INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

A Review of Color Image Vectorization Techniques

Bernard Alala
Student, Department of Computing, Jomo Kenyatta University of Agriculture and Technology, Kenya
Waweru Mwangi
Lecturer, Department of Computing, Jomo Kenyatta University of Agriculture and Technology, Kenya
George Okeyo
Lecturer, Department of Computing, Jomo Kenyatta University of Agriculture and Technology, Kenya

Abstract:
Digital image processing of gray scale or color images has become an important research and investigation tool in many areas of science and engineering. As the resolution of output devices has risen over the past 20 decades, the demand of high resolution contents has also increased. However, most images are stored in raster format. This format represents images on the computer screen using a rectangular grid of pixels, arranged in rows and columns and the images have a fixed resolution; once the image has been defined at a specific resolution, it cannot be scaled further. Therefore, the image is degraded when enlarged or scaled down which is a major problem in computer graphics, vision, and imaging. Since images with higher pixel density are enviable in many applications, there is an increasing demand to acquire high resolution raster images from low resolution. Raster images are resolution independent therefore; there is a need to convert raster images to vector. Vector graphics use geometrical primitives to express a raster image. These primitives are more compact, editable, scalable, resolution independent and also smaller in file size. Several vectorization algorithms have been developed for the last three decades. This paper presents a review of three categories of raster to vector algorithms, triangulation, mesh-based and parametric patches and algorithms developed using these techniques by various academic authors. Our findings reveal that important algorithms for converting raster images into vector have been developed. However, some of these algorithms perform better than others. Some deficiencies in the algorithms are caused by many possible outlines that can be obtained from the same bitmap image and the technique used by authors. The techniques are powerful but they also have some limitations when representing pixels. Most of the methods produce acceptable results for non-photorealistic images but some are worse or the same as the problem since they blur the image as well especially in photorealistic images. Further research should focus on how to use techniques discussed in this paper to develop a raster to vector algorithm that can convert photorealistic images to vector without compromising image quality.

Keywords: *Color image, Vector image, Raster image, Vectorization.*

1. Introduction

Raster images are represented on the computer screen using a rectangular grid of pixels, arranged in rows and columns. The pixels are small enough that they are not easy to see individually. They can be created in a wide variety of formats. The common formats are *.jpg, *.gif, *.bmp, *.tiff, and *.png. The major advantage of raster image is that they are simple, understandable, and easy to use. Nevertheless, specifying individual pixel color is not the best way to create an image. Therefore, raster file format has a number of drawbacks when representing graphics on the web. Some of the drawbacks are:

- Image Size: The size of a raster image is defined by its width, height and the number of bits allocated to each pixel in the image. The transfer of images stored in this format is a major bottleneck when accessing websites.
- Linear Features: It is difficult to adequately represent linear features depending on the cell resolution. This means that network linkages are difficult to establish.
- Fixed resolution: Raster images have a fixed resolution. That is, once the image has been defined at a specific resolution, it cannot be scaled further. Therefore, the image is degraded when enlarged or scaled down. In other words, scaling raster image will not reveal more detail as illustrated in figure 1.0.

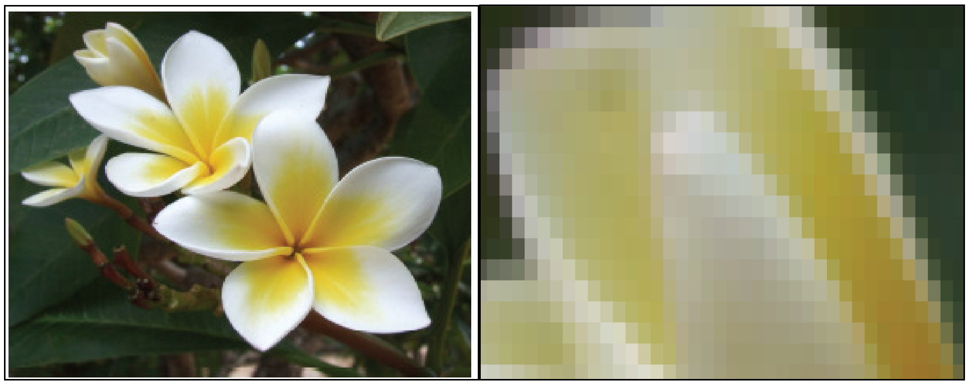
Figure 1: (a) Flower.bmp (File size: 187KB)(b) Flower.bmp (File size: 3.18MB)

Figure 1.0:The image (a) is the original raster image, and the image (b) is the same Raster image when zoomed at 130%. The image gets distorted when zoomed as indicated in Figure 1.0(b). The original image was extracted from the internet, courtesy of Lai et al., (2009).
Images with higher pixel density are enviable in many applications such as autonomous vehicles, computer graphics, computer vision, sensors etc.; therefore, the demand to acquire high resolution raster images from low resolution is increasing. The only option to acquire high resolution raster image from low resolution is to convert it into vector. The conversion of raster into vector is known as vectorization. Vectorization converts pixels into basic geometric shapes such as lines, circles, triangles, and rectangles.

1.1. Benefits of a Vector Image

- Scaling:A vector image can be enlarged and still have smooth edges, which is resolution independent. Therefore, users can zoom in or out without image degradation. The ability to resize an image without altering its proportions is essential in image processing.
- Color: Vector can render large areas of color with relatively small file sizes. This feature is necessary in websites where high resolution images are inevitable.
- Storage space: Saving vector data requires low storage space since the only part that changes is the mathematical factor attached to each image size. Therefore, it describes the same image with less information than raster.
- Image quality: Vectors are defined by equations; thus, they always render at the highest quality.
- Ability to edit: Vector graphic is very easy to edit. The ability to edit an image enables image post processing.The usefulness of vector graphic has led to the development of several algorithms that convert color bitmap image into vector. However, there isn't a perfect raster to vector algorithm. This paper provides a review of some of the vectorization algorithms developed by various authors. The review takes a systematic approach whereby the vectorization techniques are classified into categories.

2. Methodology

The objective of this study is to provide an understanding of raster to vector algorithms. The study employs descriptive research design for it to portray an accurate profile of situations. The target population for this review comprised of academic authors of raster to vector algorithms. The study used secondary data from selected scholarly work.The collected data contributed towards the formation of background information needed for the reader to comprehend the review outcome. Data analysis was done by first editing the information for consistency and completeness and then categorized systematically. This paper categorized raster to vector algorithms according to three techniques:

1) Triangulation
2) Mesh-based
3) Parametric Patches

The three are the most used techniques in raster to vector algorithms; a brief description of each technique is given below:

2.1. Triangulation

Triangulation approach approximates pixels by triangles. The technique uses Lawson's and Watson's algorithm to maximize the smallest angle over all triangles.

<u>2.1.1. Watson's Algorithm</u>

This algorithm starts by forming a super triangle which is a triangle encompassing all the given points of the domain. At first the super-triangle is flagged as incomplete. The algorithm then proceeds by incrementally inserting new points in the existing triangulation. A search is then made for all the triangles whose circumcircles contain the new point and they are deleted to give insertion polygon. This gives the new triangulation. The process continues till all points to be inserted are exhausted and then all triangles having the vertices of the super-triangle are deleted (Zimmer, 2005).

Watson Algorithm

- Form super triangle, enclosing all points $p \in V$
- As long as not all vertices of V have been treated, do:

1. Insert vertex $p \in V$ into triangulation
2. Find circumcircles containing p with corresponding triangles
3. Remove triangles to get insertion polygon
4. Retriangulate insertion polygon by simply adding edges to p

- Remove super triangle

Figure 2: A summary of Watson Algorithm

<u>2.1.2. Lawson Algorithm</u>

The algorithm first adds a point to an existing triangular mesh then circumcircles are formed for all new triangles formed. If any of the neighboring points lie inside the circumcircle of any triangle, then a quadrilateral is formed using the triangle and its neighbor. Thereafter, the diagonals of this quadrilateral are swapped to give a new triangulation. This process is continued till there are no more faulty triangles and no more swaps are required (Zimmer, 2005). The method makes use of edge flipping to insert points and therefore avoids a possibly faulty degenerated mesh which is a problem when using Watson's algorithm. Lawson algorithm does not find circumcircles and delete triangles to obtain an insertion polygon like Watson. The algorithm has a method for incremental insertion to make use of edge flipping to achieve the same results and thus avoids a possibly faulty, degenerated mesh. If there is no triangulation, a super triangle enclosing all vertices in V is used. In every insertion steps a vertex $p \in V$ is inserted. A simple re-triangulation is made where the edges are inserted between p and the corner vertices of the triangle containing p.

Lawson Algorithm

- Form super triangle, enclosing all points $p \in V$
- As long as not all vertices of V have been treated, do:

1. Insert vertex $p \in V$ into triangulation
2. Triangulate new p (draw edges to p from enclosing triangle, creating triangles)
3. For all new triangles t created recursively:
   Check circumcircle of t, if containing neighboring vertex, flip.
4. Remove super triangle

- Remove super triangle

Figure 3: A summary of Lawson Algorithm

2.2. Mesh-based Technique

This technique differs from triangulation approach because it treats the color difference as propagation from features that are detected from curvilinear. According to Chew (1993), meshes can be generated as follows:

1) Build an initial crude triangulation with the property that the surface normal in a region around each adjacent pair of triangles vary by less than $\frac{\pi}{2}$. The technique uses edge flips to make the initial triangulation into a curved-surface CDT. Some subdivision of triangles may be necessary.
2) Grade any triangles that are currently ungraded. A triangle passes only if it is both well-shaped and well sized.
3) If all triangles pass then halt. Otherwise choose the largest triangle that fails (call it Δ) and determine its surface-circumcenter (call it c).
4) Travel across the CDT from any vertex of Δ toward c (any path will work as long as we stay within the circumcircle of Δ) until we either run into a source edge or find the triangle containing c.
5) If we find the triangle containing c then insert c into the CDT. Go to step 2.
6) If we run into a source-edge with lengths within, a particular range can be split in thirds to simplify some proofs. In practice, splitting in half appears to always work. Note that since boundaries may be curved, splitting in half here means finding a point on the boundary (and on the surface) that is equidistant from each endpoint. Let l be the (straight-line) length of the new edges and consider the one or two new vertices produced by the split operation. Delete each circumcenter-vertex of the CDT that is closer than l (line-of-sight distance: and intervening source edge implies infinite distance) to a new vertex. Go to step 2.

2.3. Parametric Patches

This category of vectorization technique aims for the use of curves in vector representation. The technique involves the use of Bezier curves (de Casteljau's algorithm), B-spline curves (de Boor and Cox de Boor algorithm) and Diffusion curves.

2.3.1. Bezier curves

A Bezier curve is a parametric curve that is used in computer graphics to model smooth curves that can be scaled indefinitely. Bezier curves are evaluated using de Casteljau algorithm. The geometrical nature of the algorithm solves any point on a Bezier curve of any degree.

deCasteljau's algorithm

- For any n^{th} order Bezier curve, the algorithm works as follows:

1. It starts with the point set {p_o = first point, p_1 = first control point, p_2 = second control point and p_n = last point}, which contains n+1 elements.
2. It replaces this set with an interpolated set {p_0' = point between p_o and p_1 at distance t, p_1' = point between p_1 and p_2 at distance t and p_{n-1}' = point between p_{n-1} and p_n at distance t}.
3. Step 2 is repeated until there is only one point left; the point will lie on the original

Figure 4: A summary of de Casteljau's algorithm

The following are the steps in de Casteljau's algorithm:

2.3.2. B-spline Curve

B-splines are a more general type of curve than Bezier curves. B-spline is generated using De Boor's algorithm. The De Boor's algorithm is a generalization of de Casteljau's algorithm. It provides a fast and numerically stable way for finding a point on a B-spline curve. B-spline curve uses basis function unlike Bezier curve that uses Bernstein polynomials. The description of the de Boor's algorithm is given below:

De Boor Algorithm

$$N_i^k(u) = \frac{u-u_i}{u_{i+k-1} - u_i} N_i^{k-1}(u) + \frac{u_{i+k}-u}{u_{i+k}-u_{i+1}} N_{i+1}^{k-1}(u)$$

$$N_i^1(u) = \begin{pmatrix} 1, u_i \leq u < u_{i+1} \\ 0, otherwise \end{pmatrix}$$

Figure 5: A summary of the De Boor's Algorithm

2.3.3. Diffusion Curves

Diffusion curves present vector graphics for smoothly-shaded images and stores an image as a set of 2D Bezier curves with colors defined on either side.

We studied five algorithms that use Triangulation technique, four algorithms that use Mesh-based technique and seven algorithms that use Parametric patches. We then give both strong and weak features of each algorithm. The exact nature of the algorithms and their authors are represented in Table 1.

	Author(s)	Technique	Characteristic	Source
1.	Battiato et al., (2005)	Triangulation	Converts raster image into SVG. Generates a data dependent triangulation (DTT). Uses a wavelet based Triangulation (WBT). The number of actual triangles is larger than the number of pixels. The output vector image has larger file size than the original raster image.	http://www.computer.org/csdl/proceedings/camp/2005/2255/00/22550333-abs.html
2.	Lecot and Levy (2006)	Triangulation	Computes a set of vector primitives & gradients that best approximate the image.	http://dl.acm.org/citation.cfm?id=2383939

			More image details can be produced in user selected regions. The input image is divided into regions of varying colors. The output is a triangulation of the input regions. The algorithm cannot handle pixel art images.	
3.	Xiat et al., (2009)	Triangulation	Decompose image plane into non-overlapping parametric triangular patches. A subset of curved patch boundaries is dedicated to faithfully represent curvilinear features. Achieves accurate & compact vector-based representation. The algorithm cannot handle pixel art images.	http://dl.acm.org/citation.cfm?id=1618461
4.	Boy'e et al., (2012)	Triangulation	The algorithm Identifies vector primitives & their type in an image. Generates a triangulation which finds color and gradient that approximate input image inside each triangle. Cannot be adapted to more complex domain.	http://dl.acm.org/citation.cfm?id=2366192
5.	Lia et al., (2012)	Triangulation	The algorithm is based on triangular decomposition of the image plane and a piecewise smooth loop subdivision surfaces. Has multi-resolution representation. It is not user friendly since many control points which are difficult to manipulate are exposed by triangular subdivisions.	http://www.computer.org/csdl/trans/tg/2012/11/ttg2012111858-abs.html
	Sun et al., (2007)	**Mesh-based**	The algorithm obtains an optimized gradient mesh by formulating energy minimization problem. The gradient mesh is editable, scalable and automatic. The algorithm does not guarantee closed regions. Infeasible to perform region based shape editing	http://dl.acm.org/citation.cfm?id=1276391
	Lai et al., (2009)	Mesh-based	The algorithm automatically extracts gradient meshes from raster images. Controls error fitting to balance quality with storage. Imposes redundant restrictions to achieve a highly adaptive spatial layout. Less portable since it's not standard primitive.	http://dl.acm.org/citation.cfm?id=1531391
	Xia (2010)	Mesh-based	Based on segmentation. Perceives input image as both a 2D grid of pixels & a connected gird. Inefficient on images with regions that have abundant texture details.Uses candidate query points which are randomly generated, therefore, some regions are under sampled.	https://www.ideals.illinois.edu/bitstream/handle/2142/16586/defense_new.pdf?sequence=2
	Pang et al., (2011)	Mesh-based	The algorithm uses mean value coordinates interpolation on a triangle mesh to greater images comparable in appearance to DCIs. The method provides only approximate solutions to the DCI (Diffusion Curve Images). It's unclear how to efficiently perform integration over some regions.	http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6051406&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6051406
3	**Parametric Patches**			
	Price & Barrett (2006)	Bezier Curves	The algorithm creates editable vector graphics from an object in a raster image. Cubic Bezier grid is generated for each selected image object by recursive graph cut segmentation & an error-driven subdivision. The algorithm produces several tiny patches in images with rapidly changing regions.	http://dl.acm.org/citation.cfm?id=1165413

	Xia eta al., (2009)	Bezier Curve	The algorithm uses triangular Bezier patches. The image plane is decomposed into non-overlapping parametric triangular patches with curved boundaries. A subset of the curved patch boundaries is dedicated to faithfully represent curvilinear features. The algorithm is lossy.	http://dl.acm.org/citation.cfm?id=1618461
	Armstrong (2006)	B-spline curve	The algorithm segments raster image. The output vector image is made up of layered B-spline surfaces. The algorithm is not automatic as it requires human intervention. The method takes a longer time to approximate image color.	http://www.researchgate.net/publication/250006403_VECTORIZATION_OF_RASTER_IMAGES_USING_B-SPLINE_SURFACES
	Kopf & Lischinski (2011)	B-spline curve	The algorithm extracts a resolution independent vector representation from pixel art images. It enables scaling without image degradation. It reduces pixel aliasing artifacts. It fits spline curves to contours in the image and then it optimizes their control points. The technique does not work well with anti-aliased input images.	http://dl.acm.org/citation.cfm?id=1964994
	Finch et al., (2011)	Diffusion Curves	The algorithm uses thin plate splines that define vector images using diffusion between colored curves. It increases editing power by expanding a single curve into multiple offsets of various basic types. The thin-plate spline extrapolates beyond specified value constraints. The vectorization process is manual.	http://dl.acm.org/citation.cfm?id=2024200
	Sun et al., (2012)	Diffusion Curves	The algorithm is based on Green's functions for mapping diffusion curves images onto arbitrary surfaces. It allows the texture value of any rectangular region to be solved in closed form that facilitates anti-aliasing. The algorithm has an overhead for calculating the weight of the Green's functions.	http://dl.acm.org/citation.cfm?id=2185570
	Orzan et al., (2013)	Diffusion Curves	The algorithm creates smooth shaded images. It partitions the space through which it is drawn, defining different colors on either side. The color may vary smoothly along the curve. The resulting output image is compact, editable and can process highly complex image content. The algorithm depends on the global solution to (bi-) harmonic equation, which is an expensive operation. It is feasible for interactive use only on a powerful GPU system.	http://dl.acm.org/citation.cfm?id=2483873

Table 1: A review of raster to vector techniques

3. Discussion

A raster to vector algorithm is normally developed to solve the problems of raster images. Our findings reveal that important algorithms for converting raster images into vector have been developed. However, some of these algorithms perform better than others. Some deficiencies in the algorithms are caused by the technique used by authors. The techniques are powerful but they also have some limitations when representing pixels. In triangulation technique, several tiny line segments are used to approximate curves; this gives results that are not attractive. The technique also uses Lawson's algorithm to control error bounds in polynomial interpolation schemes. The main challenge of Lawson's algorithm is that it cannot minimize the largest edge over everytriangle.

The drawback of Mesh-based technique is creating a new mesh. This needs much skill and patience, because the artist needs to accurately predict the mesh resolution and topology required to set in the preferred smooth features. Drawing effective meshes and performing accurate manual color sampling involves a lot of effort.

Our third technique is parametric patches; this includes Bezier curve, B-spline curve and Diffusion curve. Bezier curve produces smooth representation and they are easy to edit, but the curve has global influence of the control points. Therefore, every change of a control point changes the appearance of the curve in all points of the curve. The time required to compute a large set of control points is also very high. B-spline curves allow one to create a wider variety of curves. Nevertheless, it is more complicated to compute compared to Bezier curve. B-spline curve is also a polynomial curve and these curves reveal poor tradeoff between shape and degree which can result to an unstable model. Diffusion curve is a powerful tool for producing and storing 2D image. Rendering anti-aliasing diffusion curve surfaces in real time is still a challenge, especially with highly detailed surfaces which have many control points.

4. Conclusion

Raster image vectorization is increasingly important since vector-based graphical contents have been implemented in personal computers and on the internet. This is because raster image produces some unwanted artifacts when scaled. In recent years, a number of important developments have significantly improved in the area of image vectorization. However, some of the methods produce results that are worse or the same as the problem since they blur the image as well especially in photorealistic images. In absolute sense, developing a perfect vectorization algorithm is a doubting task as many possible outlines can be obtained from the same bitmap image. Some algorithms have produced acceptable results in non-photorealistic images. Nevertheless, image quality for photorealistic images is still deficient. Further research should be carried out on developing a raster to vector algorithm that can produce high quality vector images for photographic images.

5. References

i. Adobe. (2014). https://www.adobe.com/. Accessed 29 July 2014.
ii. Arbenz, P. (2012). Numerical Methods for Computational Science and Engineering. Lecture Notes, computer science department.
iii. Battiato, S., Di, B., Gallo G., Messina G., Nicotras S. (2005). SVG Rendering for Internet Imaging. In Proceedings of IEEE CAMP'05, International Workshop on Computer Architecture for Machine Perception.
iv. Boy'e, S., Barla, P., Guennebaud, G. (2012). Avectorial solver for free-form vector gradients. ACM Transaction on Graphics.
v. Brian, P., Barrett, W. (2006). Object-based Vectorization for Interactive Image Editing. International Journal of Computer Graphics, 22, 661-670.
vi. Chew, P. (1993). Guaranteed-Quality Mesh Generation for Curved Surfaces. Department of Computer Science. Cornell University, Ithaca, NY 14853.
vii. Corel Corporation. (2014). http://www.coreldraw.com/us/. Accessed 29 July 2014.
viii. Curtis, A. (2006). Vectorization of Raster Images using B-spline Surfaces, Department of Computer Science, Brigham Young University.
ix. Finch, M., Snyder, J., Hoppe, H. (2011). Freeform graphics with controlled thin-plate splines. ACM Transactions on Graphics, 30.
x. Kopf, J., Lischinski, D. (2011). Depixelizing Pixel Art. In: ACM SIGGRAPH 2011 papers. SIGGRAPH 11, 99.
xi. Lai, Y., HU, B., Martin, R. (2009). Automatic and Topology Preserving Gradient Mesh Generation for Image Verification. ACM Trans. Graph.28.
xii. Lecot, G., Levi B. (2006). Automatic region detection and conversion. Paper No. 349-360, In Proc. EGSR'06 349-360.
xiii. Liao, Z., Hoppe, H., Forsyth, D., Yu, Y. (2012). A subdivision-based representation for vector image editing. IEEE Transactions on Visualization and Computer Graphics: 18, 11.
xiv. Orzan, A., Bousseau, A., Barla, P., Thollot, J. (2013). Diffusion Curves: A vector Representation for Smooth-Shaded Images. Communication of the ACM, 56, 101-108.
xv. Pang, W., Qin J., Cohen, M., Heng, P., Choi, K. (2012). Fast rendering of diffusion curves with triangles. IEEE Computer Graphics and Applications, 32, 68-78.
xvi. Sun, J., Liang, L., Wen, F., Shum, H. (2007). Image Vectorization using Optimized Gradient Meshes. ACM Transaction on Graphics, 26.
xvii. Sun, X., Dong, Y., Lin, S., Xu, W., Wang, W., Tong, X., Guo, B. (2012). Diffusion curve texture for resolution independent texture mapping. ACM Transactions on Graphics, 31.
xviii. Xia, T. (2010). A hybrid Approach to Problems in Image Processing, University of Illinois at Urbana-Champaign, 94pages; 3452288.
xix. Xia, T., Liao, B., Yu, Y. (2009). Patch-Based Image Vectorization with Automatic Curvilinear Feature Alignment. ACM Transaction on Graphics, 115, 28.
xx. Zhao, K., Zhang, W., Li, W. (2012). Object-based Multi-Level Objects Based Image Vectorization with Semantic Interaction. International Journal of Advancements in Computing Technology (IJACT), 4, 1.
xxi. Zimmer, H. (2005). Voronoi and Delaunay Techniques, Lecture Notes, Computer Sciences VIII, RWTH Aachen.