HAL
open science

State of the "Art": A Taxonomy of Artistic Stylization Techniques for Images and Video

Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, Tobias Isenberg

To cite this version:

Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, Tobias Isenberg. State of the "Art": A Taxonomy of Artistic Stylization Techniques for Images and Video. IEEE Transactions on Visualization and Computer Graphics, 2013, 19 (5), pp.866-885. 10.1109/TVCG.2012.160 . hal-00781502

HAL Id: hal-00781502

https://inria.hal.science/hal-00781502v1

Submitted on 19 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

State of the 'Art': A Taxonomy of Artistic Stylization Techniques for Images and Video*

Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenberg

Abstract—This paper surveys the field of non-photorealistic rendering (NPR), focusing on techniques for transforming 2D input (images and video) into artistically stylized renderings. We first present a taxonomy of the 2D NPR algorithms developed over the past two decades, structured according to the design characteristics and behavior of each technique. We then describe a chronology of development from the semi-automatic paint systems of the early nineties, through to the automated painterly rendering systems of the late nineties driven by image gradient analysis. Two complementary trends in the NPR literature are then addressed, with reference to our taxonomy. First, the fusion of higher level computer vision and NPR, illustrating the trends toward scene analysis to drive artistic abstraction and diversity of style. Second, the evolution of local processing approaches toward edge-aware filtering for real-time stylization of images and video. The survey then concludes with a discussion of open challenges for 2D NPR identified in recent NPR symposia, including topics such as user and aesthetic evaluation.

Index Terms—Image and Video Stylization, Non-photorealistic Rendering (NPR), Artistic Rendering.

1 Introduction

As the advent of photography stimulated artistic diversity in the late 19th century, so did the successes of photorealistic computer graphics in the early nineties motivate alternative techniques for rendering in non-photorealistic styles. Two decades later, the field of non-photorealistic rendering (NPR) has expanded into a vibrant area of research covering a plethora of expressive rendering styles for the visual communication: exploded diagrams [88], false color [124], [126], and artistic styles such as painterly [10], [168] and constrained palette rendering [106], [167]. It is this latter category of *artistic rendering* (AR) that forms the subject of this survey; specifically, techniques focusing on artistic stylization of two-dimensional content (photographs and video) to which we refer as *image-based artistic rendering* (IB-AR).

IB-AR's origins reach back to seminal works exploring the emulation of traditional artistic media and styles [25], [47], [55], [90], [130]. Today, IB-AR has diversified into a highly cross-disciplinary activity, which builds upon computer vision (CV), perceptual modeling, human computer interaction (HCI), and computer graphics. Many classic IB-AR problems have been found to closely relate to long-standing problems in computer graphics or computer vision; for example, video cartooning [21], [156] and its relationship to video matting and automated rotoscoping [2]. In many cases computer graphics problems have benefited from or motivated entirely new computer vision research. Similarly, the goal of much IB-AR research—that of producing a creative or artistic tool—demands a careful, user-led HCI design process.

Despite several years of discipline convergence and the resulting improvements in aesthetic quality and diversity, there have been few surveys of the IB-AR literature in the past decade. Common references for IB-AR are the texts of Gooch and Gooch [42] and Strothotte and Schlechtweg [148], both of which surveyed pre-2000 techniques (Sec. 3). The majority of other survey material takes the form of conference tutorials; yet these primarily focus upon illustrative visualization [95] or NPR for 3D graphics and games [100]. This survey follows up a recent tutorial [18] by some of the authors at Eurographics 2011, prior to which the most recent major conference tutorials on the topic were by Hertzmann et al. [95] in 2003 and Green et al. [44] in 1999. Also, a number of web-based curated bibliographies are available via Reynolds [127] (to 2004), Schlechtweg [133] (to 2007), and Stavrakis [145].

This article delivers a comprehensive view of the IB-AR landscape, covering classical and contemporary techniques while offering two perspectives. First, we provide an up-to-date taxonomy of IB-AR techniques in which algorithms are grouped according to the family of techniques used (e. g., nonlinear filters, region segmentation) or design characteristics (e. g., local greedy, or global optimization approaches to rendering).

Second, we present IB-AR's development in chronological order, from the early nineties to the modern day (c. 2011), to reflect the contemporaneous development of techniques clustered together in our taxonomy; for example local methods, followed later by global methods. We first document 'classical' (pre-2000) IB-AR and so introduce the key concepts and algorithms that continue to underpin and influence more contemporary methods (Sec. 3). These classical algorithms focused on the *stroke-based rendering* (SBR) paradigm [47], [58] with increasing

- *J. E. Kyprianidis is with the Computer Graphics Systems Group of the Hasso-Plattner-Institut, University of Potsdam, Germany.*
- *T. Wang and J. Collomosse are with the Centre for Vision, Speech and Signal Processing, University of Surrey, UK.*
- *T. Isenberg is with the University of Groningen's Johan Bernoulli Institute, the Netherlands, and with DIGITEO/CNRS/INRIA, France.*

* *This is the authors' version of the work. The definitive version was published in IEEE Transactions on Visualization and Computer Graphics. Vol. 19, No. 5, pp. 866–885, 2013. doi: 10.1109/TVCG.2012.160.*

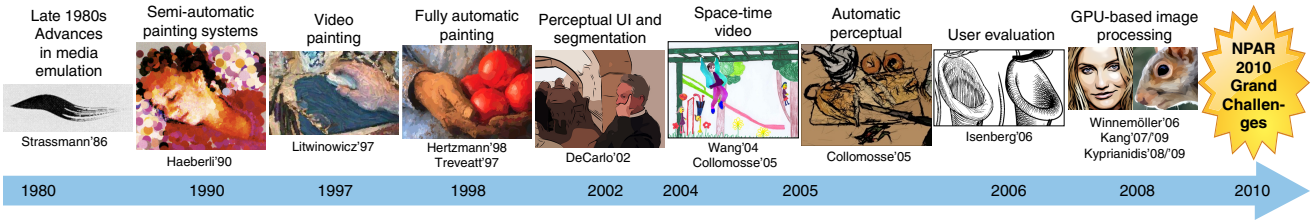

Fig. 1. Chronology of IB-AR development. From the semi-automated SBR systems of the early nineties, to increasingly automated systems drawing upon image processing. Later the aesthetic gamut is enhanced through more sophisticated computer vision and edge-aware filtering. Recently attention returns to user interaction, raising new questions around the evaluation of aesthetics and usability.

levels of automation and sophistication in stroke placement and driven by low-level image processing (typically the Sobel operator).

Next, we describe how the early convergence of computer graphics and image processing developed, enabling IB-AR to draw increasingly upon the more sophisticated image analysis offered by contemporary computer vision algorithms (Sec. 4). One consequence of the increasingly sophisticated interpretation or 'parsing' of the image was a divergence from SBR to alternative forms of rendering primitives: the use of regions and tiles which, in turn, unlocked greater diversity in the gamut of styles available to IB-AR. In line with the trend toward more complex image analysis, we also observe IB-AR to be defined increasingly as a goal-directed task—drawing upon global optimization rather than local approaches. Although these goals were initially defined at the low level of image artifacts (e. g., image gradient), the description of these goals later evolved to include higher level concepts such as perceptual salience measures and even emotional or 'affective' contexts.

In parallel with the trend toward more sophisticated scene analysis, IB-AR benefited from the emerging popularity of anisotropic and edge-preserving forms of filters in computer graphics (Sec. 5). On the one hand, such operations lacked high-level image 'understanding', limiting their artistic gamut to painterly, sketchy, and cartoon styles. On the other hand, their simplicity led to real-time speeds on GPU hardware, making them practical for video processing—and applicable to footage (e. g., water, smoke, fur) that is otherwise challenging to parse using vision methods such as segmentation.

Concluding, we catalog a number of challenges that remain outstanding in IB-AR (Sec. 6).

2 Taxonomy of IB-AR Techniques

Early prototype IB-AR systems followed the SBR paradigm and synthesized artistic renderings by incrementally compositing virtual brush strokes whose color, orientation, scale, and ordering were derived from semi- [47] or fully automated processes [55], [90], [151]. The aesthetics of the output generated by a SBR algorithm is, therefore, a function of both the media simulation applied to render each brush stroke and the process by which strokes are positioned and their attributes are set (referred to hereafter as the *stroke placement algorithm*). Although sometimes described simultaneously in early IB-AR papers, the problems of media emulation and stroke placement may be considered de-coupled. The curved spline strokes placed by Hertzmann's [55] algorithm could be rendered by sweeping various brush models along their trajectories, to emulate thick oil paint, crayon, charcoal, or pastel, to name but a few different media. It is, therefore, not surprising that IB-AR has evolved in parallel with increasingly sophisticated media emulation models; from simple simulations of hairy brushes [146] to full multi-layered models of pigment diffusion and bi-directional transfer between brush and canvas [25].

A detailed exposition of media simulation warrants a survey in its own right, but in this work we focus only on the problem of stroke placement, or more generally, the placement of artistic rendering primitives (regions, strokes, stipples, tiles). We also survey nonlinear filters that introduce an anisotropy that conveys the impression of stroke placement. Accordingly, our taxonomy avoids the categorization of IB-AR purely in terms of media (painterly, sketch, cartoon shading) and instead clusters the space of IB-AR algorithms by the elementary rendering primitive or stylization mechanism employed. We then expand the lower branches of the taxonomy by considering similarities in the nature of the algorithm; local approaches vs. global arrangement strategies, or approaches that address the rendering of outlines vs. the interior of image regions.

2.1 Stroke-based Rendering (SBR)

SBR algorithms cover a 2D canvas with atomic rendering primitives according to some process or desired end goal, designed to simulate a particular style. In many SBR algorithms these primitives are the eponymous virtual brush stroke, but the definition of SBR has diversified to primitives including tiles, stipples and hatch marks [58].

2.1.1 *Brush Stroke Techniques*

The most prevalent form of IB-AR are perhaps SBR algorithms using either short dabs of paint, or long curved brush strokes as rendering primitives. The process of covering the canvas can be categorized broadly as local or global. Local approaches typically drive stroke placement decisions based on the pixels in the spatial neighborhood of the stroke; this can be explicit in the algorithm (e. g., image moments within a window [140], [151]) or implicit due to a prior convolution (e. g., Sobel edges). An alteration to the image would thus affect only strokes in the locality. Global methods optimize

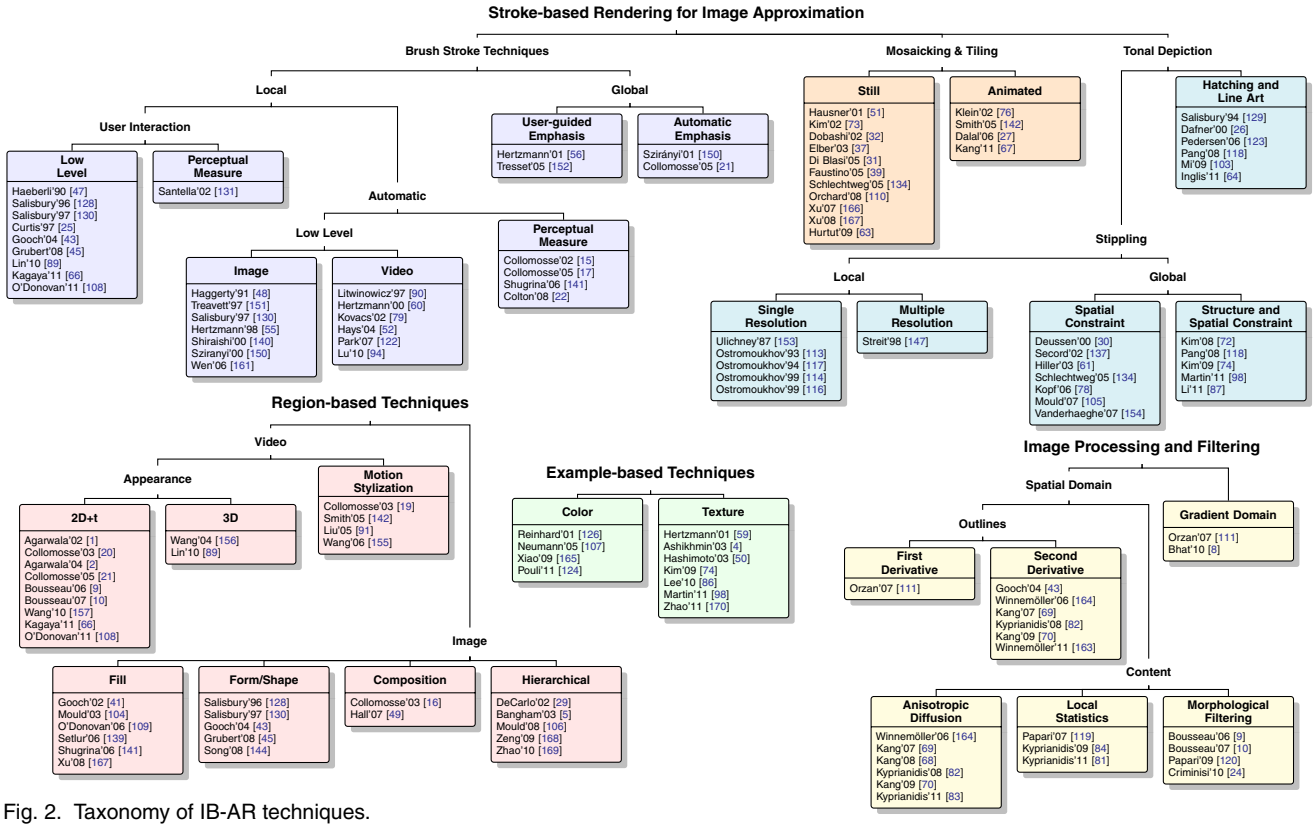

Fig. 2. Taxonomy of IB-AR techniques.

the placement of all strokes to minimize some objective function. Various strategies have been applied from snake relaxation [56], to evolutionary algorithms [17], and Monte-Carlo optimization [150]. In all cases the desired objective relates to retention of detail, for example, encouraging maximal retention of visual detail [56], [150] using low-level operators (e. g., Sobel gradient) or higher-level measures such as image salience to retain only perceptually important detail [17].

On the more heavily populated 'local' branch of the SBR taxonomy, we partition algorithms into user-assisted and automatic processes—the former typically pre-dating the latter, pointing to a trend toward automation post-nineties. The mechanism behind the automation can, as with 'global' SBR, be divided into lower- and higher-level analysis according to the definition of the 'importance' field that guides the emphasis of features in the artwork. In the parallel SBR branch of semi-automated (i. e., user-assisted) algorithms, the low/high-level distinction is again mirrored; with early techniques relying on image filters to orient brush strokes [47] and later work—pre-dating automated measures for emphasis—using gaze trackers to directly harness the perceptual measures inherent in the human visual system [131]. In some recent automated algorithms, stroke placement is influenced by even higher-level contextual parameters such as emotion and mood [22], [141]. Most recently, there has been a trend back toward interaction, producing semi-automated tools for painterly video that enable keyframing of the fields used to arrange strokes [66], [89], [108].

For automatic techniques, a clear distinction can be made between those operating over images versus video content. Video extensions of SBR are non-trivial as strokes must not scintillate (flicker) and their motion must match the underlying video content. In the SBR branch of the taxonomy this problem has largely been addressed—though by no means solved—using optical flow. Elsewhere, nonlinear filters and segmentation have been applied.

2.1.2 *Mosaicking, Tiling and Stippling*

A further sub-category of SBR aims to approximate the image using a medium other than colored pixels or paint, packing image regions with a multitude of atomic rendering primitives. The techniques approximate the image content by either (i) *stippling*, the distribution of small points (stipples) often for the purpose of *tonal depiction*; (ii) *hatching*, the use of line patterns or curves for the same; and (iii) *mosaicking* algorithms that pack small tiles together.

Stippling IB-AR techniques are closely related to digital half-toning and dithering algorithms that locally approximate regions using dot patterns, either with the sole goal of representing a local brightness or with an additional artistic intent [114]. Many early half-toning techniques developed heuristically informed greedy strategies for populating regions with stipples to avoid artifacts due to aliasing. Such techniques operate at either single or multiple scales, placing dots using local decision making. This culminated most recently in techniques designed to emphasize image structure [118], following

the trend toward perceptual analysis in SBR. In contrast to half-toning, stippling does not simply decide whether to use a black or a white pixel on a regular grid but tries to place larger dots, with the shared goal to represent the brightness and to (typically) avoid visible patterns. Early stippling used a number of brush-based techniques [30]. However, much as local SBR painterly approaches evolved into global relaxation approaches, so image stippling began to adopt a more global strategy for stipple placement. Recently, goals in stippling are to capture and replicate aspects of the stippling style of artists [74], [98] or to be able to reproduce non-repetitive patterns [78]. A smaller subset of IB-AR explored the approximation of images using lines and curves. Aside from dedicated image-based hatching approaches [129], some techniques grow labyrinthian patterns using space-filling curves [26] or reaction diffusion processes [123] that adapt to the intensity of the image.

Artistic mosaicking algorithms are closely related to packing problems, and so are approached almost universally as global optimization problems. While packing strategies vary widely, they can be categorized into those obeying purely spatial or spatio-temporal constraints. The latter are especially challenging since a balance must be maintained between a faithful approximation of frame content and the introduction of flicker (temporal incoherence) due to frequent update of the tile or glyph chosen to represent a particular spatial region.

2.2 Region-based Techniques

Much as SBR in the 1990s relied increasingly on low-level image processing (e. g., intensity gradient, moments, optical flow), a trend post-2000 was the emergence of mid-level computer vision in IB-AR. Segmentation is frequently incorporated as step toward parsing image structure, enabling the adaptation of rendering according to the content in regions. In some techniques, SBR algorithms are applied to render the interiors of regions independently [41], [141], [157]. However, the use of regions as rendering primitives in their own right has also given rise to additional styles including cartoon 'flat' shading [21], [156], new materials such as stained glass [104], [139], felt [109], and even emulation of abstract artistic styles [16].

For images, we categorize region-based approaches into those considering the arrangement of rendering primitives (e. g., strokes) within the interiors of regions and those manipulating shape, form, and composition of regions. A further category explores techniques based on image pyramids. Various interactive techniques (human gaze-trackers [29], importance maps [5]) are used to browse a region containment hierarchy constructed by segmenting successively lower resolution versions of the source image. An image can be rendered at a high level of abstraction by drawing only coarse large regions near the top of the hierarchy, or particular regions can be rendered in greater detail at lower levels. This enables local control over the level of detail. Such methods were among the first region-based IB-AR algorithms and are significant by being among the first to consider perceptual importance.

The consideration of regions in IB-AR has also benefited video stylization, offering an alternative to SBR techniques dependent on optical flow. Video segmentation is a well-studied problem in computer vision and is broadly separated into two categories: techniques that segment frames independently and associate regions over time (2D+t) and those segmenting video as a spatio-temporal (x, y, t) volume (3D). Both methodologies have seen applications to IB-AR for the purpose of cartooning or otherwise stylizing the appearance of video. All techniques share the observation that once video has been coherently segmented into regions (a non-trivial problem), the problem of hatching, sketching, or painting with temporal coherence can be solved by attaching strokes to a rigid [21] or deforming [2] region. This frames the problem of IB-AR as one of automated rotoscoping. Finally, when considering regions, it is possible to track and analyze the motion of objects. This gives rise to a complementary form of video stylization—that of artistically manipulating object motion.

2.3 Example-based Rendering

Most IB-AR algorithms encode a set of heuristics, typically emulating artistic practice with the goal of faithfully depicting a prescribed style. A complementary approach to IB-AR—example-based rendering pioneered by Hertzmann et al. [59]—learns the mapping between an exemplar pair: a source image and an artist's rendering of that image. The learned mapping can then be applied to render arbitrary images in the exemplar style.

Example-based rendering (EBR) can be categorized as performing either texture or color transfer. Color EBR typically performs a piecewise mapping between the color histograms of two images to effect a non-photorealistic recoloring. Often there is only weak enforcement of spatial coherence in the color mapping process. By contrast, texture-based EBR shares similarities with patch-based texture in-filling techniques [35], [36], which seek to fill holes in images by searching for visually similar patches elsewhere in the image. However, in the case of EBR the patches are not matched within the source image to be rendered but instead within the exemplar source image. The corresponding patch from the exemplar artistic image is then pasted into place in the output rendering. As with texture in-filling, a careful balance must be maintained between fidelity of the patch matching and the spatial coherence in the rendering.

2.4 Image Processing and Filtering

Many image processing filters have been explored for IB-AR but few have been recognized so far to produce interesting results from an artistic point of view. This is probably because these filters are often concerned with the restoration and recovery of photorealistic imagery. By contrast, IB-AR generally aims for simplification.

Among the filtering approaches to IB-AR, we distinguish two major categories depending on the domain the techniques operate on. The first is the classical spatial domain where the gray or color value of a pixel is replaced based on the values of its neighboring pixels. Most approaches that have been derived from classical image processing techniques fall into this category. In the second category, techniques operate in the gradient domain [3], a relatively sparsely researched area of IB-AR. In the spatial domain, most techniques are automatic processes. Therefore, we subdivide techniques operating in this domain by their type of output; either outlines or stylization of solid areas. Techniques creating outlines can be further classified by the type of edge detector used. We adopt the usual distinction between first and second order derivative methods.

Techniques modifying areas or image regions are further classified into edge-preserving smoothing approaches that employ some form of anisotropic diffusion [159], approaches based on local image statistics, and approaches based on morphological filtering. Techniques based on the bilateral filter fall into the anisotropic diffusion category since the bilateral filter can be interpreted as fast filter-based approximation of anisotropic diffusion. While fast isotropic variants of the bilateral filter have seen application elsewhere in graphics, IB-AR variants of the bilateral filter were developed to focus on increased quality of separable approximations [82] and focused on the enhancement of anisotropic image structures [68].

3 Classical Stylization Algorithms

IB-AR arguably began to gain momentum in the early 1990s with semi-automated paint systems such as Haeberli's [47] that enabled photos to be transformed into impressionist-like paintings with minimal labor. People would click on a photo, each click prompting the generation of a virtual brush stroke of the underlying image color. The introduction of noise prior to this process results in stroke color variation reminiscent of an impressionist painting. Haeberli's system [47] was motivated by a desire to enrich digital painting by automating the color selection process (reducing the 'time to palette'). Yet the longer-term significance of this work has been to introduce the paradigm of an artificial painting as an ordered list of strokes—whose attributes may be derived partially or fully from photorealistic source content. This concept of *stroke-based rendering* (SBR) [58] underpins almost all of the IB-AR work developed in the nineties, which sought increasingly to automate and to enhance the sophistication of stroke placement.

3.1 Local Algorithms for Stroke Placement

Haeberli's framework [47] automates the selection of stroke color, and for non-circular brush strokes can also decide stroke orientation by painting strokes orthogonal to the intensity gradient in the source image. However, the system relies upon the user to determine the order and scale of strokes. The size and sequencing of stroke over-painting is crucial to producing results with an acceptable aesthetic and without any loss of salient detail. An early attempt to automate Haeblerli's pipeline is described by Haggerty [48] using pseudo-random stroke size and painting order. With this solution strokes can be painted at sizes disproportionate to the features they represent. Also, large strokes painted in non-salient regions may over-paint nearby finer-scale strokes. Both result in a loss of detail in the painting (Fig. 3).

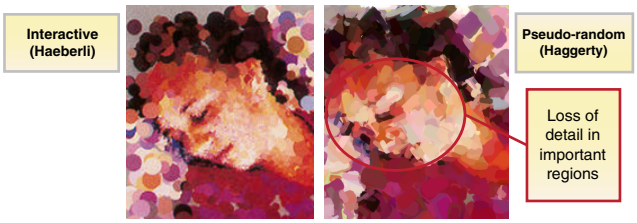

Fig. 3. Adapting Haeberli's framework [47] to randomly assign stroke size and order leads to loss of salient detail [48] and motivated later the use of image processing operators for stroke placement.

3.1.1 *Early Pen-and-Ink Hatching Algorithms*

Early semi-automated systems for rendering in pen-and-ink and cross-hatched styles follow a similar pattern of development. Salisbury et al. [129] developed a semi-automated hatching system that oriented textures according to the underlying image gradients, much as Haeberli's oriented brush strokes [47]. A multi-scale extension of the system [128] offered aesthetic improvements in the viewing of hatching patterns at multiple scales. Region-based editing and manipulation of the underlying image gradient was later introduced by Salisbury et al. [130], enabling discontinuities and swirl effects to be manually introduced, improving tool expressiveness.

3.1.2 *Early Painterly Rendering Algorithms*

The first automatic solution to IB-AR described in full detail within the literature was a painterly rendering tool proposed by Litwinowicz [90]. This system placed rectangular brush strokes at regular intervals on the canvas—retaining a random painting order but introducing the innovation of clipping strokes to thresholded Sobel edges in the source image. Strokes are rectangular, and oriented using Sobel gradients as done previously [47], [48]. The clipping process results in crisp edges that mitigate against strokes from unimportant regions over-painting more important regions. A further contribution was the interpolation of image gradient within flat, near textureless regions. Such regions result in spurious gradient directions, which create a chaotic orientation of strokes. Litwinowicz applied thin-plate splines to interpolate stroke orientation from strokes at strong edges. Later, Hays and Essa [52] adopted a similar technique for interpolation in their video painting algorithm.

Most subsequent SBR algorithms follow Litwinowicz's [90] approach; driving the stroke placement as a function of the Sobel gradient. A notable exception is

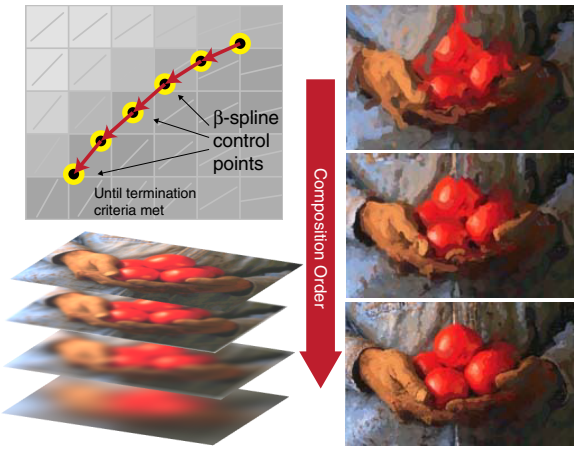

Fig. 4. Illustrating the coarse to fine rendering and curved path tracing components of Hertzmann's painting algorithm [55].

Treavett and Chen's [151] adoption of image moments computed local to each pixel. Analogous to the behavior of gradient-based approaches, their algorithm paints rectangular strokes orthogonal to the direction of maximal variance (i.e., the minor eigenvector).

3.2 Local Coarse-to-fine IB-AR Algorithms

Constant-sized rectangular strokes, even after clipping, generate an artificial regularity that could degrade the resulting aesthetic. Hertzmann [55] was the first to address this issue by proposing curved brush strokes of multiple sizes. The algorithm operates over a low-pass pyramid—a successive Gaussian blurring of the source image at increasing scales. Each scale of the pyramid corresponds to a layer in the painting; the coarsest scale is painted as the first layer with large strokes. Successively finer scales of the pyramid are painted with proportionately smaller strokes. In each iteration of the layer painting process, strokes are placed according to the gradient information at the stroke's position. However, if the image content in a given neighborhood does not differ significantly from the corresponding region in the coarser scale, then no strokes are placed there. Thus large coarse strokes often remain visible in flatter areas, whereas fine strokes appear around edge detail.

The use of curved brush strokes was a further key innovation of this technique. Given a starting or seed pixel, a sequence of spline control points is generated by iteratively hopping between pixels normal to the direction of the image gradient (Fig. 4). The pixel hopping process terminates when a significant change in color occurs or a predefined maximum length is reached. The resulting sequence of control points are rendered using a piecewise cubic β-spline, with color averaged from the visited pixels. A thick brush stroke along the spline can then be produced, either by sweeping a circular brush along its trajectory or by following a triangulation scheme based on normals to the spline trajectory. The latter enables texturing or bump-mapping of the stroke to produce a convincing oil-paint effect [57]. Alternatively, more sophisticated brush-canvas models may be driven by the spline trajectories, e. g., to create watercolor effects [25].

Hertzmann's algorithm [55] has influenced many subsequent IB-AR approaches incorporating coarse-to-fine rendering [52], [56], [89], [141], [168]. Shiraishi and Yamaguchi [140] also extended the image moment-based painterly technique of Treavett and Chen [151] to operate in multiple passes using windows of decreasing size. Multi-resolution rendering image analysis was adopted soon afterwards by half-toning algorithms that similarly distributed rendering primitives (stipples or short lines) across a low-pass pyramid [147]. Until this time, most stippling algorithms had focused on single-scale local decision-making for stroke placement [113], [117] (much as early SBR did [90]). The curve tracing algorithm became a common approach for stroke placement [17], [52], [168]. Moreover, the curve tracing process may be considered an approximation to an Euler integration scheme, common in flow field visualization. This is significant because many of the image filtering approaches to IB-AR developed in more recent years (Sec. 5) adapted a similar approach to perform, for instance, line integral convolution (LIC) [11] guided by a smoothed vector field.

3.3 Video Stylization

The extension of SBR to video stylization is non-trivial, since independent per-frame rendering of the image sequence will result in a distracting flickering or scintillation in the animation. This arises either due to noise in the source imagery, exacerbated by differential-based operators (such as the Sobel filter) that drive the stroke placement, or due to non-determinism introduced by the IB-AR algorithm itself. Holding the stroke placement constant according to the first frame removes this issue, but leaves the viewer with the impression of content moving behind frosted glass (coined by Meier [102] as the *shower door effect*). It is, therefore, desirable that:

1) the motion of brush strokes both matches the motion of the underlying video content, and
2) the animated sequence is flicker free.

If both of these properties are satisfied, then the stylized animation is said to be *temporally coherent*. Flicker is a particularly sensitive issue in painterly animation [60] and in video stylization more broadly where temporal coherence remains an open challenge (Sec. 6).

Litwinowicz's seminal paper [90] also contributed the first extension of an IB-AR painting algorithm to video. The algorithm places constant-sized rectangular strokes upon the first frame (as per Sec. 3.1.2), and translates those strokes to new locations according to an estimate of the optical flow between frames. A secondary process is introduced to govern stroke density because, over time, stroke motion may result in regions becoming too densely or too sparsely populated. The solution is to perform a Delaunay triangulation over stroke centers, and to identify bounded regions that fall below or exceed an area threshold. In the case of overly dense regions, strokes are deleted at random until the density reaches acceptable

levels. In the case of sparsely populated regions, strokes are similarly introduced at random locations within the region. The rendering order of strokes is maintained between frames to reduce flicker and, when strokes are added, they are assigned a random ordering within the list of existing strokes.

When a reliable optical flow estimate is available, the technique performs well. However, forward propagation causes errors in the flow estimate to accumulate. This creates flicker and causes strokes to drift unnaturally across the rendered frames. This is especially noticeable in flat areas where optical flow tends to perform poorly. The approach was applied in the movie *What Dreams May Come*, winning an Academy Award in 1999. Green et al. [44] report that over 1000 man-hours of manual correction to optical flow fields were required to produce the short painterly scenes in the movie. Nevertheless, optical flow features as a key component of many video painting algorithms [10], [52], [94], [122], [157].

In the late nineties real-time optical flow was impractical, leading Hertzmann and Perlin [60] to present an alternative video painting technique amenable to interactive rendering: a "Living Painting." They proposed the use of frame differencing to identify regions exhibiting significant change over time. By subtracting and thresholding consecutive video frames, a map of changed areas requiring repainting is produced. Hertzmann's [55] curved stroke rendering algorithm (Sec. 3.2) is used to render the first frame, and to 'paint over' the regions exhibiting large changes in subsequent frames.

A limitation of most SBR algorithms is that stroke placement decisions have to be performed in sequential order, making parallelization difficult. Notable examples addressing this issue are Lu et al.'s [94] modeling of stroke placement using a stochastic process and interactive SBR techniques like Schlechtweg et al.'s [134] *RenderBots* system and the work by Schwarz et al. [136], which like Renderbots, also relies upon autonomous agents interacting in parallel.

4 Vision for Stylization

An increasing reliance upon local image processing techniques (predominantly the Sobel gradient operator) was instrumental in transforming the interactive IB-AR systems of the early nineties into fully automatic rendering systems. Continuing this trend towards deeper image analysis, a major trend post-nineties was the tendency to rely increasingly upon higher-level computer vision to guide artistic rendering. This trend began with the adoption of mid-level computer vision methods, specifically the use of image segmentation.

4.1 Perceptual Measures for Stylization

DeCarlo and Santella [29] were among the first to apply image segmentation in IB-AR. Images were segmented using a variant of mean-shift [23], [101] at multiple down-sampled resolutions. A scale-space hierarchy was formed by identifying regions at a coarse scale that overlapped

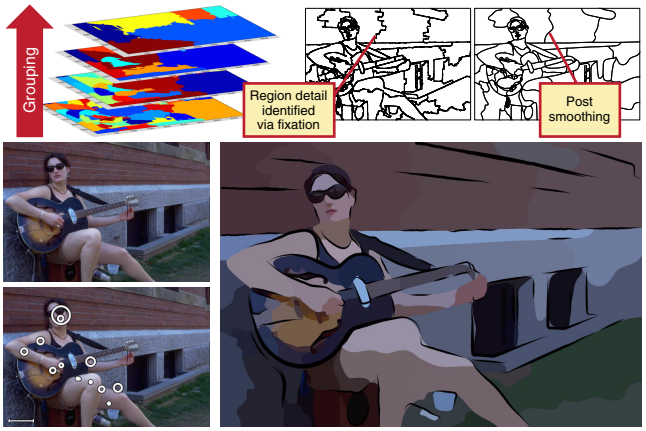

Fig. 5. Interactive abstraction system of Decarlo and Santella [29]. Images are segmented in a scale-space pyramid (top-left). The viewer's gaze is drawn to particular image regions; these regions are locally decomposed into finer segments traversing the pyramid (bottom). Outlines are smoothed and superimposed to delineate region boundaries (top-right).

and shared color similarity with regions at the adjacent finer scale. This hierarchical representation enabled an image to be rendered in a highly abstract form (using coarse regions from the top of the pyramid), or for certain regions to be locally decomposed into finer grain regions by descending the hierarchy. The process was driven by gaze tracking. The viewer was presented with the source image, and the duration and location of their fixations recorded. The level of rendering detail within each region of the image was determined (Fig. 5). The regions were rendered using flat color shading, with a smoothed black outline derived from Canny edges. A similar hierarchical rendering system was developed by Bangham et al. [5] using a 2D extension of sieves—an alternative form of scale space arising from morphological erosion and dilation at varying spatial scales. Sieves are advantageous due to their preservation of sharp corners and discontinuities, which are lost in a Gaussian scale space. Bangham et al. use a binary mask rather than gaze to indicate region importance.

The use of perceptual importance to drive emphasis in IB-AR marked a departure from the image gradient driven algorithms of the late nineties. Such techniques scaled strokes in inverse proportion to edge magnitude and so conserved all fine (i. e., high frequency) detail in the painting unless interactively down weighted, e. g., using manually specified masks [56]. By emphasizing only the *perceptually important* detail, IB-AR began to more closely emulate the practice of artists, who depict only the salient detail of a scene whilst abstracting away extraneous detail.

While Decarlo and Santella harnessed the power of the human visual system to generate their importance maps, a number of IB-AR algorithms were developed using fully automated measures of salience to drive emphasis in renderings. The first example of such an approach [15] used a local algorithm to place strokes, scaled and ordered in inverse proportion to image salience. Salience

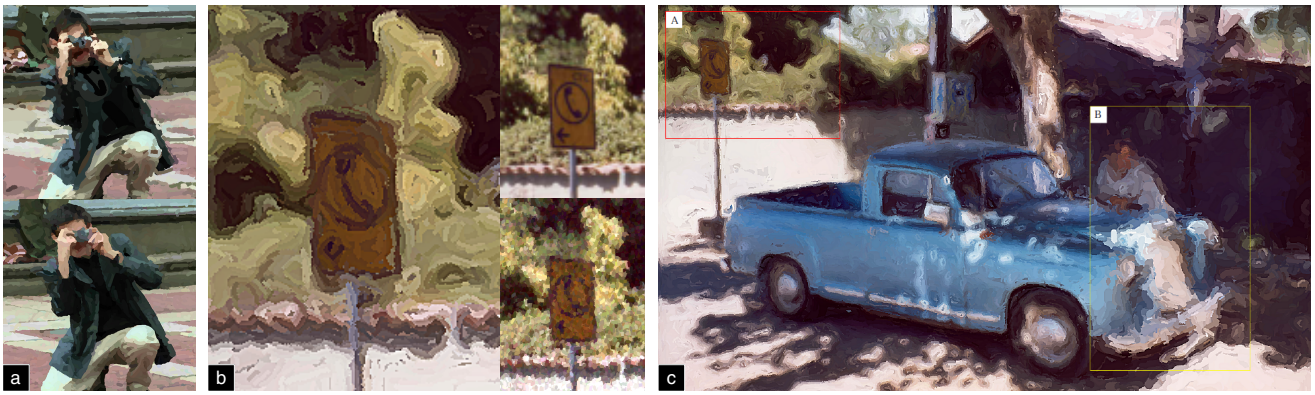

Fig. 6. Global optimization algorithms for painting. (a) Curved $\beta-$spline stroke paintings produced by Hertzmann's greedy algorithm [55] and global optimization [56]. Note the improvements in precision for all edge detail. (b)–(c) Global optimization for painting using GA [17] guided by a salience field (manually damped in region B). Note the difference in emphasis between the non-salient shrubbery and the salient sign detail; enlargement for region A compares against source (upper) and Litwinowicz's [90] method (lower).

was correlated with rarity, by identifying outliers within a distribution of Gaussian derivatives of varying scale and orientation. This definition was later extended to take visibility and local image features into account [17].

4.2 Artistic Rendering as a Global Optimization

Most IB-AR algorithms in the late 1990s treated painting as a *local process*: pixels in the image are examined in turn and strokes placed according to various heuristics. Although the image might be processed repeatedly at different scales (producing successive coarse-to-fine layers of strokes [55], [60], [140]), once a particular stroke is placed there is no subsequent adjustment of its position or attributes to improve the painting in a global sense. Each stroke is placed according to information in its local spatial neighborhood only. By contrast, *global approaches* to IB-AR iteratively optimize the position of rendering elements (e.g. brush strokes, or stipples) to minimize some objective function defined to describe the 'optimality' according to one or more heuristics. Although such an IB-AR approach was first mentioned by Haeberli [47], it was not until a decade later that the first algorithmic solution was described for painterly rendering [56].

4.2.1 Global Approaches to SBR: Brush-based

Hertzmann [56] extended his local curved stroke painterly algorithm [55] by treating each stroke as an active contour or snake. A snake is a piecewise curve, whose control points are iteratively updated to minimize an energy function. The energy function for a snake is a weighted sum of 'internal' energy parameters, guarding against sharp discontinuities or 'kinks' appearing along the curve and 'external' energy parameters. For example, the original snakes algorithm [71] moves the curve incrementally closer to an edge over time by minimizing the distance between the curve and edges in the image.

In Hertzmann's optimization [56], a single painting is created from the source photograph and iteratively updated to converge toward an aesthetic ideal. Strokes are placed in their initial positions on the canvas using the existing curved stroke painting algorithm [55]. An iterative optimization phase then begins in which snake strokes are added, deleted, or moved over the canvas to minimize the objective function evaluating painting quality. A high quality painting is deemed to be one that matches the source image as closely as possible, using a minimal number of strokes but covering the maximum area of canvas in paint. The objective function is a weighted summation assessing the current painting P with respect to these attributes:

$$E_{app}(P) = \omega_1 \sum_{x=1}^{\text{width}} \sum_{y=1}^{\text{height}} |P(x,y) - G(x,y)| \quad (1)$$

$$E_{area}(P) = \omega_2 \sum_{S \in P} \text{Area}(S) \quad (2)$$

$$E_{nstr}(P) = \omega_3 \cdot (\text{strokes in } P) \quad (3)$$

$$E_{cov}(P) = \omega_4 \cdot (\text{unpainted pixels in } P) \quad (4)$$

The weights $\omega_{1..4}$ control the influence of each quality attribute and are determined empirically. The first term E_{app} reflects how closely the painting depicted by the strokes resembles the source image. The other energy terms refer to area of strokes (E_{app}), number of strokes (E_{nstr}), and coverage of the canvas (E_{cov}). Expression $S \in P$ refers to all strokes comprising painting P. Summing the areas of strokes in Eq. (2) yields a value analogous to the quantity of paint used in the painting. A similar model of stroke redundancy was presented in the global approach of Szirányi et al. [150], using a Monte-Carlo Markov Chain (MCMC) optimization.

Fig. 6(a) demonstrates the improvements in stroke accuracy obtained using optimization. However, the high level of detail returned within the painting can cause the rendering to tend back toward photorealism and is inconsistent with the selective process by which artists emphasize salient detail. Collomosse and Hall [17] make use of a *genetic algorithm* (GA) to search the space of possible paintings, locating the 'optimal' painting for a given photograph. The optimality criterion measures the correlation between detail in the painting and the salience map of the source image (Fig. 6(b)).

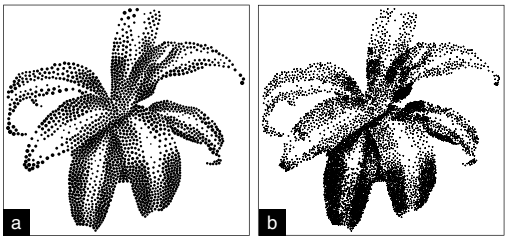

Fig. 7. Stippling examples: (a) Secord's [137] approach (2000 dots); (b) Kopf et al.'s [78] non-repetitive stippling (7700 dots).

Optimization techniques have been also applied to non-painterly styles, for example, Agarwala's 'Snake-Toonz' technique for rendering cartoons semi-automatically from video using snakes [1]. To form a set of closed regions, the initial video frame is interactively annotated by a user with curves that are 'snapped' to adjoin neighboring curves. The regions are tracked over time by iteratively relaxing the snakes on each video frame and flat-shaded to create a cartoon appearance.

Global strategies for rendering became increasingly common in IB-AR as the processing of input footage increased in sophistication; especially tiling and packing IB-AR algorithms that create mosaic art (Sec. 4.4).

4.2.2 Global Approaches to SBR: Tonal Depiction

A purely tonal IB-AR depiction is achieved using stippling. Although placing stipple dots is inherently a spatially local process, a number of algorithms seek to selectively enhance their output according to more global constraints. Such constraints reflect the behavior of stipple artists who strive for a globally even distribution of stipple dots that still reflects the local gray value of the image to be represented. The earliest stipple techniques achieved this goal using *Lloyd's method* [93], [99], which computes the Voronoi diagram of a point distribution and then moves each dot to its respective Voronoi region's center of mass. After iterating this process a number of times, the point distribution becomes even; and thus Lloyd's method yields better results than a straightforward randomized dart throwing. Lloyd's method, however, does not take grey values into account and, thus, was used initially only in a local, brush-based manner [30]. Secord [137] solved this problem by introducing *weighted Voronoi-based stippling* to enable a global and automatic processing (Fig. 7(a)). This method was later extended to shapes beyond simple points [27], [61], [138].

One issue with Lloyd's method [93], [99] is that it introduces a regularity of stipple dot placement, leading to visual artifacts such as lines or circles (Fig. 7(a)) which are typically avoided by artists [62]. These artifacts can be mitigated in IB-AR by adding randomness as done, e. g., by the *RenderBots* algorithm [134]. In practice, however, random noise can only partially remove artifacts. A better way to avoid the patterns is to employ *non-repetitive stippling* (Fig. 7(b)). Kopf et al. [78] employ non-repetitive Wang tiling, compute an even distribution per tile, and describe how to seamlessly match the distributions of neighboring tiles. The result is an artifact-free distribution that also facilitates zooming into the stipple images. When stippling in a more advanced animated fashion (e. g., in videos), however, it is also important to ensure that the stipples follow the depicted objects, which was achieved, for example, by Vanderhaeghe et al. [154].

In addition to globally uniform coverage, researchers have guided stipple distributions according to perceptual salience or image structure. Qu et al. [125], for example, preserve the visual richness of color photographs by applying a range of stippling and related bitonal techniques to different regions in the image. In an even more local approach, Mould [105] proposes a path search within a weighted regular graph that represents the image in order to place stipple dots on chains to represent meaningful edges in the image. In a later approach, Li and Mould [87] use a priority-based error diffusion technique to place those stipples first that represent and preserve the image's structure best. A related sub-domain of stippling deals with the specific subject domain of human faces in which it is particularly important to preserve certain structures. Inspired by hand-made hedcut illustrations, several approaches have addressed this issue [72], [75], [143], aiming to place the stipple dots so as to introduce visible artifacts, in order to emphasize and support the shape of the depicted faces. By contrast, most stippling algorithms seek to minimize such artifacts—and in this sense such techniques are related to image-based hatching approaches [115], [130] that also take structure into account in placing marks.

For many classes of greyscale image content, the above stippling methods generate aesthetically acceptable distributions. However, these may not be representative of those human artists would generate. Therefore, another recent trend in stippling is to use example-based approaches. Kim et al. [74] generate stipple dot distributions with the same statistical properties as those created by artists. While their approach is able to handle sparse point distributions well, it does have issues with point densities in which stipple dots start to merge. To address this issue, Martín et al. [98] treat the stipples not simply as black dots but use greyscale dot scans and aim to reproduce stippling at the correct output resolution. They base the distribution on a globally applied half-toning technique and resort to randomness to avoid the appearance of a grid. The approach achieves satisfactory stipple merging, with distributions matching that of manual stippling [96]. The use of learnable rather than procedurally prescribed artistic models as in these last two techniques has seen increased adoption within IB-AR. We now describe this area of *example based rendering* in greater detail.

4.3 Rendering by Example

Hertzmann et al. [59] coupled machine learning and IB-AR to learn and reproduce artistic styles from visual examples. They introduce the concept of *image analogies*: given a training image pair—a source image A and artistic depiction of that image A'—it should be possible

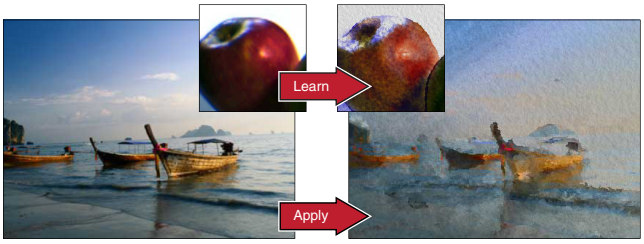

Fig. 8. Image analogies extend patch based texture in-filling techniques to match between a source image and its artistic rendering. Images are rendered in analogous styles using the learned mapping.

to learn the analogous transformation $A \mapsto A'$ such that new source images B might be transformed into artistic renderings B' in the same visual style as A'. There are two main approaches to such *example-based rendering* (EBR): methods seeking to perform texture transfer (typically performed by modulating the luminance channel) and those focusing on color transfer leaving texture constant.

4.3.1 Texture by Analogy

The majority of artistic EBR algorithms focus on the transfer of artistic texture, and borrow from the non-parametric patch-based methods used for texture synthesis and photo in-painting. Such methods (e. g., due to Efros et al. [35], [36]) in-fill from the edges of 'holes' in an image—iteratively copying patches from elsewhere in the image that share similarity with adjacent texture. Typically such methods seek to minimize pixel difference locally at the copied patch and more globally within the image using a smoothness constraint.

In Hertzmann et al.'s image analogy work [59], a mapping is learned between patches in A and corresponding patches in A'. Approximate nearest-neighbor (ANN) is then used to look up each patch in B and replace with the corresponding patch in A'. PCA is used to reduce the dimensionality of the search, which can be time-consuming for ANN over large dimensions (patch sizes). Only the normalized luminance channel is considered. The global smoothness prior enforced in this framework resembles that of Ashikhmin [4]. A recent extension of image analogies incorporates edge orientation to influence patch choice [86], echoing the gradient alignment strategies of SBR algorithms. Stippling patterns have also been learned and applied through example (see Sec. 4.2.2). In recent work, Zhao et al. [170] learn stroke patterns for the domain specific task of portrait painting by example.

Due to their application in texture synthesis, photorealistic EBR algorithms have attracted significant attention over the past decade. A number of extensions to animated sequences have been presented for general video [135], [158]. Video EBR is challenging due to the problem of constraining patch choice to satisfy not only local and global spatial coherence terms but also temporal coherence. Hashimoto et al. [50], for example, adopt optical flow as a guide to propagate choices over time.

4.3.2 Color Transfer

Manipulating color tone can affect the mood of an artistically rendered image, and forms a useful addition to the IB-AR toolbox. The majority of color transfer techniques are based on modifying the color histogram of a target image to resemble that of a source. Typically, this is performed in an (idealized) de-correlated space such as CIELab. Early approaches model the histogram as unimodal, equalizing the mean and variance of the source and target image (either as three 1D per-channel operations [126] or in 3D space [107]). More sophisticated approaches adapt to edges by considering image gradients [165] or perform matching of the histogram at multiple scales [124].

4.4 Region-based IB-AR Algorithms

Initially proposed by DeCarlo and Santella [29] as a mechanism for interactive abstraction of photographs (Sec. 4.1), image segmentation has become a cornerstone of many automatic IB-AR algorithms that make rendering decisions based on mid-level structure parsed from the image. The ability to harness structural representations of image content led to greater diversity of style (unlocking styles such as stained glass rendering or compositional artwork such as pseudo-Cubism). Arguably, aesthetics were also open for improvement as style and emphasis could be controlled at a higher level (e. g., regions) rather than in response to low-level features.

4.4.1 Region Painting and Texturing

The earliest *region-based* IB-AR algorithms focused on painterly rendering and were essentially SBR algorithms that used the shape of the region rather than an image gradient field (as common in pre-2000 SBR) to guide the placement of strokes [41], [77]. This produces visually distinct stroke patterns for each homogeneous region, often corresponding to distinct semantic objects in the image. Gooch and Gooch extracted the medial axis of each region and placed strokes along paths aligned to this axis [41]. Kollioupoulus used PCA to determine the principal axis of each 2D region and painted brush strokes parallel to this axis [77]. Santella and DeCarlo [132] painted regions with Hertzmann's curved brush stroke algorithm [55] using the image gradient but controlled the level of detail in the rendering using an eye tracker (Sec. 4.1). Shugrina et al. [141] filled region interiors with brush strokes aligned with the principal axis but placed brush strokes on the region boundaries for outlines.

The systems described so far only make use of the color and gradient information within regions. They do not leverage the semantic value of the segmentation, i. e., that image regions map approximately to objects. Zeng et al. [168] described a region-based painterly rendering system that classifies texture within regions. The classification drives the type of stroke placed, based on a pre-digitized database of stroke textures from real brushwork mapped to each texture category. In the 'Sisley' system [169], the perceptual relevance of regions was

analyzed to selectively abstract the structure of painterly renderings through geometric warping.

Regions have been rendered with more diverse approaches than SBR alone. Mould [104] and later Setlur and Wilkinson [139] rendered regions with textures to produce a stained glass effect. The latter method sought to classify source image texture within a region, which then drove database selection of a characteristic glass texture used to render that region. In this sense, the technique may be considered a forerunner of the material classification described in Zeng et al.'s [168] painterly system. IB-AR algorithms generating mosaicked effects are frequently based on an initial image segmentation—treating rendering as a packing problem (Sec. 4.5.1).

In contrast to algorithms seeking to fill regions with small primitives such as tiles or strokes, a further class of algorithm applies fill effects within regions. Straightforward cartoon or 'flat-shaded' effects may be produced by uniformly filling regions with the mean color of underlying content [21], [29], [156] and optionally vectorizing and rendering the boundaries. Variants of flat shading using only black and white were presented by Xu and Kaplan [167] and sought to depict the underlying image tone whilst discouraging connected regions of similar tone. A greedy optimization is used to solve a 2-class map coloring problem. Multi-pass (coarse to fine) global optimization approaches to black and white region shading were presented by Mould and Grant [106]. Anisotropic shading effects have also been explored such as felting [109] or color sketching [161].

4.4.2 Deformation and Composition

Although most region-based IB-AR algorithms focus on the shading or texturing of regions, some manipulate geometry for artistic effect. Song et al. [144] classify regions into one of several canonical shapes and replace regions with those shapes to create a simplified shape rendering resembling a paper cut out. Region deformation was also employed to warp regions into superquadric shapes reminiscent of Cubist renderings [16]. This work also re-arranges the position of regions in order to create abstract compositions; arguably styles such as Cubism could not be generated without region-based analysis. Shape simplification was also explored by Mi et al. [103] through decomposition into parts rather than substitution with simpler shapes [144].

4.5 Region Tiling and Packing Algorithms

A considerable volume of IB-AR literature addresses the arrangement of a multitude of small tiles (from regular shapes to irregular pictograms) to form artistic representations. These *mosaicking* algorithms are typically phrased as optimization problems seeking to maximize coverage of a 2D region, whilst minimizing tile overlap. The tile placement is content-aware, penalizing solutions that misalign tiles to cross edges in the image. A spatial coherence term is often introduced to encourage smoothly varying scale and orientation over the tiled region.

4.5.1 Photo and Video Mosaics

The recti-linear tiling of small image thumbnails to approximate a larger image (so called *photomosaics*) were among the earliest form of synthetic mosaic, inspired by early physical macro-artwork such as Dali's *Abraham Lincoln*. Thumbnails are often chosen to have a semantic connection to the larger image being created, as in Dali's work. The IB-AR literature describes optimized search strategies for expedited rendering of photomosaics [6] as well as alternative optimization strategies such as evolutionary search [14]. Klein et al. [76] extended photomosaics to video, updating elements of the mosaic to approximate video content whilst penalizing frequent changes of a given tile to prevent flicker. Work approximating images with irregular tiles (e. g., jigsaw image mosaics [73]) can be considered extensions of photomosaicking.

4.5.2 Voronoi Methods

The earliest mosaic-like renderings relied on Voronoi diagrams constructed from points randomly seeded over the image [47]. Each Voronoi region was shaded with the mean color of the underlying image. Dobashi et al. [32] modified this approach to iteratively relax the position of the Voronoi seeds to better approximate the image using a mean-squared error (MSE) between the source and rendered image. However, the greedy nature of the optimization (iteratively moving seeds within an 8-connected neighborhood) led to loss of detail in image areas exhibiting local minima. Several more sophisticated optimization strategies based on Voronoi diagrams evolved from this early work. Faustino et al. [39] place regular tiles instead of relying on Voronoi segments but guide tile placement using Voronoi regions. Voronoi seeds are scattered more heavily along image edges, leading to increased granularity in areas of high-frequency detail. The tiles are scaled in proportion to image size to preserve detail. Grundland et al. [46] form Voronoi segments according to both edge strength and image intensity.

4.5.3 Packing and Tessellation methods

Hausner et al. [51] were the first to address irregular tile shapes through an energy minimization scheme for shape packing. Kim et al.'s [73] *jigsaw image mosaics* (JIM) extended this approach using an active contour based optimization scheme to minimize the energy function to allow moderate tile deformation. The JIM energy function is a weighted summation of a color term (penalizing deviation of the mosaic from the source image), a gap term (penalizing spaces between tiles), an overlap term (penalizing tile overlap), and a deformation term (penalizing deformation of tiles). The optimization process identifies the contours of tile boundaries initially via a centroidal Voronoi diagram and, treating these contours as a mesh of snakes, relaxes the contours to minimize the JIM energy function. Branch and bound heuristics are used to improve search efficiency (Fig. 9(d)).

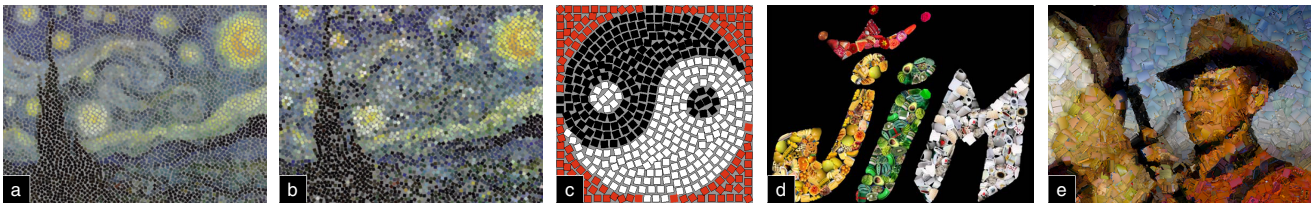

Fig. 9. Mosaicking algorithms aim to tightly pack tiles without overlap. (a) A greedy approach to tiling [6]. (b) Tiling by global graph-cut based optimization [92]. (c) Mosaicking with regular [51] and (d)–(e) irregular overlapping tiles [73], [110].

Orchard and Kaplan [110] describe a fast technique for mosaicking images with irregular tiles, capable also of cropping partial regions from the image database to use as tiles. The work follows up on an earlier specific case of irregular tiling: calligraphic (text) packing [166]. Stylization through text packing was further considered in recent work by [97]. Orchard and Kaplan's [110] speed-up leverages the FFT for cross-correlation between tiles and underlying content. The frequency domain was also leveraged by Dalal et al. [27] to deliver an efficient packing method rather than to match tiles. The technique extends the earlier algorithm of Smith et al. [142]; and whilst slightly slower, allows affine transformation of the tiles during optimization leading to tighter mosaic arrangements. Hurtut et al. [63] combined the principles of texture modeling and mosaicking to learn statistical distributions of tiles. Their mosaics are sparse tile distributions drawn from a set of tile types (e. g., stars, lines, spirals). Given a sample distribution, their system is able to learn the relative densities and spacings of a mosaic and synthesize similar distributions. Further optimization strategies using graph-cut were also explored by Liu et al. [92] (Fig. 9(b)).

4.6 Computer Vision for Video Stylization

A major goal in video stylization is temporal coherence; requiring video to exhibit minimal flicker and the rendering primitives (e. g., strokes) to move with the underlying video content. The issue of temporal coherence is discussed in the context of 3D animation in [7]. Early algorithms for 2D video stylization were based on per-pixel analysis using optical flow and frame differencing (Sec. 3.3). Temporal incoherence is common in such algorithms [60], [90] since stroke placement decisions are being made on a spatially (per-pixel) and temporally (per-frame) local basis. Higher-level analysis of visual structure, e. g., through computer vision can lead to improved coherence. We now survey two categories of post-nineties algorithms: techniques based on optical flow and segmentation-based methods. A third branch of the video stylization literature applying nonlinear filtering is discussed in Sec. 5.

4.6.1 *Visual Stylization through Optical Flow*

Approaches that employ optical flow to stylize video were revisited by Hays and Essa [52]. To mitigate against temporal incoherence arising from flow estimates, strokes were categorized as *weak* or *strong*; the latter in edge areas where gradients are higher. Strong strokes were propagated forward over time as by Litwinowicz [90] and their orientations used as the basis for interpolating a dense gradient field over the image. Thus, so long as the motion of strong strokes varied smoothly, stroke orientation also varied smoothly. Park and Yoon [122] adopted a similar strong-weak categorization. As with all approaches, the regularization of stroke density must be addressed. Hays and Essa's [52] further innovation was to introduce a smooth fade-in/out by varying the opacity of strokes as they were added and deleted, so suppressing flicker.

Bousseau et al. [10] described an algorithm for rendering watercolor video, adapting the texture advection approaches used in scientific visualization to render flow fields. In this system, blended texture patches were moved not only forward but also backward in time using a bi-directional estimate of optical flow. This mitigated against the cumulative errors inherent in the forward propagation strategies of prior approaches.

4.6.2 *Visual Stylization through Segmentation*

Segmentation is now a common component in IB-AR, and by leveraging a similar mid-level representation for video, the consistent motion of strokes within an object can be enforced. This offers improved coherence over estimating the motion of each pixel independently, as optical flow can be unreliable in untextured regions, e. g. object interiors. Region-based representations are also amenable to stylization in a broader gamut of styles, offering the option of stylizing object motion. However, these benefits come at the cost of generality; not all object are amenable to segmentation (e. g., smoke or water).

Video segmentation approaches can be categorized into spatio-temporal (3D) analysis, and frame-to-frame segmentation (2D+t) creating associations between regions over time (Fig. 10). Wang et al. [156] presented *video tooning*, an interactive stylization system following the 3D model. The system is underpinned by an anisotropic extension of mean-shift to video. The system over-segments the video volume into many partial object fragments, grouped by a user. These groups are drawn over and the drawings rotoscoped to deform automatically with the group's exterior boundary. Collomosse et al. [20], [21] adopted the complementary (2D+t) segmentation model in *stroke surfaces*. Regions were associated over time using a space-time region adjacency graph that pruned sporadic association to improve stability. Cross-sections through the resulting space-time surfaces are sampled

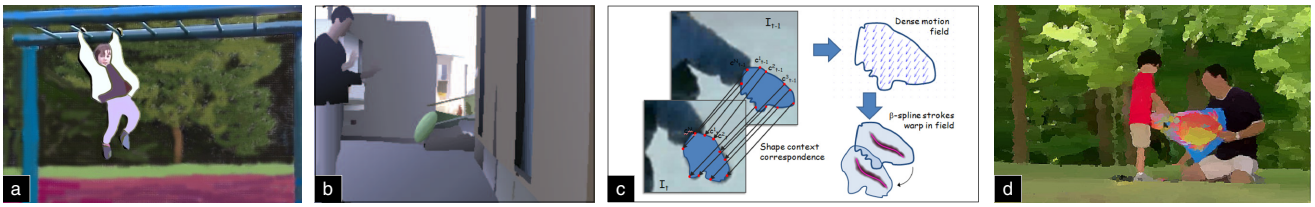

Fig. 10. Video stylization driven by coherent segmentation. (a) 3D approach. (b) 2D+t approach. (c) Cartoon and painterly styles arising from rotoscoping the coherent regions [157]. (d) Example of coherent painterly rendering [157].

at each frame and rendered to form strokes depicting the boundaries of regions. Similar to Wang et al. [156], interaction facilitates a correction of the segmentation. Painterly and cartoon effects were demonstrated by filling regions with strokes and textures that deform coherently with the boundary.

Both Wang et al. [156] and Collomosse et al. [21] echo the automated rotoscoping work of Agarwala et al. [2], which demonstrates similar attachment of rendering primitives to deforming regions. These works indicate the equivalence of automated rotoscoping to the video SBR problem. Recent work by Wang et al. [157] combines both segmentation and optical flow in a 2D+t framework. The system uses a multi-label graph-cut approach to segment individual video frames, propagating labeling priors forward in time with an optical flow variant.

Semi-automated video painting systems have recently been developed and can be considered to be advanced rotoscoping tools, permitting both high-level control over groupings of strokes whilst also allowing fine-grain modification of stroke detail. Lin et al. [89] and, recently, O'Donovan and Hertzmann [108] developed systems that enable video to be interactively segmented into layers, each of which is populated with strokes. As with Agarwala et al. [2], stroke positions deform with the supporting layer and may be dampened to reduce flicker. In the system of Kagaya et al. [66], the video is first segmented into spatial-temporal coherent regions. Users can assign style and orientation parameters as key frames to these regions to be then interpolated over space-time.

4.6.3 *Motion Stylization*

Video analysis at the region level enables not only consistent rendering within objects, but also facilitates the analysis of object motion. This motion may then be stylized using a variety of motion emphasis cues borrowed from classical animation such as mark-making (speed-lines, ghosting, or 'skinning' lines) and deformation [85].

Automated methods to generate speed-lines in video require camera motion compensation, as the camera typically pans to track objects. This can be approximated by estimating inter-frame homographies. Points on trailing object edges may be sampled over time and visualized as speed-lines [19]. Chenney et al. [13] presented early work automatically deforming objects to emphasize motion. This work introduced the 'squash and stretch' effect, scaling 3D objects along their trajectories. A similar effect can be applied to 2D video using non-uniform scaling performed within a curvilinear basis along the object's trajectory [19]. Other distortions warping the object according to velocity or acceleration emphasized drag or inertia. Liu et al. [91] segment video into distinctly moving layers using unsupervised clustering of motion vectors, and perform similar warps. Animators frequently manipulate the timing and trajectory of object motion to introduce 'snap' into an action and this effect has also been incorporated into deformation models [19], [155].

5 Image Processing and Filtering

Many of the techniques described in the previous sections are infeasible for real-time rendering and cannot be trivially adapted for multi-core CPUs or GPUs. Image processing techniques performing local filtering operations provide an interesting alternative since parallelization and GPU implementations are straightforward in most cases. Moreover, a number of filtering techniques have been shown to perform with reasonable temporal coherence when processed frame by frame. These advantages, however, come at the expense of style diversity afforded by higher-level interpretation of content.

5.1 Bilateral Filter and Difference of Gaussians

A fully automatic pipeline for the stylization of cartoon renderings based on images and videos was first proposed in the seminal work by Winnemöller et al. [164]. Their pipeline employs the *bilateral* and *difference of Gaussians* (DoG) filter and contains several influential ideas that other researchers later built upon. Computation of the bilateral filter [121] is comparatively expensive, leading Winnemöller et al. to iteratively apply a separable implementation of the bilateral filter. Although this brute force separation is prone to horizontal and vertical artifacts, it provides a reasonable trade-off between quality and speed, enabling real-time processing on consumer GPUs at the time. In addition, smooth quantization of the luminance channel in CIELab space is performed to achieve a strong cartoon-like effect similar to cel shading. While previous work mainly focused on extracting zero crossings, Winnemöller et al. showed that comparatively simple thresholding of the DoG filter leads to artistically appealing results. Moreover, in a recent survey, Winnemöller [163] showed the versatility of the DoG filter and how a large variety of NPR effects can be created by properly adjusting parameters.

A schematic overview of a generalized form of the Winnemöller et al. [164] pipeline is shown in Fig. 11. In

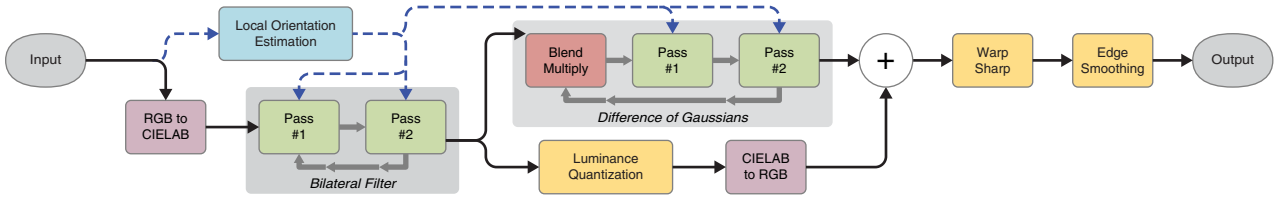

Fig. 11. Generalized pipeline for creating cartoon-like effects by local filtering [70], [82], [164]. After the conversion to CIELab, the input is iteratively abstracted using the bilateral filter. First, 1–2 bilateral filter iterations suppress noise, and outlines are extracted from the intermediate result using a DoG filter. Further iterations of the bilateral filter are performed, with luminance quantization applied afterwards. DoG edges and the output of the luminance quantization are then composited, followed by optional sharpening by warping and smoothing of the edges.

the original pipeline the local orientation estimation step was not present. This step was added later to adapt the bilateral and DoG filter to the local image structure [69], [70], [82]. Also not present were the iterative application of the DoG filter [69] and the final smoothing pass to further reduce aliasing of edges. The introduction of the *flow-based* DoG filter [69], [70], [82] increased the quality of the produced outlines and made the warp-based sharpening step of the original pipeline less important and obsolete in later work (Fig. 12(a)–(c)).

The bilateral filter smoothes low-contrast regions while preserving high-contrast edges, but may fail for high-contrast images where either no abstraction is performed or salient visual features may be removed. The latter effect can also be observed in low-contrast images. Furthermore, iterative filtering may blur edges resulting in a washed-out appearance (Fig. 12(d)). These limitations can be alleviated by overlaying the output of the bilateral filter with outlines of the DoG filter. Accordingly, the bilateral filter is rarely applied independently for IB-AR. Another possibility is to use a more sophisticated edge-preserving filtering approach, such as weighted least squares (WLS) [38]. The next section discusses a further popular approach.

5.2 Kuwahara Filter

An interesting class of edge-preserving filters that perform comparatively well on high-contrast images are variants of the Kuwahara filter [80]. Based on local area flattening, these filters properly remove detail in high-contrast regions and protect shape boundaries in low-contrast regions, resulting in a roughly uniform level of abstraction across the image (Fig. 12(e)). The original Kuwahara filter divides the local filter neighborhood into four rectangular subregions that overlap by one pixel. For all subregions, the variance—the sum of the squared distances to the mean—is computed and the response of the filter is then defined as the mean of a subregion with minimum variance. This *subregion selection* avoids averaging between differently colored regions for corners and edges. For flat or homogeneous regions the variances of different subregions are very similar. Therefore, a subregion with minimum variance is generally not well-defined and highly unstable in the presence of noise. For small filter sizes, the Kuwahara filter produces reasonable results. For IB-AR, however, comparatively large filter sizes are necessary to achieve an abstraction, resulting in noticeable artifacts. These are due to the unstable subregion selection and rectangular subregions.

Several methods address these limitations. The first approach suitable for applications in IB-AR is the *generalized Kuwahara filter* by Papari et al. [119], which introduces two important ideas. First, the rectangular subregions are replaced with smooth weighting functions that are constructed over sectors of a disc such that their sum yields a 2D Gaussian and neighboring weighting functions overlap smoothly. Using these weighting functions, the *weighted mean* and *weighted variance* can be computed for every sector. Second, a the subregion selection is replaced by taking the weighted sum of the weighted means of the sectors.

For highly anisotropic image regions, the flattening effect applied by the generalized Kuwahara filter is typically too aggressive, resulting in blurred anisotropic structures. Moreover, pixels tend to form clusters proportional to the filter size. The *anisotropic Kuwahara filter* by Kyprianidis et al. [81], [84] addresses these issues by replacing the weighting functions defined over sectors of a disc with weighting functions defined over ellipses. By adapting shape, scale, and orientation of these ellipses to the local structure of the input, artifacts are avoided. Directional image features are also better emphasized, resulting in sharper edges and the enhancement of anisotropic image features such as hair or fur.

5.3 Diffusion and Shock Filter

Osher and Rudin [112] as well as Weickert [160] recognized the artistic merit of shock filtered imagery, but the work of Kang and Lee [68] was the first to apply diffusion in combination with shock filtering for IB-AR. *Mean curvature flow* (MCF), which evolves isophote curves under curvature speed in normal direction, is chosen as diffusion method, resulting in simplified isophote curves with regularized geometry. In contrast to other popular edge-preserving smoothing techniques, MCF smooths not only irrelevant color variations while protecting region boundaries but also simplifies the shape of those boundaries. However, it also creates blurred edges, leading Kang and Lee [68] to perform de-blurring with a shock filter after some MCF iterations, which helps to preserve edges. From an aesthetic point of view, shock filtered MCF is typically too aggressive and does not preserve directional image features. Therefore, Kang and Lee [68] constrained the MCF using a vector field to penalize

Fig. 12. *Left:* Different results that were all created with the generalized cartoon pipeline. (a) Thresholded output of the separable implementation of the flow-based DoG [82]. (b) Flow-based DoG with XDoG thresholding [163]. (c) Cartoon-style abstraction generated with bilateral and flow-based DoG filter [70], [82]. *Right:* A selection of popular image abstraction techniques. (d) Bilateral filter (4 iterations) [121]. (e) Anisotropic Kuwahara filter [84]. (f) Shape-simplifying image abstraction [68]. (g) Coherence-enhancing filtering [83].

diffusion that deviates from the local image structure (Fig. 12(f)).

MCF and its constrained variant contract isophote curves to points. For this reason, important image features must be protected by a user-defined mask. A further limitation is that the technique is not stable against small changes in the input and, therefore, not suitable for per-frame video processing. In order to avoid these problems, Kyprianidis and Kang [83] combine curvature-preserving flow-guided smoothing and shock filter-based sharpening orthogonal to the flow, but instead of modeling the process by a PDE, approximations that operate as a local neighborhood filter are used. This improves stability and so suitability for per-frame video processing (Fig. 12(g)).

5.4 Morphological Filtering

Mathematical morphology (MM) provides a set-theoretic approach to image analysis and processing. Fundamental operations in MM are *dilation* and *erosion*. For grayscale images, dilation is equivalent to a maximum filter and erosion corresponds to a minimum filter. From these, other operators can be derived—notably *opening* and *closing*. These are related to order-statistics filters and applying opening and closing in sequence results in a smoothing operation that is often referred to as *morphological smoothing*.

Morphological smoothing is applied in Bousseau et al.'s [9], [10] work on watercolor rendering and in Bangham's et al.'s [5] oil paintings to simplify input images and videos before rendering. In the case of video, Bousseau et al. [10] use a spatio-temporal kernel aligned to the motion trajectory derived from optical flow. Applying opening and then closing generally results in a darkened result. Since watercolor paintings typically have light colors, Bousseau et al. [10] proposed to swap the order of the morphological operators and apply closing followed by opening. Because opening and closing are dual, this is equivalent to inverting the output of morphological smoothing applied to the inverted image.

Papari and Petkov [120] described another technique that applied morphological filtering in the context of IB-AR. Motivated by *glass patterns* and similar to line integral convolution (LIC) [11], they performed a one-dimensional dilation in form of a maximum filter over noise along the integral curves defined by a vector field. In contrast to LIC, this technique is more capable of producing thick piecewise constant coherent lines with sharp edges, resulting in a stronger brush-like effect. Moreover, it can also be applied to color images by using the location of the first maximum noise value along the integral curve as a look-up position.

Some morphological operators can be efficiently implemented by using distance transforms. Criminisi et al. [24] recently demonstrated that edge-sensitive smoothing based on the *generalized geodesic distance transform* (GGDT) can be used for the creation of cartoon-style abstractions. The image is first clustered into a fixed number of colors. Then, for every pixel the probability of the pixel's value belonging to a certain cluster is defined. These probabilities form a soft mask to which the GGDT is applied. The output is then defined as the weighted sum of the cluster's mean values where the weights are defined based on the corresponding distances.

5.5 Gradient Domain Techniques

In recent years, *gradient domain* methods have become very popular in computer vision and computer graphics [3]. The basic idea behind such methods is to construct a gradient field representing the result. Since such constructed fields are rarely conservative the result needs to be found as an approximation by solving an optimization problem (Poisson's equation).

Orzan et al. [111] were the first to apply gradient domain image editing for IB-AR. Using scale-space analysis, they extracted a multi-scale Canny edge representation with lifetime and best scale information, which is used to define the gradient field and allows for image operations such as detail removal and shape abstraction. Moreover, line drawings can be extracted from the multi-scale representation and overlaid with the reconstructed image. A limitation of the technique is that handling contrast is problematic and requires correction. Besides being computationally expensive, this technique is also known not to create temporally coherent output for video.

Bhat et al. [8] have presented a robust optimization framework that allows for the specification of zero-order (pixel value) and first-order (gradient value) constraints over space and time. The resulting optimization problem is solved using a weighted least squares solver. By using temporal constraints, the framework is able to create temporally coherent video output. The framework makes use of several computationally expensive techniques such as steerable filters and optical flow and currently limited to offline processing.

6 Future Challenges

Over the past two decades, IB-AR has delivered many high-quality expressive rendering algorithms and interactive systems. Several high-quality symposia have also emerged, notably *Non-Photorealistic Animation and Rendering* (NPAR, established in 2000) and *Computational Aesthetics* (CAe, established in 2005). As the field gathered momentum, researchers sought to identify the key emerging challenges. David Salesin identified seven Grand Challenges in a keynote delivered at NPAR 2002. We first recap these for the record, as many relate to IB-AR and still remain relevant. The challenges were phrased as seven research questions:

1) **Algorithmic Aesthetics.** How can we quantify success and how can aesthetic 'beauty' be defined?
2) **Abstraction.** How do you capture the 'essence' of an object, image, or painterly style? The reduction of an image to a sketch or structure was discussed.
3) **Visualization.** How can you use NPR to communicate information most effectively?
4) **Interactivity.** How can you develop automated tools that support creativity and the artist?
5) **Artistic Turing Test.** Can you make artistic computer images and animations that are indistinguishable from those created by hand?
6) **New Art Forms.** Can NPR be used to create some entirely new and original forms of art?
7) **Naming the Field.** In a closing aside, Salesin voiced the views of many that NPR is an unfortunately broad term. A decade later, nevertheless, it is still the dominant label for the field [40]. Artistic rendering or Artistic stylization is also in common parlance, whilst *illustrative visualization* is being used for approaches in Salesin's third challenge.

Salesin's challenges were revisited at NPAR 2010 by Gooch et al. [40], demonstrating progress by citing work contributing to each research question. They express the view that (6) remains the most promising direction; that NPR should "not just imitate and emulate styles of the past but create styles for the future." They also observe that Salesin's research questions regarding definitions of aesthetics and the artistic Turing test should be given equal weight in terms of new artistic styles emerging as a consequence of NPR. Examples of such artwork cited include Draves' *Electric Sheep* [33].

Further positions regarding directions for NPR were presented at NPAR 2010 by DeCarlo and Stone [28] and Hertzmann [54]. DeCarlo and Stone's discussion focused on *visual explanations*, that IB-AR can enhance communication by simplification through structural abstraction. This echoes the trend towards perceptually grounded IB-AR [17], [132] and the original motivations for NPR in the early 1990s where NPR and visualization shared common roots, e. g., Strothotte et al.'s [149] transmitted vs. transputed image. Structural abstraction algorithms, for example, are an emerging trend in IB-AR [103], [144].

6.1 Evaluation

DeCarlo and Stone [28], Hertzmann [54], and Gooch et al. [40] all revisit the important issue of aesthetic evaluation in IB-AR. Almost one decade since Salesin's panel discussion of this problem, few papers present structured methodologies for evaluation. This is increasingly important as many traditional styles are now covered in early IB-AR papers and researchers need to either innovate new styles [40] or demonstrate that their work improves over prior art. In some cases (e. g., animation flicker), it is practical to objectify improvement. However, in most papers, evaluation relates to aesthetics only, in the form of a subjective side-by-side visual comparison. While some researchers had previously looked at evaluating single aspects of IB-AR such as aesthetics (e. g., [53]), Isenberg et al. [65] were the first to present a concrete methodology for gathering data on participants' views on artistic renderings, proposing an unconstrained pile sort of renderings based on user preference. In doing so, they also drew conclusions with respect to some of Salesin's challenges such as the artistic Turing test by examining techniques that largely did not pass at the time. Evaluation work more closely aligned with Salesin's visual communication challenge was proposed by Gooch et al. [43] and Winnemöller et al. [164] in their portrait abstractions. It was shown that abstracted faces were more quickly recognized than photorealistic imagery, implying improved comprehension through IB-AR. Methodologies have been developed to evaluate specific aspects of IB-AR such as visual interest [132] and stippling aesthetics [96]. However, no gold standard methodology has emerged for NPR evaluation.

6.2 Interaction

Although a few IB-AR systems of the early nineties cited their motivation as emulating the artist (i. e., passing the artistic Turing test), the frequently stated motivation of contemporary IB-AR work is to retain human creativity and to deliver useful tools and new artistic media. As IB-AR has matured to improve the quality and sophistication of output, the field has begun to positively impact the creative industries. Early IB-AR examples include tools that created painterly effects in the movie *What Dreams May Come* (1998), and Rotoshop, which was used to propagate painted contours with motion in the movies *Waking Life* (2001) and *A Scanner Darkly* (2008). This increased uptake has motivated the development of effective interactive systems for working with IB-AR techniques. Thus, the trend towards automation that started with basic image-processing in the 1990s, through to increasingly sophisticated computer vision in the mid-2000s, is now trending back toward interactive techniques [45], [66], [108]. This trend also reflects the limitations of contemporary computer vision and shows that, by carefully designing minimal but well-placed interaction, a high-quality automated visual effects workflow can result. As such, the need for evaluation in NPR goes

beyond automated assessments of aesthetics to ensure the creation of effective tools for artists.

Hertzmann [54] reflects on both the usability and aesthetic aspects of evaluation in his position paper, and also notes that automated evaluation of aesthetics is useful not only in comparing results but also in global optimization approaches to IB-AR [17], [56], [79] where an objective function capturing some definition of aesthetic 'beauty' must be maximized. However, aesthetics by its nature is challenging to capture through an 'objective' function and such a definition is not certain to appeal to people in the general case.

The drive toward creative tools for IB-AR implies a closer collaboration with artists and end-users. Yet, few demonstrate any collaboration or validation with creative professionals. Addressing this is especially important if, as Gooch et al. [40] suggest, IB-AR's priority is to develop new artistic media and tools. Carroll's *Task-Artifact Cycle* [12] reminds us that users often apply new tools in ways unintended by the designer, generating tasks to be supported in further development. Collaboration with end-users is essential in closing this cycle. More generally, there is potential to draw upon HCI's best practice in structured user-studies for requirement elicitation (e. g., design ethnography) or in co-operative evaluation of the resulting tools. Connections could be forged with research communities studying computational creativity and evolutionary art. Such fields study aesthetics and separately contribute toward Salesin's measure of 'beauty.'

6.3 Technical directions

The technical direction of algorithmic research in IB-AR is challenging to predict for a longer term but may develop in the direction of several established mid-term trends. IB-AR has evolved from image processing toward more sophisticated interpretations of image structure (Sec. 4). However, output remains essentially a stylized version of the source. Analysis of scene structure could enable manipulation of imagery at the composition level. Initial forays into composition indicate the potential of extending the gamut of IB-AR to include abstract styles such as Cubism [16], Futurism [49] and nonlinear projection [34]. The manipulation of global structure for abstraction has also been explored recently [169]. Willats and Durand [162] clearly differentiate between such renderings and current IB-AR when writing about the distinction between *spatial* and *depictive* systems. Abstract visual art is only one potential beneficiary of structural analysis. Portraits are an example of subject matter currently rendered poorly by general purpose stylization algorithms since they are particularly sensitive to distortion or detail loss in facial regions. Imposing high-level models of faces may mitigate this problem and facilitate convincing caricatures and other high-level manipulations [152], [170].

A long-standing challenge in video IB-AR is that of temporal coherence (Sec. 4.6). Nonlinear, edge-aware filtering has been shown to be robust to noise and to generalize over wider classes of footage than computer vision-based methods. However, the gamut of styles achievable by filtering methods is limited to cartoon or painterly styles. By contrast, video stylization approaches based on computer vision can perform more aggressive abstraction through mid-level scene parsing (e. g., segmentation) at the cost of generality. There is a tendency for complex image processing decisions to become less stable in the presence of noise. Ensuring stability of vision-based video stylization remains a largely unsolved problem in automatic IB-AR—in most practical cases relying upon user-assisted markup of footage [89], [108] (which can also be desirable in the context of a creative tool). Nevertheless, maximizing temporal coherence while minimizing user correction and manual labor remains a difficult compromise.

A further trend is toward increased realism in media modeling that one might provocatively call the *photorealism in NPR*. The scope of this survey has necessarily focused on the algorithms for abstraction that drive the placement of rendering primitives. However, overall aesthetics are heavily influenced by media realism, especially in the emulation of traditional artistic styles. Taking painting as an example, strokes have evolved from rectangular stamps [90], to textured curved splines [55], to bump-mapped splines [57], and even to digitized real brushes selected from a database according to image texture [168]. Although this paper has focused on image and video stylization algorithms, complementary research directions investigating media modeling and interactive creative tools remain highly active. Inevitably, as IB-AR and the applications drawing these areas continue to mature, so too will the simulations of artistic media and tool sets that support them.

References

[1] A. Agarwala, "SnakeToonz: A semi-automatic approach to creating cel animation from video," in *Proc. NPAR*, 2002, pp. 139–146.

[2] A. Agarwala, A. Hertzmann, D. Salesin, and S. M. Seitz, "Keyframe-based tracking for rotoscoping and animation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 584–591, 2004.

[3] A. Agrawal and R. Raskar, "Gradient domain manipulation techniques in vision and graphics," in *ICCV Course*, 2007.

[4] M. Ashikhmin, "Fast texture transfer," *IEEE Comput. Graph. Appl.*, vol. 23, no. 4, pp. 38–43, 2003.

[5] J. A. Bangham, S. E. Gibson, and R. Harvey., "The art of scale-space," in *Proc. BMVC*, 2003, pp. 569–578.

[6] S. Battiato, G. Di Blasi, G. Gallo, G. Guarnera, and G. Puglisi, "Artificial Mosaic by Gradient Vector Flow," in *Proc. EG (Short Presentations)*, 2008, pp. 53–56.

[7] P. Bénard, A. Bousseau, and J. Thollot, "State-of-the-Art Report on Temporal Coherence for Stylized Animations," *Comput. Graph. Forum*, vol. 30, no. 8, pp. 2367–2386, 2011.

[8] P. Bhat, C. L. Zitnick, M. F. Cohen, and B. Curless, "GradientShop: A gradient-domain optimization framework for image and video filtering," *ACM Trans. Graph.*, vol. 29, no. 2, pp. 10:1–14, 2010.

[9] A. Bousseau, M. Kaplan, J. Thollot, and F. X. Sillion, "Interactive watercolor rendering with temporal coherence and abstraction," in *Proc. NPAR*, 2006, pp. 141–149.

[10] A. Bousseau, F. Neyret, J. Thollot, and D. Salesin, "Video watercolorization using bidirectional texture advection," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 104:1–7, 2007.

[11] B. Cabral and L. C. Leedom, "Imaging vector fields using line integral convolution," in *Proc. SIGGRAPH*, 1993, pp. 263–270.

[12] J. M. Carroll, W. A. Kellogg, and M. B. Rosson, "The task-artifact cycle," in *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge University Press, 1991, pp. 74–102.

[13] S. Chenney, M. Pingel, R. Iverson, and M. Szymanski, "Simulating cartoon style animation," *Proc. NPAR*, pp. 133–138, 2002.

[14] V. Ciesielski, M. Berry, K. Trist, and D. DeSouza, "Evolution of animated photomosaics," in *Proc. EvoWorkshops*, 2007, pp. 498–507.

[15] J. Collomosse and P. M. Hall, "Painterly rendering using image salience," in *Proc. EGUK*, 2002, pp. 122–128.

[16] ——, "Cubist style rendering from photographs," *IEEE Trans. Vis. Comput. Graphics*, vol. 4, no. 9, pp. 443–453, 2003.

[17] ——, "Genetic paint: A search for salient paintings," in *Proc. EvoMUSART*, 2005, pp. 437–447.

[18] J. Collomosse and J. E. Kyprianidis, "Artistic stylization of images and video," in *Tutorial at Eurographics*, 2011.

[19] J. Collomosse, D. Rowntree, and P. M. Hall, "Video analysis for cartoon-style special effects," in *Proc. BMVC*, 2003, pp. 749–758.

[20] ——, "Stroke surfaces: A spatio-temporal framework for temporally coherent nonphotorealistic animations," University of Bath, UK, Tech. Rep. CSBU-2003-01, 2005.

[21] ——, "Stroke surfaces: Temporally coherent non-photorealistic animations from video," *IEEE Trans. Vis. Comput. Graphics*, vol. 11, no. 5, pp. 540–549, 2005.

[22] S. Colton, M. F. Valstar, and M. Pantic, "Emotionally aware automated portrait painting," in *Proc. DIMEA*, 2008, pp. 304–311.

[23] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.

[24] A. Criminisi, T. Sharp, C. Rother, and P. Pérez, "Geodesic image and video editing," *ACM Trans. Graph.*, vol. 29, no. 5, pp. 134:1–15, 2010.

[25] C. Curtis, S. Anderson, J. Seims, K. Fleischer, and D. Salesin, "Computer-generated watercolor," in *Proc. SIGGRAPH*, 1997, pp. 421–430.

[26] R. Dafner, D. Cohen-Or, and Y. Matias, "Context-based space filling curves," *Comput. Graph. Forum*, vol. 19, no. 3, pp. 209–218, 2000.

[27] K. Dalal, A. Klein, Y. Liu, and K. Smith, "A spectral approach to NPR packing," in *Proc. NPAR*, 2006, pp. 71–78.

[28] D. DeCarlo and M. Stone, "Visual explanations," in *Proc. NPAR*, 2010, pp. 173–178.

[29] D. DeCarlo and A. Santella, "Stylization and abstraction of photographs," in *Proc. SIGGRAPH*, 2002, pp. 769–776.

[30] O. Deussen, S. Hiller, C. Van Overveld, and T. Strothotte, "Floating points: A method for computing stipple drawings," *Comput. Graph. Forum*, vol. 19, no. 3, pp. 41–50, 2000.

[31] G. Di Blasi and G. Gallo, "Artificial mosaics," *The Visual Computer*, vol. 21, no. 6, pp. 373–383, 2005.

[32] Y. Dobashi, T. Haga, H. Johan, and T. Nishita, "A method for creating mosaic images using Voronoi diagrams," in *Proc. EG (Short Presentations)*, 2002, pp. 341–348.

[33] S. Draves, "The Electric Sheep screen-saver: A case study in aesthetic evolution," in *Proc. EvoMUSART*, 2005, pp. 458–467.

[34] F. Durand, "An invitation to discuss computer depiction," in *Proc. NPAR*, 2002, pp. 111–124.

[35] A. Efros and W. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. SIGGRAPH*, 2001, pp. 341–346.

[36] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proc. ICCV*, vol. 2, 1999, pp. 1033–1038.

[37] G. Elber and G. Wolberg, "Rendering traditional mosaics," *The Visual Computer*, vol. 19, no. 1, pp. 67–78, 2003.

[38] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 67:1–10, 2008.

[39] G. M. Faustino and L. H. de Figueiredo, "Simple adaptive mosaic effects," in *Proc. SIBGRAPI*, 2005, pp. 315–322.

[40] A. Gooch, J. Long, L. Ji, A. Estey, and B. Gooch, "Viewing progress in non-photorealistic rendering through Heinlein's lens," in *Proc. NPAR*, 2010, pp. 165–171.

[41] B. Gooch, G. Coombe, and P. Shirley, "Artistic vision: Painterly rendering using computer vision techniques," in *Proc. NPAR*, 2002, pp. 83–90.

[42] B. Gooch and A. Gooch, *Non-photorealistic Rendering*. AK Peters, 2001.

[43] B. Gooch, E. Reinhard, and A. Gooch, "Human facial illustrations: Creation and psychophysical evaluation," *ACM Trans. Graph.*, vol. 23, no. 1, pp. 27–44, 2004.

[44] S. Green, D. Salesin, S. Schofield, A. Hertzmann, P. Litwinowicz, A. Gooch, C. Curtis, and B. Gooch, "Non-photorealistic rendering," in *SIGGRAPH courses*, 1999.

[45] J. Grubert, S. Carpendale, and T. Isenberg, "Interactive stroke-based NPR using hand postures on large displays," in *EG 2008 Short Papers*, 2008, pp. 279–282.

[46] M. Grundland, C. Gibbs, and N. Dodgson, "Stylized multiresolution image representation," *J. Electron. Imaging*, vol. 17, no. 1, pp. 013 009:1–17, 2008.

[47] P. Haeberli, "Paint by numbers: Abstract image representations," in *Proc. SIGGRAPH*, 1990, pp. 207–214.

[48] P. Haggerty, "Almost automatic computer painting," *IEEE Comput. Graph. Appl.*, vol. 11, no. 6, pp. 11–12, 1991.

[49] P. M. Hall, J. Collomosse, Y.-Z. Song, P. Shen, and C. Li, "RTcams: A new perspective on nonphotorealistic rendering from photographs," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 5, pp. 966–979, 2007.

[50] R. Hashimoto, H. Johan, and T. Nishita, "Creating various styles of animations using example based filtering," in *Proc. CGI*, 2003, pp. 312–317.

[51] A. Hausner, "Simulating decorative mosaics," in *Proc. SIGGRAPH*, 2001, pp. 573–580.

[52] J. Hays and I. Essa, "Image and video based painterly animation," in *Proc. NPAR*, 2004, pp. 113–120.

[53] C. Healey and J. Enns, "Perception and painting: A search for effective, engaging visualizations," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 10–15, 2002.

[54] A. Hertzmann, "Non-photorealistic rendering and the science of art," in *Proc. NPAR*, 2010, pp. 147—-157.

[55] ——, "Painterly rendering with curved brush strokes of multiple sizes," in *Proc. SIGGRAPH*, 1998, pp. 453–460.

[56] ——, "Paint by relaxation," in *Proc. CGI*, 2001, pp. 47–54.

[57] ——, "Fast paint texture," in *Proc. NPAR*, 2002, pp. 91–96.

[58] ——, "Tutorial: A survey of stroke-based rendering," *IEEE Comput. Graph. Appl.*, vol. 23, no. 4, pp. 70–81, 2003.

[59] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. Salesin, "Image analogies," in *Proc. SIGGRAPH*, 2001, pp. 327–340.

[60] A. Hertzmann and K. Perlin, "Painterly rendering for video and interaction," in *Proc. NPAR*, 2000, pp. 7–12.

[61] S. Hiller, H. Hellwig, and O. Deussen, "Beyond stippling—Methods for distributing objects on the plane," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 515–522, 2003.

[62] E. R. S. Hodges, Ed., *The Guild Handbook of Scientific Illustration*, 2nd ed. John Wiley & Sons, 2003.

[63] T. Hurtut, P.-E. Landes, J. Thollot, Y. Gousseau, R. Drouillhet, and J.-F. Coeurjolly, "Appearance-guided synthesis of element arrangements by example," in *Proc. NPAR*, 2009, pp. 51–60.

[64] T. C. Inglis and C. S. Kaplan, "Generating op art lines," in *Proc. CAe*, 2011, pp. 25–32.

[65] T. Isenberg, P. Neumann, S. Carpendale, M. C. Sousa, and J. A. Jorge, "Non-photorealistic rendering in context: An observational study," in *Proc. NPAR*, 2006, pp. 115–126.

[66] M. Kagaya, W. Brendel, Q. Deng, T. Kesterson, S. Todorovic, P. J. Neill, and E. Zhang, "Video painting with space-time-varying style parameters." *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 1, pp. 74–87, 2011.

[67] D. Kang, Y.-J. Ohn, M.-H. Han, and K. Yoon, "Animation for ancient tile mosaics," in *Proc. NPAR*, 2011, pp. 157–165.

[68] H. Kang and S. Lee, "Shape-simplifying image abstraction," *Comput. Graph. Forum*, vol. 27, no. 7, pp. 1773–1780, 2008.

[69] H. Kang, S. Lee, and C. K. Chui, "Coherent line drawing," in *Proc. NPAR*, 2007, pp. 43–50.

[70] ——, "Flow-based image abstraction," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 1, pp. 62–76, 2009.

[71] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[72] D. Kim, M. Son, Y. Lee, H. Kang, and S. Lee, "Feature-guided image stippling," *Comput. Graph. Forum*, vol. 27, no. 4, pp. 1209–1216, 2008.

[73] J. Kim and F. Pellacini, "Jigsaw image mosaics," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 657–664, 2002.

[74] S. Kim, R. Maciejewski, T. Isenberg, W. M. Andrews, W. Chen, M. C. Sousa, and D. Ebert, "Stippling by example," in *Proc. NPAR*, 2009, pp. 41–50.

[75] S. Kim, I. Woo, R. Maciejewski, and D. Ebert, "Automated hedcut illustration using isophotes," in *Proc. Smart Graphics*, 2010, pp. 172–183.

[76] A. Klein, T. Grant, A. Finkelstein, and M. F. Cohen, "Video mosaics," in *Proc. NPAR*, 2002, pp. 21–28.

[77] A. Kolliopoulos, "Image Segmentation for Stylized Non-Photorealistic Rendering and Animation," Master's thesis, University of Toronto, 2005.

[78] J. Kopf, D. Cohen-Or, O. Deussen, and D. Lischinski, "Recursive Wang tiles for real-time blue noise," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 509–518, 2006.

[79] L. Kovács and T. Szirányi, "Creating animations combining stochastic paintbrush transformation and motion detection," in *Proc. ICPR*, vol. 2, 2002, pp. 1090–1093.

[80] M. Kuwahara, K. Hachimura, S. Ehiu, and M. Kinoshita, "Processing of ri-angiocardiographic images," in *Digital Processing of Biomedical Images*. Plenum Press, 1976, pp. 187–203.

[81] J. E. Kyprianidis, "Image and video abstraction by multi-scale anisotropic Kuwahara filtering," in *Proc. NPAR*, 2011, pp. 55–64.

[82] J. E. Kyprianidis and J. Döllner, "Image abstraction by structure adaptive filtering," in *Proc. EG UK TPCG*, 2008, pp. 51–58.

[83] J. E. Kyprianidis and H. Kang, "Image and video abstraction by coherence-enhancing filtering," *Comput. Graph. Forum*, vol. 30, no. 2, pp. 593–602, 2011.

[84] J. E. Kyprianidis, H. Kang, and J. Döllner, "Image and video abstraction by anisotropic Kuwahara filtering," *Comput. Graph. Forum*, vol. 28, no. 7, pp. 1955–1963, 2009.

[85] J. Lasseter, "Principles of traditional animation applied to 3D computer animation," in *Proc. SIGGRAPH*, vol. 21, no. 4, 1987, pp. 35–44.

[86] H. Lee, S. Seo, S. Ryoo, and K. Yoon, "Directional texture transfer," in *Proc. NPAR*, 2010, pp. 43–50.

[87] H. Li and D. Mould, "Structure-preserving stippling by priority-based error diffusion," in *Proc. GI*, 2011, pp. 127–134.

[88] W. Li, M. Agrawala, B. Curless, and D. Salesin, "Automated generation of interactive 3D exploded view diagrams," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 101:1–7, 2008.

[89] L. Lin, K. Zeng, H. Lv, Y. Wang, Y. Xu, and S.-C. Zhu, "Painterly animation using video semantics and feature correspondence," in *Proc. NPAR*, 2010, pp. 73–80.

[90] P. Litwinowicz, "Processing images and video for an impressionist effect," in *Proc. SIGGRAPH*, 1997, pp. 407–414.

[91] C. Liu, A. Torralba, W. Freeman, F. Durand, and E. H. Adelson, "Motion magnification," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 519–526, 2005.

[92] Y. Liu, O. Veksler, and O. Juan, "Generating Classic Mosaics with Graph Cuts," *Comput. Graph. Forum*, vol. 29, no. 8, pp. 2387–2399, 2010.

[93] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[94] J. Lu, P. V. Sander, and A. Finkelstein, "Interactive painterly stylization of images, videos and 3D animations," in *Proc. I3D*, 2010, pp. 127–134.

[95] K.-L. Ma, A. Hertzmann, V. Interrante, and E. B. Lum, "Recent advances in non-photorealistic rendering for art and visualization," in *SIGGRAPH courses*, 2002.

[96] R. Maciejewski, T. Isenberg, W. M. Andrews, D. Ebert, M. C. Sousa, and W. Chen, "Measuring stipple aesthetics in hand-drawn and computer-generated images," *IEEE Comput. Graph. Appl.*, vol. 28, no. 2, pp. 62–74, 2008.

[97] R. Maharik, M. Bessmeltsev, A. Sheffer, A. Shamir, and N. Carr, "Digital micrography," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 100:1–11, 2011.

[98] D. Martín, G. Arroyo, M. Victoria Luzón, and T. Isenberg, "Scale-dependent and example-based grayscale stippling," *Computers & Graphics*, vol. 35, no. 1, pp. 160–174, 2011.

[99] M. McCool and E. Fiume, "Hierarchical Poisson disk sampling distributions," in *Proc. GI*, 1992, pp. 94–105.

[100] M. McGuire, H. Halen, J.-F. St-Amour, A. Thibault, and B. Martel, "Stylized rendering in games," in *SIGGRAPH courses*, 2010.

[101] P. Meer and B. Georgescu, "Edge detection with embedded confidence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 12, pp. 1351–1365, 2001.

[102] B. J. Meier, "Painterly rendering for animation," in *Proc. SIGGRAPH*, 1996, pp. 477–484.

[103] X. Mi, D. DeCarlo, and M. Stone, "Abstraction of 2D shapes in terms of parts," in *Proc. NPAR*, 2009, pp. 15–24.

[104] D. Mould, "A stained glass image filter," in *Proc. EGSR*, 2003, pp. 20–25.

[105] ——, "Stipple placement using distance in a weighted graph," in *Proc. CAe*, 2007, pp. 45–52.

[106] D. Mould and K. Grant, "Stylized black and white images from photographs," in *Proc. NPAR*, 2008, pp. 49–58.

[107] L. Neumann and A. Neumann, "Color style transfer techniques using hue, lightness and saturation histogram matching," in *Proc. CAe*, 2005, pp. 111–122.

[108] P. O'Donovan and A. Hertzmann, "AniPaint: Interactive painterly animation from video." *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 3, pp. 475–487, 2012.

[109] P. O'Donovan and D. Mould, "Felt-based rendering," in *Proc. NPAR*, 2006, pp. 55–62.

[110] J. Orchard and C. S. Kaplan, "Cut-out image mosaics," in *Proc. NPAR*, 2008, pp. 79–87.

[111] A. Orzan, A. Bousseau, P. Barla, and J. Thollot, "Structure-preserving manipulation of photographs," in *Proc. NPAR*, 2007, pp. 103–110.

[112] S. Osher and L. I. Rudin, "Feature-oriented image enhancement using shock filters," *SIAM Journal on Numerical Analysis*, vol. 27, no. 4, pp. 919–940, 1990.

[113] V. Ostromoukhov, "Pseudo-random halftone screening for colour and black&white printing," in *Int. Congr. in Non Impact Printing Technologies*, 1993, pp. 579–582.

[114] ——, "Artistic halftoning: Between technology and art," in *Proc. SPIE*, vol. 3963, no. 1, 1999, pp. 489–509.

[115] ——, "Digital facial engraving," in *Proc. SIGGRAPH*, 1999, pp. 417–424.

[116] V. Ostromoukhov and R. D. Hersch, "Multi-color and artistic dithering," in *Proc. SIGGRAPH*, 1999, pp. 425–432.

[117] V. Ostromoukhov, R. D. Hersch, and I. Amidror, "Rotated dispersed dither," in *Proc. SIGGRAPH*, 1994, pp. 123–130.

[118] W.-M. Pang, Y. Qu, T.-T. Wong, D. Cohen-Or, and P.-A. Heng, "Structure-aware halftoning," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 89:1–8, 2008.

[119] G. Papari, N. Petkov, and P. Campisi, "Artistic edge and corner enhancing smoothing," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2449–2462, 2007.

[120] G. Papari and N. Petkov, "Continuous glass patterns for painterly rendering." *IEEE Trans. Image Process.*, vol. 18, no. 3, pp. 652–64, 2009.

[121] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "Bilateral filtering: Theory and applications," *Foundations and Trends in Computer Graphics and Vision*, vol. 4, no. 1, pp. 1–73, 2008.

[122] Y. Park and K. Yoon, "Motion map generation for maintaining the temporal coherence of brush strokes," in *Proc. GRAPP*, vol. 4, 2007, pp. 139–152.

[123] H. Pedersen and K. Singh, "Organic labyrinths and mazes," in *Proc. NPAR*, 2006, pp. 79–86.

[124] T. Pouli and E. Reinhard, "Progressive color transfer for images of arbitrary dynamic range," *Computers & Graphics*, vol. 35, no. 1, pp. 67–80, 2011.

[125] Y. Qu, W.-M. Pang, T.-T. Wong, and P.-A. Heng, "Richness-preserving manga screening," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 155:1–8, 2008.

[126] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Colour transfer between images," *IEEE Comput. Graph. Appl.*, vol. 21, pp. 34–41, 2001.

[127] C. Reynolds, "Stylized depiction in computer graphics." [Online]. Available: http://www.red3d.com/cwr/npr/

[128] M. Salisbury, C. Anderson, D. Lischinski, and D. Salesin, "Scale-dependent reproduction of pen-and-ink illustrations," in *Proc. SIGGRAPH*, 1996, pp. 461–468.
[129] M. Salisbury, S. E. Anderson, R. Barzel, and D. Salesin, "Interactive pen-and-ink illustration," in *Proc. SIGGRAPH*, 1994, pp. 101–108.
[130] M. Salisbury, M. T. Wong, J. F. Hughes, and D. Salesin, "Orientable textures for image-based pen-and-ink illustration," in *Proc. SIGGRAPH*, 1997, pp. 401–406.
[131] A. Santella and D. DeCarlo, "Abstracted painterly renderings using eye-tracking data," in *Proc. NPAR*, 2002, pp. 75–82.
[132] ——, "Visual interest and NPR: An evaluation and manifesto," in *Proc. NPAR*, 2004, pp. 71–150.
[133] S. Schlechtweg, "NPR literature." [Online]. Available: http://isgwww.cs.uni-magdeburg.de/~stefans/npr/
[134] S. Schlechtweg, T. Germer, and T. Strothotte, "RenderBots—Multi-agent systems for direct image generation," *Comput. Graph. Forum*, vol. 24, no. 2, pp. 137–148, 2005.
[135] A. Schödl, R. Szeliski, D. Salesin, and I. Essa, "Video textures," in *Proc. SIGGRAPH*, 2000, pp. 489–498.
[136] M. Schwarz, T. Isenberg, K. Mason, and S. Carpendale, "Modeling with rendering primitives: An interactive non-photorealistic canvas," in *Proc. NPAR*, 2007, pp. 15–22.
[137] A. Secord, "Weighted Voronoi stippling," in *Proc. NPAR*, 2002, pp. 37–43.
[138] A. Secord, W. Heidrich, and L. Streit, "Fast primitive distribution for illustration," in *Proc. EGWR*, 2002, pp. 215–226.
[139] V. Setlur and S. Wilkinson, "Automatic stained glass rendering," in *Proc. CGI*, 2006, pp. 682–691.
[140] M. Shiraishi and Y. Yamaguchi, "An algorithm for automatic painterly rendering based on local source image approximation," in *Proc. NPAR*, 2000, pp. 53–58.
[141] M. Shugrina, M. Betke, and J. Collomosse, "Empathic painting: Interactive stylization using observed emotional state," in *Proc. NPAR*, 2006, pp. 87–96.
[142] K. Smith, Y. Liu, and A. Klein, "Animosaics," in *Proc. SCA*, 2005, pp. 201–208.
[143] M. Son, Y. Lee, H. Kang, and S. Lee, "Structure grid for directional stippling," *Graphical Models*, vol. 73, no. 3, pp. 74–87, 2011.
[144] Y. Song, P. M. Hall, P. Rosin, and J. Collomosse, "Arty shapes," in *Proc. CAe*, 2008, pp. 65–72.
[145] E. Stavrakis, "Non-photorealistic computer graphics library." [Online]. Available: http://www.npcglib.org/
[146] S. Strassmann, "Hairy brushes," in *Proc. SIGGRAPH*, 1986, pp. 225–232.
[147] L. Streit and J. Buchanan, "Importance driven halftoning," *Comput. Graph. Forum*, vol. 17, no. 3, pp. 207–217, 1998.
[148] T. Strothotte and S. Schlechtweg, *Non-photorealistic Computer Graphics*. Morgan Kaufmann, 2002.
[149] T. Strothotte, B. Preim, A. Raab, J. Schumann, and D. R. Forsey, "How to render frames and influence people," *Comput. Graph. Forum*, vol. 13, no. 3, pp. 455–466, 1994.
[150] T. Szirányi, Z. Tóth, M. Figueiredo, J. Zerubia, and A. Jain, "Optimization of paintbrush rendering of images by dynamic MCMC methods," in *Proc. EMMCVPR*, 2001, pp. 201–215.
[151] S. M. F. Treavett and M. Chen, "Statistical techniques for the automated synthesis of non-photorealistic images," in *Proc. EGUK*, 1997, pp. 201–210.
[152] P. Tresset and F. F. Leymarie, "Generative portrait sketching," *VSMM'05*, pp. 739–748, 2005.
[153] R. Ulichney, *Digital Halftoning*. The MIT Press, 1987.
[154] D. Vanderhaeghe, P. Barla, J. Thollot, and F. X. Sillion, "Dynamic Point Distribution for Stroke-based Rendering," in *Proc. EGSR*, 2007, pp. 139–146.
[155] J. Wang, S. M. Drucker, M. Agrawala, and M. F. Cohen, "The cartoon animation filter," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1169–1173, 2006.
[156] J. Wang, Y. Xu, H.-Y. Shum, and M. F. Cohen, "Video tooning," *ACM Trans. Graph.*, vol. 23, no. 3, p. 574, 2004.
[157] T. Wang, J. Collomosse, D. Slatter, P. Cheatle, and D. Greig, "Video stylization for digital ambient displays of home movies," in *Proc. NPAR*, 2010, pp. 137–146.
[158] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proc. SIGGRAPH*, 2000, pp. 479–488.
[159] J. Weickert, *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, Germany, 1998.
[160] ——, "Coherence-enhancing shock filters," in *DAGM-Symposium*. Springer, 2003, pp. 1–8.
[161] F. Wen, Q. Luan, L. Liang, Y.-Q. Xu, and H.-Y. Shum, "Color sketch generation," in *Proc. NPAR*, 2006, pp. 47–54.
[162] J. Willats and F. Durand, "Defining pictorial style: Lessons from linguistics and computer graphics," *Axiomathes*, vol. 15, no. 3, pp. 319–351, 2005.
[163] H. Winnemöller, "XDoG: Advanced image stylization with eXtended difference-of-Gaussians," in *Proc. NPAR*, 2011, pp. 147–155.
[164] H. Winnemöller, S. Olsen, and B. Gooch, "Real-time video abstraction," in *Proc. SIGGRAPH*, 2006, pp. 1221–1226.
[165] X. Xiao and L. Ma, "Gradient-preserving color transfer," *Comput. Graph. Forum*, vol. 28, no. 18, pp. 79–86, 2009.
[166] J. Xu and C. S. Kaplan, "Calligraphic packing," in *Proc. GI*, 2007, pp. 43–50.
[167] ——, "Artistic thresholding," in *Proc. NPAR*, 2008, pp. 39–47.
[168] K. Zeng, M. Zhao, C. Xiong, and S.-C. Zhu, "From image parsing to painterly rendering," *ACM Trans. Graph.*, vol. 29, no. 1, pp. 2:1–11, 2009.
[169] M. Zhao and S.-C. Zhu, "Sisley the abstract painter," in *Proc. NPAR*, 2010, pp. 99–107.
[170] ——, "Portrait painting using active templates," in *Proc. NPAR*, 2011, pp. 117–124.

Jan Eric Kyprianidis is a research scientist with the Computer Graphics Systems Group of the Hasso-Plattner-Institut (HPI) at the University of Potsdam, Germany. He received the diploma degree in Mathematics from the University of Hamburg, Germany in 2005. Until 2007, he was a senior software engineer at Adobe Systems. Jan Eric's research interests include NPR, digital image processing, and GPU computing.

John Collomosse is a lecturer (asst. professor) within the Centre for Vision Speech and Signal Processing (CVSSP) at the U. Surrey. Prior to joining CVSSP he held a lectureship at U. Bath for 4 years, where he also obtained his PhD in 2004 on the topic of NPR and Computer Vision. John's research explores the relationships between artistic depiction and real-world imagery, focusing on NPR and sketch based visual search. John is a member of the IEEE and a Chartered Engineer.

Tinghuai Wang is a PhD candidate based in CVSSP at the University of Surrey and HP Labs. He received a MSc in Electrical Engineering from the Royal Institute of Technology (KTH) and RWTH Aachen in 2007. He has worked for brief periods in Chinese Academy of Science, HP Labs and Sony R&D. His research interests lie in the convergence area between Computer Graphics and Vision.

Tobias Isenberg is an assistant professor for computer graphics and interactive systems at the University of Groningen, the Netherlands. Since September 2010, he also holds a Digiteo Chair of Excellence in collaboration with the French research institutions CNRS and INRIA in Orsay. He works on topics in interactive non-photorealistic and illustrative rendering as well as computational aesthetics and explores applications in scientific visualization.

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive feedback.