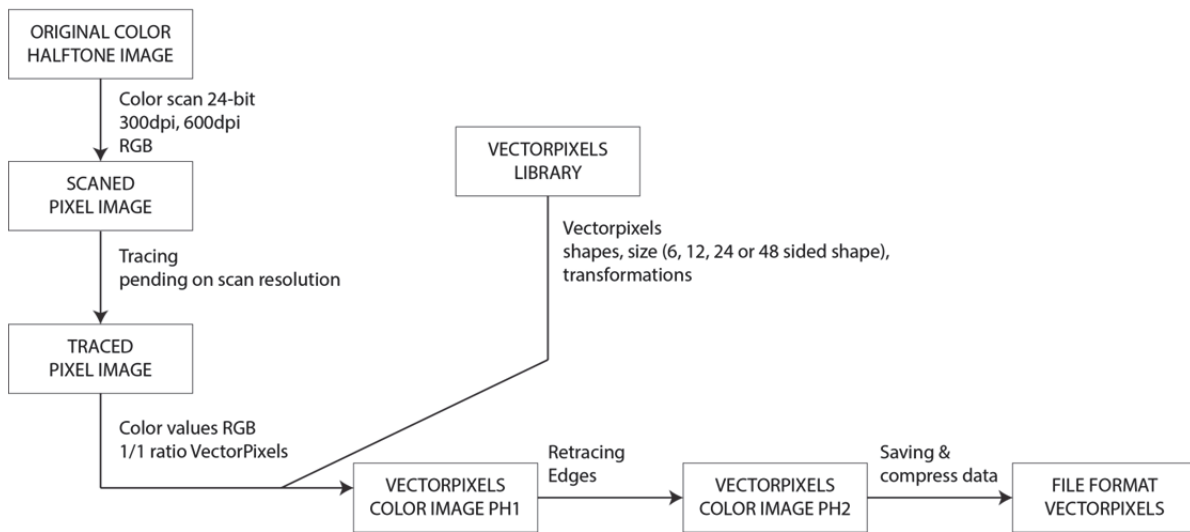
Our solution is a new algorithm called VectorPixels and differs much from solutions proposed in the past by other researchers. Despite this it uses existing technologies. VectorPixel starts from a point that is defined by vectors and there is no such thing as a resolution grid. Instead of using a bitmap grid to position pixels, VectorPixels are positioned relative to a reference point. This reference point will be in the left upper corner of the image. A logarithm set up for VectorPixels can be seen in figure 4. This algorithm consists of five parts:

- Tracing mode
- Converting pixels to VectorPixels.
- Retracing edges.
- Smoothing.
- Export.

Because there are no capturing devices available to output VectorPixels based images, tracing of existing images is necessary. For testing purposes a simple line tracing is used. At a later stage other tracing techniques could be used or tested. Tracing will detect edges in the image by difference in color values of pixels. These edges can be rebuilt to be resolution independent with VectorPixels. Note a main difference with existing solutions using a closed path to represent an area of similar pixels (Figure 6).

## ALGORITHM VECTORPIXELS

Version 1.0



**Figure 4. Algorithm for VectorPixels.**

With VectorPixels the area is built up with similar “vectorized” pixels. The shape and position of a VectorPixel can also vary depending on the detail needed in an image when enlarged. Changing resolution is already embedded in VectorPixels when they are created. VectorPixels can also include a gradation in order to give softness to edges.

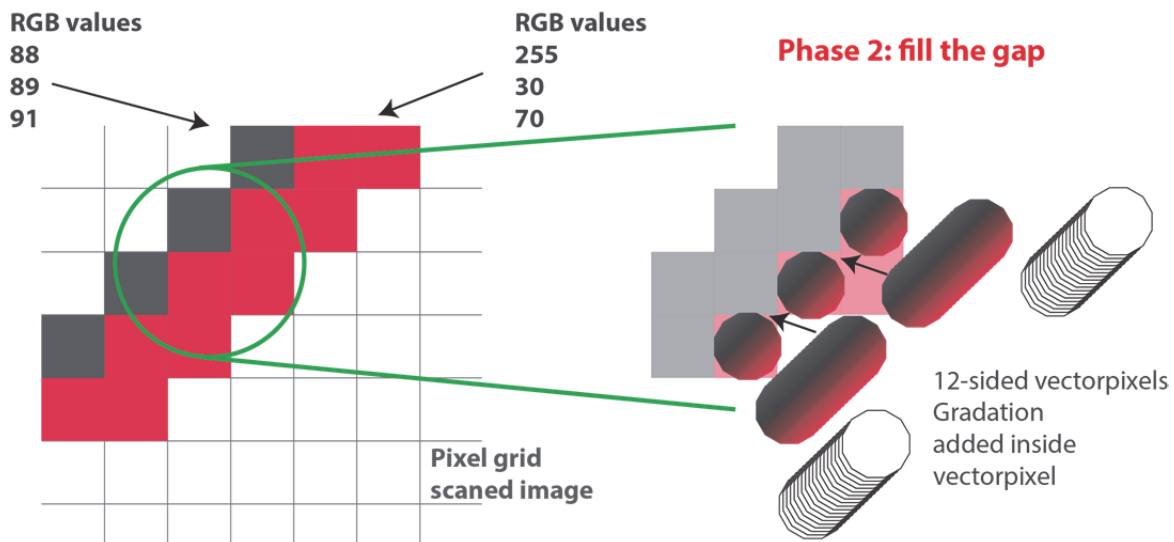
A library can be provided where different sizes or shapes of VectorPixels can be archived and used as references. This can offer great advantages when saving VectorPixels based images especially with compression. In this case only the position of a VectorPixel and a reference, need to be saved. Of course this is together with the properties of the specific VectorPixel such as Fill Color or gradient and transformations. Making use of a gradient gives the possibility to make a smooth looking edge without additional programming. Because it is not

working with a grid as with bitmaps, it is possible to position VectorPixels anywhere on the screen. For good order a maximum area (width and height) needs to be defined. Instead of using a fixed size in pixels a percentage could be used. VectorPixels can also overlap each other when a smooth curved line needs to be established (Figure 5). An algorithm (Figure 4) has been setup for first testing this new technique. Secondly this algorithm has to be realised in a programming environment which will have to deliver evidence that VectorPixels work effectively.

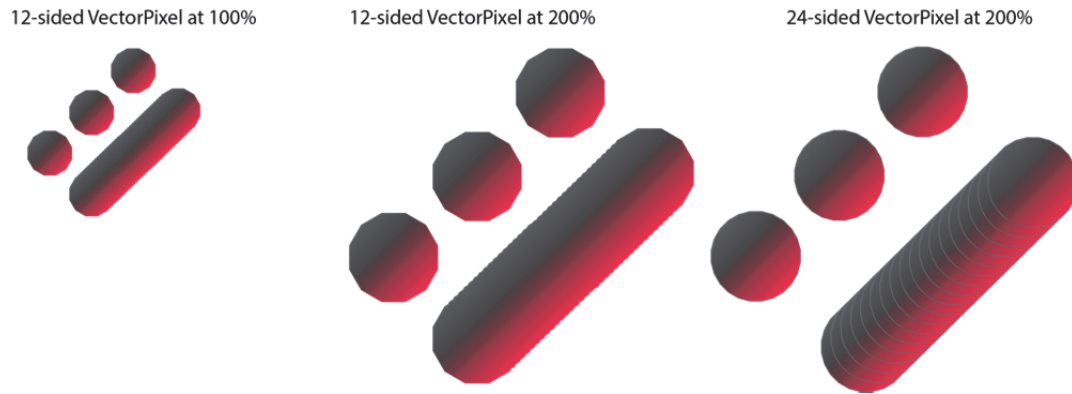
## 4. RESULTS

Our new developed algorithm is a theoretical one for the moment. Experimental research has now been carried out but useful results will be available by summer 2014. There is greater expectation in the area of smoothing edges to be achieved more easily than it

### Phase 1: VectorPixels creation



**Figure 5. Tracing and smoothing VectorPixels.**



**Figure 6. VectorPixels shape define quality. Gradation included for smooth transition.**

happens with current technologies. Also reducing file size is an important goal. When implemented in a 2D environment VectorPixels will be extended for use in a 3D environment [4]. Experiments will be conducted to measure the elapsed time for computation of VectorPixel algorithm. This could become one of the biggest drawbacks of VectorPixels due to the large processing requirements of calculations.

## 5. CONCLUSIONS

A lot of experimental research has to be done yet. Once first results become available a preliminary conclusion can be made. A schedule has been setup to get the first results published by the summer of 2014. The proposed algorithm is the baseline and could be refined when testing VectorPixels. This research is part of a larger whole. VectorPixels were first started to be used in 3D texturing. When VectorPixels are finally defined and working for 2D rich color images it can be transferred for use in 3D texturing. Especially in the area of shading it could be a great addition when Phong or Lambert shading needs to be calculated. This work will also benefit from the antialiased voxelization described in Prakash [11], [12].

## 6. REFERENCES

- [1] A. Orzan, A. Bousseau, P. Barla, et al. "Diffusion Curves: A Vector Representation for Smooth-Shaded Images", New York, ACM 56 (7), pp.101-108, July 2013.
- [2] J. Canny. "A Computational Approach to Edge Detection", MIT, IEEE Transactions on Pattern and Machine Intelligence PAMI-8 (6), pp. 679-696, November 1986.
- [3] J.H. Elder and R. Goldberg. "Image Editing in the Contour Domain", Canada, York University, IEEE Transactions on Pattern and Machine Intelligence, 23 (3), pp. 291-294, September 2000.
- [4] J. Behr et al. "A Scalable Architecture for the HTML5/X3D Integration Model X3DOM", *Web3D'10*, Los Angeles CA, 2010, pp.185-194.
- [5] J.C. Mong and D.F. Brailsford. "Using SVG as the Rendering Model for Structured and Graphically Complex Web Material", Electronic Publishing Research Group, University of Nottingham, ACM symposium on document engineering, pp. 88-91 November 2003.
- [6] W3C. (2010). Scalable Vector Graphics. Available: <http://www.w3.org/Graphics/SVG/>. Last accessed 29th March 2014.
- [7] Adobe Developer Center. (2013). Scalable Vector Graphics. Available: <http://www.adobe.com/devnet/svg.html>. Last accessed 29th March 2014.
- [8] E. Kim, X. Huang and G. Tan, "Markup SVG-A Online Content-Aware Image Abstraction and Annotation Tool", IEEE Transactions on Multimedia 13 (5), pp. 993-1006, July 2011.
- [9] D. Aydin, "Transitions on Computational Collective Intelligence", Springer Lecture Notes in Computer Science. 6220, pp. 39-55, May 2010.
- [10] R. Kunal, "Unsupervised edge detection and noise detection from a single image". Elsevier Pattern Recognition 46 (8), pp. 2067-2077, August 2013.
- [11] C. E. Prakash and S. Manohar, "Error Measures and 3D Anti-aliasing of Voxel Data", in Proc. of the Pacific Graphics 95, Taejon, Korea, 1995, pp. 225-239.
- [12] C. E. Prakash et al., "Voxel-Based Modeling for Layered Manufacturing", Computer Graphics and Applications, IEEE, 15(6), pp. 42-47, November 1995.