Noname manuscript No.
(will be inserted by the editor)

Artistic Rendering Enhancing Global Structure

Yu-Kun Lai · Paul L. Rosin

Received: date / Accepted: date

Abstract Non-photorealistic rendering techniques usually produce abstracted images. Most existing methods consider local rendering primitives, and global structures may be easily obscured. Inspired by artists, we propose a novel image abstraction method that considers preserving or even enhancing global structures in the input images. Linear structures are particularly considered due to their wide existence and the availability of techniques for their reliable detection. Based on various computer vision techniques, the algorithm is fully automatic. As demonstrated in the paper, artistic looking results are obtained for various types of images. The technique is orthogonal to many non-photorealistic rendering techniques and can be combined with them.

Keywords non-photorealistic rendering · global structure · multiple support · Hough transform · deformation · snapping

1 Introduction

A large variety of techniques for non-photorealistic rendering (NPR) have been developed [24]. Most of these use local (i.e. relatively small) rendering primitives, such as strokes [18,43], stipples [19], mosaic tiles [17] and stained glass [28], which are determined on a local basis. In this paper we also use local primitives – namely regions – but wish to incorporate larger scale geometric information to control the rendering. In particular, we are inspired by how artists take the shapes and positions of objects and extend them out to create a larger scale (irregular) grid for the image. The shapes and positions of other objects in the scene are then influenced and modified by this grid, which provides a level of coherence at the global level of the image. Probably the most famous exponents of this technique were the Cubists, who emphasised the flat, two-dimensional surface of the picture plane, and the attributes of line and form. The Cubists often used straight-line constructions to divide the artwork into multiple perspectives.[1] See figure 2 for some examples which contain many straight lines, abstracted regions, and alignment of linear structures. While we were inspired by such pictures, note that unlike Collomosse and Hall [10], we were not aiming to generate renderings that exactly mimic their style.

Various methods have been proposed to simulate different artistic styles. To achieve this, abstraction or simplification is usually needed. One approach is to use non-linear filtering [41,21,12,23]. Although efficient, this is limited in the styles that can be produced. Many previous efforts rely on automatic or interactive segmentation and produce artistic effects at the local level of regions. In Gooch *et al.* [15], automatically extracted regions are used to help produce strokes in a painterly

Yu-Kun Lai
School of Computer Science and Informatics, Cardiff University, UK
Tel.: +44(0)2920 876353
Fax: +44(0)2920 874598
E-mail: Yukun.Lai@cs.cardiff.ac.uk

Paul L. Rosin
School of Computer Science and Informatics, Cardiff University, UK
E-mail: Paul.Rosin@cs.cardiff.ac.uk

[1] Some relevant examples of Cubists art that inspired our work are the following. Pablo Picasso: Still-life with Fruit-dish on a Table (1914-15), Harlequin and Woman with a Necklace (1917); Georges Braque: Pedestal Table (1913), Harlequin with Guitar (1919); Juan Gris: Violin and Checkerboard (1913), Harlequin with Guitar (1919), Jean Metzinger: Sailboats (Scène du port) (1912), Playing Cards, Coffee Cup and Apples (1917), Marc Chagall I and the village (1911), Homage to Apollinaire (1912), The Poet (Half past three) (1911).

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

Fig. 1 Several outputs from our rendering pipeline. (a) input image; (b) one dominant straight line with multiple sections of support (drawn in green) has been detected, the red and blue lines were rejected. This line provides the constraints for segmentation; (c) the output rendering with the global structure enhanced. (d) an alternative rendering with a deformation, determined by the blue and green lines, such that approximate horizontal and vertical alignment of structures is exaggerated for artistic effect.

(a) (b)

Fig. 2 Cubist paintings by (a) Juan Gris: Violin and Checkerboard (1913) and (b) Jean Metzinger: Playing Cards, Coffee Cup and Apples (1917).

style rendering. In DeCarlo and Santella [14], simplified regions are used to produce artistic abstraction, with the abstraction level controlled by fixation data from eye-trackers. Nice results are obtained but required the effort of collecting extra data. Wen *et al.* [40] proposed a system for colour sketch generation, based on interactive improvement of automatically produced segmented regions. To produce a stronger artistic effect, the work of Song *et al.* [37] approximates each extracted region with optimally simplified shapes. In [34], dark and light tonal regions are extracted and used for the overall tonal balance in a rendering style using only a few tones. Region-based abstraction has also been used for video processing [11,39] in which case a key requirement is to ensure temporal coherence. In all of these works, global structures are not explicitly preserved. Structure preservation has been considered in [30] for non-photorealistic rendering, or for image enhancement [32, 25] and image editing [26], but their focus is local structures related to edges.

Some previous work has demonstrated that preserving large scale or global structure can produce attractive NPR results [17,42]. However, in such cases the structure curves are *manually* specified. Son et al. [36] use a global structural grid guided by the edge vector field for control of stippling. While structures are better preserved, it contains lots of detailed grid structure, and thus is more suitable for controlling a large number of primitives (e.g. stippling) and less effective for regions. Also, it may not be able to capture structures implied by severely fragmented features. In their work on generating Cubist style renderings, Collomosse and Hall [10] concluded that it would be desirable to enforce straight boundaries between regions, but were put off by the algorithmic complexity. Our NPR method takes on that challenge. Some other work that shows the importance of global structure in different applications is the manga layout system [7] and the picture story generation system [44]. The former determines appropriate placement of images using learning-based optimisation, while the latter takes text and user specified sketches as input to create a stylised sequence of images. However, neither of them take a single image and produce a stylised version.

In this work, we propose a novel algorithm that is based on various computer vision techniques, and produces an abstracted artistic rendering which not only preserves, but may also enhance global structures. The global structures could potentially be any significantly noticeable structures, such as lines, arcs etc. Among all the different structures, linear structures are probably most common in images, in particular since they remain linear under different viewing positions. Thus in this work, we focus on producing artistic abstraction with significant lines preserved and even enhanced.

Inspired by the concept of non-accidental feature alignment (collinearity, co-termination, etc.) which is used by both perceptual and computer vision systems [6] as a cue to identify salient structures, (perceptual rules were also applied by [29] to direct abstraction of architectural drawings), our approach selects lines with "multiple support" as the features which will provide the means to stylise images. Each of them corresponds to multiple, approximately collinear sections in the image. Not only do such lines tend to represent meaningful structures, but making fragmented linear structures stand out in the stylisation produces a pleasing artistic effect. Without semantic input, the approach is not perfect, but it is generally more robust and meaningful than simply using long edges.

Our algorithm is not suitable for images with too many (thus cluttered) or too few (thus not significant) linear structures, but as demonstrated by various examples throughout the paper, many images can benefit from the proposed technique to produce interesting results. Figure 1 demonstrates our algorithm; first, straight lines are detected – note that these lines have support from more than one section of significant object boundary. Next, image abstraction is obtained by rendering flat regions from segmentation with the constraints of the straight lines. It can be seen that the colour patches tend to be aligned to the straight line (the vertical boundary of the kite), thereby emphasising these structures across the image. Linear structures in the image may also be emphasised by exaggerating certain lines to be horizontal or vertical, creating some artistic effect (as shown in figure 1d).

An additional example is given in figure 3. Many linear structures are detected and emphasised by alignment of region boundaries. This effect can be subtle – some instances are highlighted with arrows. Also, significant structures are emphasised further by tweaking the brightness on both sides of lines.

Our contributions in this paper are

- to the best of our knowledge we are the first to use global structures in non-photorealistic rendering of general images
- to tackle this new problem required us to develop (based on existing computer vision methods) several new algorithms as part of our rendering pipeline, such as: line detection with multiple support, snapping and texture suppression.
- our results demonstrate that our proposed automatic rendering approach produces a novel and artistic effect.

Fig. 3 Our rendering has emphasised linear structures by aligning regions, as indicated by arrows (colour coded for different lines).

2 Method

Figure 4 shows the system pipeline. The first two steps are to automatically detect the global linear structure in the image. The Hough Transform is used for line detection since it is robust, and able to cope with fragmented support. We have observed that lines with multiple support are more suitable for producing pleasing artistic effects. Since there is no standard method for detecting multiple support lines we have developed a new approach. These lines are used as constraints in an image segmentation, which is performed by adapting standard techniques. Finally, the geometry and colour of the regions are simplified to produce a more artistic appearance, and the image rendered. This paper does not focus on the rendering stage, and so a simple approach is taken. Linear structures are further emphasised in rendering by tweaking the brightness on different sides of the lines using harmonic interpolation. Further details are provided as follows.

2.1 Line Detection

Line detection is performed using the Hough Transform [38] applied to edges detected using a Sobel operator with a 5×5 kernel. The polar representation of a line is used

$$x \cos\theta + y \sin\theta = \rho \tag{1}$$

with the origin at the centre of the image. An accumulator space of size $\left[-\frac{D}{2}, \frac{D}{2}\right]$ is built, where D is the length of the image diagonal. For each edge pixel at

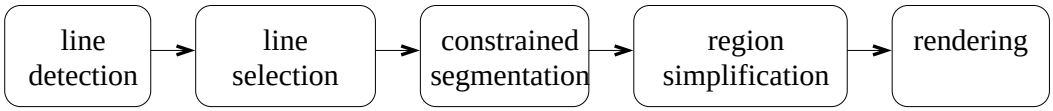


Fig. 4 System pipeline.

(x_i, y_i) with edge magnitude m_i and orientation θ_i the accumulator space is incremented by m_i over the range of bins (ρ'_i, θ'_i) defined by

$$x_i \cos\theta'_i + y_i \sin\theta'_i = \rho'_i \tag{2}$$

where $\theta'_i = [\theta_i - 5°, \theta_i + 5°]$. The value 5° is introduced to cope with errors and uncertainty in the estimation of the edge orientation. In our experiments the size of the (ρ, θ) bins in the accumulator space is set to $1° \times 1$ pixel.

The accumulator space is first smoothed slightly to remove plateaus, and then image lines are detected as maxima in a $m \times m$ square window (in our experiments we have set $m = 17$). Note that special care needs to be taken when the window straddles extreme θ values, since switching θ between 0° and 180° also involves a sign change in ρ. Peaks are retained if their magnitude is above a threshold, which has been set as 0.3 times the maximum accumulator value.

2.2 Multimodal Line Support

One of the strengths of the Hough Transform is that it can successfully detect lines (and other features) even from fragmented and cluttered data. In fact, for our purposes the most interesting situations are when support for a line is distributed across several sections, since we wish to tie these sections together using the common underlying line. For the contrary case, when the support for the line is not split, the extended line is not needed, and should generally be discarded.

To identify which lines detected by the Hough Transform are usable we analyse the distribution of edge magnitudes in the source image along the line being tested. At each point on the line the edge magnitudes are accumulated along the normal in both directions for a fixed distance (10 is used in our experiments). The desired lines with multiple sections of support will correspond to those with a multimodal distribution of edge magnitudes. To identify multimodal distributions we apply the criterion used by Otsu [31] for image thresholding, which finds the intensity threshold that divides the image into two classes so as to maximise the weighted between-class variance. For all the pixels on a line with positions $1 \ldots L$, the normalised edge magnitudes summed to 1 are treated as a probability distribution denoted as p_i. We assume that the line is segmented into two: L_1 with positions $[1 \ldots k]$ and L_2 with positions $[k+1 \ldots L]$. The idea is to find the best k that separates the line into two segments that maximise the between-class variance $\frac{\sigma_b^2(k)}{\sigma_T^2}$, where the between-class variance is

$$\sigma_b^2(k) = \omega_1(k)[\mu_1(k) - \mu_T]^2 + \omega_2(k)[\mu_2(k) - \mu_T]^2$$

the two class probabilities are

$$\omega_1(k) = \sum_{i=1}^{k} p_i \qquad \omega_2(k) = 1 - \omega_1(k)$$

the two class means are

$$\mu_1(k) = \sum_{i=1}^{k} i p_i / \omega_1(k) \qquad \mu_2(k) = \sum_{i=k+1}^{L} i p_i / \omega_2(k)$$

and the total mean is $\mu_T = \mu_1(L) = \sum_{i=1}^{L} i p_i$. The total variance $\sigma_T^2 = \sum_{i=1}^{L} (i - \mu_T)^2 p_i$ is constant over all thresholds, and so for image thresholding it is sufficient to find the threshold as $t = \arg_k \max \sigma_b^2(k)$ rather than $t = \arg_k \max \frac{\sigma_b^2(k)}{\sigma_T^2}$.

We shall set a threshold on the criterion for accepting a line, and so it seemed that using $\sigma_b^2(k)/\sigma_T^2$ rather than $\sigma_b^2(k)$ was more appropriate since it would provide a normalisation across different lines. However, this normalised value is known to be affine invariant (i.e. invariant w.r.t. any shift and scaling) [31] which means that some short and close clusters of support edges will be considered to be equal to longer and more separated edge clusters. This is not ideal in practise. We found it better to use $\sigma_b^2(k)$ with images rescaled to approximately the same size. Ideally, large values of between-class variance will correspond to multimodal distributions, while small values indicate a unimodal distribution. However, it is necessary to check for false positives, since distributions that are predominantly unimodal, but also containing well separated small modes, will also produce large values of $\sigma_b^2(k)$. Therefore we developed the following algorithm which uses the threshold t. The ratio of class probabilities is tested, and if

$$\frac{\min(\omega_1(k), \omega_2(k))}{\max(\omega_1(k), \omega_2(k))} < r$$

(where the ratio threshold has been set to $r = 0.3$ in our experiments) then the smaller class is deleted from the distribution (which is renormalised) and $\{\sigma_b^2(k), t\}$ is recomputed. This process is repeated until the ratio test

is satisfied. The between-class variance of the remaining edge magnitude distribution is then tested, and if it is above a threshold (which has been set to 10000 in our experiments for images rescaled to have about 0.5M pixels) then the line is retained as having significant multiple support. A consequence of not using the σ_T^2 scaling is that lines without multiple support may be retained if the lines are long enough to generate a large $\sigma_b^2(k)$ value. However, such an error of commission is preferable to an error of omission. Moreover, such a line is salient, and in any case, given its extent, most of the adjacent regions will already be aligned with it.

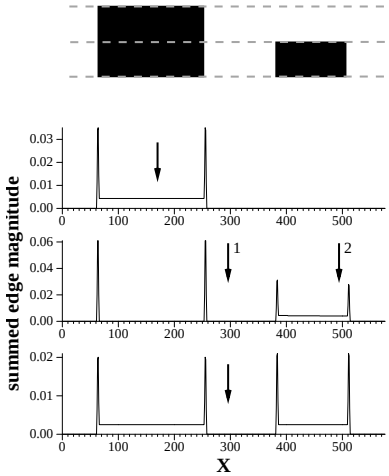


Fig. 5 Testing lines for multiple sections of support. Top: synthetic image containing two rectangles with three of the candidate lines detected using the Hough Transform, bottom: edge magnitude distributions of each of the candidate lines with threshold locations marked by arrows.

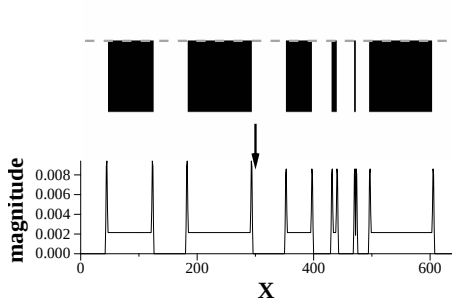


Fig. 6 Line testing with more than two sections of support. Top: synthetic image containing multiple rectangles with one of the candidate lines detected using the Hough Transform, bottom: edge magnitude distribution of the line.

The line selection process is illustrated in figure 5, which shows a simple synthetic image containing two rectangles overlaid with three of the candidate lines detected using the Hough Transform. The edge magnitude distributions of each line are displayed underneath. Edges are accumulated along the normals which creates the artifacts of large but narrow peaks when the normals are aligned with the vertical sides of the rectangles. For the top line the threshold is at the centre of the left hand rectangle, and a relatively low between-class variance is measured, causing the line to be rejected. The small isolated edge magnitude peaks in the middle line distribution cause the threshold (marked by arrow 1) to be shifted away from the main distribution. This is detected by the class ratio test, which causes the two narrow peaks on the left to be deleted. The new threshold (arrow 2) is now at the centre of the right hand rectangle, and also produces a relatively low between-class variance. The bottom line passes the ratio test and has a relatively high between-class variance, and is therefore classified as having multimodal support. Our method can cope with lines with more than two segments. An example is shown in figure 6 where more than two edge supports (including some very short responses) exist for a line. Our method successfully identifies the separation and recognises such lines (including another not shown) as well supported. In practise, to reduce the effect of spurious edges affecting the line support distribution, edges are downweighted if they have different orientations to the line being tested. The value at location i in the distribution is calculated as

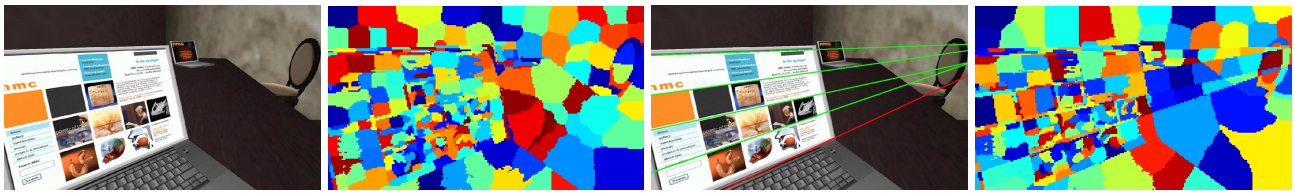
$$\sum_i m_i |\cos(\theta_i - \theta_L)|$$

where θ_L is the orientation of the line, and the summation is over the line segment normal to the line.

2.3 Constrained Segmentation

To produce abstracted rendering, we first decompose the input image into a few regions with consistent colour distribution. To enhance the artistic effect we wish the segmented boundaries to be snapped to the detected multiple support lines. This snapping will reinforce the linear structures. We adapt the multiscale normalised cut graph partitioning approach [13] to take linear constraints into account; alternative segmentation methods may also be used. The method builds a multiscale pairwise pixel affinity graph and uses a graph partitioning algorithm to derive the segmentation. Two cues are used, one based on pixel intensity and location closeness and the other based on the edge strength between pixels. For pixels i and j, the edge based affinity weight is modified with the second term in the exponent as

$$W_E(i,j) = \exp\left\{ - \max_{x \in line(i,j)} \|\nabla(x)\|^2 / \sigma_E - \lambda N_L(i,j) \right\},$$

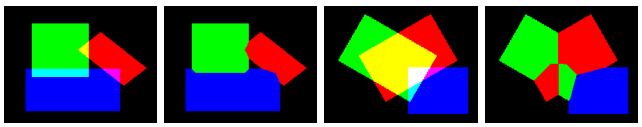


(a) (b) (c) (d)

Fig. 7 Using extracted lines to constrain segmentation. a) input image; b) segmentation result without constraint; c) lines extracted with Hough transform: those with multiple support (in green) are kept and those without are rejected (in red); d) segmentation result with constraints.

where $line(i,j)$ includes all the pixels forming the line segment connecting i and j, $\|\nabla(x)\|$ gives the edge magnitude, σ_E is the parameter controlling the drop-off of the weight (we used $\sigma_E = 1$ in the experiments), λ is a constant controlling the effect of linear constraints (we used $\lambda = 10$ for all the experiments) and N_L is the number of constraining lines the line segment between i and j crosses. The line equations for the detected multi-support lines can be precomputed and N_L can be obtained by checking how many times i and j sit on opposite sides of a line. For all the other parameters, default settings from [13] were used. An example is shown in figure 7(d) that produced segmentation regions well aligned with the detected lines in figure 7(c), compared with the result without constraints in figure 7(b); only the lines with multiple support (coloured green) are used.

2.4 Region Simplification

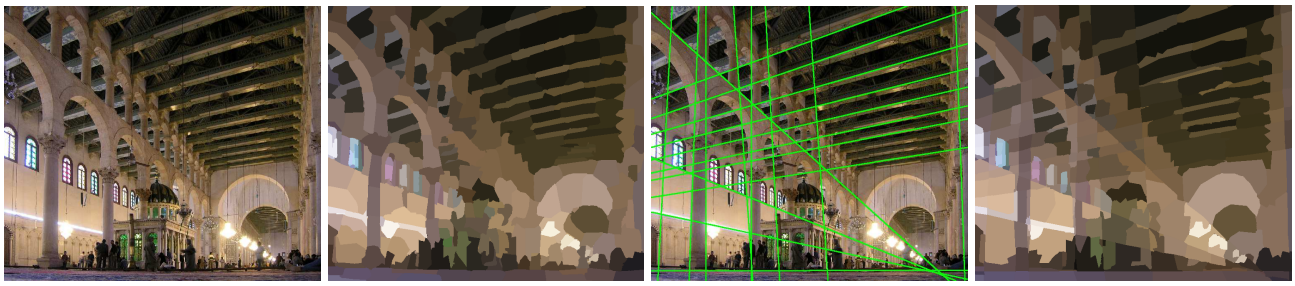


(a) (b) (c) (d)

Fig. 8 Combining overlapping regions: a,c) regions added, b,d) regions combined by our method.

To enhance the artistic effect it is desirable to simplify the region boundaries. This is a common practise by artists such as the Cubists where simplified and stylised shapes are used to enhance artistic effect. The region simplification could be guided by the underlying image to make it closer to the input; however, this will potentially restrict the level of abstraction. We use a simpler alternative strategy where simplification is performed independently of the input image. Since the simplified regions do not usually deviate much from the original regions, the recognisability of the produced abstraction is not lost while the artistic abstraction is enhanced.

A straightforward approach is to apply iterative mode filtering to an image containing the region labels. However, this has the drawback of shrinking and potentially deleting some regions. An alternative is to fit geometric primitives such as triangles and ellipses to the regions as was carried out by Song *et al.* [37]. However, this would provide too crude a representation of the image for our purposes since it would often obliterate the global line structure that we wish to enhance. Therefore we considered instead an approach for region simplification that provides sufficient fidelity and avoids shrinkage. We use standard polygonal approximation of the region boundaries [33]. Ideally, all regions should be simplified such that the interactions between regions (i.e. their common sections of boundary) should be taken into account. However, this would be excessively complicated, and despite the large literature on polygonal approximation we are not aware of any published solutions to this problem. While the closest work does optimise a polygonal approximation over multiple object boundaries, such that an appropriate variable number of breakpoints is appropriately allocated to each object [22], the object boundaries are separate and have no common sections. Therefore, we just compute the polygonal approximation of each region independently using a fixed threshold for the maximum allowable deviation. Since the simplification method treats each region independently it is likely that the resulting regions will not properly partition the image, and there will be gaps and overlaps between regions. To deal with overlaps, the innermost region label is selected as the most appropriate. The distance transform [3] is used to compute for each pixel in a region's interior the shortest distance to the region boundary. Then, for any pixel which contains multiple region labels, the label which has the maximum distance to the region boundary (i.e. the most "interior" pixel) is selected. Gaps are easily resolved by filling them with the original labels produced by the region growing. Figure 8 demonstrates the process in which three overlapping rectangular re-



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

Fig. 9 Comparative results without and with global structure enhancement. a) input image; b) rendering without global constraints; c) extracted lines; d) rendering with linear structure constrained segmentation;

gions are combined. They are shown added together in figure 8a, and after fusion in figure 8b. If there is deep inter-penetration of regions then the results of fusion are less intuitive, see figure 8c&d. However, since our region simplification is not extreme, this is not a problem for our rendering.

2.5 Rendering

Once the regions have been obtained various rendering styles can easily be applied. Our approach is a simple flat colouring, taking for each region the mean colour from the corresponding patch in the source image. The rendering style looks somewhat similar to that of [23, 45]; however, our focus is to demonstrate the effect of global linear structure and as shown later, our method can be combined with other rendering styles. The simple rendering style does have the advantage of avoiding the potential interaction between the rendering style and the enhancement. Otherwise, this can reduce the effects of the enhancement, as demonstrated by the examples in the paper.

To further emphasise the linear structures without being too distracting, we alter the rendered image by changing the brightness in different sides of each line. This ensures that the alteration is smooth other than at lines, where discontinuities are deliberately created to make the lines visible. This cannot be achieved by simply using unsharp masking since it can only be used to enhance edges already existing in the image, but cannot create new edges, which is desired in our application.

We define the adjustment field $f(p)$ for each pixel p as follows, assuming $d_{min} = \min_{l \in L} |dis(l,p)|$ is the minimum distance from p to any of the lines. L is the set of all the multi-supported lines. $dis(l,p)$ is the signed distance between point p and line l:

$$\begin{cases} f(p) = 0 & d_{min} < 1 \quad or \quad d_{min} > D_M \\ f(p) = \pm d & 1 \le d_{min} < 2 \\ \Delta f(p) = 0 & otherwise \end{cases},$$

where D_M determines the distance over which the field fades away. d specifies the step size of discontinuity at each line. We estimate the average intensity within a small distance (we used 5 pixels wide) and if the pixel satisfying $1 \le d_{min} < 2$ is located on the darker side, $-d$ is used to make it even darker and otherwise d is used. This has the effect of enhancing global contrast across the lines as the existing intensity difference is strengthened. Δ is the Laplacian operator $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, producing a harmonic field interpolating the boundary conditions. Laplacian interpolation gives a smooth field and the mean-value property ensures that $-d \le f(p) \le d$ holds for any pixel p. We used $D_M = 50$ and $d = 0.1$ for all the experiments. $f(p)$ can be efficiently obtained by solving a sparse linear system. We then modify each pixel in the HSV colour space by keeping H and S unchanged, and replace V at each pixel with

$$V(p) = clamp(V(p)(1 + f(p)), 0, 1),$$

where $clamp(\cdot, 0, 1)$ function limits the value to be within $[0, 1]$. Using multiplication instead of addition ensures more consistent perceptual change for both dark and light pixels since the human vision system senses brightness in a logarithmic manner (Weber's law). The effect is demonstrated in figure 10 where the linear structures are more clearly visible without being excessively distracting. The middle image shows the colour-coded interpolation field $f(p)$ where green is zero and red is positive (up to d) and blue is negative (up to $-d$).

3 Optional Refinements

3.1 Horizontal/Vertical Enhancement

When photographing scenes with horizontal or vertical structures, the structures will often be more prominent and the pictures more balanced if these structures are horizontal or vertical in the image. When composing

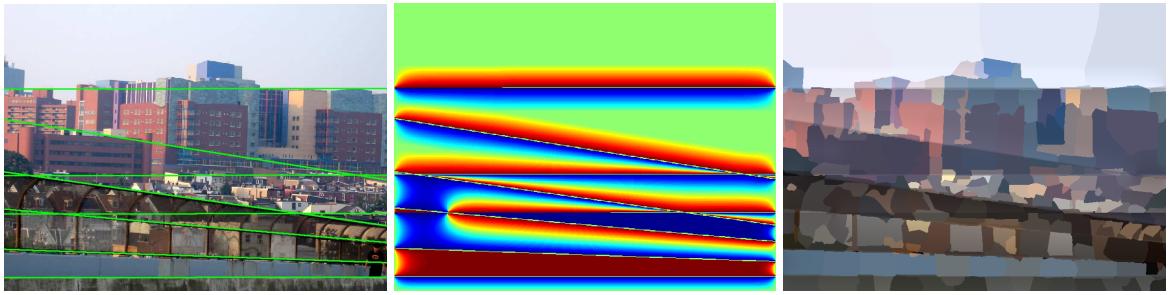


Fig. 10 Our abstracted rendering with line enhancement. From left to right: image with detected lines overlaid, colour-coded intensity adjustment field, rendering with lines enhanced.

visual images, horizontal and vertical lines have special significance. Horizontal lines suggest a feeling of rest, vertical lines suggest a feeling of grandeur, while a combination of horizontal and vertical lines suggests stability and solidity. Thus, horizontals and verticals appear frequently in images. Moreover, incorrect alignment of horizontal lines may cause the image to appear unbalanced [8].

Based on the detected lines, we propose as an optional step to further enhance or exaggerate horizontal/vertical alignment for artistic effect by applying a deformation. Deformation has often been used to provide a caricature effect [9], although has not been generally used for NPR. For this purpose, we detect all the significant lines using the Hough transform (not necessarily with multiple support) and if the lines are within θ_L from perfect horizontal or vertical (which is typically set to 6° in our experiments), we slightly deform the image such that these lines are perfectly aligned by keeping the same ρ and setting θ to either 0° or 90° from the results of Hough transform. We treat these lines before and after deformation as constraints to deform the images. Consistent intersection points between two lines of roughly orthogonal directions will be used to split lines into segments for use as constraints. This ensures that intersection points are well preserved and the deformation is natural. We use moving least squares based image deformation [35] because it is efficient and formulates directly with line segment constraints. Line detection may not be perfect thus the deformed images may still have structures slightly deviated from perfect lines. Our general pipeline further enhances this as we make our rendering snapped to detected lines during segmentation. By combining these, we can emphasise horizontal/vertical structures more substantially than the input.

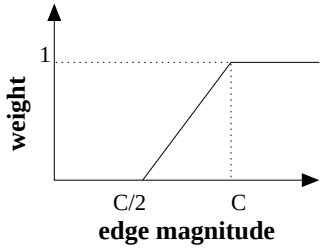


Fig. 11 Weighting function of neighbourhood pixels for texture suppression; C is the edge magnitude of the central pixel in the window.

3.2 Texture Suppression

Edge detectors are typically responsive to textures containing lots of fine detail, and so images containing substantial texture can result in the Hough Transform detecting undesirable lines. One solution is to apply a texture suppression post-processing step to reduce such spurious edge responses due to texture. We have developed a simple two part method that gives comparable or better results to the well established approach in [16].

The first part removes edges if their neighbourhood contains a large proportion of equal or stronger edges. Somewhat weaker edges than the central edge are also taken into account, but a linear weighting function is used so that they have less effect – see figure 11. An additional factor is taken into account: the coherence of edges in the neighbourhood, which is measured as the circular variance [27] of edge directions. At each location (x, y) the following is calculated

$$S_{xy} = (1 - r_{xy}) \sum_{\{s,t\} \in \mathcal{N}(x,y)} w_{xy}(s,t)$$

where $(1 - r_{xy})$ is the circular variance, $\mathcal{N}$ is the neighbourhood, and w the weighting function. The edge is retained if

$$S_{xy} < p|\mathcal{N}|.$$

In our experiments the neighbourhood was set to a 23×23 window, and proportion $p = 0.75$. The second part uses Bischof and Caelli's [2] edge lifetime. Edges

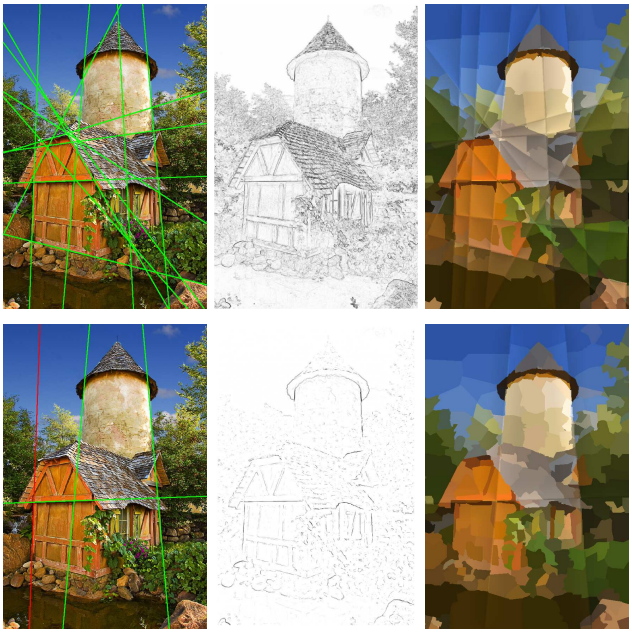


Fig. 13 Comparative results without (top row) and with (bottom row) texture suppression. Left column: original images are shown overlaid with the detected lines; middle column: edges; right column: our renderings.

are extracted from the image over a range of scales using the Canny detector. The edges are tracked across adjacent scales, and their gradient magnitudes at each scale are summed during tracking to provide a more stable measure of edge strength. Since this approach produces thin edges they were dilated and combined with the result of Canny's edges prior to the non-maximal suppression stage to produce thicker edges.

The results of these two approaches are fairly effective, although they still retain some spurious edges (see figure 12c&d). Since the errors tend to be independent the results can be combined by a `MIN` operation to produce a cleaner edge map (figure 12e) which is substantially better than the original Canny edge map (figure 12b) and also cleaner than Grigorescu *et al.*'s result (figure 12f). Further results of texture suppression are shown in figure 13.

4 Experiments

In this section, various artistic rendering results produced with our method are given. The algorithm is fully automatic. Although a few parameters exist in the algorithm pipeline, the only parameter that needs to be adjusted is the horizontal/vertical enhancement threshold θ_L, and we have shown various examples to demonstrate its effect. Except for this, for images with normalised size containing about 0.5M pixels, a fixed set

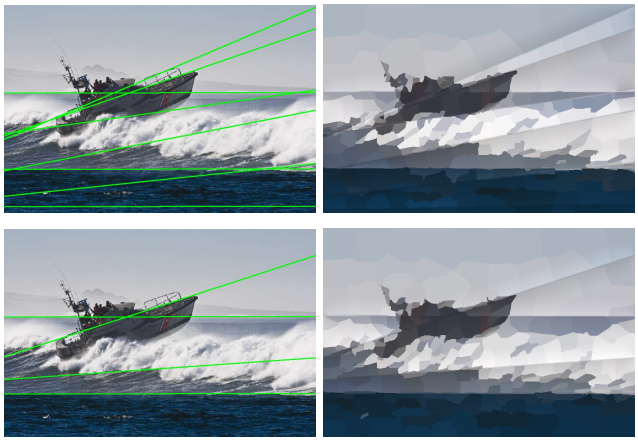
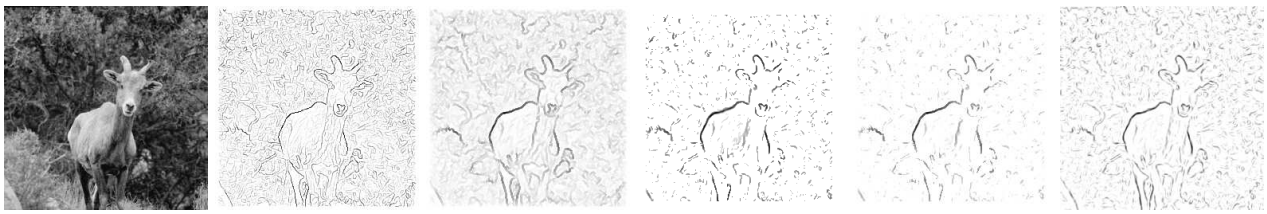


Fig. 14 Comparative results without (top row) and with (bottom row) texture suppression. Left column: original images are shown overlaid with the detected lines; right column: our renderings.

Fig. 15 Comparative results with enhancement of horizontal and vertical structures. Left column: original images; middle column: our renderings without global deformation; right column: our renderings with global deformation.

of parameters, as given in the paper, work well and are used for all the examples in the paper. We carry out experiments on a computer with a 3.4GHz Intel i7-3770 CPU. The running times depend on the linear structures detected. Using the current unoptimised code, on average the basic pipeline takes 137.1 seconds. The optional steps take the following average times: snapping with constrained segmentation 9.4 seconds, region simplification 138.6 seconds, rendering with line enhancement 6.7 seconds, horizontal/vertical enhancement 52.1 seconds and texture suppression 14.9 seconds.



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

Fig. 12 Texture suppression: a) original image; b) Canny edges; c) tracked edges displaying lifetime [2]; d) edges after suppression by testing the proportion of strong edges in the neighbourhood; e) combination of (c) and (d); f) edges after suppression with [16]

Fig. 16 Enhancement of horizontal and vertical structures. In the middle small deviations ($\theta_L = 6°$) from horizontal/vertical are realigned. For greater artistic effect the rendering on the right shows extreme deformation ($\theta_L = 22°$).

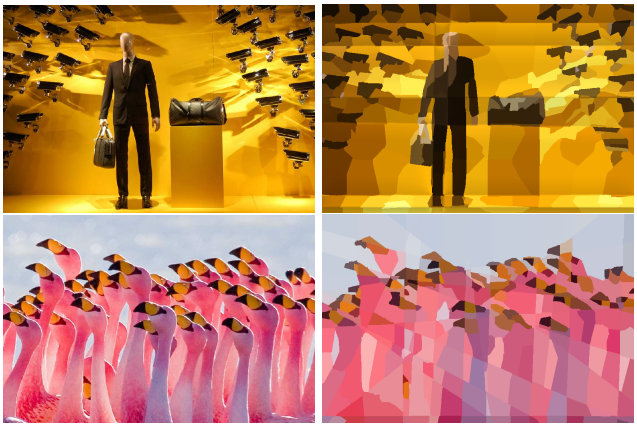


Fig. 17 More results. Left column: input images; right column: our renderings.

Our method automatically extracts linear structures from input images and identifies those with multiple support, as shown in figures 1, 7, 9, 10, 13 and 14 with lines having multiple support rendered in green and others rendered in red and ignored for further processing. As shown in these examples, significant linear structures identified are preserved or even enhanced in the highly abstracted rendering.

Figure 7 shows an example of segmentation with and without line constraints. Global linear structures shown in (c) are preserved and even enhanced as shown in (d). Comparative results without (a)(b) and with (c)(d) global linear structures are shown in figure 9. With linear structure being considered, the abstracted rendering looks more artistic as some regularities are produced. Object boundaries which are close to linear structures are likely to be snapped to them, producing more regular results.

Figure 13 shows how the foliage produces many false responses in the Sobel edge map, and consequently many spurious lines are detected by the Hough transform. Performing texture suppression eliminates the majority of these texture edges, yielding a cleaner line map. Another example is demonstrated in figure 14 in which texture suppression removes some edges arising from the ripples in the water. Note the effective use of the multiple support lines, in particular the horizontal through the boat, emphasising the horizon.

Three examples are shown in figure 15 where initially our renderings emphasise multiple support, such as the left ear with collar, and the right ear with the hat rim (top example), the two skyscrapers (middle example) and the book and bowl (bottom example). The top example demonstrates that even if the sources of multiple support are not semantically connected the resulting enhancement is still artistically effective. These linear structures are further emphasised by aligning some of them to horizontal or vertical, producing more exaggerated artistic looking results. The tolerance θ_L for deformation is set to the default of 6° unless specified, but may be adjusted, leading to different artistic effects. An example is given in figure 16 where a small threshold makes the tie vertically aligned, and a large threshold also makes both sides of the body vertical, leading to a more extreme exaggeration of the global structures.

Our artistic renderings for a variety of images are shown in figure 17. Global linear structures are well preserved. Interesting structures are emphasised, producing more artistic looking results, by slight deformation (top image) and snapping to the lines. The bottom image shows an example without line enhancement; since

most neighbouring regions are highly contrasted the effect of snapping to the lines is still clearly visible.

Our technique is orthogonal to many non-photorealistic rendering techniques [4,23,20] and thus can be combined with them. This is demonstrated in figure 18 in which the following rendering styles have been used: watercolour [4], conte, pastel and cartoon (the last three are gimp plugins) – others could also have been used. Our pipeline is modified to use the alternative rendering algorithms in place of the flat region colouring. Since the regions are not used the snapping step is also not applied. Further results are shown in the gallery in figure 19.

5 Conclusion

In this paper, inspired by artists, we propose a novel algorithm for abstracted rendering of images with global structures preserved and enhanced. Highly abstracted and artistic looking results are obtained fully automatically. The algorithm is based on various computer vision techniques to extract global (linear) structures with support from multiple boundary sections, and produce regions that consider both global structures and local colour distributions. We demonstrated that the algorithm works well on a variety of types of images. The work can potentially be improved in various ways.

- While linear structures occur most widely and can be detected most reliably, other structures such as circular or elliptical arcs may also be interesting to emphasise. However, circles in general become ellipses and therefore detecting circles are not useful for general scenes. Meanwhile, techniques for detecting elliptical sections (such as the Hough Transform) are much less reliable and computationally expensive. These problems could be overcome by introducing a modest amount of user interaction to highlight salient features in the image.
- A different type of structure to enhance would be vanishing points. This should be both effective, since vanishing points and their associated lines will be present in many images, and practical, since their detection is reasonably reliable.
- Another potential extension would be to use techniques such as [1] to detect long salient edges of arbitrary shape. However, the results are still local and cluttered, and would require substantial selection and grouping to find global structures with multiple support.

Our method is not suitable for all images. If significant line segments were distributed randomly, the chances of having coincident lines with multiple support would be small. However, in practice, we have observed a much higher chance of having multiple support lines in real images, due to for instance regular alignment of objects in the scene or occlusion, as demonstrated in the paper. Since no semantic information is available, the detected lines may not correspond to semantically meaningful structures; however, as shown by various examples in the paper, interesting and aesthetically pleasing results are obtained in many cases.

Humans are more sensitive to faces and may expect more details in faces so as to be more recognisable. Established face detectors can be used to identify the face areas and treat them with special care. The idea and technique proposed in this paper is general and can be combined with other non-photorealistic rendering techniques to produce artistic effects preserving global structures. Since the proposed method does not incorporate and respect semantic knowledge, it is possible for the global linear constraints to be misapplied, e.g. distorting human faces will probably look displeasing.

A potential modification to our approach is to apply the snapping and harmonic field based line enhancement to large structures with limited extent rather than globally. However, it is arguable whether global or large structure is preferred. Detecting line segments rather than lines requires estimating additional parameters, and so is more complicated and less reliable.

Currently region snapping only applies close to the global linear structure. This could be extended to neighbouring regions by propagation which may help strengthen the effect in certain cases. However, since such alignment may not look attractive or appropriate in some cases, in particular when a large number of lines are detected, we feel it is better to restrict alignment to the locality of global structures.

Similar to other non-photorealistic rendering research, it is difficult to evaluate the aesthetic aspect of the work, due to the subjective nature and the inherent challenge of quantifying aesthetics. A partial solution to this is a user study, which is left as future work. The user study may also help to improve the parameter settings to better align with perceptions [5].

References

1. Bhat, P., Zitnick, C.L., Cohen, M., Curless, B.: Gradientshop: A gradient-domain optimization framework for image and video filtering. ACM Trans. Graph. **29**(2), 10:1–10:14 (2010)
2. Bischof, W., Caelli, T.: Parsing scale-space and spatial stability analysis. Computer Vision, Graphics and Image Processing **42**, 192–205 (1988)
3. Borgefors, G.: Distance transforms in arbitrary dimensions. Computer Vision, Graphics and Image Processing **27**, 321–345 (1984)

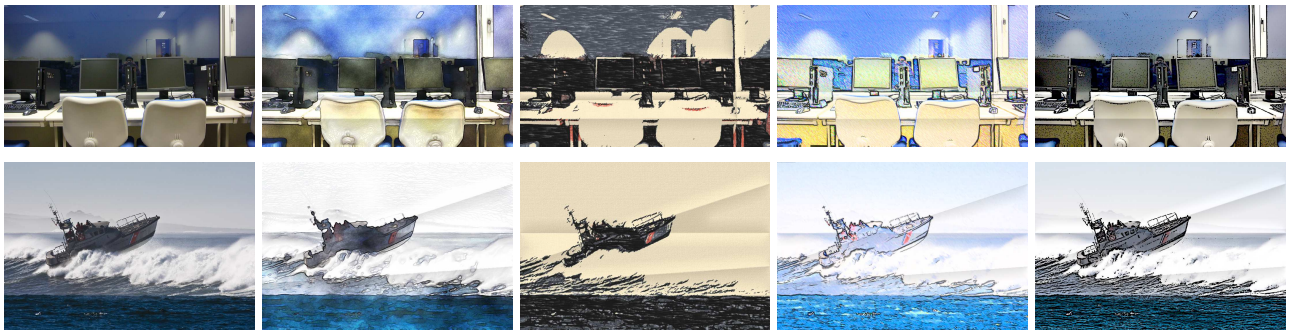


Fig. 18 Applying our global structure enhancement to various rendering styles (watercolour, conte, pastel and cartoon).

4. Bousseau, A., Kaplan, M., Thollot, J., Sillion, F.X.: Interactive watercolor rendering with temporal coherence and abstraction. In: ACM Symp. Non-photorealistic Animation and Rendering, pp. 141–149 (2006)
5. Bousseau, A., O'shea, J.P., Durand, F., Ramamoorthi, R., Agrawala, M.: Gloss perception in painterly and cartoon rendering. ACM Trans. Graph. **32**(2), 18:1–18:13 (2013)
6. Boyer, K.L., Sarkar, S.: Perceptual organization in computer vision: status, challenges, and potential. Computer Vision and Image Understanding **76**(1), 1–5 (1999)
7. Cao, Y., Chan, A.B., Lau, R.W.H.: Automatic stylistic manga layout. ACM Trans. Graph. **31**(6), 141:1–141:10 (2012)
8. Child, J.: Studio Photography: Essential Skills. Taylor and Francis (2008)
9. Clarke, L., Chen, M., Mora, B.: Automatic generation of 3D caricatures based on artistic deformation styles. IEEE Trans. Vis. Comp. Graph. **17**(6), 808–821 (2011)
10. Collomosse, J.P., Hall, P.M.: Cubist style rendering from photographs. IEEE Trans. Vis. Comp. Graph. **4**(9), 443–453 (2003)
11. Collomosse, J.P., Rowntree, D., Hall, P.M.: Stroke surfaces: temporally coherent non-photorealistic animations from video. IEEE Trans. Vis. Comp. Graph. **11**(5), 540–549 (2005)
12. Cong, L., Tong, R., Dong, J.: Selective image abstraction. The Visual Computer **27**(3), 187–198 (2011)
13. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: Proc. Comput. Vis. Patt. Recog., pp. 1124–1131 (2005)
14. DeCarlo, D., Santella, A.: Stylization and abstraction of photographs. In: SIGGRAPH, pp. 769–776 (2002)
15. Gooch, B., Coombe, G., Shirley, P.: Artistic vision: painterly rendering using computer vision techniques. In: ACM Symp. Non-photorealistic Animation and Rendering, pp. 83–90 (2002)
16. Grigorescu, C., Petkov, N., Westenberg, M.: Contour and boundary detection improved by surround suppression of texture edges. Image Vision Comput. **22**(8), 609–622 (2004)
17. Hausner, A.: Simulating decorative mosaics. In: SIGGRAPH, pp. 573–580 (2001)
18. Hertzmann, A.: Painterly rendering with curved brush strokes of multiple sizes. In: SIGGRAPH, pp. 453–460 (1998)
19. Hiller, S., Hellwig, H., Deussen, O.: Beyond stippling – methods for distributing objects on the plane. Comput. Graph. Forum **22**(3), 515–522 (2003)
20. Huang, H., Fu, T.N., Li, C.F.: Painterly rendering with content-dependent natural paint strokes. Vis. Comput. **27**(9), 861–871 (2011)
21. Kang, H., Lee, S.: Shape-simplifying image abstraction. Computer Graphics Forum **27**(7), 1773–1780 (2008)
22. Kolesnikov, A., Fränti, P.: Data reduction of large vector graphics. Pattern Recognition **38**(3), 381–394 (2005)
23. Kypianidis, J.E., Kang, H.: Image and video abstraction by coherence-enhancing filtering. Comp. Graph. Forum **30**(2), 593–602 (2011)
24. Kyprianidis, J.E., Collomosse, J., Wang, T., Isenberg, T.: State of the "art": A taxonomy of artistic stylization techniques for images and video. IEEE Trans. Vis. Comp. Graph. **19**(5), 866–885 (2013)
25. Li, X.Y., Gu, Y., Hu, S.M., Martin, R.: Mixed-domain edge-aware image manipulation. IEEE Trans. Image Processing **22**(5), 1915–1925 (2013)
26. Ma, L.Q., Xu, K.: Efficient antialiased edit propagation for images and videos. Computers & Graphics **36**(8), 1005–1012 (2012)
27. Mardia, K.: Statistics of Directional Data. Academic Press (1972)
28. Mould, D.: A stained glass image filter. In: Eurographics Workshop on Rendering Techniques, pp. 20–25 (2003)
29. Nan, L., Sharf, A., Xie, K., Wong, T.T., Deussen, O., Cohen-Or, D., Chen, B.: Conjoining gestalt rules for abstraction of architectural drawings. ACM Trans. Graph. **30**(6), 185 (2011)
30. Orzan, A., Bousseau, A., Barla, P., Thollot, J.: Structure-preserving manipulation of photographs. In: ACM Symp. Non-photorealistic Animation and Rendering, pp. 103–110 (2007)
31. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Sys., Man, and Cyber. **9**, 62–66 (1979)
32. Paris, S., Hasinoff, S.W., Kautz, J.: Local laplacian filters: edge-aware image processing with a laplacian pyramid. ACM Trans. Graph. **30**(4), 68:1–68:11 (2011)
33. Ramer, U.: An iterative procedure for the polygonal approximation of plane curves. Computer, Graphics and Image Processing **1**, 244–256 (1972)
34. Rosin, P.L., Lai, Y.K.: Towards artistic minimal rendering. In: ACM Symp. Non-photorealistic Animation and Rendering, pp. 119–127 (2010)
35. Schaefer, S., McPhail, T., Warren, J.D.: Image deformation using moving least squares. ACM Trans. Graph. **25**(3), 533–540 (2006)
36. Son, M., Lee, Y., Kang, H., Lee, S.: Structure grid for directional stippling. Graphical Models **73**(3), 74–87 (2011)
37. Song, Y., Hall, P., Rosin, P.L., Collomosse, J.: Arty shapes. In: Comp. Aesthetics, pp. 65–73 (2008)

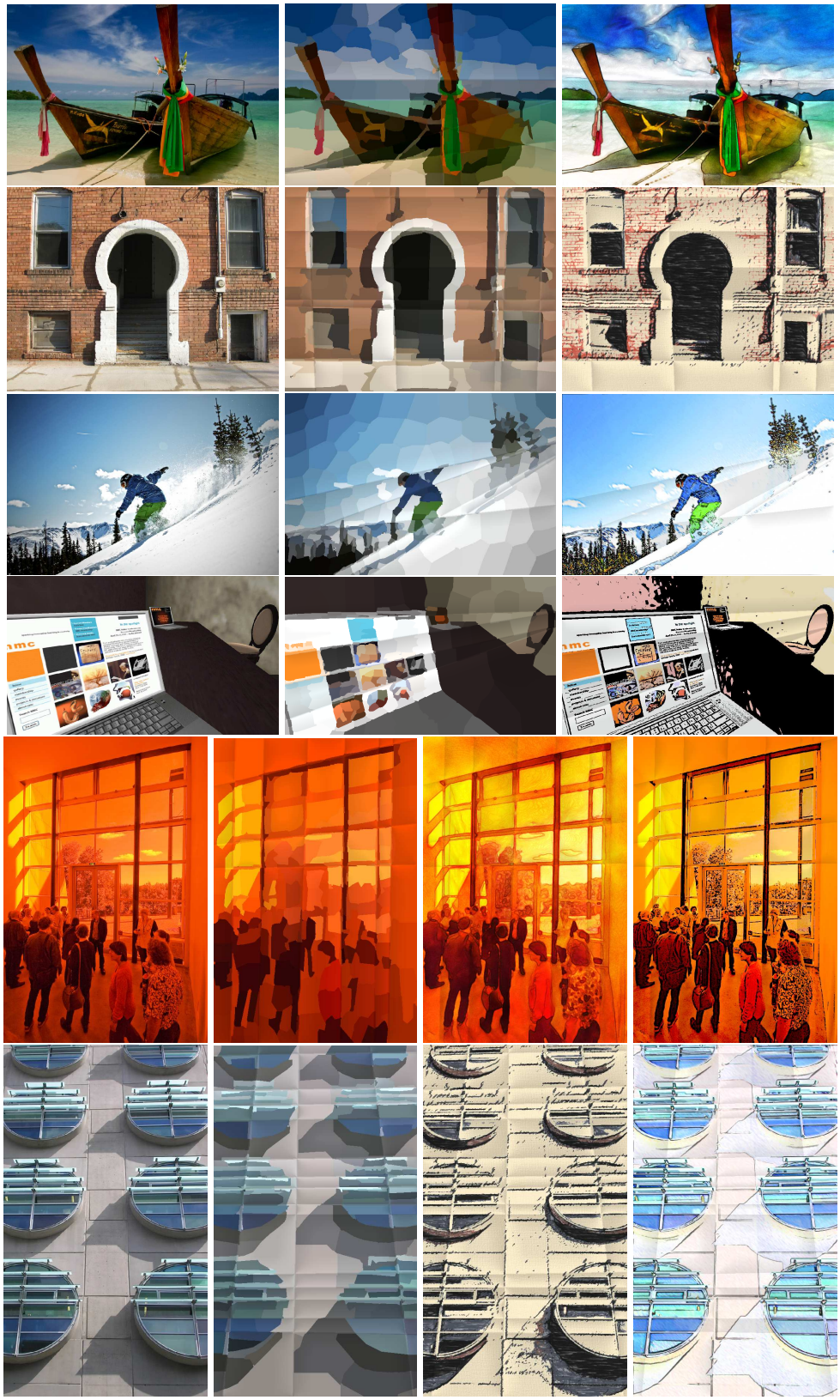


Fig. 19 Gallery of results. Rows 1-4 (left to right): input image, global structure enhancement of images rendered in the style described in the paper, enhanced images rendered in watercolour, conte, pastel and cartoon styles respectively. Rows 5-6: as above, but each with two alternative rendering styles.

38. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Thomson-Engineering (2007)
39. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.F.: Video tooning. ACM Trans. Graph. **23**(3), 574–583 (2004)
40. Wen, F., Luan, Q., Liang, L., Xu, Y.Q., Shum, H.Y.: Color sketch generation. In: ACM Symp. Non-photorealistic Animation and Rendering, pp. 47–54 (2006)
41. Winnemöller, H., Olsen, S., Gooch, B.: Real-time video abstraction. ACM Trans. Graph. **25**(3), 1221–1226 (2006)
42. Xu, J., Kaplan, C.S.: Calligraphic packing. In: Graphics Interface, pp. 43–50 (2007)
43. Yang, C.K., Yang, H.L.: Realization of Seurat's pointillism via non-photorealistic rendering. The Visual Computer **24**(5), 303–322 (2008)
44. Zhang, J., Hao, Y., Li, L., Sun, D., Yuan, L.: StoryWizard: a framework for fast stylized story illustration. The Visual Computer **28**(6-8), 877–887 (2012)
45. Zhang, S.H., Li, X.Y., Hu, S.M., Martin, R.R.: Online video stream abstraction and stylization. IEEE Trans. Multimedia **13**(6), 1286–1294 (2011)