DOI: 10.1111/j.1467-8659.2011.02075.x
The definitive version is available at www.interscience.wiley.com

State-of-the-Art Report on

Temporal Coherence for Stylized Animations

Pierre Bénard[†1,2], Adrien Bousseau[‡3], Joëlle Thollot[§1,2].

[1]Grenoble University [2]INRIA Grenoble Rhône-Alpes [3]REVES / INRIA Sophia-Antipolis

Abstract

Non-photorealistic rendering (NPR) algorithms allow the creation of images in a variety of styles, ranging from line drawing and pen-and-ink to oil painting and watercolor. These algorithms provide greater flexibility, control and automation over traditional drawing and painting. Despite significant progress over the past 15 years, the application of NPR to the generation of stylized animations remains an active area of research. The main challenge of computer generated stylized animations is to reproduce the look of traditional drawings and paintings while minimizing distracting flickering and sliding artifacts present in hand-drawn animations. These goals are inherently conflicting and any attempt to address the temporal coherence of stylized animations is a trade-off. This state-of-the-art report is motivated by the growing number of methods proposed in recent years and the need for a comprehensive analysis of the trade-offs they propose. We formalize the problem of temporal coherence in terms of goals and compare existing methods accordingly. We propose an analysis for both line and region stylization methods and discuss initial steps toward their perceptual evaluation. The goal of our report is to help uninformed readers to choose the method that best suits their needs, as well as motivate further research to address the limitations of existing methods.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation— I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—

Keywords: non-photorealistic rendering, temporal coherence, stylization

1. Introduction

While research in stylized computer animations dates back to 15 years [Mei96], its results have reached mainstream media only recently and partially [MESA*10]. This is in part because the issue of *temporal coherence* prevents the widespread adoption of research algorithms by the industry. Temporal incoherence can introduce several artifacts into the animation (flickering, popping, sliding) that are disturbing for the audience and hardly controllable by artists. Research is yet to propose an algorithm that would ensure perfect temporal coherence for a large variety of styles. Nevertheless, significant advancements have been made in controlling style coherence and minimizing the visibility of the artifacts. This paper attempts to organize current non-photorealistic stylization methods according to their approach on temporal coherence. It offers a global vision on the available techniques, which we hope will help readers choose the appropriate algorithm for their needs and motivate further research.

Following the definitions introduced by Willats and Durand [WD05, Dur02], we describe stylization methods as rendering algorithms that use *marks* (brush strokes, hatches, watercolor pigments, ...) with *attributes* (color, thickness, opacity, ...) to draw *2D primitives* (points, lines and regions) depicting a 3D scene. Existing methods allow the generation of still images with the look of traditional drawings and paintings [WS94, CAS*97, ZZXZ09]. However, a direct application of these algorithms on each frame of an animation produces distracting artifacts as the motion of the marks does

† pierre.benard@laposte.net
‡ adrien.bousseau@inria.fr
§ joelle.thollot@imag.fr

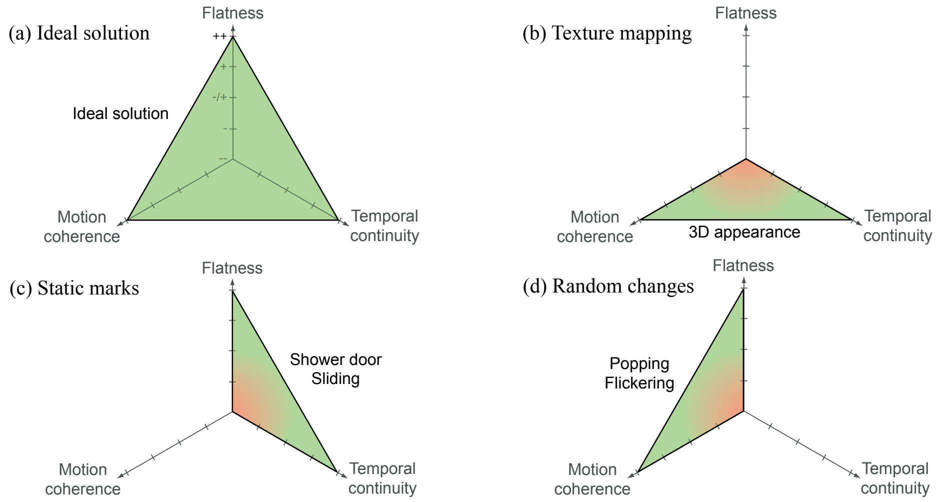


Figure 1: The temporal coherence problem involves three goals represented by the axes of these diagrams. Fully ignoring one of them produces the opposite artifacts.

not match the motion of the depicted scene. Conversely, animating the marks in accordance with the 3D scene motion can alter the impression that each frame is drawn on a flat canvas. This contradiction is at the heart of the temporal coherence problem, for which we provide a formal definition in Section 2.

We make the distinction between methods devoted to the coherent animation of line drawings (Section 3) and methods that address the animation of color regions (Section 4), since these two sub-problems often require different solutions. The organization of this paper highlights the two common steps shared by most methods. The first step is the *extraction* of the 2D primitives (lines or regions) on which the marks will be applied (Section 3.1 and 4.1). The extraction phase often includes the definition of correspondences between the primitives of successive frames. Simplification can be applied during the extraction step in order to remove small primitives that would typically not be drawn in traditional illustrations. The second step, that we call *rendering*, applies the style marks over the extracted primitives (Section 3.2 and 4.2). Various strategies have been developed to ensure that this rendering step produces a temporally coherent animation.

The constraints raised by the temporal coherence problem are inherently contradictory. Existing methods provide different trade-offs to fulfill these constraints. Evaluating the quality of the proposed trade-offs remains challenging for many reasons. Most of the time, no absolute ground truth exists and user aesthetics expectations may not be well defined. Moreover, multiple perceptual effects are involved and they may either emphasize or mask temporal artifacts. In Section 5 we review the first attempts to evaluate the quality of temporal coherence algorithms, and discuss potential directions for future research in this area.

2. Problem Statement

The temporal coherence problem in non-photorealistic rendering encompasses both spatial and temporal aspects of the marks. We propose a formulation of this problem in terms of the concurrent fulfillment of three goals: *flatness*, *motion coherence* and *temporal continuity*. These goals are represented in Figure 1 as three axes of a radar chart whose scales are purely qualitative. In Figure 1a, we show the theoretical ideal solution: an equilateral triangle fully covering the axes, i.e., perfectly fulfilling the three goals. The subsequent sub-figures (Figure 1b-d) show three naïve solutions which completely neglect one goal and produce the opposite artifacts. We define these goals as follow.

Flatness gives the impression that the image is drawn on a flat canvas rather than painted over the 3D objects of the scene [Mei96]. Flatness is a key ingredient in generating computer animations that appear similar to traditional hand-drawn animations. Several properties of the marks must be preserved to produce such a 2D appearance. In particular the size and distribution of marks should be independent of the underlying geometry of the scene. As a typical example, the size of the marks should not increase during a zoom.

Motion coherence is the correlation between the apparent motion flow of the 3D scene and the motion of the marks. A low correlation produces sliding artifacts and gives the impression that the scene is observed through a semi-transparent layer of marks, also referred to as the *shower door effect* [Mei96].

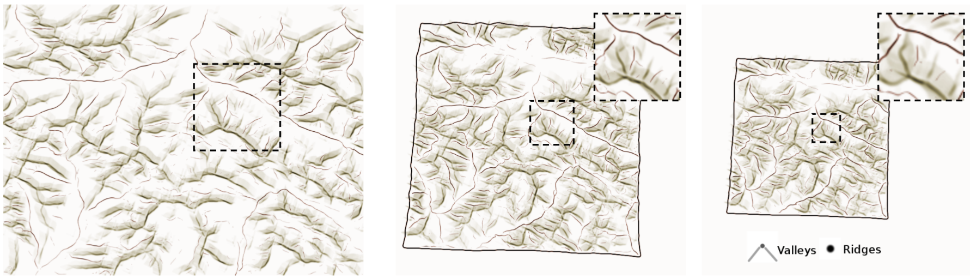


Figure 2: Three frames from a zoom on a terrain model rendered with the "Implicit Brushes" of Verge et al. Surface features exhibit natural LOD and coherence. From [VVC*11].

Temporal continuity minimizes abrupt changes of the marks from frame to frame. Perceptual studies [YJ84, SS09] have shown that human observers are very sensitive to sudden temporal variations such as popping and flickering. The visibility and attributes of the marks should vary smoothly to ensure temporal continuity and fluid animations.

Unfortunately these goals are inherently contradictory and naïve solutions often neglect one or more criteria. While texture mapping can be used to apply the marks over the scene with high motion coherence and temporal continuity (Figure 1b), perspective projection makes the size of the marks vary with depth which breaks the impression of flatness. Keeping the marks static from frame to frame (Figure 1c) ensures flatness and temporal continuity but produces a strong shower door effect since the motion of the marks has no correlation with the motion of the scene. Finally, processing each frame independently as in hand-drawn animation leads to strong flickering and popping since the position of the marks varies randomly from frame to frame (Figure 1d).

The methods we review in this paper address the problem of temporal coherence using different *trade-offs* between these three goals. We highlight these trade-offs, and discuss the variety of styles, performance and implementation complexity of each method.

3. Temporal Coherence of Line Drawings

Line drawings can effectively depict complex information with simple means. In traditional line drawing, artists use ink, pencil or charcoal to draw image discontinuities such as silhouettes, contours or shadow boundaries. Line drawing algorithms often replicate this artistic workflow by first identifying the lines, and then rendering them with a particular medium. Both steps require special consideration to produce temporally coherent animations.

3.1. Extracting Coherent Lines

Many researchers have focused their efforts on the extraction of lines from images and 3D scenes (see the annotated bibliography of Rusinkiewicz et al. [RCDF08]). We classify existing line extraction algorithms as either performed in image space or object space. In each case we review algorithms to extract the lines and methods to build correspondences between lines in multiple frames.

3.1.1. Image Space Lines

Image processing is the simplest and most efficient way to extract lines from 3D scenes, making it particularly suitable for video games [MESA*10]. The seminal work of Saito and Takahashi [ST90] filters depth and normal maps to extract contours and creases. The modern GPU implementation of Nienhaus and Döllner [ND05] uses depth peeling to extract both visible and hidden lines in real-time. These methods produce line drawings with natural level of detail (LOD) based on screen space proximity (Figure 2). LOD improves the readability of the drawing at every scale by preventing clutter.

Lee et al. [LMLH07] propose a more involved filtering technique allowing lines of controllable thickness. They extract luminance features from shaded 3D models using a 2D local fitting approach. Generalizing this method, Vergne et al. [VVC*11] use differential geometry on view-centered buffers (surface gradient and curvature tensor) and cubic polynomial fitting to extract various features and their profile: surface edges, ridges, valleys, inflections, occluding contours and luminance edges. Their method also applies to videos.

Image space methods produce lines made of independent, unconnected pixels, offering neither parameterization nor correspondence across frames. The computer-aided rotoscoping approach of Agarwala et al. [AHSS04] address this limitation by explicitly tracking edges in videos with Bézier curves. These curves can serve as a scaffold to animate brush strokes drawn by the artist. The system is however designed for offline processing and relies on heavy-weight optimization and manual correction.

3.1.2. Object Space Lines

Line drawings can also be generated by extracting feature lines on the surface of 3D models. However, animating such

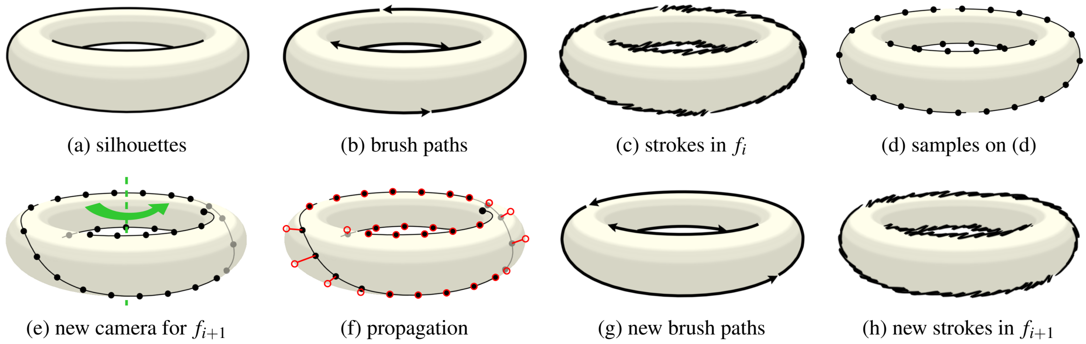


Figure 3: Overview of Kalnins et al. "Coherent Stylized Silhouettes". The input silhouettes (a) are split into continuous brush paths, parameterized and textured (b-c). This parameterization is sampled and propagated by reprojection in the new camera (d-e) and local search in 2D (f). New coherent brush paths are recovered from the votes (g); their parameterization is optimized to compromise between uniformity in 2D and coherence in 3D. From [KDMF03].

feature lines without explicit control of temporal coherence produces a variety of artifacts such as sliding, popping, and splitting.

Some abrupt temporal changes originate from instabilities in the line extraction with respect to small changes in the viewpoint. DeCarlo et al. [DFR04] propose line trimming and fading policies which greatly improve coherence when animating suggestive contours. They analyze the speed of motion of lines over the surface and discard the ones that are moving too fast.

Visibility. Line visibility computation based on 2D buffers (an *item buffer* [NM00, KMM*02, CDF*06] or a *depth buffer* [IHS02]) can also be responsible for some popping due to aliasing. Algorithms for hidden line removal in object space [MKG*97, HZ00, GTDS10] avoid this problem at the price of high computational complexity. The *segment atlas* of Cole and Finkelstein [CF10] provides similar quality at interactive frame rates by exploiting graphics hardware.

Connectivity. The connectivity of the extracted lines can be inferred from their 3D structure. For polygonal meshes, the connectivity information between edges can be used to grow paths by linking together visible pieces of lines [NM00, IHS02]. Simple heuristics on distances and angles are defined to solve ambiguities at lines intersection. These linking algorithms may however be unstable under change of viewpoint.

After projection and clipping, feature lines can be parameterized in each frame using image-space arc-length. Such parameterization allows a broad range of styles like textured lines, wiggles or tapering [NM00, IHS02, GVH07, GTDS10] but the lack of correspondence in parameterization from frame to frame prevents temporally coherent animations. View-independent lines such as creases, ridges and valleys can rely on the underlying surface parameterization to ensure such a correspondence. In contrast, view-dependent lines such as silhouettes, suggestive contours, and apparent ridges move over the surface and have different 3D geometry and topology for each viewpoint [RCDF08]. Two main classes of approaches have been proposed to build a correspondence between view-dependent lines: the first class focuses on real-time applications, while the second one targets precomputed animations.

Parameterization by 2D sample propagation. Building on the seminal work of Masuch et al. [MSS98] and Bourdev [Bou98], Kalnins et al. [KDMF03] propose the first complete method to tackle coherent parameterization of feature lines, using the connectivity of smooth silhouettes [HZ00] to propagate parameterization from frame to frame.

The three key ideas of their system are:

1. propagating the line parameterization through samples in image space from one frame to the next to provide temporal continuity (Figure 3(d-f)),
2. splitting continuous paths into possibly multiple brush paths corresponding to the paths used in previous frames (Figure 3(b,g)), and
3. optimizing an energy function that includes the competing goals of uniform image-space arc-length parameterization, coherence on the object surface, and attempting to merge multiple paths where possible.

This approach offers temporally coherent parameterization for feature lines extracted from simple, smooth objects. Unfortunately, models of moderate complexity, for example the Stanford bunny, generate silhouettes made of many tiny fragments (Figure 4). The method of Bénard et al. [BCGF10] enforces a common parameterization for nearby lines, thereby making short segments behave as parts

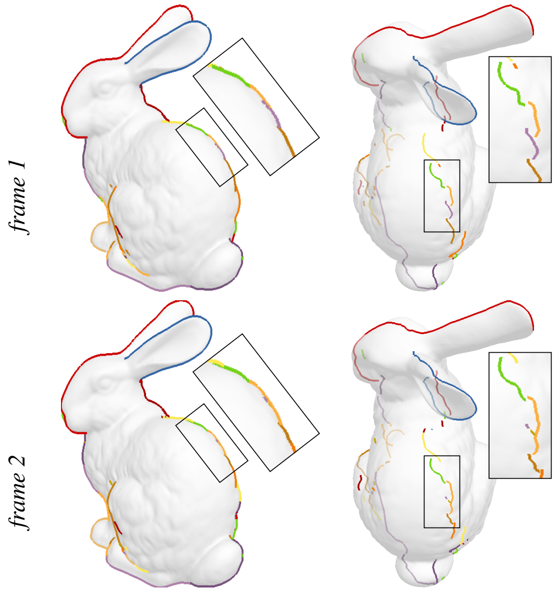


Figure 4: Silhouette coherence. Frame 1: (a) Silhouettes appear to be continuous in the image, but (b) form many disconnected parts in object space. Frame 2: (a) The silhouettes are visually similar to the ones in frame 1, but are (b) composed of a different set of disconnected parts.

of a longer line. This solution is mostly suitable for nearby lines that are also parallel and may lead to popping artifacts when multiple lines merge in a single one.

Karsch and Hart [KH11] use *snaxels* to extract and track visual, shadows and shading contours. Snaxels are closed active contours [KWT88] that minimize an energy on a 3D meshed surface. They define this energy as an implicit contour function corresponding to silhouettes or isophotes. This approach deals robustly with topological events, provides long connected paths and natural correspondences across frames of an animation. These paths can be used as input for the system of Kalnins et al. [KDMF03] to determine a coherent parameterization.

Spatio-temporal formulation. Buchholz et al. [BFP*11] formulate the line correspondence problem as the parameterization of the space-time surface that is swept by the lines during the animation (Figure 5). They propose a robust algorithm to construct this surface, taking into account merging and splitting events of silhouettes. The surface is then parameterized by an optimizer that ensures motion coherence and flatness. Users can control the trade-off between these constraints with few meaningful parameters. The robustness of this approach comes however at the price of expensive computations (several minutes for few seconds of animation).

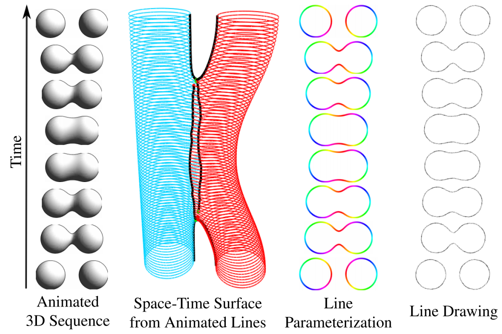


Figure 5: A temporally coherent parameterization for line drawings is computed by parameterizing the space-time surface swept by the line over time. From [BFP*11].

3.2. Line Drawing Rendering

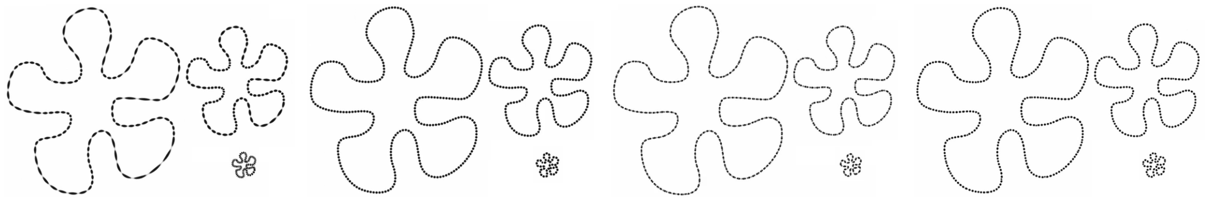
Convolution. Although the lack of parameterization limits the range of styles of image space lines, Vergne et al. [VVC*11] extend the work of Lee et al. [LMLH07] and formulate the mark rendering process as a spatially-varying convolution that mimics the contact of a brush with the line. This implicit style definition ensures flatness (Figure 2) and produces temporally continuous stylized lines from dynamic 3D scenes and videos in real-time.

Texture Mapping. Parameterized lines allow the use of texture mapping to produce dots and dashes or to mimick paint brushes, pencil, ink and other traditional media.

There are two simple policies for texturing a path. The first approach, that we call the *stretching policy* (Figure 6a), stretches or compresses the texture so that it fits along the path a fixed number of times. As the length of the path changes, the texture deforms to match the new length. The second approach, called the *tiling policy* (Figure 6b), establishes a fixed pixel length for the texture, and tiles the path with as many instances of the texture as can fit. Texture tiles appear or disappear as the length of the path varies.

The two policies are appropriate in different cases. The tiling policy is necessary for textures that should not appear to stretch, such as dotted and dashed lines. Because the tiling policy does not stretch the texture, it is also usually preferred for still images. Under animation, however, the texture appears to slide on or off the ends of the path similarly to the shower door effect (Figure 6b) . In contrast, the stretching policy produces high motion coherence under animation, but the stroke texture loses its character if the path is stretched or shrunk too far (Figure 6a). Kalnins et al. [KMM*02] combine these two policies with 1D texture synthesis using Markov random field to reduce repetitions.

The *artmap* method [KLK*00] (Figure 6c) is an alternative to the simple stretching and tiling policies. This method uses texture pyramid, where each texture has a particular tar-



(a) Stretching (b) Tiling (c) Artmap (d) SLAM

Figure 6: Three strokes texture mapping policies. (a) Stretching ensures coherence during motion but deforms the texture. Conversely (b) tiling perfectly preserves the pattern, but produces sliding when the path is animated. (c) Artmap [KLK*00] avoids both problems at the price of fading artifacts because the hand-drawn texture pyramid is usually too sparse. (d) Self-Similar Line Artmap [BCGF10] solves this problem.

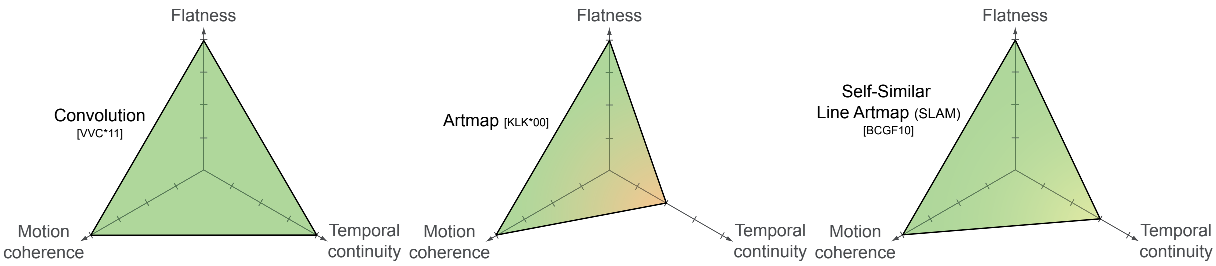


Figure 7: Diagrams of the temporal cohrence trade-offs for line drawings rendering.

		Coherent Parameterization	**Level of Details**
Image space lines	Filtering [ST90, ND05, LMLH07, VVC*11]	−−	++
	Rotoscoping [AHSS04]	+	+
Object space lines	Sample propagation [KDMF03]	+	−−
	Vote splatting [BCGF10]	+/−	−−
	Snaxels [KH11]	+	−−
	Spatio-temporal formulation [BFP*11]	++	−−

Table 1: Summary of the properties of the different line extraction approaches.

		Flatness	**Coherent motion**	**Temporal continuity**
Convolution [VVC*11]		++	++	++
Texture mapping	Tiling	++	−−	++
	Stretching	−−	++	++
	Artmap [KLK*00]	++	++	+/−
	SLAM [BCGF10]	++	++	+

Table 2: Summary of the trade-offs made by the different line rendering approaches.

get length in pixels. At each frame, the texture with the target length closest to the current path length is selected and drawn. This mechanism ensures that the brush texture never appears stretched by more than a constant factor (often 2×).

Nevertheless, stretching artifacts still appear when the length of the path extends beyond the length of the largest texture in the artmap. Fading or popping artifacts can also occur during transitions between levels of the texture pyramid. Finally, a major drawback of the artmap method resides in the manual construction of the texture pyramid. Artists need to draw each level of the pyramid, tacking care of the coherence across levels. As a result, current artmap implementations such as "Styles" in Google SketchUp© use as few as four textures, which accentuates artifacts during transitions.

To reduce transition artifacts and automate the creation process, Bénard et al. [BCGF10] propose *Self-Similar Line Artmap (SLAM)*, an example-based artmap synthesis approach that generates an arbitrarily dense artmap based on a single exemplar. Their synthesis not only guarantees that each artmap level blends seamlessly into the next, but also provides continuous infinite zoom by constructing a self-similar texture pyramid where the last level of the pyramid is included in the first level. The synthesis takes a few minutes as a preprocess and provides good results for many brush textures, although fine details might be lost for very complex patterns.

3.3. How to choose a line extraction and rendering method?

Table 1 compares the various line extraction techniques we have presented with respect to two key properties: their ability to provide a coherent parameterization and to deal with level of detail. Object space methods provide line parameterization which allows complex rendering effects, in particular stroke texture mapping, but may introduce strong temporal artifacts. Image space lines do not offer easy texturing, but naturally support LOD control.

Table 2 and Figure 7 summarize the compromise made by line rendering approaches. Note that the temporal continuity of the texture mapping methods highly depends on the continuity of the input parameterization.

4. Temporal Coherence of Color Regions

Color regions can be represented in a variety of styles including watercolor, oil painting, cross hatching and stippling. This wide range of appearance has conducted researchers to propose different rendering algorithms adapted to the properties and constraints of each style. We first review methods to extract coherent regions in an animation and then discuss the different families of algorithms to render marks over these regions. We highlight for each type of algorithm the range of styles it can produce.

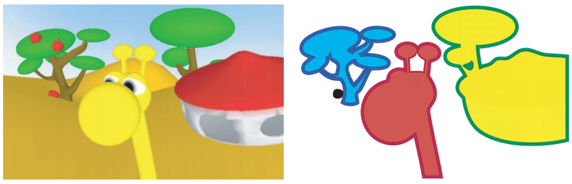


Figure 8: Bezerra et al. "3D dynamic grouping". Objects of the input scene (left) are grouped based on their 3D spatial proximity (right). Adapted from [BEDT08].

4.1. Extracting Coherent Color Regions

Maintaining the temporal coherence of stylized color regions often requires the definition of a correspondence between the regions from frame to frame. In the case of 3D scenes, the correspondence can be directly obtained from the scene geometry. In the case of videos, computer vision algorithms can be used to estimate region correspondence from video streams.

Extraction from 3D scenes. Most algorithms dealing with the stylization of 3D scenes assign a unique identifier to each object of the scene. Rendering this identifier in an *ID buffer* provides a segmentation of the image that evolves coherently with time. This simple approach can however produce many small color regions when objects are distant from the camera. Kolliopoulos et al. [KWH06] propose to abstract out such small details by grouping pixels using a spectral clustering algorithm. Their method clusters pixels based on their similarity in color, ID, normal and depth. They achieve temporal coherence by linking each frame to its predecessor, so that the segmentation of a frame is biased to be similar to the previous frame. Temporal incoherence may however still occur in case of fast motion or occlusions. Bezerra et al. [BEDT08] address these issues with an object-space grouping method based on the mean-shift clustering algorithm (Figure 8). Their implementation of the mean-shift algorithm offers real-time performances, while the spectral clustering of Kolliopoulos et al. [KWH06] requires a few seconds of processing per frame.

Extraction from videos. Extracting coherent regions from videos remains an active area of research in computer vision and several algorithms from this domain have been applied to the stylization of videos.

The *Video Tooning* system [WXSC04] relies on a spatio-temporal mean-shift algorithm to extract 2D+t volumes from a video. Smoothing this volume simplifies the corresponding color regions and allows the creation of different levels of detail. The extracted volume can also be used to interpolate user-specified brush strokes between key-frames. Note however than an additional spatio-temporal parameterization in the spirit of [BFP*11] would be needed to ensure high temporal coherence. Collomosse et al. [CRH05]

		Coherent Parameterization	Level of Details
3D scenes	ID buffer	++	−−
	Image-space clustering [KWH06]	+/−	+
	Object-space clustering [BEDT08]	++	+
Videos	Spatio-temporal segmentation [WXSC04, CRH05, SBCv*11]	+	+
	Image filters [WOG06, KK11]	−−	++
	Optical flow [Lit97, HP00, HE04, SZK*06, BNTS07, KBD*10]	+	+
	Rotoscoping [AHSS04, LZL*10]	++	+

Table 3: Summary of the trade-offs made by the different families of methods for extracting color regions.

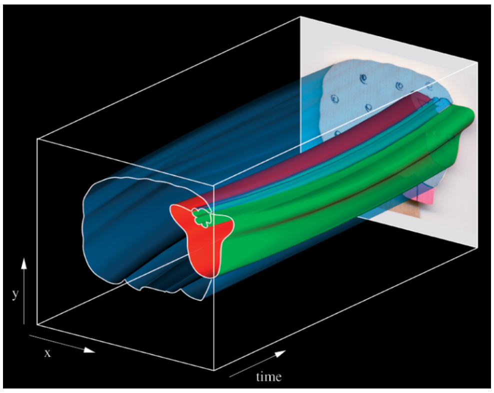


Figure 9: Collomosse et al. "Stroke surfaces". Visualization of the spatio-temporal volume extracted from an animation. Intersecting the volume with a time plane generates a temporally coherent segmentation. From [CRH05].

describe a similar system that first segments the video on a frame-by-frame basis. The algorithm then links the segmented regions from frame to frame (Figure 9). This frame-by-frame approach requires less memory and reduces over-segmentation of the spatio-temporal volume in case of fast motion. Segmentation-based methods remain however computationally demanding and often need user assistance.

Color regions can also be simplified by means of local image filters, similar in spirit to image space line extration. Winnemöller et al. [WOG06] apply a bilateral filter followed by soft quantization to produce cartoon animations from videos in real time. The soft quantization produces results with higher temporal coherence than hard quantization. Various filters have been proposed for similar purpose, such as the coherence-enhancing filter of Kyprianidis and Kang [KK11]. These filtering approaches are fast to compute as they can be implemented on the GPU. However color regions generated this way are defined implicitly, which prevents more complex stylization effects.

Finally, frame-to-frame correspondence can be specified on a pixel basis using motion estimation. Optical flow [BB95] and related methods have been used to maintain temporal coherence for painterly rendering [Lit97, HP00, HE04, KBD*10], watercolor rendering [BNTS07] and hatching [SZK*06]. Bousseau et al. [BNTS07] also use optical flow to orient a spatio-temporal filter along motion trajectories. Their morphological filter removes small features from the video with limited popping and flickering but are slower than filters that operate in the image domain only.

While optical flow provides per-pixel motion estimates, local errors in the motion field can lead to swimming artifacts in the stylized output. Feature tracking produces sparser but more robust estimates [LZL*10]. The *particle video* algorithm [ST06] produces motion estimates that are both dense and robust, but it has not yet been used for non-photorealistic rendering to the best of our knowledge. Complementary to these automatic approaches, user-assisted rotoscoping allows refinement and additional control over the motion estimation [AHSS04, OH11].

For the special case of 2D cartoon animations, Sýkora et al. [SBCv*11] extract dense correspondences between hand-made drawings using as-rigid-as-possible image registration [SDC09]. These correspondences allow the propagation of color and texture coordinates over the animation. In case of disocclusions, the algorithm extrapolates texture coordinate to cover the appearing pixels, which can produce sliding artifacts.

In summary, clustering and segmentation-based approaches explicitly extract color regions that can be used as input for subsequent mark- and texture-based rendering. However, they are usually computationally expensive, and can require user guidance. Conversely, filtering methods perform in real-time with intuitive controls, but they don't provide time coherent parameterization.

4.2. Color Regions Rendering

We classify previous work on coherent region rendering into two main categories: mark-based and texture-based methods. Choosing one or the other has important consequences. The first class of methods tends to compromise temporal continuity, whereas the latter tends to compromise either motion coherence or flatness.

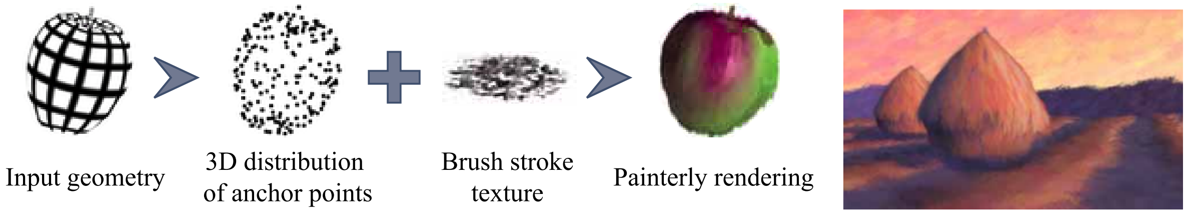


Figure 10: Meier's painterly rendering algorithm. The input geometry is sampled into a 3D distribution of anchor points. For each visible sample, a brush stroke texture is drawn in screen space. Adapted from [Mei96].

4.2.1. Mark-based Methods

Few-marks methods. Methods based on few marks are mostly used for styles where the individual marks are strongly visible, typically painterly rendering and stippling where the marks are the brush strokes and stipples respectively.

Meier [Mei96] introduced the idea of attaching anchor points to the 3D surface of an object and using them to draw overlapping marks in 2D (Figure 10). Since the marks are usually small with respect to object size, their motion remains very close to the original motion field of the 3D scene, providing good motion coherence. Flatness is preserved by drawing the marks as 2D sprites. In its original algorithm, Meier draws anchor points from a static uniform distribution over the object surface. Such an object space distribution is however non-uniform after projection in the image. Consequently, anchor points can be too dense for objects that are far from the camera, and too sparse for nearby objects. Many subsequent work address the issue of anchor point distribution, providing various trade-offs between flatness and temporal continuity. Methods have been proposed to process 3D scenes [KGC00, CL06, PFS03, VBTS07] and videos [Lit97, Her01, CRL01, HE04, VBTS07].

A first solution proposed by Daniels [Dan99] gives control to the user. In this workflow, artists first draw strokes over the object in a given frame of the animation. The strokes are projected and stored on the 3D surface and back-projected in the subsequent frames. Artists can then add or remove strokes in other frames to fill-in holes and avoid clutter. Schmid et al. [SSGS11] extend this approach by allowing strokes to be embedded anywhere in a 3D implicit canvas surrounding proxy surfaces. These methods provide precise control over the final look of the animation but require significant time and expertise.

Dynamic distributions aim at automating this process by adapting the distribution of anchor points in each frame. The goal is to maintain a uniform spacing between anchor points (Poisson disk distribution) while avoiding sudden appearance or disappearance of marks.

Extending the Poisson disk tiling method of Lagae et al. [LD05], Kopf et al. [KCODL06] propose a set of recursive Wang tiles which allows to generate 2D point distributions with blue noise property in real-time and at arbitrary scale. This approach relies on precomputed tiles, handling 2D rigid motions (zooming and panning inside stills). The subdivision mechanism ensures the continuity of the distribution during the zoom, while the recursivity of the scheme enables infinite zoom.

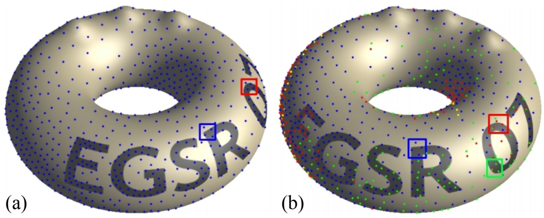


Figure 11: (a) Points are distributed in 2D and projected on the 3D torus. (b) Points are back-projected in the next frame. Red points are rejected based on a Poisson-disk criterion; blue points are still valid; green points are added; and yellow points are removed due to visibility. From [VBTS07].

A dynamic distribution can be precomputed as a hierarchical 3D distribution over the objects surface [CRL01, PFS03, NS04]. At each frame, anchor points are selected as a cut in the hierarchy according to depth and surface orientation. These approaches ensure a good temporal continuity as long as the amount of zoom does not exceed the limit of the hierarchy. Although the dynamic cut prevents the distribution from becoming too sparse or too dense, the distribution is often not Poisson disk after projection in image space [PFS03].

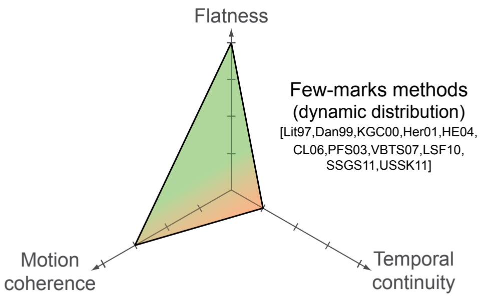
Focusing on hatching patterns, Umenhoffer et al. [USSK11] start from a dense 3D point distribution and propose a deterministic rejection sampling algorithm based on low-discrepancy sequences to control the image space density. A local weighting scheme smooths the rejection of samples due to changes in viewpoint or shading. Their 2D distributions are more uniform than the ones produced by hierarchical approaches, but the zooming range still needs to be defined *a priori*, which limits interactive navigation.

Vanderhaeghe et al. [VBTS07] propose a hybrid technique which finds a more balanced trade-off. They compute the distribution in 2D – ensuring blue noise property – but

Figure 12: Zoom sequence on a sphere textured with NPR Gabor noise [BLV*10] before (top) and after binary threshold (bottom). At each level of the zoom, the statistical properties of the noise are well preserved thanks to their LOD mechanism.

move the points according to the 3D motion of the scene by projecting the points on the 3D geometry (Figure 11). At each frame, the distribution is updated to maintain a Poisson-disk criterion. The temporal continuity is enhanced further by (1) fading appearing and disappearing points over subsequent frames; and (2) allowing points in overly dense regions to slide to close under-sampled regions. Their approach also applies to videos using optical flow to move the anchor points.

The data structure required to manage the anchor points and the rendering of each individual stroke makes few-marks methods complex to implement and not very well-suited to real-time rendering engines. Nevertheless, Lu et al. [LSF10] proposed a full GPU implementation with a simplified stochastic stroke density estimation which runs at interactive framerates but offers less guarantees on the point distribution.

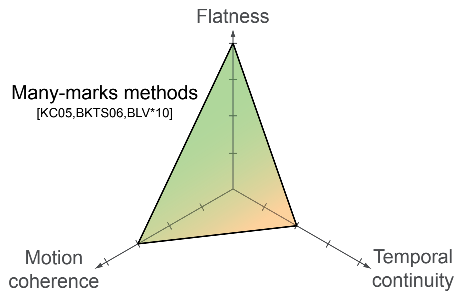


In summary, all these methods offer coherent motion at the price of either a poor flatness due to non-uniform anchor point distribution, or poor temporal continuity due to appearance and disappearance of anchor points. Most methods limit popping by carefully blending new strokes or fading out deleted ones, but such visibility events may still be perceivable.

Many-marks methods. Blending a large number of marks, many-marks methods allow the animation of continuous textures where each individual mark is indistinguishable from its neighbors. These methods often produce less popping artifacts than few-marks methods because each individual mark has much less influence on the final image.

Many-marks methods can be seen as a form of sparse convolution noise used in procedural texturing (see the recent overview of Lagae et al. [LLC*10]). In NPR, Kaplan and Cohen [KC05] use fiber marks to animate a canvas, and Bousseau et al. [BKTS06] approximate Perlin noise with Gaussian kernels to represent watercolor pigments. Bénard et al. [BLV*10] extend the Gabor noise [LLDD09] to create complex non-photorealistic patterns. Although many-marks methods were initially developed to produce continuous patterns, Bénard et al. obtain a wider range of textures by applying color maps over their procedural noise. They demonstrate a variety of styles including cross-hatching, watercolor, stippling and oil painting, although the individual stipples and brush strokes are not as convincing as with few-marks methods. Color maps can also reveal popping artifacts that are not perceivable in the original noise. More studies are needed to better understand the influence of subsequent processing when dealing with procedural stylization.

Kaplan and Cohen [KC05] and Bousseau et al. [BKTS06] distribute marks in each frame using a precomputed hierarchical 3D distribution. Although popping artifacts are greatly reduced compared to few-marks methods, some artifacts remain under close inspection. Bénard et al. [BLV*10] describe a GPU-based algorithm to generate point distributions at interactive rates. Their algorithm relies on a level-of-detail mechanism to adapt the density of the distribution while preserving the statistical properties of Gabor noise (Figure 12). This approach offers a better temporal continuity and reduces popping.



In summary, these approaches bridge the gap between marks-based and texture-based methods by generating a

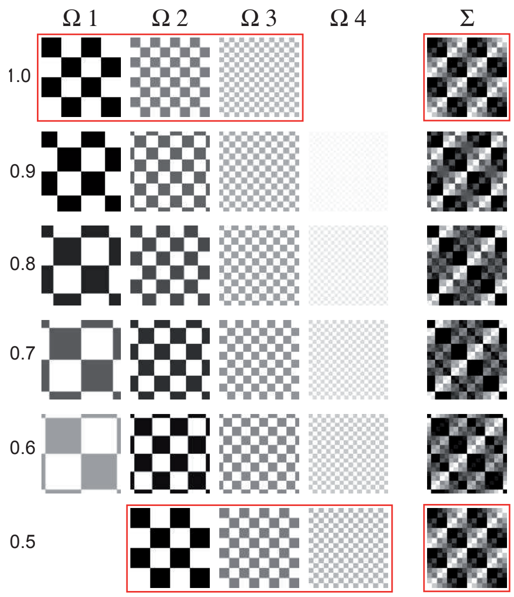


Figure 13: "Dynamic Canvas" infinite zoom mechanism applied on a checkerboard texture. For each zoom factor (left), four scales are blended to produce a texture with an apparent constant scale (right). From [CTP*03].

continuous pattern from discrete marks. It allows them to improve temporal continuity while preserving strong flatness and good motion coherence.

4.2.2. Texture-based Methods

Texture-based approaches are mostly used for continuous textures (canvas, watercolor) or highly structured patterns (hatching). By embedding multiple marks, textures facilitate and accelerate rendering compared to mark-based methods. Textures can be applied over the entire image or over the 3D objects in the scene.

Image space approaches. Applying textures over the image favors flatness over motion coherence. The challenge is to deform the texture so that it follows the scene motion while preserving the appearance of the original pattern. We can distinguished between two sub-families of approaches: planar methods that deform the texture with global as-rigid-as-possible transformations, and texture advection and filtering approaches that work at the pixel level.

Planar approaches. Cunzi et al. [CTP*03] apply a 2D paper texture over the image in order to stylize a 3D environment during a real-time walkthrough. They approximate the 3D motion of the camera with 2D transformations of the texture. This approach provides a convincing trade-off between motion coherence and flatness but is limited to navigation in static scenes with a restricted set of camera motions. Cunzi et al. also introduce an infinite zoom mechanism called *fractalization* to produce a convincing feeling of zoom during camera motion. Taking inspiration from sound illusions [She29], the infinite zoom generates a dynamic texture by alpha-blending multiple scales of the original texture (Figure 13). The scaling factors and blending weights are derived from the camera motion to cycle over the multiple scales and produce the illusion of a globally constant scale.

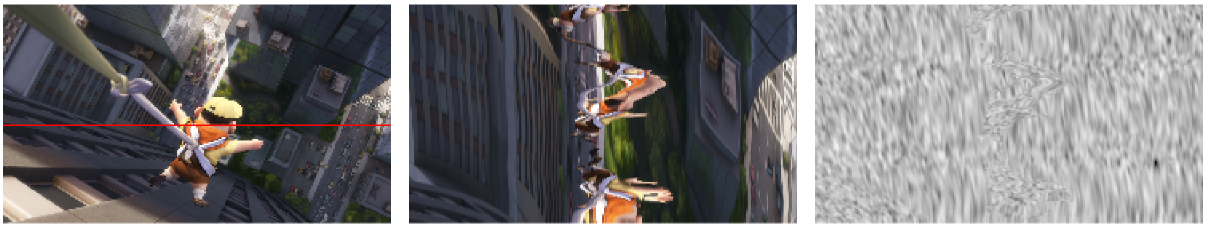
Texture fractalization achieves a very good impression of flatness, but alters the original pattern by replicating and blending multiple scales. While replicates are almost unnoticeable for self-similar stochastic textures (paper fibers, watercolor pigments), artifacts are more visible for structured patterns (brush strokes, stipples). The checkerboard in Figure 13 is an extreme example of such blending artifacts. To better preserve the original pattern, Han et al. [HRRG08] propose an example-based texture synthesis algorithm that generates a texture at multiple scales during a zoom. This method has yet to be applied to non-photorealistic walkthrough scenarios.

The image space mechanism proposed by Cunzi et al. models the scene as a single plane which leads to sliding artifacts for strong parallax. A better approximation of the scene motion can be obtained by modeling the scene as multiple local planes [CDH06, BSM*07]. In such local approaches, the projected 3D transformation of each part of the scene is approximated by the closest 2D rigid transformation in the least-square sense. The 2D rigid transformations preserve the flatness of the texture but sliding artifacts still occur for extreme 3D motions.

Texture advection and filtering. Bousseau et al. [BNTS07] apply non-rigid deformations to animate a texture according to the optical flow of a video [BB95]. This approach extends texture advection methods used in vector field visualization [Ney03] by advecting the texture forward and backward in time to follow the motion field. This *bidirectional* advection allows the method to deal with occlusions where the optical flow is ill-defined in the forward direction but well defined in the backward direction.

The non-rigid deformations can however distort the texture and alter the original pattern. Bousseau et al. propose an advanced blending scheme that periodically regenerates the texture to cancel distortions and favor at each pixel the advected texture with the least distortion. Similarly to texture fractalization, the method is very effective for stochastic patterns, but blending artifacts and small distortions are visible for structured patterns. In addition the bidirectional advection scheme requires the entire animation to be known in advance, which prevents the use of this method for real-time applications.

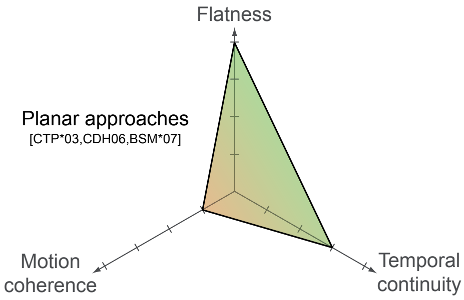
To overcome this limitation, Kass and Pesare [KP11] propose to filter a white or band-pass noise along the motion trajectory of each pixel (Figure 14). Their recursive filter produces a coherent noise with stationary statistics within

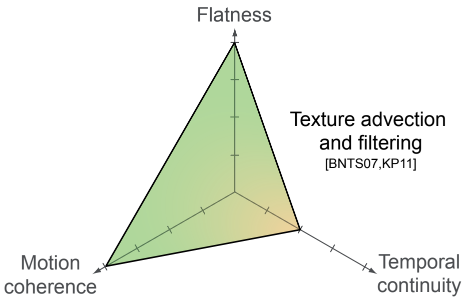


(a) Selected line of original (b) Selected line through time (c) Same line of noise through time

Figure 14: Kass and Pesare "coherent noise". A scan line of the original image (a) is represented over time (from bottom to top) in image (b). Figure (c) displays the evolution of the noise for the same line. The smooth variations of the noise along time and the high correlation with the scene motion illustrate the temporal coherence of this approach. From [KP11].

a frame (high flatness) and high correlations between frames (high motion coherence). This approach is fast enough for real-time application, but is restricted to isotropic procedural noise.





In summary, image-space methods ensure very good flatness but introduce either sliding artifacts, blending artifacts or local distortions. These methods are especially well suited to stochastic patterns such as watercolor pigments and canvas where blending and distortions are less noticeable.

Object space approaches. Applying textures over the 3D objects of the scene favors strong motion coherence at the price of reduced flatness. With texture mapping, the pattern is perfectly attached to the object surface but is severely distorted and scaled by perspective projection. The NPR algorithms in this category propose various mechanisms to compensate for such distortions and maintain a near-constant size of the pattern in image space.

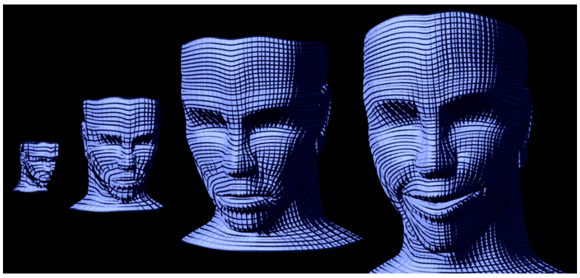


Figure 15: Veryovka "threshold textures". The spacing of hatching lines adapts to orientation, scale and deformation of the face model. Adapted from [Ver02].

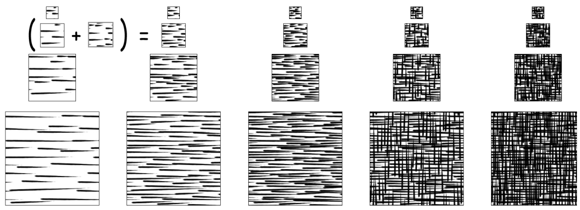


Figure 16: Praun et al. "Tonal Art Map". The hatching pattern of one texture level appears in all the subsequent – above and to the right – levels of the pyramids, ensuring continuity when zooming. From [PHWF01].

Johnston [Joh99] generates procedural hatching patterns as a function of texture coordinates. The texture coordinates are re-mapped according to projection to maintain a near-constant size of lines in image space. Veryovka [Ver02] extends this approach to obtain a more uniform distribution of lines at all scales and better filtering (Figure 15). These approaches are however restricted to procedural patterns and are highly dependent on the input texture parameterization.

The *artmaps* solution [KLK*00] relies on *mipmapping* [Wil83] to adapt the scale of the texture according to the distance to the camera. This approach corrects the texture compression induced by depth and can be ex-

		Flatness	**Coherent motion**	**Temporal continuity**
Naïve	Static marks	++	−−	++
	Texture mapping	−−	++	++
	Random marks	++	++	−−
Few-marks	Fixed distribution [Mei96]	−/+	+	++
	Dynamic distribution [Lit97, USSK11]	++	+	−
Many-marks	Discrete LOD [KC05, BKTS06]	++	+	−
	Continuous LOD [BLV*10]	++	+	−/+
Image space texture	Planar [CTP*03, CDH06, BSM*07]	++	−	+
	Advection and filtering [BNTS07, KP11]	++	++	−/+
Object space texture	Artmaps [KLK*00, PHWF01, FMS01]	−	++	−/+
	Fractalization [BBT09]	−	++	+

Table 4: Summary of the trade-offs made by the different families of methods for rendering color regions.

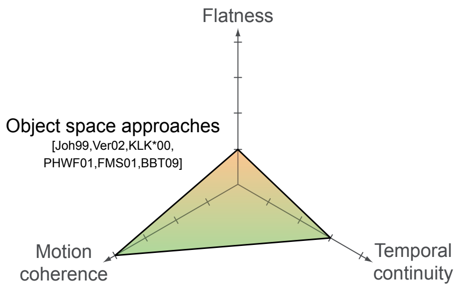
		Stochastic (pigments, canvas)	**Irregular** (hatching, halftoning)	**Near regular** (paint strokes, stipples)
Marks	Few	−−	++	++
	Many	++	+	−
Textures	Procedural	−−	++	−−
	Artmap	+/−	++	−
	Fractalization	++	+/−	−−
	Advection	++	−	−−

Table 5: Summary of the media and patterns best-supported by the different families of methods for rendering regions.

tended to the correction of perspective deformations using the more complex *ripmaps* mechanism. As noted by Praun et al. [PHWF01], higher temporal coherence can be obtained for hatching styles by including the hatches of one "tonal artmap" level into the next level (Figure 16). Popping artifacts may however remain during transitions between levels. Fung and Veryovka [FVA03] propose an automatic algorithm to generate pen-and-ink tonal artmaps from grayscale textures. Freudenberg et al. [FMS01] prove the performance of these methods by integrating an artmap-based rendering in the Fly3D game engine.

Bénard et al. [BBT09] extend the texture fractalization algorithm of Cunzi et al. [CTP*03] to object space. Central to their approach is an object space infinite zoom mechanism that guarantees a quasi-constant size and density of the pattern in image space for any distance to the camera. This approach produces less popping than artmaps, but the method suffers from the same blending artifacts as the original image space method.

In summary, object space methods offer a perfect motion coherence and compensate for most perspective distortions. They are however less effective for structured patterns for which popping or blending artifacts are more visible. Thanks to the hardware acceleration of texture mapping these approaches work in real-time and fit seamlessly into most rendering pipelines.



4.2.3. How to choose a region extraction and rendering method?

Table 3 compares extraction methods according to their ability to compute a coherent parameterization and different levels of detail. Clustering and segmentation-based methods can greatly simplify color regions but as a side effect do not preserve fine details well. Although most motion estimation methods were not designed for the particular goal of simplification, Bousseau et al. [BNTS07] demonstrate how they can be used to reduce flickering and popping when removing small details in videos.

Table 4 summaries the trade-offs made by the different region rendering techniques with respect to the three goals of temporal coherence. However, additional constraints can

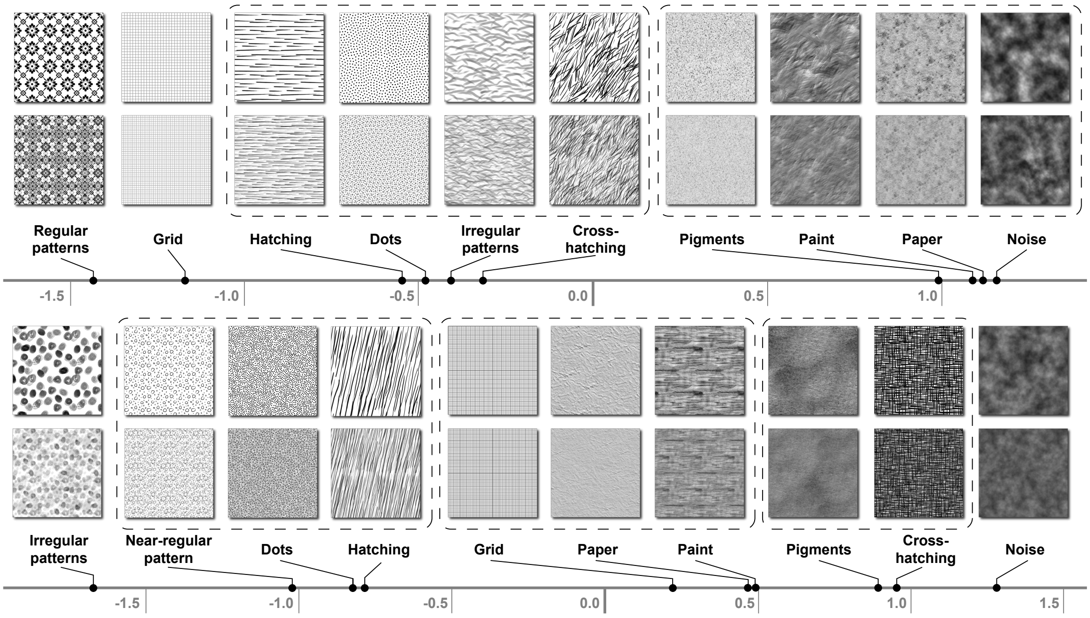


Figure 17: Two series of texture pairs in decreasing order of perceived distortion derived from users rankings. Texture pairs surrounded by the same dashed line may be considered as perceptually equivalently distorted. From [BTS09]

be taken into account when choosing between methods. Besides performance and ease of implementation, the range of styles that can be produced by each approach needs to be considered carefully. Table 5 compares the aptitude of each class of approach at handling stochastic (pigments, canvas), irregular (hatching, halftoning) and near regular (paint strokes, stipples) patterns.

5. Evaluating Temporal Coherence

Evaluation is a longstanding question in non-photorealistic rendering, and up to now most authors only evaluate their methods through visual inspection. In photorealistic rendering and image processing, various approaches have been proposed to assess the fidelity of an image or a video sequence to a reference [Win05, AvMS10]. However, most of the time, no such ground truth exists for non-photorealistic rendering techniques, in particular when dealing with animations. Moreover, the goals of flatness, motion coherence and temporal continuity are contradictory, and therefore no perfect solution exists.

Nevertheless, each goal has specific artifacts that a subject can observe. Perceptual measurements of these artifacts should give an indication of the performance of each solution for a given goal and a given style. It could also provides an indication of the relative importance of the different criteria involved in the compromise. Finally, starting from these perceptual measurements, objective metrics could be derived to automatically assess visual quality.

Perceptual evaluations have been used in expressive rendering for specific styles or applications [SD04, CSD*09], but only two studies focused on temporal coherence. Bénard et al. [BTS09] investigate the effect of fractalization on various 2D textures of media. They define the texture *distortion* as the dissimilarity between the original and transformed texture. They perform a study in which subjects are asked to rank two series of ten pairs of original/distorted textures from the most distorted pair to the least distorted pair (Figure 17). From the subjects' rankings, their goal is to quantify the perceived dissimilarity between texture pairs on a perceptually-linear scale. Using Thurstone's *law of comparative judgment*, they derive for each series an interval scale corresponding to differences in perceived distortion (axes of Figure 17). These plots show that stochastic textures (noise, pigment, paper) are more robust to fractalization. On the contrary, structured textures (near-regular patterns, grids) are judged as the most severely distorted. Finally, they propose the *gray level co-occurrence error* [CRT01] as a predictor showing strong correlation with subjects' judgments.

In a second study, Bénard et al. [BLV*10] compare six rendering methods (shower door, texture mapping, [BSM*07], [BNTS07], [BBT09] and [BLV*10]) with respect to the three goals of flatness, coherent motion and temporal continuity. They evaluate the criteria independently on both a simple and a complex scene for three basic motions (translation, rotation, zoom) and a walkthrough in the complex scene (Figure 18). For each stimulus, they ask subjects to rank images or video sequences rendered with the dif-

(a) A participant performing the ranking experiment

(b) Simple stimuli stylized with "Dynamic Solid Textures" [BBT09] (left) and "Dynamic 2D Patterns" [BSM*07] (right)

(c) Complex stimuli stylized by "texture advection" [BNTS07]

Figure 18: Setup and some stimuli of the user study of Bénard et al. Adapted from [BLV*10].

ferent methods according to a given criteria. A last ranking assess the overall aesthetic of the sequence. The variance in the results for flatness and temporal continuity indicates the complexity of these questions. Conversely, the responses concerning motion coherence are consistently in favor of object-space approaches. Moreover, motion coherence and pleasantness are strongly correlated, which tends to indicate that coherent motion is the most important criteria to preserve in the overall trade-off.

These two studies are first steps toward a formal and perceptually grounded evaluation. They only investigate the stylization of color regions and are restricted to a few techniques and styles. Additional studies are needed to better understand the pros and cons of the available methods, in particular for stylized lines. Finally, new objective measures could be defined to quantify the compromise of each solution. For example, one could use an optical flow analysis to quantify motion coherence and temporal continuity.

6. Discussion and Conclusion

This state-of-the-art report illustrates the large amount of work addressing the issue of temporal coherence of stylized animations. It also highlights a number of limitations that represent interesting directions for future research. The requirements implied by temporal coherence are both contradictory and ill-defined, which in our sense is one of the challenges of this field. We propose a formulation of the temporal coherence problem in terms of three goals in order to facilitate the concurrent analysis of existing methods. The downside of such a classification is that it cannot encompass all the complexity of the involved criteria. We especially found hard to define the goal of *flatness* and chose to restrict it to its geometric definition for the sake of clarity. An alternative definition could be to aim for images that look like hand-made drawings and paintings, although this is harder to quantify. We hope that this state-of-this-art report will encourage the NPR community to refine and complement these definitions.

As a possible starting point, we are convinced that human perception should play a greater role in these definitions. Flatness is related to the perception of *shape from texture*, although in our context we seek to minimize the perception of 3D shapes. Temporal continuity involves *visual attention* which can explain human sensibility to popping. Motion coherence could benefit from studies on *motion transparency* to describe more precisely sliding effects. Beyond the comprehension of the temporal coherence problem, these connections could also help to design new algorithms that better deceive human perception.

The need for a formal evaluation of temporal coherence algorithms is also striking. Perceptual measurement of the artifacts produced by each trade-off should highlight objective trends from individual subjective judgments. Quantitative measurements could be deduced from perceptual evaluations, paving the way to the formulation of temporal coherence as a numerical optimization problem. Such a formulation would give users precise control on the different goals of temporal coherence.

Acknowledgments

The authors would like to thank Pascal Barla, Alexandrina Orzan, Gaurav Chaurasia, Pierre-Édouard Landes and the reviewers for their valuable comments and suggestions. This work has been sponsored by the Animare (ANR-08-JCJC-0078-01) project. Adrien Bousseau thanks the Eurographics PhD award committee for the invitation to submit this State-of-the-Art paper.

References

[AHSS04] Agarwala A., Hertzmann A., Salesin D. H., Seitz S. M.: Keyframe-based tracking for rotoscoping and animation. *ACM Transactions on Graphics 23* (2004), 584–591. 3, 6, 8

[AvMS10] Aydin T. O., Čadík M., Myszkowski K., Seidel H.-P.: Video quality assessment for computer graphics applications. *ACM Transactions on Graphics 29*, 6 (2010), 161:1–161:12. 14

[BB95] Beauchemin S. S., Barron J. L.: The computation of optical flow. *ACM Computing Surveys 27*, 3 (1995), 433–466. 8, 11

[BBT09] Bénard P., Bousseau A., Thollot J.: Dynamic solid textures for real-time coherent stylization. In *Proceedings of the 2009 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (Boston, USA, 2009), ACM press, pp. 121–127. 13, 14, 15

[BCGF10] Bénard P., Cole F., Golovinskiy A., Finkelstein A.: Self-similar texture for coherent line stylization. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France, 2010), ACM press, p. 91. 4, 6, 7

[BEDT08] Bezerra H., Eisemann E., Décoret X., Thollot J.: 3d dynamic grouping for guided stylization. In *Proceedings of the 6th International Symposium on Non-photorealistic Animation and Rendering* (Annecy, France, 2008), ACM Press, pp. 89–95. 7, 8

[BFP*11] Buchholz B., Faraj N., Paris S., Eisemann E., Boubekeur T.: Spatio-temporal analysis for parameterizing animated lines. In *Proceedings of the 10th International Symposium on Non-Photorealistic Animation and Rendering* (Vancouver, Canada, 2011), ACM press, pp. 85–92. 5, 6, 7

[BKTS06] Bousseau A., Kaplan M., Thollot J., Sillion F. X.: Interactive watercolor rendering with temporal coherence and abstraction. In *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering* (Annecy, France, 2006), ACM Press, p. 141. 10, 13

[BLV*10] Bénard P., Lagae A., Vangorp P., Lefebvre S., Drettakis G., Thollot J.: A Dynamic Noise Primitive for Coherent Stylization. *Computer Graphics Forum 29*, 4 (2010), 1497–1506. 10, 13, 14, 15

[BNTS07] Bousseau A., Neyret F., Thollot J., Salesin D.: Video watercolorization using bidirectional texture advection. *ACM Transactions on Graphics 26*, 3 (2007), 104:1–104:7. 8, 11, 13, 14, 15

[Bou98] Bourdev L.: *Rendering Nonphotorealistic Strokes with Temporal and Arc-Length Coherence*. Master's thesis, Brown University, 1998. 4

[BSM*07] Breslav S., Szerszen K., Markosian L., Barla P., Thollot J.: Dynamic 2D patterns for shading 3D scenes. *ACM Transactions on Graphics 26*, 3 (2007), 20:1–20:5. 11, 13, 14, 15

[BTS09] Bénard P., Thollot J., Sillion F.: Quality assessment of fractalized NPR textures: a perceptual objective metric. In *Proceedings of the 6th Symposium on Applied Perception in Graphics and Visualization* (Chania, Greece, 2009), ACM press, pp. 117–120. 14

[CAS*97] Curtis C. J., Anderson S. E., Seims J. E., Fleischer K. W., Salesin D. H.: Computer-generated watercolor. In *Proceedings of SIGGRAPH 97* (Los Angeles, USA, 1997), ACM press, pp. 421–430. 1

[CDF*06] Cole F., DeCarlo D., Finkelstein A., Kin K., Morley K., Santella A.: Directing Gaze in 3D Models with Stylized Focus. In *Proceedings of the 17th Eurographics Symposium on Rendering* (Nicosia, Cyprus, 2006), Eurographics Association, pp. 377–387. 4

[CDH06] Coconu L., Deussen O., Hege H.-C.: Real-time pen-and-ink illustration of landscapes. In *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering* (Annecy, France, 2006), ACM Press, p. 27. 11, 13

[CF10] Cole F., Finkelstein A.: Two fast methods for high-quality line visibility. *IEEE transactions on visualization and computer graphics 16*, 5 (2010), 707–17. 4

[CL06] Chi M.-T., Lee T.-Y.: Stylized and abstract painterly rendering system using a multiscale segmented sphere hierarchy. *IEEE transactions on visualization and computer graphics 12*, 1 (2006), 61–72. 9

[CRH05] Collomosse J. P., Rowntree D., Hall P. M.: Stroke surfaces: temporally coherent artistic animations from video. *IEEE transactions on visualization and computer graphics 11*, 5 (2005), 540–9. 7, 8

[CRL01] Cornish D., Rowan A., Luebke D.: View-dependent particles for interactive non-photorealistic rendering. In *Proceedings of Graphics Interface* (Ottawa, Canada, 2001), A K Peters Ldt, pp. 151–158. 9

[CRT01] Copeland A. C., Ravichandran G., Trivedi M. M.: Texture synthesis using gray-level co-occurrence models: algorithms, experimental analysis, and psychophysical support. *Optical Engineering 40*, 11 (2001), 2655–2673. 14

[CSD*09] Cole F., Sanik K., DeCarlo D., Finkelstein A., Funkhouser T., Rusinkiewicz S., Singh M.: How well do line drawings depict shape? *ACM Transactions on Graphics 28*, 3 (2009), 28:1–28:9. 14

[CTP*03] Cunzi M., Thollot J., Paris S., Debunne G., Gascuel J., Durand F.: Dynamic canvas for non-photorealistic walkthroughs. In *Proceedings of Graphics Interface* (Halifax, Canada, 2003), A K Peters Ldt. 11, 13

[Dan99] Daniels E.: Deep canvas in Disney's Tarzan. In *Proceedings of SIGGRAPH 99* (Los Angeles, USA, 1999), ACM Press. 9

[DFR04] DeCarlo D., Finkelstein A., Rusinkiewicz S.: Interactive rendering of suggestive contours with temporal coherence. In *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France, 2004), ACM press, pp. 15–24. 4

[Dur02] Durand F.: An invitation to discuss computer depiction. In *Proceedings of the 2nd International Symposium on Non-photorealistic Animation and Rendering* (Annecy, France, 2002), ACM press, pp. 111–124. 1

[FMS01] Freudenberg B., Masuch M., Strothotte T.: Walk-Through Illustrations: Frame-Coherent Pen-and-Ink Style in a Game Engine. *Computer Graphics Forum 20*, 3 (2001), 184–191. 13

[FVA03] Fung J., Veryovka O., Arts E.: Pen-and-ink textures for real-time rendering. In *Proceedings of Graphics Interface* (Halifax, Canada, 2003), A K Peters Ldt, pp. 131–138. 13

[GTDS10] Grabli S., Turquin E., Durand F., Sillion F. X.: Programmable rendering of line drawing from 3D scenes. *ACM Transactions on Graphics 29*, 2 (2010), 1–20. 4

[GVH07] Goodwin T., Vollick I., Hertzmann A.: Isophote distance: a shading approach to artistic stroke thickness. In *Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering* (San Diego, USA, 2007), ACM Press. 4

[HE04] Hays J., Essa I.: Image and video based painterly animation. In *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering* (Annecy, France, 2004), ACM Press, p. 113. 8, 9

[Her01] Hertzmann A.: Paint by relaxation. In *Computer Graphics International* (Hong-Kong, July 2001), IEEE Computer Society, pp. 47–54. 9

[HP00] HERTZMANN A., PERLIN K.: Painterly rendering for video and interaction. In *Proceedings of the 1st International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France, 2000), ACM Press, pp. 7–12. 8

[HRRG08] HAN C., RISSER E., RAMAMOORTHI R., GRINSPUN E.: Multiscale texture synthesis. *ACM Transactions on Graphics 27*, 3 (2008), 51:1–51:8. 11

[HZ00] HERTZMANN A., ZORIN D.: Illustrating smooth surfaces. In *Proceedings of SIGGRAPH 2000* (New Orleans, USA, 2000), ACM Press, pp. 517–526. 4

[IHS02] ISENBERG T., HALPER N., STROTHOTTE T.: Stylizing Silhouettes at Interactive Rates: From Silhouette Edges to Silhouette Strokes. *Computer Graphics Forum 21*, 3 (2002), 249–258. 4

[Joh99] JOHNSTON S.: *Nonphotorealistic rendering with Renderman*. Morgan Kaufmann Publishers Inc., San Francisco, USA, 1999, ch. 16, pp. 441–480. 12

[KBD*10] KAGAYA M., BRENDEL W., DENG Q., KESTERSON T., TODOROVIC S., NEILL P. J., ZHANG E.: Video painting with space-time-varying style parameters. *IEEE transactions on visualization and computer graphics 17*, 1 (2010), 74–87. 8

[KC05] KAPLAN M., COHEN E.: A Generative Model For Dynamic Canvas Motion. In *Proceedings of the First Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging* (Girona, 2005), Neumann L., Casasayas M. S., Gooch B., Purgathofer W., (Eds.), Eurographics Association, pp. 49–56. 10, 13

[KCODL06] KOPF J., COHEN-OR D., DEUSSEN O., LISCHINSKI D.: Recursive Wang tiles for real-time blue noise. *ACM Transactions on Graphics 25*, 3 (2006), 509–518. 9

[KDMF03] KALNINS R. D., DAVIDSON P. L., MARKOSIAN L., FINKELSTEIN A.: Coherent stylized silhouettes. *ACM Transactions on Graphics 22*, 3 (2003), 856–861. 4, 5, 6

[KGC00] KAPLAN M., GOOCH B., COHEN E.: Interactive artistic rendering. In *Proceedings of the 1st International Symposium on Non-photorealistic Animation and Rendering* (Annecy, France, 2000), ACM Press, pp. 67–74. 9

[KH11] KARSCH K., HART J. C.: Snaxels on a plane. In *Proceedings of the 10th International Symposium on Non-Photorealistic Animation and Rendering* (Vancouver, Canada, 2011), ACM press, pp. 35–42. 5, 6

[KK11] KYPRIANIDIS J. E., KANG H.: Image and video abstraction by coherence-enhancing filtering. *Computer Graphics Forum 30*, 2 (2011), 593–602. 8

[KLK*00] KLEIN A. W., LI W., KAZHDAN M. M., CORRÊA W. T., FINKELSTEIN A., FUNKHOUSER T. A.: Non-photorealistic virtual environments. In *Proceedings of SIGGRAPH 2000* (New Orleans, USA, 2000), ACM Press, pp. 527–534. 5, 6, 12, 13

[KMM*02] KALNINS R. D., MARKOSIAN L., MEIER B. J., KOWALSKI M. A., LEE J. C., DAVIDSON P. L., WEBB M., HUGHES J. F., FINKELSTEIN A.: WYSIWYG NPR: drawing strokes directly on 3D models. In *Proceedings of SIGGRAPH 2002* (San Antonio, USA, 2002), vol. 21, ACM Press, p. 755. 4, 5

[KP11] KASS M., PESARE D.: Coherent noise for non-photorealistic rendering. *ACM Transactions on Graphics 30* (2011), 30:1–30:6. 11, 12, 13

[KWH06] KOLLIOPOULOS A., WANG J. M., HERTZMANN A.: Segmentation-Based 3D Artistic Rendering. In *Proceedings of the 17th Eurographics Symposium on Rendering* (Nicosia, Cyprus, 2006), Eurographics Association, pp. 361–370. 7, 8

[KWT88] KASS M., WITKIN A., TERZOPOULOS D.: Snakes: Active contour models. *International Journal of Computer Vision 1*, 4 (1988), 321–331. 5

[LD05] LAGAE A., DUTRÉ P.: A procedural object distribution function. *ACM Transactions on Graphics 24*, 4 (2005), 1442–1461. 9

[Lit97] LITWINOWICZ P.: Processing images and video for an impressionist effect. In *Proceedings of SIGGRAPH 97* (Los Angeles, USA, 1997), ACM Press, pp. 407–414. 8, 9, 13

[LLC*10] LAGAE A., LEFEBVRE S., COOK R., DEROSE T., DRETTAKIS G., EBERT D., LEWIS J., PERLIN K., ZWICKER M.: A Survey of Procedural Noise Functions. In *Computer Graphics Forum* (2010), vol. 29, Blackwell Publishing Ltd, pp. 2579–2600. 10

[LLDD09] LAGAE A., LEFEBVRE S., DRETTAKIS G., DUTRÉ P.: Procedural noise using sparse gabor convolution. *ACM Transactions on Graphics 28*, 3 (2009), 54:1–54:10. 10

[LMLH07] LEE Y., MARKOSIAN L., LEE S., HUGHES J. F.: Line drawings via abstracted shading. *ACM Transactions on Graphics* (2007), 18. 3, 5, 6

[LSF10] LU J., SANDER P. V., FINKELSTEIN A.: Interactive painterly stylization of images, videos and 3D animations. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (Washington, USA, 2010), vol. 26, ACM press, pp. 127–134. 10

[LZL*10] LIN L., ZENG K., LV H., WANG Y., XU Y., ZHU S.-C.: Painterly animation using video semantics and feature correspondence. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France, 2010), vol. 1, ACM Press, p. 73. 8

[Mei96] MEIER B. J.: Painterly rendering for animation. In *Proceedings of SIGGRAPH 96* (New Orleans, USA, 1996), ACM press, pp. 477–484. 1, 2, 9, 13

[MESA*10] MCGUIRE M., EKANAYAKE C., ST-AMOUR J.-F., HALÉN H., THIBAULT A., MARTEL B.: Stylized Rendering in Games. In *ACM SIGGRAPH 2010 Courses* (Los Angeles, USA, 2010). 1, 3

[MKG*97] MARKOSIAN L., KOWALSKI M. A., GOLDSTEIN D., TRYCHIN S. J., HUGHES J. F., BOURDEV L. D.: Real-time nonphotorealistic rendering. In *Proceedings of SIGGRAPH 97* (Los Angeles, USA, 1997), ACM Press, pp. 415–420. 4

[MSS98] MASUCH M., SCHUMANN L., SCHLECHTWEG S.: Animating Frame-to-Frame Consistent Line Drawings for Illustrative Purposes. In *SimVis* (Magdeburg, Germany, 1998), SCS Publishing House e.V., pp. 101–112. 4

[ND05] NIENHAUS M., DÖLLNER J.: *Blueprint Rendering and Sketchy Drawings*. Addison-Wesley Professional, 2005, ch. 15, pp. 235–252. 3, 6

[Ney03] NEYRET F.: Advected Textures. In *Proceedings of Eurographics / SIGGRAPH Symposium on Computer Animation* (San Diego, USA, 2003), Eurographics Association, pp. 147–153. 11

[NM00] NORTHRUP J. D., MARKOSIAN L.: Artistic silhouettes. In *Proceedings of the 1st International Symposium on Non-photorealistic Animation and Rendering* (Annecy, France, 2000), ACM Press, pp. 31–37. 4

[NS04] NEHAB D., SHILANE P.: Stratified Point Sampling of 3D Models. In *Eurographics Symposium on Point-Based Graphics* (Zurich, Switzerland, 2004), pp. 49–56. 9

[OH11] O'DONOVAN P., HERTZMANN A.: AniPaint: Interactive Painterly Animation from Video. *IEEE transactions on visualization and computer graphics* (2011). 8

[PFS03] PASTOR O., FREUDENBERG B., STROTHOTTE T.: Real-time animated stippling. *IEEE Computer Graphics and Applications 23*, 4 (2003), 62–68. 9

[PHWF01] PRAUN E., HOPPE H., WEBB M., FINKELSTEIN A.: Real-time hatching. In *Proceedings of SIGGRAPH 2001* (Los Angeles, USA, 2001), ACM Press, p. 581. 12, 13

[RCDF08] RUSINKIEWICZ S., COLE F., DECARLO D., FINKELSTEIN A.: Line drawings from 3D models. In *ACM SIGGRAPH 2008 Courses* (2008). 3, 4

[SBCv∗11] SÝKORA D., BEN-CHEN M., ČADÍK M., WHITED B., SIMMONS M.: Textoons: Practical texture mapping for hand-drawn cartoon animations. In *Proceedings of the 10th International Symposium on Non-Photorealistic Animation and Rendering* (Vancouver, Canada, 2011), ACM press, pp. 75–84. 8

[SD04] SANTELLA A., DECARLO D.: Visual interest and NPR: an evaluation and manifesto. In *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering* (Annecy, France, 2004), ACM press, p. 150. 14

[SDC09] SÝKORA D., DINGLIANA J., COLLINS S.: As-rigid-as-possible image registration for hand-drawn cartoon animations. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering* (Los Angeles, USA, 2009), ACM press, pp. 25–33. 8

[She29] SHEPARD R.: Circularity in judgments of relative pitch. *Psychol. Rev 36* (1929), 172–180. 11

[SS09] SCHWARZ M., STAMMINGER M.: On predicting visual popping in dynamic scenes. In *Proceedings of the 6th Symposium on Applied Perception in Graphics and Visualization* (Chania, Greece, 2009), ACM Press, p. 93. 3

[SSGS11] SCHMID J., SENN M. S., GROSS M., SUMNER R. W.: OverCoat: an implicit canvas for 3d painting. *ACM Transactions on Graphics 30* (2011), 28:1–28:10. 9

[ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-D shapes. *Computer Graphics (Proceedings of SIGGRAPH 90) 24*, 4 (1990), 197–206. 3, 6

[ST06] SAND P., TELLER S.: Particle video: Long-range motion estimation using point trajectories. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (New York, USA, 2006), IEEE Computer Society, pp. 2195–2202. 8

[SZK∗06] SNAVELY N., ZITNICK C. L., KANG S. B., , COHEN M. F.: Stylizing 2.5-d video. In *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering* (Annecy, France, 2006), ACM Press. 8

[USSK11] UMENHOFFER T., SZÉCSI L., SZIRMAY-KALOS L.: Hatching for motion picture production. *Computer Graphics Forum 30* (2011), 533–542. 9, 13

[VBTS07] VANDERHAEGHE D., BARLA P., THOLLOT J., SILLION F.: Dynamic point distribution for stroke-based rendering. In *Proceedings of the 18th Eurographics Symposium on Rendering 2007* (Grenoble, France, 2007), Eurographics Association, pp. 139–146. 9

[Ver02] VERYOVKA O.: Animation with Threshold Textures. In *Proceedings of Graphics Interface* (Calgary, Canada, 2002), A K Peters Ldt, pp. 9–16. 12

[VVC∗11] VERGNE R., VANDERHAEGHE D., CHEN J., BARLA P., GRANIER X., SCHLICK C.: Implicit Brushes for stylized Line-based rendering. *Computer Graphics Forum 30* (2011), 513–522. 3, 5, 6

[WD05] WILLATS J., DURAND F.: Defining pictorial style: Lessons from linguistics and computer graphics. *Axiomathes 15* (2005), 319–351. 1

[Wil83] WILLIAMS L.: Pyramidal parametrics. In *Computer Graphics (Proceedings of SIGGRAPH 83)* (Detroit, USA, 1983), ACM press, pp. 1–11. 12

[Win05] WINKLER S.: Perceptual Video Quality Metrics – A Review. In *Digital Video Image Quality and Perceptual Coding*. CRC Press, Boca Raton, USA, 2005. 14

[WOG06] WINNEMÖLLER H., OLSEN S. C., GOOCH B.: Real-time video abstraction. *ACM Transactions on Graphics 25*, 3 (2006), 1221 – 1226. 8

[WS94] WINKENBACH G., SALESIN D. H.: Computer-generated pen-and-ink illustration. In *Proceedings of SIGGRAPH 94* (Orlando, USA, 1994), ACM press, pp. 91–100. 1

[WXSC04] WANG J., XU Y., SHUM H.-Y., COHEN M. F.: Video tooning. *ACM Transactions on Graphics 23*, 3 (2004), 574–583. 7, 8

[YJ84] YANTIS S., JONIDES J.: Abrupt visual onsets and selective attention: evidence from visual search. *Journal of Experimental Psychology: Human Perception and Performance 10*, 5 (1984), 601–621. 3

[ZZXZ09] ZENG K., ZHAO M., XIONG C., ZHU S.-C.: From image parsing to painterly rendering. *ACM Transactions on Graphics 29* (2009), 2:1–2:11. 1