Computational Aesthetics in Graphics, Visualization, and Imaging (2008)
P. Brown, D. W. Cunningham, V. Interrante, and J. McCormack (Editors)

Arty Shapes

Yi-Zhe Song[1], Paul L. Rosin[2], Peter M. Hall[1] and John Collomosse[1]

[1]MTRC, University of Bath, Bath, UK
[2]Department of Computer Science, University of Cardiff, Cardiff, UK

Abstract

This paper shows that shape simplification is a tool useful in Non-Photorealistic rendering from photographs, because it permits a level of abstraction otherwise unreachable. A variety of simple shapes (e.g. circles, triangles, squares, superellipses and so on) are optimally fitted to each region within a segmented photograph. The system automatically chooses the shape that best represents the region; the choice is made via a supervised classifier so the "best shape" depends on the subjectivity of a user. The whole process is fully automatic, aside from the setting of two user variables to control the number of regions in a pair of segmentations — and even these can be left fixed for many images. A gallery of results shows how this work reaches towards the art of later Matisse, of Kandinsky, and other artists who favored shape simplification in their paintings.

Categories and Subject Descriptors (according to ACM CCS): I.3.4 [Graphics Utilities]: Paint systems

1. Introduction

In recent years, the image-based Non-Photorealistic Rendering (NPR) literature has become increasingly populated with methods for producing abstract synthetic art. As our background section (Section 2) makes clear, this trend is a perfectly natural one; pioneering work relied on relatively simple image processing to support figurative art in painterly styles such as pointillism. Today, image-based NPR is capable of non figurative art, rendered in a wide range of different media.

This paper continues the trend toward abstraction. There are two crucial technical contributions:

- We optimally fit shapes such as triangles or rectangles to regions in segmented images
- We provide a novel classifier that automatically decides which of the fitted shapes "best" represents the region.

The advances allow us to emulate works by artists like Kandinsky, Miró, and the later works of Matisse. These artists transform complex geometric shapes into much simpler forms, typically circles, squares or triangles, or else shapes resembling convex hulls, for example. It is these type of artworks that influenced our research in this paper; we know of no previous NPR work using shape fitting.

Our system works, broadly, as follows. Given a segmented image, we optimally fit simple geometric shapes such as rectangles, triangles, circles, superellipses, convex hull and so on to each segment. Which of these shapes is used when rendering can be specified by a user, or else can be chosen automatically. Details of our shape fitting and automatic shape selection technique, which represent our principle contribution, are to be found in Section 3. Once shapes are fitted, we can render them, as Section 4 explains. In practice we use more than one segmentation, to give fitted shapes of different scales or "granularity". These different granularities are rendered so as to preserve salient detail. The final piece can optionally be painted over using any one of the standard NPR algorithms designed for rendering areas; Section 5 provides a gallery of results.

2. Background

Many image-driven NPR algorithms have been proposed over the last decade. Such algorithms can be traced back to the semi-automated paint systems of the early nineties [Hae90, SadRBS94] which construct artwork as a sequence of virtual brush strokes. Media emulation [CPE92] followed soon after, producing algorithms capable of rendering photographs into oil paintings [Lit97, Her98], hatchings [SWHS97, Hal99], and watercolors [CAS*97], to name but a few. This fascination with media continues [Bro07] with excellent results.

Deciding where to place strokes was seen as the principal barrier to producing (synthetic) art automatically. It quickly became clear that low-level image processes such as edge-maps [Lit97] or color variance maps [TC97] are not sufficient, because these act locally. Salience maps that took image-wide information into account [CH02] allowed the location of each stroke to be decided on image-wide (global) information. But each stroke was manufactured locally, so the look of images was improved, but the gamut of artistic styles with reach of NPR was not extended.

Only recently have the means to extend the gamut of NPR to abstract styles been recognized: NPR systems require more sophisticated image processing than linear filters used in the early days. For example, Mould [Mou03], and later Setlur et al. [SW06] synthesize stained glass renderings by generating translucent texture patches, driven by a region segmentation. In the latter case, image content within each region determines the appearance of glass shards by visually querying a texture database. Collomosse and Hall [CH03] cut out, re-arranged and distorted segmented image regions to create Cubist-like renderings from photographs. The same authors have recently shown how to introduce non-linear (artistic) perspective into photographs [HCS*07]. Others reflect the intuition that art making depends on inferring perceptual structure from images so as to facilitate their rendering [CH06, OBBT07]; simply put, art is a matter of abstraction.

This paper explores the fusion of image segmentation and shape fitting to create abstract artwork, and so continues this recent trend. Like others before us [GCS02, DS02, BGH03, CH06, OBBT07], we depend on an ability to identify regions within a photograph; that is, to segment it. But we contrast with this prior work in two important ways: First we do not use an explicit hierarchy of regions, but instead rely on being able to segment a photograph into regions of different granularity — segmentations of large sized or of small sized regions. Second we do not use the results to drive low level stroke placement in painting algorithms. Instead we simplify the shape of the regions in a segmentation so as to produce higher level "abstract" artistic compositions that are advocated by artists such as Kandinsky and later Matisse, which previously was beyond the reach of image-based NPR.

3. The Method

Our proposal is that shape simplification steers NPR away from image segmentations towards yet higher levels of abstraction. This section describes our approach to shape simplification. Subsection 3.1 explains how we segment images into disjoint regions of interest. Our methods for fitting various types of shapes to a given region are then discussed in the following subsection (Subsection 3.2). The users can predefine which particular shapes they want for a given region, or can be chosen automatically using the method described in Subsection 3.3.

Figure 1: *Left: Original Color Image; Middle and Right: Segmentation Results with N = 5 and 120, respectively*

3.1. Segmentation

We start by segmenting a color image into disjoint regions of interest, using a modern image segmentation algorithm. However, given an image, we are not only interested in a single segmentation, but multiple ones with different levels of details. There are many decent image segmentation algorithms in the computer vision literature, we use a multi-scale normalized cut algorithm by Cour et al. [CBS05]. It has the benefit of operating in various image scales and offering a single parameter N, which is the number of regions (i.e. cuts) to make in the image, specified a priori by the user. A smaller N yields larger and coarser regions, whereas large N returns smaller and more detailed regions. Figure 1 shows an example color image and its segmentation results when N is changed.

3.2. Fitting Simple Shapes to Regions

Having segmented an image, we are able to fit a wide selection of shapes to each region. Specifically, we fit five shapes: circles, rectangles, triangles, superellipses and a "robust" version of the convex hull, as now explained.

Voss and Süße described a powerful method for fitting a variety of geometric primitives by the method of moments [VS97]. The data is first normalized by applying an appropriate transformation to put it into a canonical frame. The fitted geometric primitive is then simply obtained by taking the geometric primitive in the canonical frame and applying the inverse transformation to it. We have applied this approach to fit circles, rectangles and triangles.

To fit superellipses a closed form solution is not available using the above approach, and so we use the least squares method described in [RW95]. The ideal distance measure to minimise would be the shortest Euclidean distances between each point and the superellipse, but this is expensive to compute. Instead the ray from the center of the superellipse to each data point is intersected with the superellipse, and the summed distances along these rays is minimized using Powell's method for non-linear optimization [PFTV90]. The optimization is initialized by fitting an ellipse.

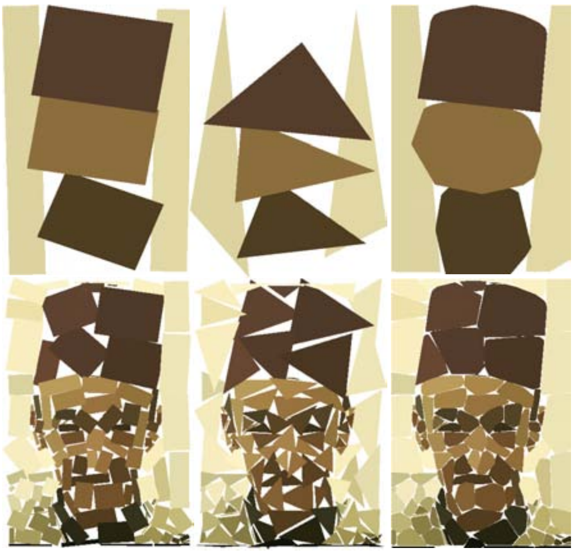

Figure 2: *Results of fitting shapes of a single type. Left Column: Fitting Rectangles; Middle Column: Fitting Triangles; Right Column: Fitting Robust Convex Hulls*

The convex hull is an attractive symbolic representation of a shape on two counts. It is generally more compact (using only a subset of the original polygonal vertices), and also perceptually simpler since all indentations have been removed. However it has two limitations: it is insensitive to the size and shape of all indentations, and is also too sensitive to protrusions. To overcome these problems Rosin and Mumford [RM06] suggested a "robust" version of the convex hull, which is the convex polygon that maximises the area overlap with the input polygon. To compute the robust convex hull they used a genetic algorithm; alternatively a dynamic programming solution has been described [KF05].

Results of fitting various user defined shapes to the two segmentation results shown in Figure 1 (N = 5 and 120) are given in Figure 2.

3.3. Automatic Shape Type Selection

We are now able to optimally fit a collection of simple shapes to each region within a segmented image. The problem now is how to choose amongst them. Interaction is one approach, but not only is this tedious for the user but, we argue, is less interesting than considering automatic choice. One key contribution of this paper is our proposal of automatic shape selection. We considered several varieties of information theoretic approaches, but found none that corresponded with our subjective judgment (much as rms is known to give a poor estimate of image decompression quality). So we opted in favor of a trained classifier; training allows some subjectivity into the process. We now explain our classifier and the training regime.

Automatically selecting appropriate shape models is done using a supervised classification paradigm; specifically, a C4.5 decision tree [Qui93] is learnt from a training set of regions which is then applied to new unseen data. The basis of a decision tree is that each feature can be used to make a decision that splits the data into smaller subsets, partitioning feature space into equivalence classes using axis-parallel hyperplanes. C4.5 builds decision trees by selecting the most informative feature (that is not yet considered in the path from the root) to split each subset. An entropy measure — Normalized Information Gain — determines the effectiveness of each feature. The regions are described by a feature vector and are manually labelled into shape categories. These features are the basis for making the decision regarding which is the most appropriate model. The feature vector consists of the errors between the region and each of the fitted shape models. To compute the errors at each data point the shortest distance to the fitted shape is determined using the distance transform. However, the summed error is not a sufficient descriptor – it is easy to construct examples in which the best shape model (according to aesthetics and perceptual criteria) does not have a lower summed error. Instead, the distribution of point errors, which is more informative, is considered, and summarized by the following lower order statistics: mean, standard deviation, skew, and kurtosis. Thus, for five shape models and four statistical terms, each region is described by a total of twenty features.

Following this approach, we selected 2 training images other than the ones used in this paper (an image of a Buddha and an English cottage photo) and segmented them using the segmenter described in Section 3.1. We also deliberately segmented each image into different granularities to get more regions with a larger variety of shapes. Among over 500 segmented regions, we extracted their feature vectors and labelled 81 of them as our training data. A C4.5 decision tree is then built using those 81 pieces of training data. The learnt decision tree is used to generate all results in this paper.

Results of the automatic shape selection step is shown in Figure 3. As can be seen, there is now a variety of shapes, comparing to the set of results shown in Figure 2. For example, regions such as the eyes of the man have triangles fitted to them, which was expected as their corresponding image segments are somewhat triangular.

4. Rendering Shapes

We can now fit shapes to each region from a given segmentation and automatically select the best fit among those. Segmentations at a coarser level can yield large and more abstract shapes; whereas, detailed segmentations often result in shapes that are too small and overly detailed. What we really want is to preserve an appropriate amount of detail, while keeping the abstractness. We resolve this issue by treating the layer of larger/coarser shapes as "background" and the

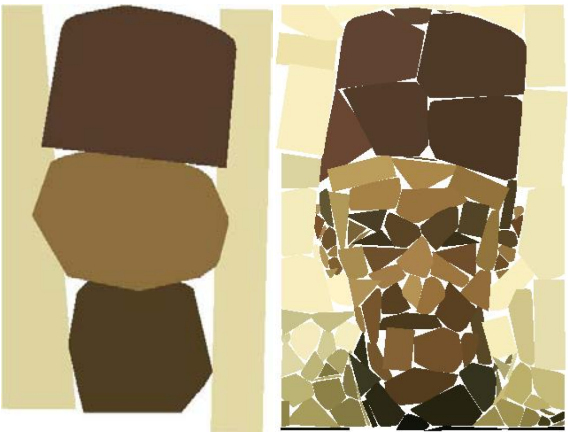

Figure 3: *Results of automatic shape selection*

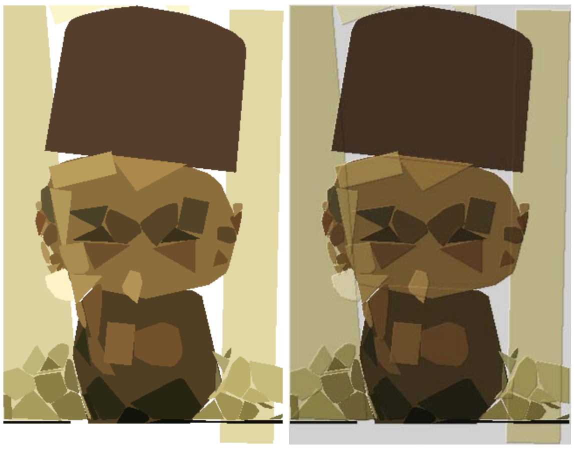

Figure 4: *Left: Result of combining two layers of shapes from segmentations of different granularities; Right: rendering shapes as paper cutouts.*

one with smaller and detailed shapes as "foreground". Doing this naively will result in the top layer completely covers the bottom one. We solve this by filtering the detailed shapes on the top layer by their corresponding shapes underneath. More specifically, we only render shapes from the top layer whose color deviates from that of the shape underneath above a certain threshold. Hertzmann also used color differences to place strokes on top of those already painted in his stroke rendering work [Her98]. Unlike him, we measure color differences in terms of just noticeable difference (jnd) in CIELAB color space. For instance, take two colors, (L_1, a_1, b_1) and (L_2, a_2, b_2), we define their color difference ΔE_{12} as follows

$$\Delta E_{12} = \frac{\sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}}{jnd}$$

where $jnd \approx 2.3$ in CIELAB color space [Sha03]. Therefore, in general, ΔE measures how many jnds one color deviates from another. By placing a threshold on ΔE, we can control the level of detail to render on the top layer; increasing the threshold results in fewer shapes being rendered and vice versa. The left of Figure 4 shows such a result of merging two layers of shapes. It is clear that in the merged result (shown on the left), features like the "hat" reside in one single shape inherited from the bottom layer, whereas, details such as facial features are taken from the top layer. As with the result shown in Figure 4 and all other results in this paper, a constant threshold of 5 is used on ΔE.

When it comes to rendering shapes into a framebuffer, the ordering of shapes can play a role in the final output as well. This is because of the fact that shapes fitted often overlap each other and which one comes on top can confuse viewer's perception. We tackled this problem by introducing a shape fitting error τ. Given a shape model s and its corresponding region r, we denote the area of s by S and similarly R for r, then τ is defined as the following ratio, $|\bigcap(S,R)|/|\bigcup(S,R)|$, which is a form of Tanimoto similarity score, calculated on a per pixel basis. The idea is then to lay down shapes according to their assigned fitting error. Shapes with large fitting errors are rendered before those with smaller errors.

To create an embossed look, to the paper cuts we simply counted the number of shapes lying over each pixel; the resulting height field became a bump map. To create transparent paper we simply used "alpha" colour channel.

5. Gallery of Renderings

This section exhibits a collection of shape rendering results. Divided into two parts, we first demonstrates examples of NPR images that can be generated with no additional parameters other than a choice of rendering style such as "flat", "embossed" or "transparent". The second part includes a collection of synthetic art works that applies modern NPR stoke-rendering techniques on top of our shape renderings. Specifically, we used the technique described in [SBC06] to generate all results shown in this part. These images show that it is possible to create abstract paintings in different media, but do not directly add to the main contribution of this paper.

The right of Figure 4 shows the "man" rendered as "paper-cuts"; paper cut outs appear in the later Matisse, exemplified by artworks such as "L'escargot". A paper cutout rendering of "bird feeding" is shown in the middle of Figure 5, where the original image is shown on the left. As can been seen, there is a nice balance of shapes in the final rendering; relatively large entities in the scene such as the trunk of tree has a single rectangle fitted, while a combination of small shapes together compose the nest. A highly abstract version of "bird feeding" is shown on the right of the same figure, where the user chose to fit circles across every region. This

Figure 5: *Left: Original color image; Middle: shapes rendered as paper cutouts; Right: an abstract result of fitting circles.*

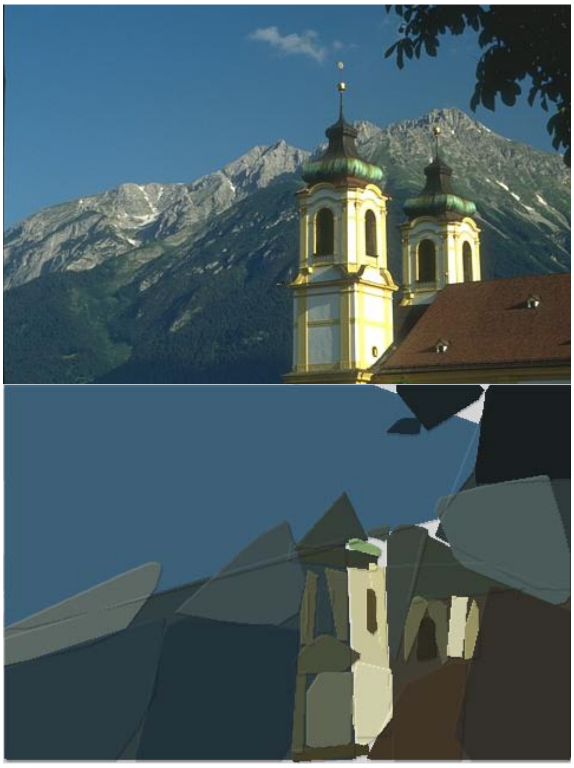

Figure 6: *Top to Bottom: Original color image and chapels rendered as paper cutouts.*

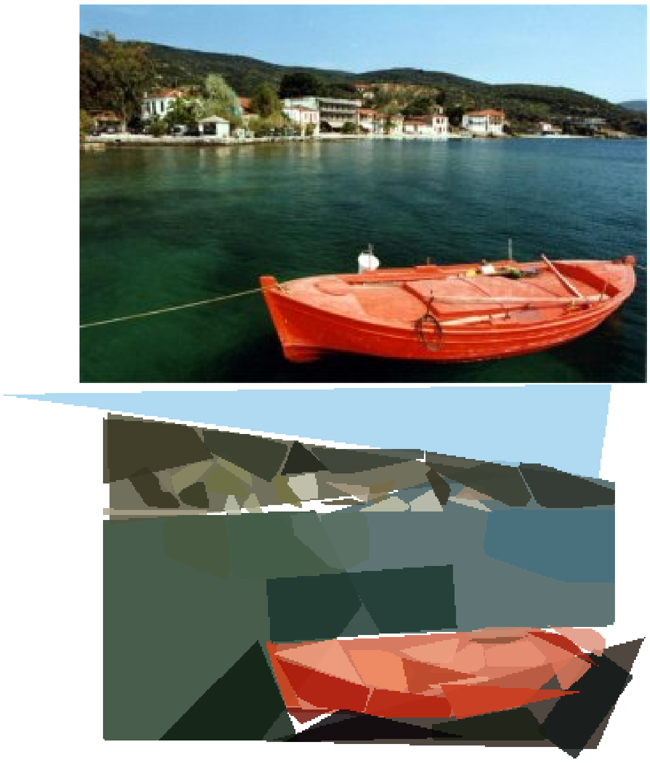

Figure 7: *Top to Bottom: Original color Image; shapes rendered as translucent paper cutouts.*

Figure 8: *Les Ballons*

may not be to everyone's taste, but we liked it and the result is an extreme example of how shape simplification enables abstraction that goes well beyond stroke modeling.

In a similar fashion to the "bird feeding" example, Figure 6 shows how a landscape scene is rendered into a piece of artwork where paper cutouts were used as basic elements. Again, large objects such as the mountain at the back and the sky have rather large shapes fitted; but both "towers" are composed of a rather interesting combination of smaller shapes. The sense shown at the top of Figure 7 is rendered as a combination of transparent paper cutouts, shown beneath. It is interesting here to note how various shapes are fitted to represent the boat itself, whereas the "sky" is represented as an abstract triangle. The method in this paper favors broad, clean colours; so examples such as hot air ballons make pretty pictures (Figure 8).

All the above rendering results are obtained solely from shape simplification, with simple effects like paper cutouts put on top. But, of course, we can make of stroke renderers to further enhance the aesthetic appeal of our synthetic artwork. Two oil paintings are included in Figure 9. Figure 10 shows both an oil painting and a crayon painting of the "man".

Finally, we offer up a rather whimsical version of the Matisse's snail, in Figure 11, as a reminder of the source of our motivation that led to this paper.

6. Conclusion

This paper's contribution to NPR is that we have moved towards automatically creating more abstract art than was previously possible. More specifically, the art we synthesized was influenced by artists such as Kandinsky and later Mattise who advocate the use of geometric shapes. Shape simplification is the key to delivering the level of abstraction resembled in such type of art. Importantly, we can automatically select which shape fits the best among a few that we can fit. We are also able to combine shapes of various granularities and so preserve appropriate amount of detail.

The whole process only takes two parameters during the segmentation step, these being integers to specify how many segments the user wants on each layer. The automatic shape selection step involves supervised learning, however, can be re-used once trained. No further input is required unless one wishes to paint with an advanced media style — but control of that part lies rightfully with other literature. In the gallery of Section 5 we exhibit direct output that is similar in abstraction to Kandinsky et al, a cut-paper effect as used by Matisse in his later works, as well a more traditional media of oil and crayon paintings.

It is clear that shape simplification is able to support the creation of synthetic artworks of a more abstract nature than before. There is plenty of scope for future work in this area. The picture frame was, of course, composited on by hand.

References

[BGH03] BANGHAM J. A., GIBSON S. E., HARVEY R.: The art of scale space. In *Proc. British Machine Vision Conference* (2003), pp. 569–578.

[Bro07] BROOKS S.: Mixed media painting and portraiture. *IEEE Trans. on Visualization and Computer Graphics 13*, 5 (2007), 1041–1054.

[CAS*97] CURTIS C., ANDERSON S., SEIMS J., FLEISCHER K., SALESIN D. H.: Computer-generated watercolor. In *Proc.* 24th *Intl. Conference on Computer Gr aphics and Interactive Techniques (ACM SIGGRAPH)* (1997), pp. 421–430.

[CBS05] COUR T., BENEZIT F., SHI J.: Spectral segmentation with multiscale graph decomposition. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 1124–1131.

[CH02] COLLOMOSSE J. P., HALL P. M.: Painterly rendering using image salience. In *Proc. 20th Eurographics UK Conference* (2002), pp. 122–128.

[CH03] COLLOMOSSE J. P., HALL P. M.: Cubist style rendering from photographs. *IEEE Transactions on Visualization and Compute r Graphics (TVCG) 4*, 9 (Oct. 2003), 443–453.

[CH06] COLLOMOSSE J. P., HALL P. M.: Salience-adaptive painterly rendering using genetic search. *Intl. Journal on Artificial Intelligence Tools (IJAIT) 15*, 4 (Aug. 2006), 551–576.

[CPE92] COCKSHOTT T., PATTERSON J., ENGLAND D.: Modelling the texture of paint. *Computer Graphics Forum 11*, 3 (1992), 217–226.

[DS02] DeCarlo D., Santella A.: Abstracted painterly renderings using eye-tracking data. In *Proc.* 29th *Intl. Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH)* (2002), pp. 769–776.

[GCS02] Gooch B., Coombe G., Shirley P.: Artistic vision: Painterly rendering u sing computer vision techniques. In *Proc.* 2nd *ACM Symposium on Non-photorealis tic Animation and Rendering* (June 2002), pp. 83–90.

[Hae90] Haeberli P.: Paint by numbers: abstract image repre sentations. In *Proc.* 17th *Intl. Conference on Computer Gr aphics and Interactive Techniques (ACM SIGGRAPH)* (1990), vol. 4, pp. 207–214.

[Hal99] Hall P.: Non-photorealistic rendering by Q–m apping. *Computer Graphics Forum 1*, 18 (1999), 27–39.

[HCS*07] Hall P. M., Collomosse J. P., Song Y. Z., Shen P. Y., Li C.: Rtcams: A new perspective on non-photorealistic rendering. *IEEE Trans. on Visualization and Computer Graphics 13*, 5 (2007), 966–979.

[Her98] Hertzmann A.: Painterly rendering with curved brush strokes of multiple sizes. In *Proc.* 25th *Intl. Conference on Computer Gr aphics and Interactive Techniques (ACM SIGGRAPH)* (1998), pp. 453–460.

[KF05] Kolesnikov A., Fränti P.: Optimal algorithm for convexity measure calculation. In *Int. Conf. Image Processing* (2005).

[Lit97] Litwinowicz P.: Processing images and video for an imp ressionist effect. In *Proc.* 24th *Intl. Conference on Computer Gr aphics and Interactive Techniques (ACM SIGGRAPH)* (Los Angeles, USA, 1997), pp. 407–414.

[Mou03] Mould D.: A stained glass image filter. In *Rendering Techniques* (2003), pp. 20–25.

[OBBT07] Orzan A., Bousseau A., Barla P., Thollot J.: Structure preserving manipulation of photographs. In *Proc. ACM NPAR* (2007), pp. 103–110.

[PFTV90] Press W. H., Flannery B. P., Teukolsky S. A., Vettering W. T.: *Numerical Recipes in C*. Cambridge University Press, 1990.

[Qui93] Quinlan J. R.: *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[RM06] Rosin P. L., Mumford C. L.: A symmetric convexity measure. *Computer Vision and Image Understanding 103*, 2 (2006), 101–111.

[RW95] Rosin P. L., West G. A. W.: Curve segmentation and representation by superellipses. *Proc. IEE: Vision, Image, and Signal Processing 142* (1995), 280–288.

[SadRBS94] Salisbury M. P., an d Ronen Barzel S. E. A., Salesin D. H.: Interactive pen-and-ink illustration. In *Proc.* 21st *Intl. Conference on Computer Gr aphics and Interactive Techniques (ACM SIGGRAPH)* (Florida, USA, 1994), pp. 101–108.

[SBC06] Shugrina M., Betke M., Collomosse J.: Empathic painting: interactive stylization through observed emotional state. In *NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2006), ACM, pp. 87–96.

[Sha03] Sharma G.: *Digital Color Imaging Handbook*. CRC Press, 2003.

[SW06] Setlur V., Wilkinson S.: Automatic stained glass rendering. In *Proc. Computer Graphics Intl. (CGI)* (2006), pp. 682–691.

[SWHS97] Salisbury M. P., Wong M. T., Hughes J. F., Salesin D. H.: Orientable textures for image-based pen-and-ink illustration. In *Proc.* 24th *Intl. Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH)* (Los Angeles, USA, 1997), pp. 401–406.

[TC97] Treavett S. M. F., Chen M.: Statistical techniques for the automated synthesis of non-photorealistic images. In *Proc. 15th Eurographics UK Conference* (1997), pp. 201–210.

[VS97] Voss K., Süsse H.: Invariant fitting of planar objects by primitives. *IEEE Trans. on Patt. Anal. and Mach. Intell. 19*, 1 (1997), 80–84.

Figure 9: *Two oil Paintings; above a church tower, below a red boat.*

Figure 10: *Left to right: a painted man and a crayoned man*

Figure 11: *The Snail*