Towards Artistic Minimal Rendering

Paul L. Rosin *
Cardiff University

Yu-Kun Lai†
Cardiff University

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

Figure 1: *Mona Lisa rendered in different styles. (a) line drawing, (b) image abstraction, (c)(d)(e): three styles of the proposed approach in this paper (single level, texture and pyramid).*

Abstract

Many nonphotorealistic rendering techniques exist to produce artistic effects from given images. Inspired by various artistic work such as Warhol's, interesting artistic effects can be produced by using a minimal rendering, where the minimum refers to the number of tones as well as the number and complexity of the primitives used for rendering. To achieve this goal, based on various computer vision techniques, our method uses a combination of refined lines and blocks, as well as a small number of tones, to produce abstracted artistic rendering with sufficient elements from the original image. There is always a trade-off between reducing the amount of information and the ability to represent the shape and details of the original images. Judging the level of abstraction is semantic-based, so we believe that giving users this flexibility is probably a good choice. By changing some intuitive parameters, a wide range of visually pleasing results can be produced. Our method is usually fully automatic, but a small amount of user interaction can optionally be incorporated to obtain selective abstraction.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation; I.4.6 [Image Processing]: Segmentation—Region growing, partitioning

Keywords: non-photorealistic rendering, line drawing, image abstraction

*e-mail: Paul.Rosin@cs.cf.ac.uk
†e-mail: Yukun.Lai@cs.cf.ac.uk

1 Introduction

Two-Dimensional (i.e. image based) non-photorealistic rendering (NPR) has proliferated over the last few years, and applications such as Photoshop contain numerous effects for modifying images to provide a more artistic effect. Many of these effects either operate on a region basis (e.g. mosaic [Hausner 2001], stained glass [Mould 2003]) or build up the output image by applying a vast number of strokes (e.g. [Hertzmann 1998; Hiller et al. 2003; Yang and Yang 2008]). However, while such approaches are effective, they are limited in terms of their ability to perform substantial simplification whilst retaining the essence of the image. Generating a *sparse* (and possibly abstracted or caricatured) rendering of an image such as the line drawings by Matisse or Picasso is a more challenging task. Even when 3D models are available the results in the current literature are not ideal [Sousa and Prusinkiewicz 2003; Cole et al. 2008]. The task of extracting good lines from images is substantially more difficult, and it is likely that domain specific models are required to produce work aiming at this calibre, so that high level image interpretation would enable some minimal set of semantically essential features to be selected for rendering.

In this paper we continue with the traditional bottom-up approach, and in particular take Kang *et al.* [2007] as our starting point. They extract lines by applying a difference of Gaussians (DoG) filter followed by thresholding. Their contribution is to reduce fragmentation and extract highly coherent lines by adaptively determining the shape and orientation of the filter kernel from the local image characteristics. The DoG is applied in the direction perpendicular to the largest intensity contrast, as specified by the local edge flow. In turn, this is determined by performing non-linear vector smoothing of the edge tangents, using a bilateral filtering type approach, so that strong edge directions are preserved while weak edge directions are modified to follow dominant edges. At each pixel in the image the curvilinear centre line of the kernel is created by following the local edge flow a fixed distance upstream and downstream. The second dimension of the kernel is formed by expanding the centre line in the direction perpendicular to the edge flow. The line drawings produced by Kang *et al.*'s method can be attractive, but as with the previously cited methods, they cannot be directly modified to perform substantial abstraction, as shown in figure 1(a).

In contrast, the work by Song *et al.* [2008] is specifically designed to produce highly abstract images, reminiscent of the later period of art by Matisse consisting of paper cutouts. An example is given in figure 1(b). The image is segmented at multiple scales using graph cut, and a variety of simple shapes (e.g. circles, triangles, squares, superellipses and so on) are fitted to each region. 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. A consequence of this approach (and many others that aim for high levels of abstraction [Wen et al. 2006]) is that it is unable to capture the nuances necessary to produce, for instance, an identifiable portrait.

The aim of this paper is to use Kang *et al.*'s method to provide the detailed content from an image in the form of coherent lines, and to combine it with dark and light blocks that capture the overall tonal balance of the image, and which, inspired by Song *et al.*, are extracted from the image and then simplified and rendered in various ways. By combining abstraction along with a limited amount of readily recognisable detail, we can produce effects similar to Warhol's portraits from the mid 1980's which mixed and overlaid line drawings, photographs and coloured geometric blocks such as rectangles. In addition, we consider applying some decorative effects to the blocks, in a manner loosely inspired by Matisse who sometimes included a field of patterns overlaid on a flat background. Three different rendering styles of the Mona Lisa have been given in figure 1(c-e).

The contribution of the paper is to introduce a novel insight of a class of artistic rendering as a perceptually minimal rendering model. A variety of techniques from computer vision have been incorporated to approach this goal. More specifically, a careful combination of refined lines and blocks, together with a small number of tones have been combined to produce visually pleasing artistic rendering with recognisable information using a set of "least complicated" primitives. During this process, some aesthetic criteria are also naturally taken into account, e.g. tonal balance is retained since significant dark or light regions are reproduced using blocks in the output image. This is related to the work by Rivotti *et al.* [2007] where combinations of different primitives are used for aesthetic nonphotorealistic rendering from 3D models.

2 Method

The motivation of our algorithm is to model the open problem of artistic rendering as using a set of "least complicated" primitives to approximate the input image, while maintaining the rendered image as recognisable as possible. We use dark and light blocks (potentially simplified) to lay out the overall tone balance. A small number (typically between 3 and 10) of tones is used instead of two as our observation shows that much more information can be preserved, using only the small perceptual overhead of one or a few extra tones. Refined and simplified coherent lines are further overlaid to enhance the visual richness while keeping the perceptual overhead well controlled.

These essential components are discussed in details in the first three subsections, including post-processing refinement of Kang *et al.*'s lines, three tone drawing and extraction of refined dark and light blocks. We then discuss how a multi-scale pyramid can be used to generate even richer results, at the cost of slightly more tones. For certain images with cluttered background, a foreground mask can optionally be created to produce an artistic focusing effect. The foreground mask can be created automatically, but our observation is that, by using a small amount of user scribbles, the mask can be made far more reliable and effective.

Since the output of algorithms similar to our method use a small number (or even just two) grayscale tones, such algorithms tend to use grayscale images as input. Our observation is that even though a grayscale image is sufficient to produce reasonable results, having the extra information from colour (which is widely available) can often make the tonal blocks better aligned with the important image boundaries, which can be poorly defined when converted to grayscale images. We utilise this in various places of the algorithm pipeline; details will be given later in this section.

2.1 Post-processing Kang *et al.*'s lines

We have considered applying a range of well known techniques from the computer vision literature as a post-processing step to enhance the output of Kang *et al.*'s method. Since their method is basically a (sophisticated) edge detector the large literature relating to edge detection is directly applicable. The aim of this post-processing, following from our philosophy of minimal rendering, is to clean up and simplify the line rendering. This will help improve the coherence of the DoG output such that it appears more like strokes rather than independent pixels.

Hysteresis: Kang *et al.* applied a constant threshold value at each edgel, however there are many other edge thresholding approaches [Rosin 1997]. In particular, the incorporation of hysteresis for thresholding edge maps by Canny [1986] meant that the results of his edge detector were less fragmented and more coherent than previous thresholded edge maps. Canny's method of hysteresis employs two thresholds T_U and T_L, where it was experimentally determined that $T_L \approx \frac{T_U}{2}$. Two sets of edgels E_U and E_L are determined, whose gradient magnitudes are greater than T_U and T_L respectively. The final binary edge map contains all edgels E_U and those edgels in E_L that are connected to an edgel from E_U by a path of E_L edgels. The rationale is that edgels of intermediate strength (i.e. between T_U and T_L) are only retained if there is supporting evidence from connected strong edges. Kang *et al.*'s detector has been specifically designed to produce coherent lines. Nevertheless, figure 2 shows that when hysteresis is applied to the DoG output there is further improvement. Many small disconnected edges are removed, and in some places lines are extended by retaining edgels that were previously removed by the constant threshold.

Deleting connected components: Another way to clean up the thresholded edge map is to perform connected component analysis to group adjacent edgels together, and then remove connected components smaller than a given size. While that is our basic approach, in the following sections we will be extending the rendering from binary to images containing more intensities. Therefore we use a more general approach, namely connected set morphology. Here, rather than use fixed structuring elements as in standard mathematical morphology, the operations are applied to connected components in the gray level image. Thus, a gray scale closing sets each pixel to the lowest threshold value at which it belongs to a connected background component which has an area greater than or equal to the given threshold. We use the efficient algorithm by [Meijster and Wilkinson 2002] for computation of connected set opening and closing.

Scale space tracking: Scale space analysis has been applied to edge analysis and involves blurring the image over a range of scales, detecting edges at each scale, and tracking them across scale. The longer an edge persists before annihilation (i.e. its lifetime) the more likely it is to be significant. A new measure of edge strength can now be taken as its lifetime (i.e. as an indication of "stability"), or the gradient magnitudes at each scale can be summed during the

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

Figure 2: *Post-processing: a) original image, b) Kang* et al.*'s lines, c) lines after hysteresis thresholding d) lines after additional connected set morphological cleaning.*

tracking process. The latter approach was found to be especially effective for edge thresholding [Rosin 1997]. However, in experiments we found scale space tracking as described in [Bischof and Caelli 1988] to perform poorly on Kang *et al.*'s DoG output, and therefore do not include it in our system pipeline. An explanation for this failure could be the local adaptation of the DoG kernel shape led to greater edge drift over scale. Also, Bischof and Caelli tracked single pixel wide edges whereas Kang *et al.*'s lines can be thicker.

2.2 Three tone drawings

Our next development of Kang *et al.*'s method is to use it to draw both black and white lines on a gray background. Adding white lines allows highlights to be included, and can substantially enhance the impression of three dimensional structure. The effectiveness of this approach is well known to traditional artists, often appearing, for example, as chalk and charcoal drawings. An alternative way to indicate three dimensional shape would be to use cross hatching, but our philosophy of minimal rendering favours the approach we have taken of maintaining a small number of strokes with only the small perceptual overhead of one extra tone.

Processing is straightforward: lines are extracted from both the image and its inverted versions, and then combined such that at each pixel the output tone is black (white) if there only exists a black (white) line. Otherwise (if no lines or both black and white lines are present at that pixel) the output tone is gray. This can be computed using the following $\frac{lines(I)+\overline{lines(\overline{I})}}{2}$ where I is the input image, $lines(\cdot)$ is the tone output from the line detector, and bars denote inversion. (In fact, for better perceptual effect we prefer to remap the midgray intensity to 20% density.)

Figure 3 demonstrates how an additional tone enables extra details to be picked up (e.g. along the nose and lips) that are missed in the black and white drawing (see figure 2). In comparison with an optimally three tone thresholded version [Luessi et al. 2009] our three tone drawing provides a clearer, more readily identifiable depiction.

2.3 Dark and light blocks

For some images their nature is better captured by the rendering if their overall tonal balance is approximately retained. Therefore we wish to extract and render regions from the image that represent significant tonal blocks. We consider these blocks to be dark and light regions which we would like to locate automatically. Unfortunately, automatic and reliable image segmentation is in general an unsolved problem – thus whatever scheme we use we can expect some errors. With traditional NPR techniques that employ a huge number of individual strokes, individual erroneous strokes are hidden amongst the mass of their neighbours. In our minimal approach, errors, especially with regard to blocks, are much more exposed. Nevertheless, we will later describe some strategies for reducing the effects of such artifacts.

We note that the combination of lines and tonal blocks was applied to the rendering of faces over 25 years ago by [Pearson and Robinson 1985] in the context of data compression. They thresholded the image intensities and combined this – by multiplication – with extracted lines. In other words, a logical `AND` is applied such that the dark tonal blocks are retained while light tonal blocks are ignored. More recently a similar approach was taken by [Gooch et al. 2004]. However, they also only used a simple approach for extracting the tonal blocks, and stated that the intensity threshold value was "chosen manually according to taste". In contrast, we desire an automatic method which is reasonably reliable and accurate, and can also be applied to a wider range of image types beyond just faces with uncluttered backgrounds.

Since we want dark and light regions, image thresholding seems a reasonable starting point. Hundreds of algorithms exist [Sezgin and Sankur 2004] and we choose the classic algorithm by Otsu [1979] which determines the threshold that minimises the within class variance of the two groups of pixels separated by the thresholding operator. To extract significant tonal blocks a multi-threshold variant is used; i.e. n thresholds are selected to minimise the within class variance across the $n + 1$ groups of pixels. We use the efficient algorithm based on dynamic programming and the SMAWK matrix searching algorithm described in [Luessi et al. 2009] which improves on some of the time consuming solutions found in the earlier literature. If two thresholds are selected this would generate three classes of pixels which could be considered as dark, medium intensity, and light. However, given the inevitable thresholding errors that will occur for some images, we take a more conservative approach and select three thresholds so that the dark and light classes are separated by a double class of intermediate intensity pixels. The rationale is that under-representing the dark and light blocks is preferable to over representing them.

The thresholding provides a global histogram based criterion for classifying pixels into tonal classes. Due to noise and other variability this results in noisy and fragmented dark and light blocks which need to be cleaned up. Two steps of post processing are em-

(a) (b) (c)

Figure 3: *Three tone drawings: a) original image, b) three tone drawing, c) image posterised into three levels*

ployed which take local spatial information into account. First, a standard mathematical morphological opening and closing is applied which removes isolated small, or thin regions. A sufficiently large structuring element is used so as to ensure that the majority of spurious clutter is removed, and consequently the remaining regions are substantially distorted. The purpose of the second step is to rectify this, and moreover improve the segmentation of the tonal blocks by taking colour information into account rather than just intensity. Grabcut [Rother et al. 2004], initialised by the output of the mathematical morphological operations, provides an effective solution. Assume the tonal region (either dark or light) is denoted as R. We use morphological dilation and erosion operators with a fixed width to obtain expanded or shrunk regions, denoted as R_+ and R_- respectively, to initialise the grabcut model with $\overline{R_+}$ as background, R_- as foreground and $R_+ - R_-$ as the unknown region to be optimised. Note that certain pixels having similar intensity may be well separated in the 3-dimensional colour space. Having the boundary optimised to snap to significant boundaries in the colour image is thus much more reliable. Colour information is also used in the generation of foreground and background Gaussian mixture models used internally in the grabcut algorithm. Thus colour similarity helps to produce refined masks taking both tonal levels and the colour similarity into account. The grabcut approach also allows the topology of the masks to be changed, i.e. some holes in the masks may be automatically added or removed.

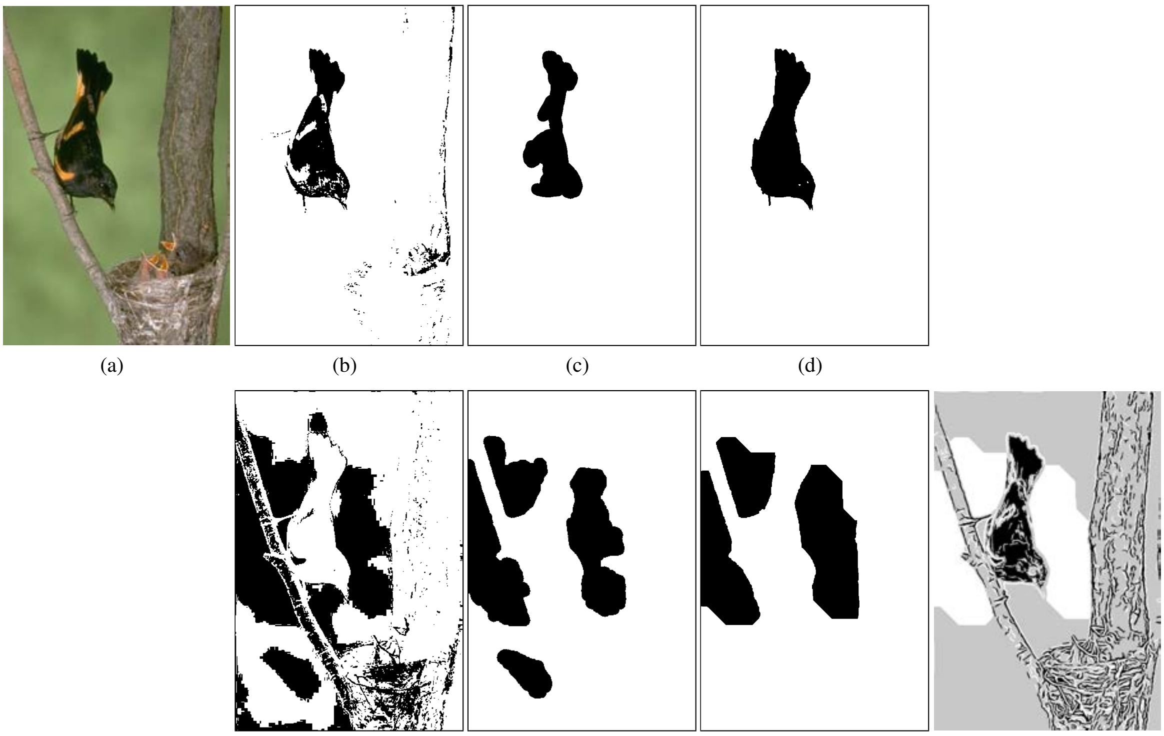
The process of extracting the masks is illustrated in figure 4. The image contains considerable JPEG artifacts, and so the initial masks produced by thresholding contain blockiness as well as noise and clutter. This is successfully removed by the opening and closing operations. Finally, the grabcut restores accurate region boundaries, and moreover fills in the yellow portions of the bird's body.

Having extracted accurately delineated blocks, they can now be rendered as they are, or simplified and modified for artistic effect. As with the stylisation in [Song et al. 2008] simple geometric primitives can be fitted to the blocks using the invariant fitting method of Voss and Süße [1997] – we have used triangles, parallelograms and ellipses. The data (i.e. the block's boundary) is first normalised by applying an appropriate transformation to put it into a canonical frame (e.g. for ellipse fitting they take the transform that maps an ellipse to the unit circle). Applying the inverse transformation to the canonical primitive produces the fitted ellipse (e.g. in this example the circle would be mapped back to produce an ellipse). Alternatively, rather than using fixed shapes, the blocks can be simplified using polygonal approximation [Ramer 1972].

Having extracted the tonal blocks, a set of rules is required for determining how the various combinations and superpositions of lines and blocks of two tones plus gray background are rendered. This is given in table 1 which can be interpreted as drawing each line by its given tone unless there is a conflict with a line of the opposite tone or a conflict with a block. In the latter case, rendering the line takes precedence over the block. A black line over a white block can still be rendered black. However, a black line over a black block needs to be modified to be visible, and is therefore rendered as gray. Inverting lines in this manner (i.e. modulated by their background) is a standard artistic technique, especially popular in woodblock illustration and also used in previous artistic thresholding [Xu and Kaplan 2008]. Perceptually more recognisable results are produced in our work than [Xu and Kaplan 2008] by using one more tone.

line		block		output
black	white	black	white	
×	×	×	×	gray
×	×	×	√	white
×	×	√	×	black
×	×	√	√	gray
×	√	×	×	white
×	√	×	√	gray
×	√	√	×	white
×	√	√	√	white
√	×	×	×	black
√	×	×	√	black
√	×	√	×	gray
√	×	√	√	black
√	√	×	×	gray
√	√	×	√	white
√	√	√	×	black
√	√	√	√	gray

Table 1: *Rules for determining at each pixel the output tone to be rendered according to the tones of the lines and blocks (if present).*



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4: *Extracting masks a) original image, b) dark mask from thresholding, c) dark mask after cleaning by mathematical morphology, d) dark mask after grabcut, e) light mask from thresholding, f) light mask after cleaning by mathematical morphology, g) light mask after grabcut, h) final rendering.*

Rather than render blocks as homogeneous we have experimented with incorporating a decorative pattern which is applied to (i.e. cut out from) each block. Although they can be attractive, such patterns have a distracting effect which disrupts the lines within the blocks. This can be counteracted to some degree by thickening the lines, although excessive thickening also becomes distracting and unattractive. Some examples using textured blocks are given in figures 1(d) and 11.

2.4 Rendering pyramid

Earlier we showed that adding a third (gray) tone allowed the rendering to better capture an image's characteristics. Continuing this theme, we consider further increasing the number of tones, as a consequence of embedding the rendering in a multi-resolution framework. We wish to capture lines at different scales since this would allow important lines to be emphasised and large scale structures to be better captured. One approach for this would be to apply Kang *et al.*'s DoG detector with a range of kernel sizes to the input image. Instead, we choose to fix the kernel size, and apply the line detector to the image at different resolutions within a pyramid. In fact, the full three tone rendering method is applied at each resolution, and the outputs are rescaled to the full image size before being combined by simple averaging. This approach means that lines drawn at low resolution are made thicker after rescaling, which emphasises the salient lines. While resizing could be done using methods such as bilinear interpolation we use instead simple pixel replication followed by a small amount of Gaussian blurring which produces a smoother, more pleasing effect. Whichever of the two resizing methods is used, the interpolation or blurring will create new intensities that were not in the downsized image. Since we wish to retain a limited tonal palette, the intensities in the resized image are mapped onto the closest intensities in the downsized image.

How many tones does the mean image contain? Consider the general case of combining n images each containing the same t tones. The upper bound on the number of resulting mean tones will occur when each unique combination of tones produces a mean that is distinct from all other tonal combinations. The number of combinations of n items with repetition allowed from t items is $\binom{n+t-1}{n}$. For the example three level pyramids used in this paper the final rendering is still relatively minimal, containing $\leq \binom{5}{3} = 10$ tones.

Figure 5 demonstrates the application of the pyramid. Three tone renders from each of the three levels (without any morphological cleaning) are shown, with parallelograms fitted to the tonal blocks. The levels are combined into the mean image (with morphological cleaning, smoothing and intensity remapping) in figure 5e. Note the emphasis of significant lines in the final rendering by their detection and thickening at the higher levels of the pyramid.

2.5 Incorporating salience

So far, we have applied rendering uniformly across the image. While this is often satisfactory, there will be some images in which it is necessary to modify the level of detail according to the saliency of objects in the image. Thus, two factors need to be considered: how to measure saliency, and a means of modifying the level of detail. Ideally these should be automatic, and for the first part there are many algorithms available. However, after experimentation we have not found any that are consistently reliable over a wide range of images (e.g. see figure 6). Therefore we take a three stage semi-automatic approach: the user "scribbles" in the image so as to very

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

Figure 5: *Rendering using the pyramid: a) original image, b) three tone rendering at bottom level, c) three tone rendering at middle level, d) three tone rendering at top level, e) mean of all renders from the pyramid. Image courtesy of Colin Swan.*

quickly and roughly indicate the whereabouts of the foreground and background. This is used to initialise a supervised watershed segmentation [Lefèvre 2007] in which colour models are build from the scribbles. As with the previously discussed segmentation of tonal blocks, such a global approach produces a good overall estimate of the foreground regions, but tends to have inaccurate boundaries. Again, we use grabcut to refine the boundaries.

Several approaches were investigated to modify the rendering based on the saliency map or foreground/background mask. The first two modified the input image before applying the basic three tone drawing method. Either the input image was blurred according to saliency, or else its contrast was reduced according to saliency. To modify the contrast the image's edge map was reduced according to saliency and then Poisson reconstruction [Pérez et al. 2003] applied. The third method takes the basic three tone drawing generated from the input image and reduces the contrast of lines according to saliency. To avoid abrupt change in the output across the mask boundary, a small width of transition is added to smoothly interpolate between 0 and 1.

The need for figure/ground extraction is demonstrated in figure 6, which is a challenging example; the highly textured wall results in a profusion of background lines that swamp the figure of the boy in the foreground (figure 6e). Salience maps from [Harel et al. 2006; Rosin 2009] are displayed in figure 6b and 6c. While they are moderately effective at highlighting the boy they are too irregular and diffuse. Consequently the rendered lines produced from the respective blurred input images are unsatisfactory (figure 6f and 6g). We have tried further to refine the automatically extracted saliency map using turbo pixels [Levinshtein et al. 2009], but the results are still not robust enough for general input images. In comparison, the scribble selected foreground mask clearly highlights the boy with only minor segmentation errors (figure 6d). The image is blurred according to the mask (figure 6i), and the rendering successfully captures the boy with the wall still indicated but not dominating; see figure 6h. For comparison, the alternative methods of using the foreground mask to modify the rendering are also shown. After the contrast of the background in the original image has been reduced using Poisson reconstruction (figure 6j) the basic three tone rendering in figure 6k is produced. It is very similar to the drawing from the blurred image (figure 6h) although the background lines are thinner. Figure 6l shows the background lines rescaled – although quite attractive (the boy is clearly highlighted) it is also more revealing of minor inaccuracies of the foreground mask.

A second example (figure 7) is taken from [DeCarlo and Santella 2002] who used eye tracking to modify the level of abstraction. Our basic three tone rendering is shown in figure 7b. Note the good use of tone inversion in which the chair and floor are drawn with gray and white lines due to the large dark tonal block. Alternatively, if a foreground mask is extracted and applied to blur the image then the majority of the background is removed, leaving just the tonal blocks and a few lines to indicate the overall structure.

3 Results

Figure 8 shows an example from [Gooch et al. 2004] where their drawing is generated by performing a logical `AND` of the thresholded raw and line detected images. Their thresholds are carefully tuned, and they are able to achieve high quality results. Our rendering using a three level pyramid also captures the subject's resemblance from the input photograph, but the extra tones enable detailing (such as the beard) and subtleties (such as the bridge of the nose) to be retained.

Figure 9 shows an example from [Mould and Grant 2008] who combine a base layer (computed using graph cuts) with a detail layer (computed by local thresholding). Their results are attractive, but have a photographic appearance to them, as opposed to our renderings which look more hand drawn. In addition, even in just our three tone version, features such as the eyes and hands have been more clearly delineated.

Figure 10 shows an examples from [Xu and Kaplan 2008] who oversegment the colour image and then optimise the assigned segment tones (black/white) according to their intensity, boundary contrast, etc. Thus segments are sometimes inverted from their natural tone so as to maintain boundaries (cf. our rules for overlaying lines and blocks). However, compared to our results, their stylisation destroys or distorts a lot of important detail (such as the eyes).

Figure 11 shows various artistic rendering results. The top row shows the original images, and the middle and bottom rows show different styles produced by our algorithm. We choose among the combinations of single level or pyramid-based, simplified or non-simplified blocks, textured or untextured blocks, to have a diverse collection of results.

Programs were run on a 2.40 GHz Intel Core 2 Duo with 4GB of RAM. For 1 megapixel images the run time was about 20 seconds using our current unoptimised code.

4 Conclusion

In this paper, aiming at producing a minimal rendering of images, we propose a new algorithm to generate artistic renderings. Lines and blocks are extracted, simplified and appropriately combined to

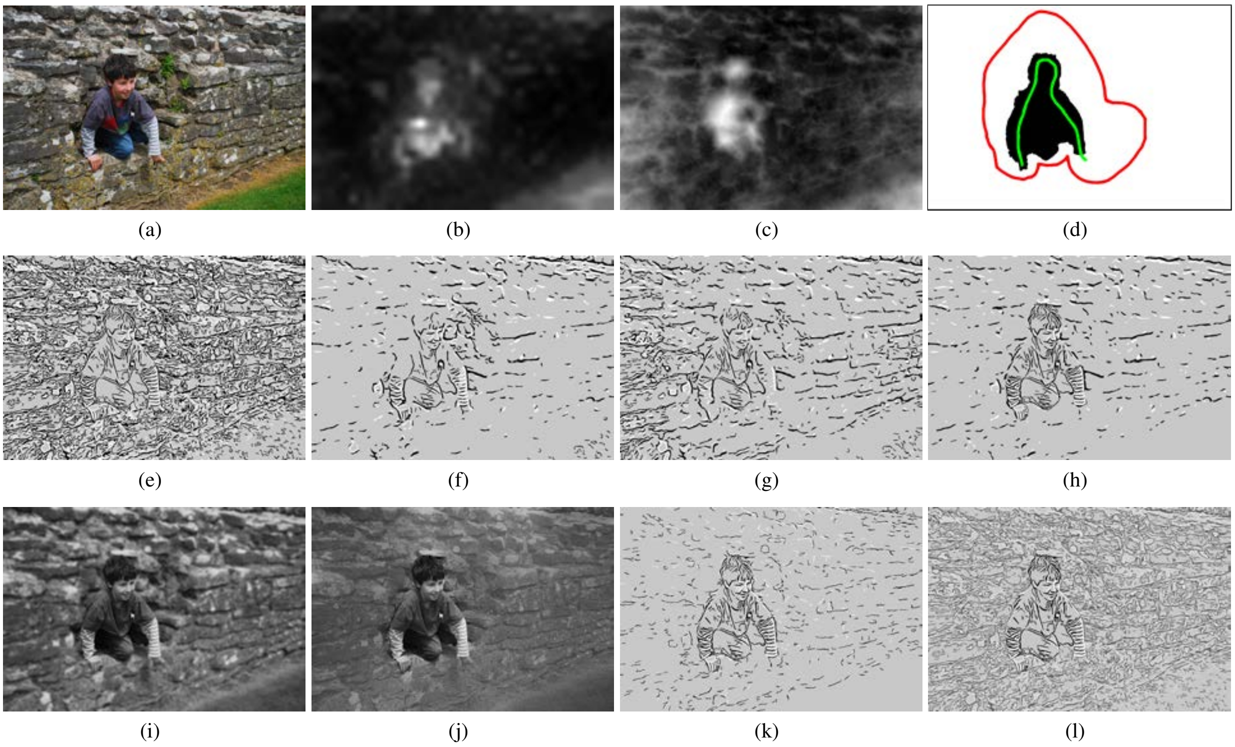


Figure 6: *Incorporating salience to modify level of detail: a) original image, b) salience map using method by [Harel et al. 2006], c) salience map using method by [Rosin 2009], d) mask showing initial scribbles for foreground and background, e) our basic three tone rendering, f) three tone rendering after image is blurred using salience map from [Harel et al. 2006], g) three tone rendering after image is blurred using salience map from [Rosin 2009], h) three tone rendering after image is blurred using foreground mask, i) image blurred using foreground mask, j) background contrast reduced using Poisson reconstruction, k) three tone rendering from Poisson reconstruction, l) background lines from our basic three tone rendering rescaled (giving five tones).*

produce visual pleasing nonphotorealistic rendering results. Our algorithm is a fully automatic method with optional interactive foreground selection to provide users with more flexibility. Most examples, unless otherwise stated are obtained automatically. Our algorithm works well on various examples, as demonstrated in the paper. A careful user evaluation similar to [Gatzidis et al. 2008] is worth exploring in the future.

The inclusion of dark and light tonal blocks greatly enhances the appearance of the renderings (this is most clearly demonstrated in figure 1). However, even with our two stage thresholding and grabcut approach it is not possible to ensure that some errors do not occur. Two techniques have been used in this paper to ameliorate these effects. The first is to replace the blocks by highly abstracted versions that simultaneously hides the faulty segmentation and provides an attractive stylisation. The second is to use a multi-resolution framework so that a faulty segmentation at one level will often be partially rectified by the results at another level.

Acknowledgements

We would like to thank Henry Kang for providing code for his line drawing algorithm, and Yi-Zhe Song for providing figure 1b.

References

BISCHOF, W., AND CAELLI, T. 1988. Parsing scale-space and spatial stability analysis. *Computer Vision, Graphics and Image Processing 42*, 192–205.

CANNY, J. 1986. A computational approach to edge detection. *IEEE Trans. PAMI 8*, 679–698.

COLE, F., GOLOVINSKIY, A., LIMPAECHER, A., BARROS, H. S., FINKELSTEIN, A., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2008. Where do people draw lines? *ACM Trans. Graph. 27*, 3, 88:1–11.

DECARLO, D., AND SANTELLA, A. 2002. Stylization and abstraction of photographs. In *SIGGRAPH*, vol. 21, 769–776.

GATZIDIS, C., PAPAKONSTANTINOU, S., BRUJIC-OKRETIC, V., AND BAKER, S. 2008. Recent advances in the user evaluation methods and studies of non-photorealisitic visualisation and rendering techniques. In *Proc. Info. Vis.*, 475–480.

GOOCH, B., REINHARD, E., AND GOOCH, A. 2004. Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. Graph. 23*, 1, 27–44.

HAREL, J., KOCH, C., AND PERONA, P. 2006. Graph-based visual saliency. In *NIPS*, 545–552.

HAUSNER, A. 2001. Simulating decorative mosaics. In *SIGGRAPH*, 573–580.

HERTZMANN, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH*, 453–460.

HILLER, S., HELLWIG, H., AND DEUSSEN, O. 2003. Beyond stippling – methods for distributing objects on the plane. *Comput. Graph. Forum 22*, 3, 515–522.

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

Figure 7: *Example from [DeCarlo and Santella 2002]. a) original image, b) our basic three tone rendering, c) foreground mask showing initial scribble, d) re-rendering after image blurring.*

(a) (b) (c)

Figure 8: *Example from [Gooch et al. 2004]. a) original image, b) rendering from [Gooch et al. 2004], c) our rendering.*

KANG, H., LEE, S., AND CHUI, C. K. 2007. Coherent line drawing. In *ACM Symp. NPAR*, 43–50.

LEFÈVRE, S. 2007. Knowledge from markers in watershed segmentation. In *Computer Analysis of Images and Patterns*, Springer, vol. 4673 of *LNCS*, 579–586.

LEVINSHTEIN, A., STERE, A., KUTULAKOS, K. N., FLEET, D. J., DICKINSON, S. J., AND SIDDIQI, K. 2009. TurboPixels: Fast superpixels using geometric flows. *IEEE Trans. PAMI 31*, 12, 2290–2297.

LUESSI, M., EICHMANN, M., SCHUSTER, G. M., AND KATSAGGELOS, A. K. 2009. Framework for efficient optimal multilevel image thresholding. *Journal of Electronic Imaging 18*, 013004+.

MEIJSTER, A., AND WILKINSON, M. 2002. A comparison of algorithms for connected set openings and closings. *IEEE Trans. PAMI 24*, 4, 484–494.

MOULD, D., AND GRANT, K. 2008. Stylized black and white images from photographs. In *ACM Symp. NPAR*, 49–58.

MOULD, D. 2003. A stained glass image filter. In *Eurographics Workshop on Rendering Techniques*, 20–25.

OTSU, N. 1979. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man, and Cyber. 9*, 62–66.

PEARSON, D. E., AND ROBINSON, J. A. 1985. Visual communication at very low data rates. *Proc. IEEE 73*, 4, 795–812.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Trans. Graph. 22*, 3, 313–318.

RAMER, U. 1972. An iterative procedure for the polygonal approximation of plane curves. *Computer, Graphics and Image Processing 1*, 244–256.

RIVOTTI, V., PROENÇA, J., JORGE, J., AND SOUSA, M. C. 2007. Composition principles for quality depiction and aesthetics. In *Proc. Computational Aesthetics in Graphics, Visualization, and Imaging*, 37–44.

ROSIN, P. 1997. Edges: Saliency measures and automatic thresholding. *Machine Vision and Applications 9*, 4, 139–159.

ROSIN, P. 2009. A simple method for detecting salient regions. *Pattern Recognition 42*, 11, 2363–2371.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23*, 3, 309–314.

SEZGIN, M., AND SANKUR, B. 2004. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging 13*, 1, 146–168.

SONG, Y. Z., HALL, P. M., ROSIN, P. L., AND COLLOMOSSE, J. P. 2008. Arty shapes. In *Proc. of Computational Aesthetics*, 65–73.

SOUSA, M. C., AND PRUSINKIEWICZ, P. 2003. A few good lines: Suggestive drawing of 3d models. *Comput. Graph. Forum 22*, 3, 381–390.

VOSS, K., AND SÜSSE, H. 1997. Invariant fitting of planar objects by primitives. *IEEE Trans. PAMI 19*, 1, 80–84.

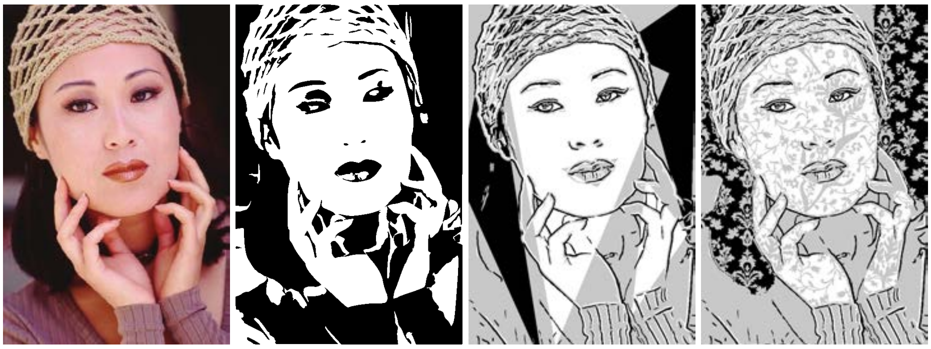
WEN, F., LUAN, Q., LIANG, L., XU, Y.-Q., AND SHUM, H.-Y. 2006. Color sketch generation. In *ACM Symp. NPAR*, 47–54.

XU, J., AND KAPLAN, C. S. 2008. Artistic thresholding. In *ACM Symp. NPAR*, 39–47.

YANG, C.-K., AND YANG, H.-L. 2008. Realization of Seurat's pointillism via non-photorealistic rendering. *The Visual Computer 24*, 5, 303–322.

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

Figure 9: *Example from [Mould and Grant 2008]. a) original image, b) rendering from [Mould and Grant 2008], c) our rendering (3 tones) d) our rendering (10 tones).*



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

Figure 10: *Example from [Xu and Kaplan 2008]. a) original image, b) rendering from [Xu and Kaplan 2008], c) our rendering (triangles fitted to tonal blocks) d) our rendering (textured tonal blocks).*

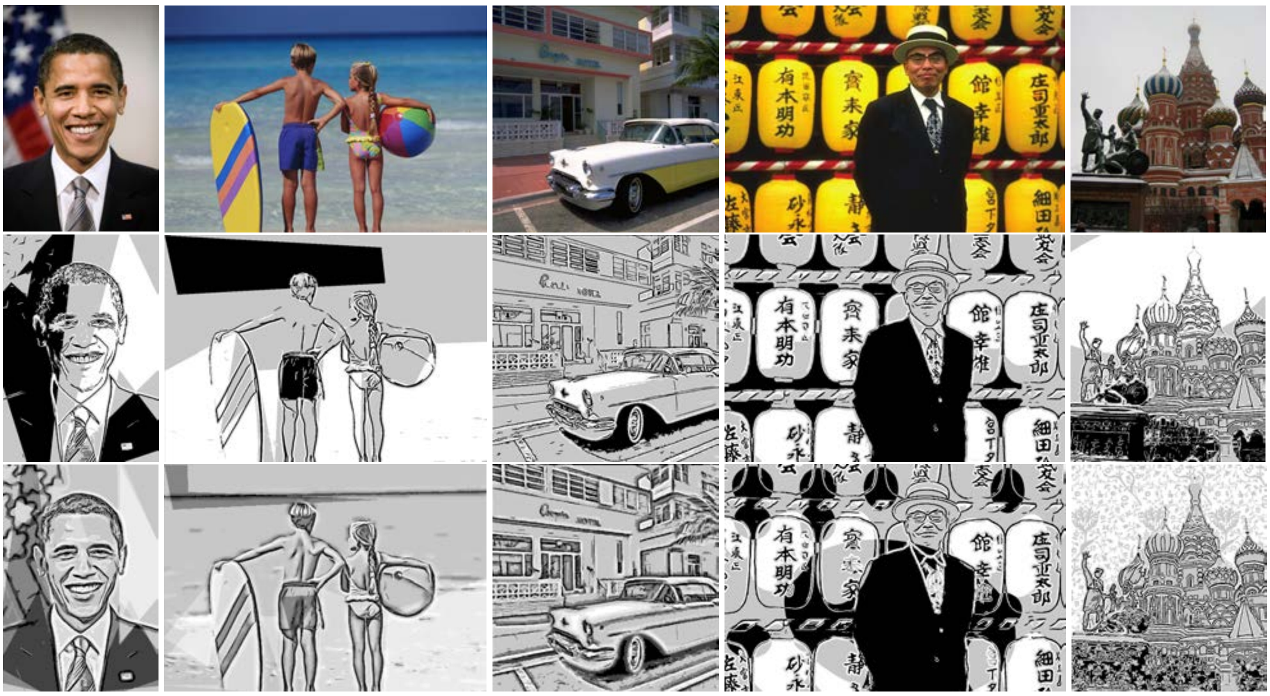


Figure 11: *Various artistic renderings produced with our algorithm. Top row: input images; middle and bottom rows: artistic rendering with two different styles. Original images from The Obama-Biden Transition Project, [Wen et al. 2006], [DeCarlo and Santella 2002] and [Xu and Kaplan 2008].*