Chapter 11
Non-Photorealistic Rendering with Reduced Colour Palettes

Yu-Kun Lai and Paul L. Rosin

Abstract In contrast to photorealistic rendering, where richer colours are likely to be preferred, non-photorealistic rendering can often benefit from some abstraction, and colour palette reduction is one direction. By using a small number of carefully selected colours the overall tonal distribution can be well expressed with less visual clutter. This is also essential to simulate certain art forms, such as cartoons, comics, paper-cuts, woodblock printing, etc. that naturally prefer or require reduced palettes. In this chapter we will summarise major techniques used in colour palette reduction, such as region segmentation, thresholding and colour palette selection. Most approaches consider images as input and generate stylised image renderings while some work also considers video stylisation, in which case temporal coherence is essential. We finish this chapter with some discussions about potential future directions.

11.1 Introduction

In fine art the limitations of the media often dictate that the works employ a small number of tones or colours, e.g. pen and ink, woodblock, engraving, posters. However, artists have worked to take advantage of such restrictions, producing masterpieces that prove that "less is more". This chapter describes NPR techniques that also work within the confines of a reduced palette, from the extreme case of binary renderings to those that use a handful of tones or colours.

Yu-Kun Lai

School of Computer Science and Informatics, Cardiff University, United Kingdom e-mail: Yukun.Lai@cs.cardiff.ac.uk

Paul L. Rosin

School of Computer Science and Informatics, Cardiff University, United Kingdom e-mail: Paul.Rosin@cs.cardiff.ac.uk

Fig. 11.1 An example similar to the abstracted black and white designs produced as spin-offs from The Blues Brothers film.

A reduced palette is also common in popular culture. A good example is the iconic representational style of The Blues Brothers that has spun out of the classic cult film. Mostly dating after the film's release, these graphics have developed a distinctive style (see Fig. 11.1 for one that we have created). They appear to be generated by applying some simple filters to source colour images taken from the original film, with the processing pipeline something like the following: foreground extraction → blurring → colour to gray scale conversion → binary thresholding → additional manual correction. Thus, the first four steps perform abstraction, while the last step preserves salient details and ensures that the aesthetic quality is satisfactory.

In fact, the above exemplifies the goals of non-photorealistic rendering with reduced colour palettes and the steps involved. As is normal for image based NPR, geometric abstraction is required to remove the noise and extraneous detail present in captured images. Moreover, as the number of tones/colours decreases and the image's information content is further reduced, ensuring that significant details are preserved becomes even more critical. Thus, the overall goal can be summarised as a combination of geometric and photometric abstraction along with detail preservation.

There is a large computer vision literature containing methods for processing images that could be applied to the task of simplifying the geometric and photometric properties of images. However, when applied naïvely they tend not to be effective for producing pleasing "artistic" renderings. For example, when a standard thresholding algorithm [23] is applied to a grey level version of the image in Fig. 11.2(a) the result is unattractive, and moreover contains blotches, speckles, and has lost many significant details (Fig. 11.2(b)). In fact, no single threshold is sufficient to produce a satisfactory result. Detail can be better retained if the global threshold is replaced by a threshold surface created by interpolating local thresholds calculated in subwindows – Fig. 11.2(c). However, a consequence is that noise and other unwanted detail is also increased. Alternatively, Fig. 11.2(d) shows the results of first segmenting [9] the image, and then re-rendering the regions so that each is given a flat colour consisting of the mean colour of that region in the source colour image. Although some details and structures have been captured better, others are lost, and the effect is unattractive. Another segmentation (using graph cuts [3]) shown in

Fig. 11.2 Unsuccessful attempts to render images using computer vision algorithms. (a) source colour image, (b) image with global thresholding, (c) image with local thresholding, (d) segmented image with regions rendered with the mean colour from the corresponding source colour image pixels, (e) alternative segmentation, (f) image with colour quantisation using median cut (10 colours).

Fig. 11.2(e) is more successful, but still contains visual clutter while losing salient details. Using adaptive colour quantisation such as the median cut algorithm [12], the result in Fig. 11.2(f) preserves some details, but is not aesthetically pleasing due to the spurious boundaries caused by quantisation; a subtle change of colour may lead to the pixel being quantised to a different colour.

Whilst describing a range of reduced palette techniques, the scope of this chapter does not cover methods that are focused on rendering using strokes or other small primitives. Some of these are described in detail elsewhere in this book—stippling and halftoning (Chapter 3), hatching (Chapter 4), pen and ink (Chapter 6), path
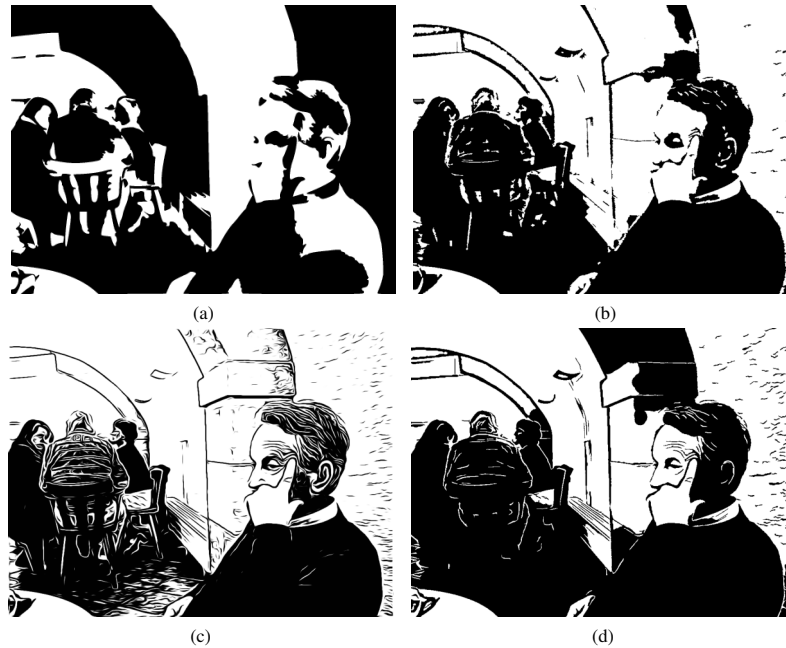
Fig. 11.3 Examples of rendering with a binary palette. (a) Xu and Kaplan [35], (b) Mould and Grant [21], (c) Winnemöller [32], (d) Rosin and Lai [25].

based rendering (Chapter 9) and mosaicing (Chapter 10)—although other highly specialised techniques such as digital micrography [19], ASCII art [37], etc. are outside the scope of this book. Another related but complimentary topic, focussing more on the creative use of regions, is provided in Chapter 7.

11.2 Binary Palettes

Converting a colour image to black and white is the most extreme version of colour palette reduction. While it is the most challenging case, it is also the most interesting, and can potentially produce the most dramatic stylisation. Although image thresholding has been popular in the image processing community for over 50 years, it is only recently that it has become of significant interest to the NPAR community. To differentiate NPAR methods from traditional thresholding algorithms, Xu and Kaplan [35] coined the term "artistic thresholding" to describe approaches to thresholding that try to preserve the visibility of objects and edges.

Xu and Kaplan's [35] method starts by over-segmenting the input image, to produce small regions (i.e. superpixels), which are connected to form a region adjacency graph. The assignment of black or white labels to the regions will define a binary segmentation of the image. Each region is described by a set of properties:

its average colour, area; also the lengths of the shared boundaries between adjacent regions are stored. Several cost functions are then designed to capture specific aspects of a potential labelling: 1/ The normalised pixelwise difference between the binary assignment and the intensity of the source image. 2/ The difference between the current proportion of black in the segmented image and a target black proportion. 3/ Two cost functions are defined to take into account the effectiveness of the labelling of adjacent regions, since adjacent regions with contrasting colours should be assigned opposite black and white labels, while adjacent regions with similar colours should be assigned the same labels. They are computed using both the difference in intensity between the labelled region and the source image pixels and the amount of common region boundary. 4/ If the image has been partitioned into a number of high-level components then there is a cost function to encourage a homogeneous distribution of the labels amongst these components. Finally, these individual cost functions are combined as a weighted sum, where the weights can be chosen to produce different artistic effects. The overall cost function is optimised using simulated annealing. Perturbing individual region labels provides too local an effect, and so sub-graphs of size 3–5 vertices are modified by enumerating all label assignments and choosing the one with the lowest cost. A post-processing step tidies up the results, applying mathematical morphology opening and closing operations to remove small regions and increase abstraction. In addition, lines are inserted between adjacent regions if their colour difference is above a threshold. An interesting consequence of the cost function is that segments are sometimes inverted from their natural tone so as to maintain boundaries; see Fig. 11.3(a).

The approach by Mould and Grant [21] initialises their binary rendering using adaptive thresholding (see Fig. 11.2(c)). Next, a cost function is minimised using the graph cut algorithm. Let the intensity at each pixel be I_p, the local mean intensity in the neighbourhood of pixel p be μ_p, and the global variation in image intensity be σ. The basic data term is defined as the distance from the local mean

$$\frac{1}{2}e^{-\left(I_p-\mu_p\right)^2/2\sigma^2}.$$

This is later modified by weighting with a penalty function in an attempt to down-weight outliers. The smoothness term has a similar form applied to image gradients, and penalises differences in neighbouring intensities. To balance abstraction and preservation of detail, Mould and Grant separate the image into a simplified base layer and a detail layer. The base layer is generated by applying the graph cut algorithm with the cost function modified to increase the smoothness term and decrease the data term. For the detail layer, as well as determining black and white pixels by adaptive thresholding, a third class is included whose intensities are as yet uncertain (a "don't know" class) if $|I_p - \mu_p| \leq 0.4 \times \sigma$. The known values in the detail layer are refined using graph cuts, and combined with the base layer by a further energy minimisation. The cost function takes its data term values from the detail layer except for those locations with "don't know" labels which use the base layer energies. The smoothness term from the detail layer is reused. This further optimisation is

only carried out in the detail region and in a 5 pixels wide band around it. Finally, the result is tidied up by removing small isolated regions; see Fig. 11.3(b).

The Difference of Gaussians (DoG) operator that has been commonly used in computer vision for edge detection has been extended by Winnemöller [32] to produce a filter capable of generating many styles such as pastel, charcoal, hatching, etc. Of relevance to this section is its black and white rendering with both standard black edges and negative white edges. The DoG is modified so that the strength of the inhibitory effect of the larger Gaussian is allowed to vary according to the parameter τ:

$$D_{\sigma,k,\tau}(x) = G_\sigma(x) - \tau \times G_{k\sigma}(x)$$

and the output is further modified by an adjustable thresholding function that allows soft thresholding:

$$T_{\varepsilon,\psi}(u) = \begin{cases} 1 & \text{if } u \geq \varepsilon \\ 1 + \tanh(\psi(u - \varepsilon)) & \text{otherwise} \end{cases}$$

τ controls the brightness of the result and the strength of the edge emphasis, ψ controls the sharpness of the black/white transitions, and ε controls the level below which the adjusted luminance values will approach black. Since τ and ψ are coupled it was found more convenient to reparameterise the filter as

$$S_{\sigma,k,\rho}(x) = (1+\rho)G_\sigma(x) - \rho \times G_{k\sigma}(x)$$

so that the new parameter controls just the the strength of the edge sharpening effect. For black and white images a large value of ψ is used. The DoG output contains both minima and maxima, and so increasing their magnitude by setting large values of ρ will encourage negative edges (i.e. white lines on a black background). To improve the spatial coherence of the results, the isotropic DoG filter is replaced by the anisotropic flow-based DoG [14]. See Fig. 11.3(c).

Rosin and Lai's [25] 3-tone method described in the next section can also be adapted to produce binary renderings. In brief, it uses the flow-based DoG to extract both black and white (negative) lines (cf. Winnemöller's [32] approach). The source image is thresholded, and the lines are drawn on top; see Fig. 11.3(d) and Fig. 11.4(b).

Meng *et al.* [20] described a specialised method to generate paper-cuts restricted to human portraits. These are effectively binary renderings with the extra constraint that the black pixels form a single connected region. They use a simplified version of the hierarchical composition model [38] which represents the face by an AND-OR graph. Nodes in this graph represent facial components (e.g. left eyebrow, right eye, nose); the AND nodes provide the decomposition of the face, while the OR nodes provide alternative instances for facial components. The latter are created by artists who manipulated photographs to extract a set of binary templates for typical mouths, eyes, etc. which form the leaves of the AND-OR graph. Facial features are located in the source image by fitting an active appearance model [7], and local thresholding is applied at each feature's bounding subwindow to produce a set of
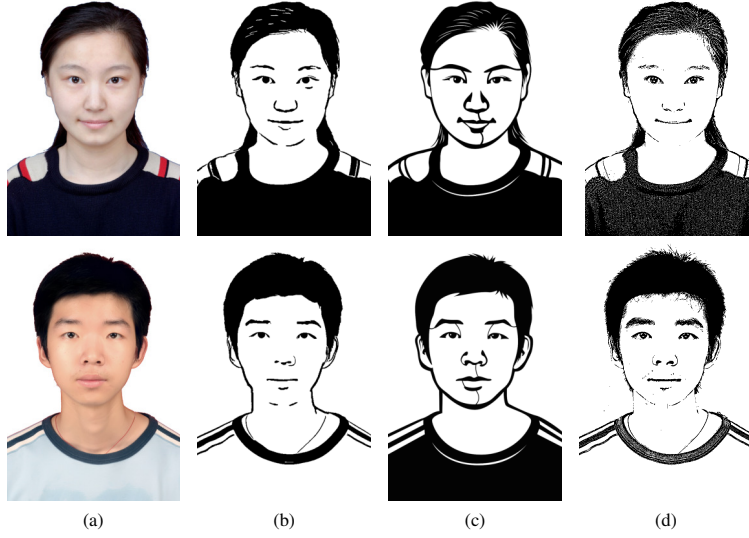
Fig. 11.4 Examples of rendering with a binary palette. (a) source image, (b) Rosin and Lai [25], (c) Meng *et al.* [20], (d) Gooch *et al.* [11].

regions (the "proposal")

$$T' = \{\text{nose}', \text{mouth}', \ldots\}.$$

These features are matched using a greedy algorithm to the AND-OR graph to minimise the following cost.

$$d(T_{i,j,\ldots}, T') + \lambda c(T_{i,j,\ldots})$$

where

$$T_{i,j,\ldots} = \{\text{nose}_i, \text{mouth}_j, \ldots\}$$

denotes a set of facial component templates from the AND-OR graph that make up a complete face, d is the distance between the template and the proposal, and c is the number of different original paper-cut template that the selected components are taken from. Finally, post-processing is applied to extract the hair and clothing using graph cut segmentation, and enforce connectivity by inserting a few curves; see Fig. 11.4(c).

Another approach that was designed and applied to faces was given by Gooch *et al.* [11]. They computed brightness according to a model of human perception involving computation of DoGs over multiple scales. A simple global threshold was then applied, typically close to the average brightness, which produced a line-like result. Finally, dark regions were extracted from the source intensity image by performing thresholding at about 3–5% intensity range. and combined with the lines by a multiplication or logical AND; see Fig. 11.4(d).
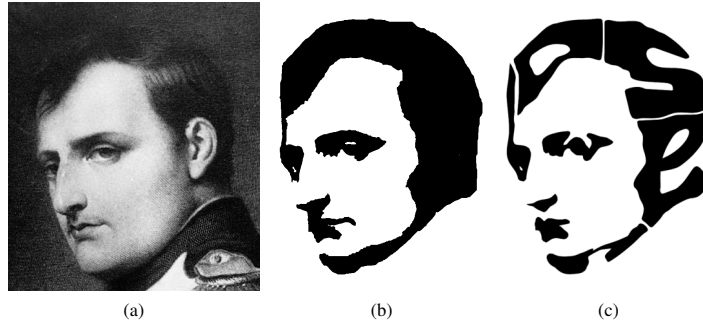
Fig. 11.5 Stages in calligraphic packing: source image, thresholded image, calligraphic packing [34] using text "Lose Win".

Xu and Kaplan [34] describe a specialised stylistic method for rendering images as calligraphic packings in which the black regions are approximated by letters which form one or several words. The method is limited to relatively simple images from which a simple foreground object can be extracted. The foreground is simplified by blurring, and thresholded, to produce the "container region" in which the letters will placed. The positions of the letters are refined using a version of Lloyd's algorithm: at each iteration the container is partitioned into subregions (one for each letter), and each letter is moved to its subregion's centroid. The letters are also scaled to fit their subregion. The subregion boundaries are simplified using mathematical morphology, and then the source letter templates are warped to fit their enclosing subregions by warping individual convex subpolygons in such a manner as to maximise the similarity between the subregion and letter shapes; see Fig. 11.5.

Comparing the methods described above for rendering using a binary palette, certain similarities and differences become apparent. The two basic approaches use either filtering or perform segmentation and labelling as part of an optimisation framework. While the former has the advantage of efficiency, the latter is more flexible regarding the terms that can be incorporated into the cost function, and is able to consider more global aspects of the rendering. As outlined in the introduction, a combination of abstraction and detail preservation is necessary to achieve good stylisation, and this is carried out by all the methods. For instance, lines can be extracted from the image or from the region adjacency graph, and inserted into the rendering. In addition to rendering with black regions and/or lines, several approaches take advantage of negative (white) lines to include extra information without increasing the palette. In a similar manner, region tones can also be inverted. Most of the techniques are general purpose, but a model based approach can be advantageous. For instance, in the second row in Fig. 11.4(a) there is low contrast around the woman's chin. This causes the contour to be missed by the general methods, but it can be captured with the facial model.

For binary rendering, one of the issues for most of the methods is to set an appropriate level for some thresholding or segmentation step in their processing pipeline, and this is done either using preset values, dynamically, or manually. This setting is

most critical for a binary palette since the limited intensities mean that an error in the setting will tend to have more drastic consequences on the results than for larger palettes. Unfortunately, unless high level semantics are available (e.g. in the form of a facial model) it is not possible to expect any automatic method to work 100% reliably, since low level image cues such as gradient, colour, shape etc may at times be misleading.

11.3 Palettes with Fixed Number of Tones

While two tone rendering is probably most interesting as it reduces the palette to an extreme and mimics some artistic forms such as pen-and-ink drawing, introducing a small number of extra tones may be beneficial, e.g. to improve the expressibility. Similar to binary palette rendering, to achieve artistic effect, various abstraction strategies have been proposed to remove extraneous detail, reduce the number of tones and the complexity of shapes. However, the availability of additional tones gives opportunities to preserve more subtle detail and gives more flexibility in determining the choice of tones, shapes of regions etc. without sacrificing recognisability. Different algorithms are needed to handle extra tones properly to achieve a balance of abstraction and recognisability using a combination of techniques. We focus on techniques that produce rendering with a fixed or perceptually fixed number of tones and cover more general rendering with variable number of tones in the next section.

Rosin and Lai [25] proposed an approach that produces a three-tone rendering where the gray background is overlaid with a combination of dark (black) and light (white) tonal blocks and lines. As the mid-tone is gray, both light and dark regions can be well delineated. A multi-level adaptive thresholding [23] is used to extract the dark and light tonal blocks which are then refined by using GrabCut [26] (an iterative graph cut based algorithm). Lines are further extracted to make the image more recognisable. a modified coherent line drawing algorithm by Kang *et al.* [13] is used. Kang's approach first finds a smooth edge tangent flow following the salient image edges which is then used to produce a local edge-aware kernel, combined with difference-of-Gaussian (DoG) and thresholding to obtain binary edge pixels that are more coherent with dominant edges. To apply this for three-tone rendering, both dark and light lines are extracted by using [13] both on the input image I and its inverse $\bar{I}$. Some further improvements involve using hysteresis thresholding [4] where edgels with intermediate strength are preserved only if there is some path connecting them to strong edgels, and deleting small connected components. These improvements lead to a cleaner edge map with less fragments and noise. Consider four set of pixels: dark lines, light lines, dark blocks and light blocks. Each pixel can belong to none or multiple sets. A quite sophisticated set of rules with all the 2^4 combinations of set membership is introduced. The rationale is to make lines more visible when combined with tonal blocks. So a black line over a white tonal block can be rendered black while a black line over a black tonal block should be rendered differently (e.g. gray) to be distinctive (see Fig. 11.6(d)). They also extended their method to ap-

(a)

(b)   (c)   (d)

(e)   (f)   (g)

Fig. 11.6 Examples of rendering using a palette with a fixed number of tones. (a) source image, (b) simple thresholding (3 tones), (c) simple thresholding (10 tones), (d) Rosin and Lai (3 tones) [25], (e) Rosin and Lai (pyramid: 10 tones) [25], (f) Kyprianidis and Döllner [16] (g) Olsen and Gooch [22]. Part g: ©2011 ACM, Inc. Included here by permission.

ply the three-tone rendering to an image pyramid with three levels and average the obtained image, leading to a 10-tone rendering (see Fig. 11.6(e)). Compared with binary rendering, the additional tones allow more detail such as highlights and shadows to be preserved. The results preserve more salient details and generally contain less unimportant details than simple thresholding (Fig. 11.6(b),(c)).

Another approach to achieve a fixed number of tones is quantisation and in practice soft quantisation is often used to avoid artefacts which produces perceptually fixed number of tones. Winnemöller *et al.* [33] proposed an approach for real-time image and video abstraction. They first produced a more abstracted image by simplifying the low-saliency regions while enhancing high-saliency features. They enhanced the image contrast by using an extended nonlinear filter [2] which in the

automatic real-time scenario reduces to the bilateral filter [28] but also allows extra saliency measures as input. This step was followed by an edge enhancement step where a difference-of-Gaussian (DoG) filter is used for edge detection. To produce more stylised images, such as cartoon or paint-like effects, they used an optional colour pseudo-quantisation step to reduce the number of tones. For each channel, q (typically $8-10$) equal-sized bins are chosen. If hard quantisation were used, a maximum of q^3 tones would be produced. To improve the time coherence for videos, their approach was instead a soft quantisation, so the exact number of tones may be more but remains perceptually similar. For each pixel location $\hat{x}$, assume the channel value is $f(\hat{x})$, the nearest bin boundary is $q_{nearest}$, and Δq is the bin size, the quantised value is calculated as

$$Q(\hat{x}, q, \varphi q) = q_{nearest} + \frac{\Delta q}{2} \tanh(\varphi q \cdot (f(\hat{x}) - q_{nearest})),$$

where $\tanh(\cdot)$ is a sigmoid (smooth step) function, φq controls how sharp the step function is. A fixed φq would create transition boundaries in regions with large smooth transitions. An adaptive approach was proposed instead to use larger φq for pixels where the local luminance gradient is large. The soft quantisation improves time coherence and thus reduces flickering in video stylisation.

Kyprianidis and Döllner [16] proposed an approach based on a similar framework, and soft quantisation as in [33] is used to produce a more stylised effect. Before that, the image abstraction is instead obtained by adapting the bilateral filter (for region contrast enhancement) and DoG (for edge enhancement) guided by a smoothed structure tensor. The input image is treated as a mapping $f : (x, y) \in \mathbb{R}^2 \to (R, G, B) \in \mathbb{R}^3$, where (x, y) is the pixel coordinate and (R, G, B) is the vector comprising three colour components. The directional derivatives (Sobel operators) in x and y directions are: $\frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial R}{\partial x} & \frac{\partial G}{\partial x} & \frac{\partial B}{\partial x} \end{pmatrix}^T$, $\frac{\partial f}{\partial y} = \begin{pmatrix} \frac{\partial R}{\partial y} & \frac{\partial G}{\partial y} & \frac{\partial B}{\partial y} \end{pmatrix}^T$. Denote the Jacobian matrix $(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$ as J and the structure tensor is computed as

$$(g_{ij}) = J^T J = \begin{pmatrix} \frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial x} & \frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial y} & \frac{\partial f}{\partial y} \cdot \frac{\partial f}{\partial y} \end{pmatrix}.$$

$J^T J$ is a symmetric positive semi-definite matrix and assume $\lambda_1 \geq \lambda_2 \geq 0$ are two eigenvalues and v_1 and v_2 are corresponding eigenvectors. v_1 shows the direction with the maximum rate of change and v_2 the minimum. So the direction field derived from v_2 follows the local edge direction. To improve smoothness and reduce discontinuities, a Gaussian smoothing operator on the local structure tensor is applied before eigen decomposition. An example is shown in Fig. 11.6(f). Details as well as the original tones are largely preserved. Region boundaries are generally smooth due to the soft quantisation.

In the settings of image simplification and vectorisation, Olsen and Gooch [22] proposed an approach to produce stylised images with three-tone rendering using soft and hard quantisation. The stylisation helps to reduce the complexity of images

and make it effective in vectorisation. Their method worked on grayscale images and started with similar blurring and unsharp masking to suppress low-saliency regions and enhance significant features. The object boundaries were then simplified using line integral convolution guided by a smoothed edge orientation field as in [16]. After that, a piecewise linear intensity remapping is applied to increase the contrast. A soft quantisation approach was then used, similar to [33] but more flexible as the bins do not need to be equal-sized. Assume feature (bin centre) values for all the bins are $\mathbf{b} = (b_1, b_2, \ldots, b_n)$, where n is the number of bins. For an arbitrary intensity value v, it is first clamped to the range of $[b_1, b_n]$. Then assuming for some j, $v \in [b_j, b_{j+1}]$, the width w_j and vertical shift c_j are defined as $w_j := \frac{1}{2}(b_{j+1} - b_j)$ and $c_j := \frac{1}{2}(b_{j+1} - b_j)$ respectively. The soft quantised output p is then defined as

$$p(v, s) := w_j \frac{\text{sig}\left(\frac{s}{w_j}(v - c_j)\right)}{\text{sig}(s)} + c_j,$$

where sig is a sigmoid function. The formula ensures C^0 continuity and since $p'(c_j, s) = \frac{s \times \text{sig}'(0)}{\text{sig}(s)}$, the derivative (sharpness) at the midpoint between two regions is solely determined by the parameter s, regardless of the bin size. As a vectorisation method, the three-tone regions are further traced and boundaries smoothed to obtain a compact representation. An example is shown in Fig. 11.6(g). The result looks more stylised as the region boundaries are smoothed.

From this section we see that to obtain renderings with a fixed number of tones, two general approaches are considered. The fixed number of tones can be a natural result of some generating model, such as combining multiple layers of rendering, mimicking the art forms of e.g. charcoal and chalk. Another general approach is to use (soft) quantisation. Using quantisation alone is usually not sufficient as it can easily mix the salient features with extraneous detail, but it can be an effective way of making the intermediate results more artistically appealing. This technique is typically combined with algorithms such as image filtering that produce smooth but abstracted rendering to produce a cartoon like flat shading.

11.4 Palettes with Variable Number of Tones

Image rendering with reduced but a variable number of tones is useful to achieve paint-like flat rendering, thus removing the non-essential detail. Segmentation is often used to find regions with homogeneous colours. A simple but widely used strategy is to use a flat colour for each region. A further improvement involves manipulating the tones and regions to enhance the stylisation. For segmentation based stylisation of videos, special attention needs to be paid to ensure temporal coherence. Alternative approaches to achieve fewer tones include image filtering or can

Fig. 11.7 Examples of rendering using a palette with a variable number of tones. (a) source image, (b) result using GIMP cartoon filter, (c) DeCarlo *et al.* [8], (d) Song *et al.* [27], (e) Zhang *et al.* [39], (f) Weickert *et al.* [30], (g) Kyprianidis *et al.* [17], (h) Xu *et al.* [36]. Part c: ©2002 ACM, Inc. Included here by permission.

be formulated as an energy optimisation problem. An example of a variable number of fixed tones produced by a GIMP cartoon filter plugin[1] is given in Fig. 11.7(b).

11.4.1 Stylisation using segmentation

Segmentation is often used to produce stylised images with flat colours. Examples shown in Fig. 11.2(d),(e) demonstrate that traditional image segmentation techniques may not be ideal for stylisation.

DeCarlo and Santella [8] proposed an approach that renders abstracted and stylised images based on eye tracker data such that regions receiving more attention are better preserved. The method starts with a hierarchical image segmentation to build a region hierarchy. The input image is repeatedly downsampled to form an image pyramid. The image at each resolution is segmented individually using an existing method. Following scale-space theory, the regions at finer scales tend to be included within regions at the coarser scales. A region hierarchy is formed using a bottom-up merging process. Starting from the regions at the finest scale, treat each as a leaf node, proceed up the pyramid until reaching the coarsest level. At each level, a finer region is generally assigned to the parent region with the most significant overlap as long as this does not violate the tree (hierarchical) structure. To render the image in a more abstracted and stylised manner, for areas receiving more attention, more detail needs to be preserved, thus finer regions in the hierarchy should be used. Similarly, coarser regions should be used for areas with less attention. This effectively defines a frontier, a separation of the tree structure determining whether a branch or its children nodes should be used for rendering. Since region boundaries are determined by the finest level segmentation, region boundaries are smoothed by treating them as a curve network and the interior nodes are smoothed using low-pass filtering while keeping the branching nodes fixed. The final rendering is obtained by showing the average colour for each region. Selected smoothed lines are overlaid if the boundary lines belong to the frontier and they are sufficiently long with respect to the fixation. The line thickness is also determined by the length, with longer lines being thicker. An example is shown in Fig. 11.7(c); effective abstraction is obtained, but this also requires eye tracking data as input.

Although a vast literature exists for image and video segmentation, it remains particularly challenging to obtain semantically meaningful regions. An alternative approach is to use some interactive segmentation instead. The work in [31] for colour sketch generation starts from the mean-shift image segmentation [6]. They then designed a user interface to allow users to merge and split regions as well as editing region boundaries. The tones and shapes of regions are then manipulated to further enhance the stylisation. More detail is given in the next subsection.

---

[1] Many cartoon effect filters are available for GIMP – we have used CarTOONize by Joe1GK.

11.4.2 Tone and region manipulation

Whilst it is natural to take the mean colour of a region to achieve abstraction, the results may not be sufficiently stylised. In an effort to learn the artist's experience of choosing appropriate colours, Wen *et al.* [31] further consider applying learning based colour shift to each region. The training data involves a set of regions from some training images, with their average colour $\mathbf{O}_k$ and artist painting colour $\mathbf{P}_k$, both vectors in a perceptually uniform (hue, chroma, value) colour space. For each region $\mathbf{R}_i$ in the current image with M regions, N candidate colours are obtained from the training set such that the hue is similar (within a difference threshold) and chroma and value are as close as possible. A graph is then built treating each region as a vertex. An edge is created for an adjacent pair of regions or regions with similar colours. Edges between a foreground region and a background region are excluded, making it two separated graphs. Following the artist's experience, different rules are applied for foreground and background regions. The heuristics are then formulated as an energy minimisation problem, considering all the following: For example, background regions tend to have the same colour as the training region with similar colour, area and perimeter. Non-adjacent background regions with similar colour tend to be coloured the same. Foreground regions often have increased chroma and value but also adjacent and homogeneous foreground regions tend to preserve the original chroma and value contrast, etc.

Instead of requiring example pairs of original and artistic colours as training data, Zhang *et al.* [39] take a reference video or reference colour distribution as input and adjust the region colour from a simple average to follow the style of the reference. HSV colour space is used. The hue is adjusted using mean shift [6] and the saturation and value are adjusted using some linear scaling. Ignoring the pixels with low S and/or V, for the reference video, the following are calculated: the colour histogram for the H channel, the mean μ_S and standard deviation σ_S for the S channel, and the mean μ_V for the V channel. For the current input frame, assuming μ_s, σ_s and μ_v are defined similarly and computed based on the whole frame. (h, s, v) is the average colour of a region being considered, and (h', s', v') is the adjusted colour. h' is obtained by updating h using mean shift for a few iterations:

$$h \Leftarrow \frac{\sum_{c \in N(h)} cD(c)}{\sum_{c \in N(h)} D(c)},$$

where $N(h)$ represents $30°$ neighbourhood in the reference histogram, and $D(c)$ is the corresponding histogram entry. s' and v' are adjusted to match the overall saturation and value:

$$s' = \mu_S + (s - \mu_s)\frac{\sigma_S}{\sigma_s}, \quad v' = v\left(\lambda + (1 - \lambda)\frac{\mu_V}{\mu_v}\right),$$

where λ is a parameter balancing the brightness variation between the input and the reference. To improve temporal coherence, the adjusted colour is further blended across time.

The region boundaries may also be manipulated to enhance artistic styles. To simulate artistic sketching, Wen *et al.* [31] observe that regions do not entirely fill the image and more specifically light pixels close to boundaries are shrunk leaving white blanks to emphasise the shading. For each region, the boundary is first smoothed and then down-sampled to obtain a list of control points. Each control point is then adjusted individually. For a control point P_0, the closest opposite boundary point along the normal direction is denoted as P_1. N points are evenly sampled along $\overline{P_0P_1}$. A weight is assigned for each point X_k as $W_k = 1 - I_k$ where I_k is the intensity at pixel X_k. This gives more weight to darker pixels. Two centres are then calculated, a geometric centre P_c as the average of all the coordinates X_k and a weighted centre P_w with W_k as the weight. If more dark pixels exist closer to P_1, i.e. $d = \|\overline{P_0P_w}\| - \|\overline{P_0P_c}\| > 0$, P_0 is moved by d along the normal direction and otherwise no adjustment is needed for P_0. Self-crossing generated in this process is detected and resolved using heuristics.

Instead of using regions directly obtained from segmentation, the work by Song *et al.* [27] considers extreme abstraction by replacing each region with some more abstracted shape (see Fig. 11.7(d) for an example). The algorithm starts by segmenting the image into regions using existing methods. Each region can be fitted with various primitives and the following are considered: circles, rectangles, triangles, superellipses and robust convex hull. Specifying the appropriate type of shape primitive is time-consuming as there may exist a large number of regions. Since aesthetics is a subjective judgement, a supervised learning approach is used. Regions in the training set are obtained by segmenting selected images. Each region used in the training set is manually labelled to indicate the ideal primitive type and a feature vector is extracted consisting of the errors between the region and fitted shape of each type. These are fed into a C4.5 decision tree [24]. For a new input image, they first segment the image into two levels of details (by changing the number of regions). Regions are rendered using the flat mean colour. Coarse-level regions are rendered first. Fine-level regions are rendered on top of that if the difference between the fine-level region and pixels underneath are larger than some threshold (measured in L_2 difference in the CIELab colour space). This provides a balance between abstraction (using large regions) and recognisability (using small regions). For regions with the same level of detail, shapes are rendered in descending order of approximation error compared with the regions they represent.

These tone and region manipulation techniques can in principle be combined with other colour palette reduction approaches to produce more stylised rendering.

11.4.3 Temporally coherent stylisation using segmentation

Segmentation based stylisation can also be used for video rendering. In this case, temporal coherence is essential.

Zhang *et al.* [39] propose a system for automatic online video stylisation. Their approach also uses segmentation and flat colour shading to produce artistic effect. In this case temporal coherence is essential to avoid flickering. To ensure real-time performance on live video streams, the algorithm works on individual frames. The first frame is segmented using some image segmentation algorithm. For each subsequent frame, as edges give important information for regions, the Canny edge detector [4] is used to extract the edge map. To balance abstraction with recognisability, important areas such as human faces need more regions to be well represented. This is accomplished by using a face detector and applying a smaller threshold for edge detection, leading to more detailed edges in these areas. To preserve coherence between frames, segmentation labels are propagated from the previous frame using optical flow. This gives for each source pixel in the previous frame a corresponding target pixel. Not all the pixels in the current frame receive consistent labels from propagation. The target pixel will remain unlabelled if any of the following conditions is true: either the colour difference between the target and source pixels is above some threshold, or the target pixel has none or multiple source pixels associated. The propagated labels are often noisy and a morphological filter is applied to improve the regions by updating the label of a pixel based on its 8-connected neighbours: if the current pixel is either unlabelled or consistent with the labels of more than half of its neighbours, it obtains/keeps this label and otherwise it is marked unlabelled. Unlabelled pixels are assigned to adjacent regions if they have consistent colours. The remaining pixels are treated as belonging to new regions and a trapped-ball segmentation is applied. This strategy deals with video shot change as a large number of pixels will be unlabelled and thus a new segmentation applied. An example where this technique is applied to an image is shown in Fig. 11.7(e).

Another technique related to segmentation based stylisation is rotoscoping, used for generating cartoon-like videos. To obtain high-quality temporally coherent regions, user assistance is often needed. Agarwala *et al.* [1] proposed a system for rotoscoping based on interactive tracking. Users are asked to draw a set of curves for a pair of keyframes. The rotoscoping curves are then interpolated between these frames. The interpolation needs to take into account smoothness of the curve, smooth transition over time, and consistency with underlying image transitions. This is formulated as an energy minimisation problem with five terms (three shape terms and two image terms) and the obtained nonlinear least squares problem is solved using the Levenberg-Marquardt (LM) algorithm. Interpolated rotoscoping curves may also be edited if users prefer. Smoothly changing rotoscoping curves allow various rendering styles, including cartoon-like flat shading or brush strokes following the curve evolution.

Wang *et al.* [29] also propose an interactive system for rotoscoping based video tooning. The user input is similar: the user needs to draw a few curves in keyframes to roughly indicate the semantic regions. Keypoints on these curves are also speci-

fied, indicating the correspondences. For time coherence, they instead treat the video as a 3D volume of pixels and a 3D generalisation of the mean-shift algorithm is first applied. The curves drawn by the user are then used to guide the merging of 3D segments. Assume a pair of loops $L(k_1)$ and $L(k_2)$ are specified in frames k_1 and k_2 respectively. $S(k_1)$ and $S(k_2)$ indicate segments that each have the majority of their pixels located within loops $L(k_1)$ and $L(k_2)$ in the corresponding frame respectively. For intermediate frame t, $k_1 < t < k_2$, to produce closed regions, pixels fully surrounded by pixels in $S(k_1) \bigcup S(k_2)$ are also included and after that segments with the majority of their pixels included form a set $S*$. Within each frame t, $S*$ indicates a mean-shift suggested boundary $L_{ms}(t)$. Simple linear interpolation also produces a boundary $L_s(t)$. An optimisation approach is then used, iteratively deforming $L_s(t)$ to be closer to $L_{ms}(t)$ while preserving smoothness. After mapping keypoints from user drawn curves to the deforming curve using local shape descriptors, the deformation is formulated as an energy minimization with two terms. For two adjacent frames, the first term E_{smooth} measures the change in movement of each control point as well as the vector between adjacent keypoints and thus prefers consistent and stable transitions. The second term simply measures the distance between corresponding control points of $L_S(t)$ and $L_{ms}(t)$. This iterates until the local minimum is reached. In addition to 3D regions, boundaries between them also form 3D edge sheets. When the video is rendered, the 3D regions and edge sheets are intersected with the frame to form a 2D image. The region is rendered in a flat colour and the strength of edges is determined by a few factors such as time in the frame sequence, or the location on the intersected curve and motion.

11.4.4 Abstraction using filtering

Filtering, in particular anisotropic filtering, is an effective approach for image abstraction. Although not guaranteed, in practice, certain filters tend to produce stylised rendering with reduced palettes. Detailed discussion about filtering is given in chapter 5, and in this section we only focus on its application for reduced palette rendering.

Anisotropic diffusion has been studied in the image processing community for a long time. Although it was not directly considered for the application of NPR, such techniques can actually produce reasonable stylisations. An example produced by Weickert *et al.* [30] is given in Fig. 11.7(f). Assume $f(x)$ represents an image with x represents the coordinate of a pixel. $u(x,t)$ is a filtered image, with t being a scale parameter, conceptually as time. $u(x,t)$ is the solution of the diffusion equation

$$\partial_t u = div \left(g(|\nabla u_\sigma|^2)\nabla u \right),$$

with the boundary conditions $u(x,0) = f(x), \partial_n u = 0$ for boundary pixels where n is the normal direction to the boundary. ∇u_σ is the gradient of u after smoothing by a Gaussian kernel with σ being the standard deviation. $g(s)$ is a nonlinear function

for anisotropic diffusion. If s is close to zero, $g(s)$ is close to 1, and $g(s)$ drops with increasing s. This ensures that diffusion is less likely to cross high gradient pixels which tend to be region boundaries.

More recently, Kyprianidis *et al.* extend Kuwahaha filter with anisotropic kernels derived from structure tensor, similar to [16] (see Fig. 11.7(g) for an example). Their experiments demonstrate that sufficiently abstracted results are obtained, even without further quantisation. A further improvement [15] is later proposed to strengthen the abstraction while avoiding some artefacts, by using an image pyramid and and propagates from coarse to fine orientation estimates and filtering results.

11.4.5 Reduced tone rendering using optimisation

Unlike most segmentation techniques where regions are obtained first, followed by an estimation of the representative colour for each region, it is possible to formulate both region segmentation and colour estimation in a uniform energy minimisation formulation. Xu *et al.* [36] propose an approach that aims at finding an optimal output image S for an input image I such that S contains as little change as possible while close to I. Assume $\partial_x S_p$ and $\partial_x S_p$ represent the colour difference around pixel p between adjacent pixels in the x and y directions, respectively. The energy to be minimised is represented as

$$E(S) = \sum_p (S_p - I_p)^2 + \lambda C(S),$$

where λ is a weight balancing the two terms. The first term ensures S is as close to I as possible, and $C(S)$ is the count of non-zero elements (a.k.a. L_0 norm) and defined as

$$C(S) = \#\{p \mid |\partial_x S_p| + |\partial_y S_p| \neq 0\}.$$

is the number of elements in the set. This formulation naturally leads to a reduced palette rendering as the number of colour changes is minimised. An example of L_0 energy minimisation is given in Fig. 11.7(h). This approach is nicely formulated, however it is an NP-hard problem to find the global optimum, and therefore in practise only an approximate solution is achieved [36].

While a variety of methods have been proposed for rendering with a variable number of tones, the use of a relatively small number of tones naturally leads to flat-shaded regions. However, these regions may be explicitly calculated or implicitly obtained. Explicit approaches use segmentation to obtain regions and assign an appropriate tone to each region. Although image segmentation has been studied for decades, it is still challenging in practice to obtain aesthetically pleasing rendering for various input images, especially when a smaller number of regions is used. In many cases, additional inputs are resorted to, such as eye tracking data and user interaction. Unlike the filtering based approaches, when video is stylised, special care needs to be taken to ensure temporal coherence. Approaches proposed involve

propagation of segmentation labels along adjacent frames or treating the temporal sequence of 2D frames as a 3D volume. Alternatively, implicit approaches do not calculate regions directly; they are rather by-products of tone reduction process. Image filtering (in particular anisotropic filtering) based approaches cannot guarantee to reduce the number of tones and thus are often combined with (soft) quantisation. However, certain filtering methods can directly obtain sufficient abstraction with perceptually reduced number of tones. Another implicit approach, which is studied more recently, is to formulate the reduced tone rendering as an energy minimisation problem. While obtaining regions explicitly seems to require more effort and is more error prone as inappropriate segmentation may lead to loss of essential detail or distracting region boundaries, these methods naturally have more flexibility in manipulating the regions, by adjusting the tones and/or region boundaries.

11.5 Pre-Processing

As part of the image stylisation pipeline it is often useful to apply various pre-processing steps to improve the suitability of the image for rendering. Previous sections have already mentioned the application of filtering to remove noise and perform abstraction. Likewise, there are other pre-processing algorithms, that while they have not necessarily been applied yet in the context of reduced palette rendering, could be usefully incorporated. Examples include: saliency detection to control levels of abstraction [8]; colorimetric correction to provide color constancy which will ensure more consistent and correct palettes; colour harmonisation to produce more attractive palettes [5]; depth estimation to control relighting and other stylisation effects [18]; intrinsic image decomposition to extract shading and reflectance from an image to enable rendering to better capture the underlying scene characteristics, conversion from a colour image to grayscale whilst attempting to preserve the colour contrast in the grayscale image [10].

We demonstrate the effect of the last example in Fig. 11.8. This is especially critical for rendering with a binary palette since limiting the final colours to just black and white means the standard colour to grayscale conversion method (a simple weighted combination of the three colour channels) is prone to missing detail that is more easily retained when a larger palette is available. The conversion that preserves colour contrast (Fig. 11.8(c)) has retained a much higher contrast between the aeroplane and the background foliage compared to the standard colour to grayscale conversion (Fig. 11.8(b)). This subsequently enables the intensity thresholding step in Rosin and Lai's [25] rendering pipeline to achieve a better foreground/background separation (see Fig. 11.8(d),(e)).
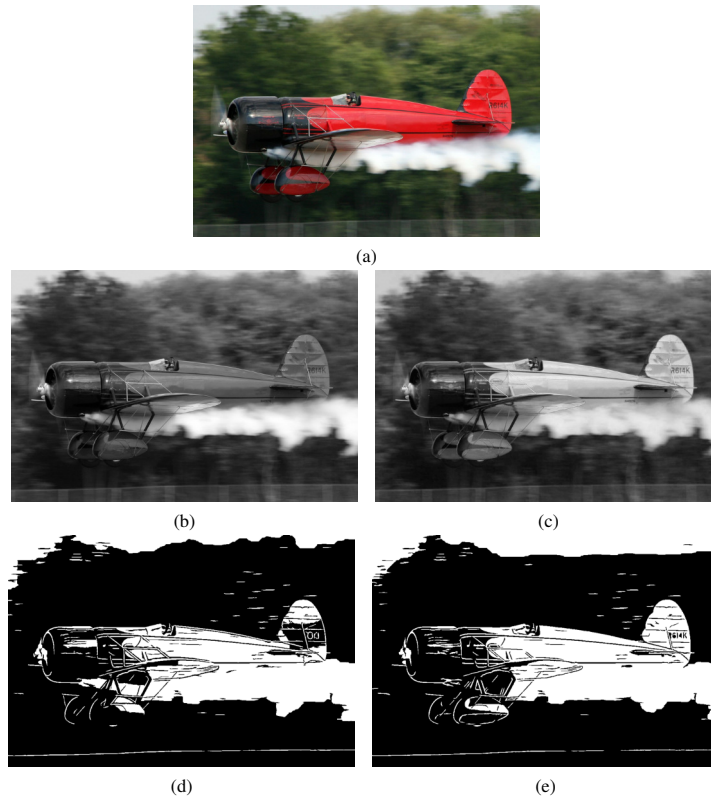
Fig. 11.8 Effects of image pre-processing before stylisation. (a) source image, (b) standard colour to grayscale conversion, (c) colour contrast preserving colour to grayscale conversion [10], (d) Rosin and Lai's [25] two tone rendering of (b), (e) Rosin and Lai's [25] two tone rendering of (c).

11.6 Conclusions

This chapter has described techniques for transforming a colour or gray level image into a reduced palette rendering for artistic effect. Although there is a large variety of techniques and stylisations (see Fig. 11.9–Fig. 11.11 for more rendering examples), it is evident that there are several common elements in the rendering pipelines. First, some degree of abstraction should be performed, otherwise a pixel-wise recolouring will tend to result in noisy, fragmented and unartistic renderings. Second, detail needs to be preserved or re-inserted, since the reduced palette and the abstraction tends to remove salient features that are necessary to fully recognise important aspects of the image/scene. Third, for arbitrary images determining the palette boundaries is non-trivial, and most algorithms are liable to make errors at least occasionally. Therefore some techniques use a soft thresholding approach which reduces the visual impact of such palette boundary errors, and has a stabilising effect which will increase temporal coherence for video stylisation.

We note that of the different classes of reduced palettes that have been discussed in this chapter, that the smaller the palette the greater the variety in the stylisations. This can be explained by the fact that large palettes enable the renderings to re-produce the original image more faithfully (while still incorporating subtle stylistic differences) than smaller palettes such as the two tone rendering. Since the latter is not aiming for fidelity, it uses relatively extreme stylisation techniques such as region and line tone inversions that are not seen in large palette renderings.

The results presented in this chapter show that many reduced palette rendering methods are available and capable of achieving effective stylised effects. How-ever, in the literature they tend to be demonstrated on good quality images and systematic evaluation is limited (see Chapter 15 and Chapter 16). Therefore it is not easy to determine the reliability of the techniques, or the effect of apply-ing them to images with varying resolution, foreground/background characteristics, colour/intensity distributions, etc. Nevertheless, even if it is difficult to develop gen-eral purpose methods that are guaranteed a 100% success rate on arbitrary images, this does not exclude them from practical applications if some manual interaction is allowed. For instance, the complete 2006 film 'A Scanner Darkly' was rendered using a reduced palette: regions were rendered with flat colours, and some linear fea-tures were included. This stylisation was an extremely time consuming task since all the keyframes were rotoscoped (i.e. hand-traced on top of the source film). However, the state of the art is now at a stage where a much less user intensive, semi-automatic solution would be possible. An alternative solution to improving performance is to design methods for specialised domains, which are then able to use domain spe-cific knowledge and constraints to overcome limitations in the data (noise, ambigu-ities, missing data). An example given in this chapter is the paper-cut rendering by Meng *et al.* [20] that used a facial model.

Until recently rendering using a binary palette was an underdeveloped topic, which may explain the recent surge of interest in developing such techniques. As the reduced palette techniques mature it is likely that there will be a trend to video stylisation, especially since related methods for ensuring temporal coherence have been developed for a variety of other stylisation effects (see Chapter 13). Further directions in the future will be to apply reduced palette rendering to a wider range of data types, such as 3D volumetric data, image plus range data (e.g. the Kinect), stereo image/video pairs, low quality images (e.g. mobile phones), etc.

References

1. Agarwala, A., Hertzmann, A., Salesin, D., Seitz, S.M.: Keyframe-based tracking for rotoscop-ing and animation. ACM Trans. Graph. **23**(3), 584–591 (2004)
2. Barash, D., Comaniciu, D.: A common framework for nonlinear diffusion, adaptive smooth-ing, bilateral filtering and mean shift. Image Vision Comput. **22**(1), 73–81 (2004)
3. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimisation in vision. IEEE Trans. Pattern Anal. Mach. Intell. **26**(9), 1124–1137 (2004)
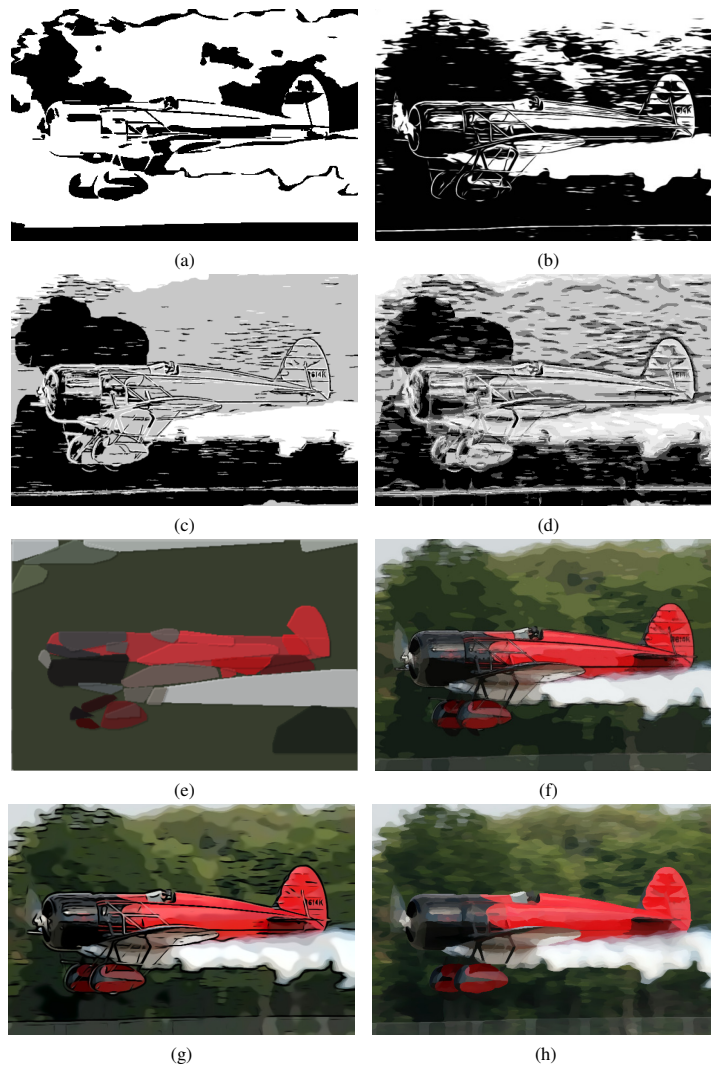
Fig. 11.9 Examples of rendering with various styles. (a) Xu and Kaplan [35], (b) Winnemöller [32], (c) Rosin and Lai [25] (3 tones). (d) Rosin and Lai [25] (pyramid: 10 tones). (e) Song *et al.* [27], (f) Zhang *et al.* [39], (g) Kyprianidis and Döllner [16], (h) Kyprianidis *et al.* [17].

4. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**, 679–698 (1986)
5. Cohen-Or, D., Sorkine, O., Gal, R., Leyvand, T., Xu, Y.Q.: Color harmonization. ACM Transactions on Graphics **25**(3), 624–630 (2006)
6. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
7. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active Appearance Models. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 681–685 (2001)

Fig. 11.10 Examples of rendering with various styles. (a) source image, (b) Xu and Kaplan [35], (c) Winnemöller [32], (d) Rosin and Lai [25], (e) Zhang *et al.* [39], (f) Wen *et al.* [31], (g) Song *et al.* [27], (h) Kyprianidis *et al.* [17].

8. DeCarlo, D., Santella, A.: Stylization and abstraction of photographs. In: Proc. ACM SIG-GRAPH, 3, pp. 769–776 (2002)
9. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Graph-Based Image Segmentation. Int. J. Computer Vision **59**(2), 167–181 (2004)
10. Gooch, A.A., Olsen, S.C., Tumblin, J., Gooch, B.: Color2Gray: salience-preserving color removal. ACM Transactions on Graphics **24**(3), 634–639 (2005)
11. Gooch, B., Reinhard, E., Gooch, A.: Human facial illustrations: Creation and psychophysical evaluation. ACM Transactions on Graphics **23**(1), 27–44 (2004)
12. Heckbert, P.: Color image quantization for frame buffer display. In: Proc. ACM SIGGRAPH, pp. 297–307 (1982)
13. Kang, H., Lee, S., Chui, C.K.: Coherent line drawing. In: ACM Symp. Non-photorealistic Animation and Rendering, pp. 43–50 (2007)
14. Kang, H., Lee, S., Chui, C.K.: Flow-Based Image Abstraction. IEEE Transactions on Visualization and Computer Graphics **15**(1), 62–76 (2009)

Fig. 11.11 Examples of rendering with various styles. (a) source image, (b) Mould and Grant [21], (c) Winnemöller [32], (d) Rosin and Lai [25], (e) Kyprianidis and Döllner [16], (f) Song *et al.* [27],

15. Kyprianidis, J.E.: Image and video abstraction by multi-scale anisotropic Kuwahara filtering. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering, pp. 55–64 (2011)
16. Kyprianidis, J.E., Döllner, J.: Image Abstraction by Structure Adaptive Filtering. In: EG UK Theory and Practice of Computer Graphics, pp. 51–58 (2008)
17. Kyprianidis, J.E., Kang, H., Döllner, J.: Image and Video Abstraction by Anisotropic Kuwahara Filtering. Comput. Graph. Forum **28**(7), 1955–1963 (2009)
18. Lopez-Moreno, J., Jimenez, J., Hadap, S., Reinhard, E., Anjyo, K., Gutierrez, D.: Stylized Depiction of Images Based on Depth Perception. In: ACM Symp. Non-photorealistic Animation and Rendering, pp. 109–118. ACM (2010)
19. Maharik, R., Bessmeltsev, M., Sheffer, A., Shamir, A., Carr, N.: Digital micrography. ACM Trans. Graph. **30**(4), 100 (2011)
20. Meng, M., Zhao, M., Zhu, S.C.: Artistic paper-cut of human portraits. In: 18th Int. Conf. on Multimedia, pp. 931–934 (2010)

21. Mould, D.: A Stained Glass Image Filter. In: Eurographics Workshop on Rendering Techniques, pp. 20–25 (2003)
22. Olsen, S.C., Gooch, B.: Image simplification and vectorization. In: ACM Symp. Non-photorealistic Animation and Rendering, pp. 65–74 (2011)
23. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Systems, Man and Cybern. **9**, 62–66 (1979)
24. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc. (1993)
25. Rosin, P.L., Lai, Y.K.: Towards artistic minimal rendering. In: ACM Symp. Non-photorealistic Animation and Rendering, pp. 119–127 (2010)
26. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23**(3), 309–314 (2004)
27. Song, Y., Hall, P., Rosin, P.L., Collomosse, J.: Arty Shapes. In: Proc. Comp. Aesthetics, pp. 65–73 (2008)
28. Tomasi, C., Manduchi, R.: Bilateral Filtering for Gray and Color Images. In: ICCV, pp. 839–846 (1998)
29. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.F.: Video tooning. ACM Trans. Graph. **23**(3), 574–583 (2004)
30. Weickert, J., ter Haar Romeny, B.M., Viergever, M.A.: Efficient and reliable schemes for non-linear diffusion filtering. IEEE Transactions on Image Processing **7**(3), 398–410 (1998)
31. 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)
32. Winnemöller, H., Kyprianidis, J.E., Olsen, S.C.: XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization. Computers & Graphics **36**(6), 740–753 (2012)
33. Winnemöller, H., Olsen, S., Gooch, B.: Real-time video abstraction. ACM Trans. Graph. **25**(3), 1221–1226 (2006)
34. Xu, J., Kaplan, C.S.: Calligraphic packing. In: Graphics Interface 2007, pp. 43–50 (2007)
35. Xu, J., Kaplan, C.S.: Artistic thresholding. In: ACM Symp. Non-photorealistic Animation and Rendering, pp. 39–47 (2008)
36. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via L_0 gradient minimization. ACM Trans. Graph. **30**(6), 174 (2011)
37. Xu, X., Zhang, L., Wong, T.T.: Structure-based ASCII Art. ACM Transactions on Graphics **29**(4), 52:1–52:9 (2010)
38. Xu, Z., Chen, H., Zhu, S.C., Luo, J.: A Hierarchical Compositional Model for Face Representation and Sketching. IEEE Trans. Pattern Anal. Mach. Intell. **30**(6), 955–969 (2008)
39. Zhang, S.H., Li, X.Y., Hu, S.M., Martin, R.R.: Online Video Stream Abstraction and Stylization. IEEE Transactions on Multimedia **13**(6), 1286–1294 (2011)