EG UK Theory and Practice of Computer Graphics (2013)
Silvester Czanner, Wen Tang (Editors)

StegoNPR: Information Hiding in Painterly Renderings

Craig Lord[1] and John Collomosse[1]

[1]Centre for Vision Speech and Signal Processing, University of Surrey, UK

Abstract

We present a novel steganographic technique for concealing information within an image. Uniquely we explore the practicality of hiding, and recovering, data within the pattern of brush strokes generated by a non-photorealistic rendering (NPR) algorithm. We incorporate a linear binary coding (LBC) error correction scheme over a raw data channel established through the local statistics of NPR stroke orientations. This enables us to deliver a robust channel for conveying short (e.g. 30 character) strings covertly within a painterly rendering. We evaluate over a variety of painterly renderings, parameter settings and message lengths.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Non-photorealistic rendering I.3.4 [Computer Graphics]: Graphics Utilities—Paint Systems D.2.11 [Software Architectures]: Information Hiding—Steganography

1. Introduction

The stylization of photographs into synthetic artwork is a fundamental research problem in Non-photorealistic Rendering (NPR) [KCWI12]. Painterly rendering has received considerable scrutiny, and is predominantly tackled through stroke based rendering (SBR) processes. In SBR, virtual brush strokes are composited on canvas with attributes (such as colour and orientation) driven by the underlying image content [Hae90]. In classical SBR models [Hae90, Lit97, Her98, CH02, CH05], stroke attributes are derived locally from a few pixels, whilst the spatial extent of a brush stroke is often tens of pixels. This leads to a non-linear distortion effect that, given carefully chosen stroke placement heuristics, results in an aesthetically pleasing painterly appearance.

In this paper we propose a novel use for SBR; that of concealing information within the pattern of strokes laid down during the painterly rendering process. As such we propose SBR as a novel channel for steganography — covert communication over an existing communication channel. Steganography has been explored in digital formats such as audio, ASCII text, network packets, as well as in images. In all cases the intention is for the sender to embed a second signal, robustly and imperceptibly within the carrier channel, that may be subsequently extracted by the recipient. Often as the volume of embedded data increases, distortion becomes observable within the carrier signal. We suggest painterly rendering — a process that intentionally introduces image distortion for the purpose of aesthetics — to be well suited for steganography, as this distortion serves to mask the embedded signal. To the best of our knowledge, NPR as a steganographic medium has not been previously explored.

Our algorithm is a variant of Hertzmann's curved brush painterly rendering algorithm [Her98]. We embed covert information by quantizing the orientation of spline strokes to create differing patterns of alignment within defined regions of the painting. This permits several hundred binary digits to be reliably embedded as raw data within a typical painting. Within the resulting raw data channel, a linear binary code (LBC) provides error correction resulting a robust channel capable of carrying around 30 characters (e.g. a short message or URL) in a 1080 × 768 resolution painting.

2. Related Work

Steganography (Greek etymology: 'secret writing') dates back millenia to wax covered messages inscribed on tablets, and missives inked upon couriers' skin underneath clothing. Digital steganography has evolved from this concept, often concealing messages within the carrier (*cover text*) signal within components that are information rich but create little or no perceptible change when altered. Examples for text documents include white-space [Kwa13], or invisible Unicode sequences used to delimit Arabic text [MJF07]. Flow control bits have been used to covertly communicate a secondary data channel over TCP/IP packets [DGPP10]. Spread

DOI: 10.2312/LocalChapterEvents.TPCG.TPCG13.065-071

spectrum techniques have been used extensively for audio steganography [ASS*10].

In digital image steganography (DIS), numerous tools exist for encoding data within the least significant bits (LSBs) of pixel RGB values (see [JK00] for a survey), or modulating these values to produce quantized levels of difference [DSMC03]. The frequency domain has also been exploited through selective manipulation of DCT coefficients [CYKB04]. Generally, as covert channel capacity increase so does detectability often via statistical measures e.g. entropy. Pragmatically this is often tackled by encrypting the data or using a secret key of some form (e.g. one time pad [Sch96]) to distribute hidden bits. However falling back upon cryptography, with associated key distribution problems, is not a steganographic solution.

Robustness is an important issue in DIS, and generally decreases with capacity as higher-frequency signal variations are required to embed the covert signal. As such, LSB techniques – which can conceal hundreds of kilobytes per image — are not only detectible, but also fragile being susceptible to lossy compression (e.g. JPEG), and image resampling during photo manipulation. The StirMark tool suite [Pet13] has been developed to evaluate robustness to such processes, comprising a set of tests such as scaling (factor 1.01), rotation (1 degree) and so on. We propose to embed signal within higher-level structure in the image, specifically brush stroke orientations. Because such artifacts exist at larger scale, they are less susceptible to loss due to minor image manipulation. Furthermore the introduction of visual artifacts (strokes) is a natural consequence of painterly rendering, and so is aesthetically plausible as mask for concealing information.

Painterly rendering is a well studied NPR problem [KCWI12], receiving attention since Haeberli's early semi-automated paint systems [Hae90]. These were later automated using a variety of low-level image processing heuristics to determine stroke attributes; for example aligning strokes perpendicular to image gradient [Lit97, Her98] or by image moments [TC97, SY00]. More recently perceptual salience measures have augmented such measures to focus emphasis in the rendering [CH02, CH05] or to use mid-level vision operators to parse and render structure in the scene [DS02, SHRC08, ZZ10]. We have chosen to adapt Hertzmann's multi-resolution curved spline painterly rendering algorithm [Her98] for our approach. Hertzmann's pixel 'hopping' procedure to establish spline control points underpins most subsequent curved stroke painterly algorithms (e.g. [Her02, HE04, CH05, ZZ10]) and so enables our steganographic technique to generalise to these more sophisticated, and to future, techniques.

3. Steganographic Painterly Rendering

We first describe how an RGB colour image $\mathcal{I}$ is converted into a painting $\mathcal{P}$, as a function of pixel colour $I(x,y)$, local gradient magnitude $|\bigtriangledown I(x,y)| = \left(|\frac{\delta I}{\delta x}|^2 + |\frac{\delta I}{\delta y}|^2\right)^{\frac{1}{2}}$ and local gradient orientation $\Theta(x,y) = atan(|\frac{\delta I}{\delta y}|/|\frac{\delta I}{\delta x}|)$. We use notation $M = \{m_1, ..., m_n\}$ to indicate the raw data bits $m_i = \{0,1\}$ we wish to embed within $\mathcal{P}$.

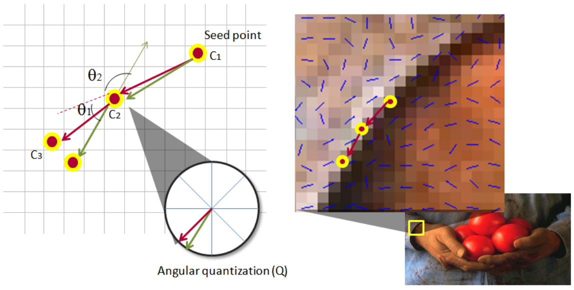

Figure 1: *Illustrating stroke placement. In this example control points $\{c_1, ..., c_3\}$ are generated iteratively by hopping between pixels in direction $Q(\Theta(c_j) \pm \pi; c_j)$ where Q is the angular quantization function local to c_j (defined in Sec. 3.2). Ambiguity $\pm\pi$ arises from the gradient direction. The hop is made in the direction minimising angle between the direction of the previous hop (dashed red line). In this example θ_1 is chosen over θ_2. Green arrows indicate $\Theta(c_j)$ and and red the chosen quantized hop angle.*

3.1. Stroke Placement Process

We represent $\mathcal{P}$ as a sequence of N Catmull-Rom spline [FvDFH90] strokes $\mathcal{P} = \{s_1, ..., s_N\}$ each of which comprises a variable number of 2D control points $s_i = \{c_1, ..., c_n\}$. A decision to seed a stroke is made stochastically at every pixel $p_i = \{c_1\}$; the probability of seeding a stroke at any given point is proportional to the high frequency content present at that location.

$$p(s|c_1) \propto |\bigtriangledown I(c_1)|. \qquad (1)$$

If seeded, a stroke proceeds to grow bi-directionally from c_1 in directions $\Theta(c_1) \pm \pi$; a single direction of growth is illustrated in Figure 1. Growth occurs by iteratively 'hopping' between pixels in the approximate direction of the local gradient orientation. Specifically, given a control point c_j the subsequent control point c_{j+1} is given by:

$$c_{j+1} = c_j + \begin{bmatrix} d\cos\theta_j \\ d\sin\theta_j \end{bmatrix}. \qquad (2)$$

where d is a constant distance proportional to the level of detail being painted (discussed shortly) and θ_j is the angular direction of the $(j-1)^{\text{th}}$ hop:

$$\theta_j = Q(\Theta(c_j) \pm \pi; c_j). \qquad (3)$$

where $Q(.;c_j)$ is an angular quantization function local to c_j, used to embed information in the stroke orientations, discussed in 3.2. The ambiguity $\Theta(c_j) \pm \pi$ (illustrated as θ_{1a}

vs. θ_{1b} in Figure 1) arises due to the lack of directionality on the intensity gradient $\left[|\frac{\delta I}{\delta x}||\frac{\delta I}{\delta y}|\right]^T$. The angle minimising $|\Theta(c_j) - \theta_{j-1}|$ is chosen.

Growth of each end of the stroke proceeds independently until one of several halting criteria are met. Specifically the strokes is terminated if:

1. Adding a new control point c_j would cause the angle between hops to exceed a threshold $|\theta_j - \theta_{j-1}| > T_a$, where we use threshold $T_a = \frac{\pi}{4}$.
2. Adding c_j would depart significantly from the mean colour of the current stroke i.e. $|c_j - \sum_{i=1}^{j-1} I(c_i)| > T_c$, where we use threshold $T_c = 0.2$ in a normalised RGB space.

Following Hertzmann's coarse-to-fine painting strategy, the entire stroke placement process is iterated several times at different spatial scales. A low-pass pyramid is formed and a painting generated for the coarsest layer, after which successively finer layers are over-painted. In each case d is varied inversely proportional to the pyramid level, so that larger strokes are painted at coarser scale. In our experiments we use two octave intervals of scale. Paint strokes are rendered using the mean colour of their control points i.e. $\frac{1}{m}\sum j = 1^m I(c_j)$ and a texture map with the appearance of brush bristles applied to enhance aesthetics.

3.2. Robust Data Embedding

Data is embedded within the painterly rendering via the angular quantization function $Q(\theta_j;c_j)$ within equation 3, where θ_j is the hop angle and $c_j = [x_j y_j]^T$ is the 2D control point location. Rather than painting a stroke with hop angles derived directly from $\Theta(.)$, θ_j is quantized to one of several bin intervals within $[0, 2\pi]$. In section 4.2 we evaluate configurations using 4 and 8 angular quantization bins as the parameter governs a trade between aesthetics and robustness. An alternating pattern of set (1) and reset (0) bit labels are distributed over the angular quantization bins as indicated in Figure 2 (a).

We divide the painting canvas into a rectilinear grid producing a set of $\mathcal{R} = \{r_1, ..., r_{w\times h}\}$ regions, each of which offer one bit of message capacity for M i.e. r_1 encodes m_1. During the message encoding process, quantization function $Q(\theta_j;c_j)$ determines the spatial region in which c_j falls. The quantization of θ_j is performed against bins labelled as matching m_i. Thus in an 8 bin quantization scheme, θ_j will be quantized into a value drawn from one of two angle sets $\{0, \frac{pi}{2}, pi, \frac{3pi}{2}\}$ or $\{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\}$ more concisely:

$$Q(\theta_j;c_j) = \frac{1}{4}m_i + 2\pi \min_{\alpha} |\theta - \frac{(m_i + 2\alpha)\pi}{4}|. \qquad (4)$$

where i is determined by spatial quantization of c_j into the closest grid region r_i. In practice it is desirable to perform angular quantization only a fraction of the grid cell, leaving the remainder with 'natural' stroke orientations to mitigate against pattern artifacts arising from the regularity of strokes Figure 2 (b,c).

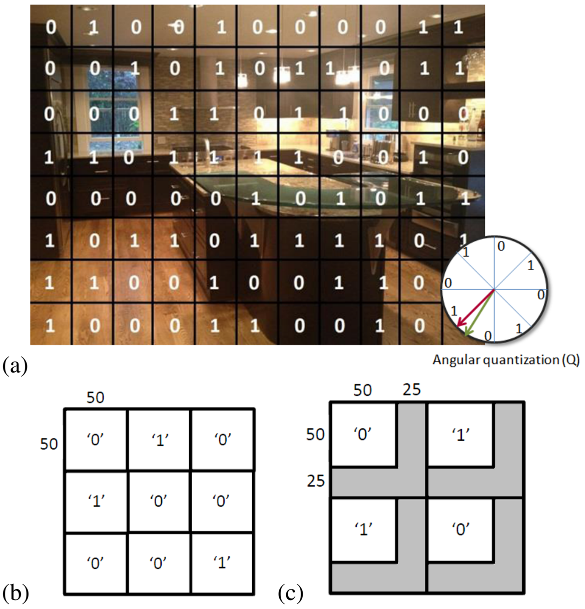

Figure 2: *Data embedding strategy. (a) The canvas is spatially divided into a grid of regions. Each region is assigned a bit of the message (following transformation of the raw message into a robust stream via Hamming coding). The bit determines the angular quantization $Q(.)$ of hops made during stroke construction. (b) grid cells are interspersed with regions on non-quantized stroke orientation which can mitigate against perceptual regularity of a tighter gridding strategy (c).*

3.2.1. Error Correction

The bit stream M can carry any data, and so it is straightforward to robustify the data embedding using a forward error correcting (FEC) code. FEC extends a bitstream by embedding additional parity bits introducing a data redundancy that enables detection and correction of errors (bit flips). Since M is a predefined constant length $w \times h$ in our application, we must sacrifice some capacity for robustness when using FEC.

A number of robust FEC schemes exist, for example Reed-Solomon coding used in digital communications and CDROMs. Whilst robust, many such algorithms cannot operate efficiently over the few tens of characters message length offered by our channel. We therefore opt for a linear binary coding system (LBC), specifically a Hamming code, in which consecutive groups of k bits within M are represented by 2^k codewords of bit length n in the encoded message M'. Hamming codes are referred to in triplet notation (n,k,d) where d indicates the minimum Hamming distance between any of the 2^k codewords. Crucially, a Hamming code is able to detect up to $k/2$ errors (the 'syndrome') and correct up to $(k/2)-1$ errors in a Hamming coded bit stream. Detection is performed by observing codewords in the bit stream that are not part of the codebook. Correction is performed by identifying the closest valid codeword to the received codeword, in terms of Hamming distance i.e. bit flips required to transform one into the other.

Hamming codes can be discovered through exhaustive search — we use a (7,4,3) code for our system, resulting in a 42.8% reduction in capacity but the ability to correct up to 12.5% errors in the stream. The codebook for our (7,4,3) code is $\{0,113,98,19,84,37,54,71,56,73,90,43,108,29,14,127\}$.

3.3. Decoding Process

Given the image of a painting P, and knowledge of the levels of spatial (grid) and angular quantization used during data embedding, the concealed bit stream M may be recovered by analysing the orientations of rendering primitives in P. As it is challenging to segment individual primitives, which are often textured for aesthetic effect, we instead recover the orientations of stroke outlines using low-level signal processing operations.

First, the gradient magnitude and orientation are computed from the painting yielding $| \bigtriangledown P(x,y)| = \left(|\frac{\delta P}{\delta x}|^2 + |\frac{\delta P}{\delta y}|^2 \right)^{\frac{1}{2}}$ and local gradient orientation $\Theta'(x,y) = atan(|\frac{\delta I}{\delta y}|/|\frac{\delta I}{\delta x}|)$. The objective is to recover a histogram of gradient orientations from pixels within each grid cell, the peak of which corresponds to the predominant gradient orientation and so directly to the relevant bit in the embedded data stream. To increase the signal to noise ratio within this histogram, pixels contribute only if their gradient magnitude exceeds a threshold (in our experiments, 15% of the maximum possible gradient magnitude). Figure 3 illustrates the importance of thresholding the edge map to create a unimodal distribution in the histogram and good signal to noise ratio.

4. Results and Discussion

We evaluate our system using five Creative Commons images (Figure 6a) containing varying levels of high frequency detail. The image resolution in all cases is 1080 × 768 pixels. We experiment with two angular quantization levels (4 and 8), and grid configurations suitable for storing messages of various lengths with and without error correction.

4.1. Qualitative Results

Figure 6 presents the results of embedding an identical 30 8-bit ASCII message within five paintings (with error correction). The length of the embedded data is 420 bits with correction (240 raw data bits). Figure 6c,d embed the image using the default angular quantization level of 8. The messages is exactly recoverable in both cases. Figure 6c illustrates that a bump map (inset) may be applied to strokes (after [Her02]) in addition to texture mapping, with the message remaining recoverable after the process. Figures 6b,d-e embed the message using a quantization level of 4. An increased regularity is observed in the stroke patterns at this quantization level.

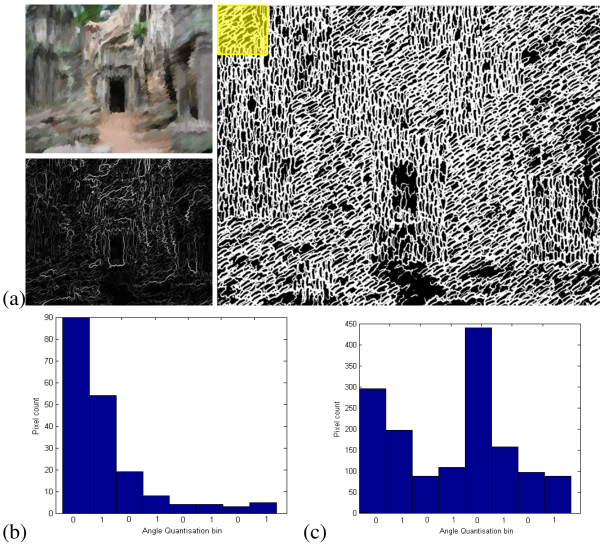

Figure 3: *Decoding the painting: gradient orientation on dominant edges is analysed to produce a frequency histogram. Yellow region indicates the grid cell analysed. (a) Edge mask and orientation field derived from a painting. (b) Orientation histogram of dominant edges. (c) Orientation histogram of all edges.*

4.2. Quantifying robustness vs. capacity

The benefits of the LBC error correction was evaluated by embedding five messages of varying bit lengths (40, 88, 144, 176, 240) within the 5 test paintings, and attempting to recover the information. The angular quantization level was 4. The test was repeated with and without error correction (Figure 5a,b respectively). In all cases where corruption occurred, error correction was shown to minimise data loss. With error correction enabled only two paintings exhibited data corruption, in the worst case 3 bits of 240 i.e. a loss rate of 1.25%.

The experiment was repeated with an angular quantization level of 8. Figure 5c indicates a higher error rate, at the cost of improved aesthetics through an improved gamut of stroke orientations (compare stroke placement in Figure 4). In both cases the Kitchen image exhibited greater error at higher capacities, possibly due to the image exhibiting larger expanses

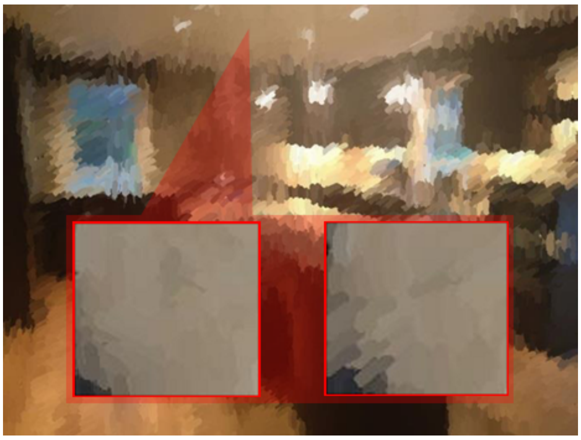

Figure 4: *The aesthetics of stroke placement improve with an 8 level (left) vs. 4 level (right) quantization scheme, at a small cost to robustness.*

of flat texture. Flat texture is likely to be problematic since the colour of neighbouring strokes will be similar, and thus a stroke edge (from which orientation may be recovered) is less likely to manifest in the generated painting.

A final experiment emulated a StirMark attack by scaling the data embedded painting by 10% and 20% in Photoshop prior to decoding. No corruption was observed. When error correction was disabled, very little corruption occurred at 10% with an average of 1.6 bits dropped. At 20% the decoder, again with error correction disabled, dropped an average of 0.75 bits. We conclude that any resampling artifacts introduced are comfortably corrected by the Hamming coding scheme.

5. Conclusion

We have presented a novel form of steganography; concealing information with the output of a painterly rendering algorithm. To the best of our knowledge this is the first time the intentional aesthetic distortions of an image-based NPR algorithm have been used to conceal information. Utilising a simple Hamming coding scheme for error correction over our covert channel, we have demonstrated the ability to robust store up to 30 characters (240 bits) within a single megapixel image. We have demonstrated that output aesthetics (fidelity of stroke orientation) may be traded for robustness and/or capacity by varying the level of angular quantization in the algorithm.

Potential avenues for further development include more sophisticated Computer Vision techniques to isolate strokes during decoding, thereby offering greater resilience to image alteration and potentially greater capacity. The finest grid size tested featured cells of size approximately 50 × 25 (for 240 error corrected bits = 420 raw bits). This is approaching the resolution of the larger individual brush strokes. The ability to robustly isolate brush strokes for analysis (perhaps of their medial axis) might improve the handling of smaller strokes and so an increase in the maximum practical grid resolution for data embedding.

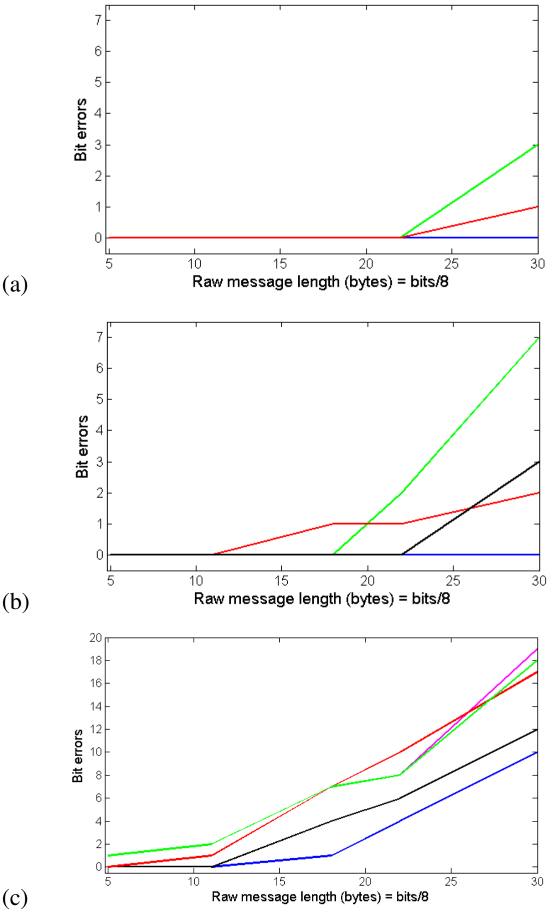

Figure 5: *Exploring the trade-off between quantization level and robustness. Plotting bit dropout across the channel with quantization level 4, with (a) and without (b) LBC error correction. Plotting bit errors vs. bytes encoded for the 5 test images. Charles (magenta), Ruins (blue), Causeway (black), Jungle (red), Kitchen (green). (c) Plotting bit dropout across an 8 quantization level channel using LBC.*

References

[ASS*10] Anying X., Shuwei L., Shan X., Juhua H., Sisi X.: Audio information hiding based on distance metric. In *Proc. Intl. Conf. Signal Processing Systems (ICSPS)* (2010), IEEE, pp. 443–446. 2

[CH02] Collomosse J. P., Hall P. M.: Painterly rendering using image salience. In *Proceedings 20th Eurographics UK Con-*

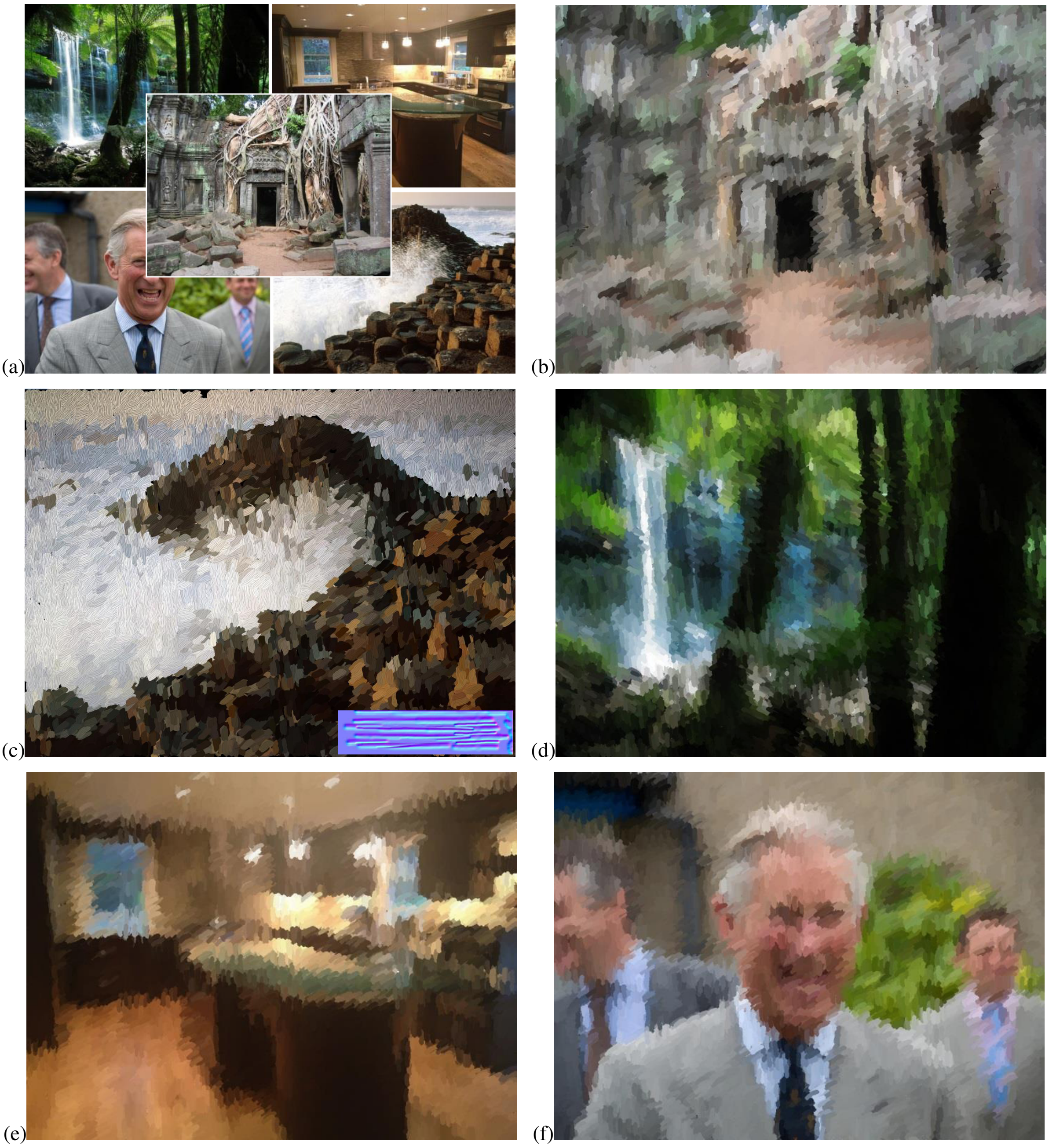

Figure 6: *A gallery of steganographic paintings (b-f): Ruins, Causeway, Jungle, Kitchen, Charles, generated from the corresponding five photographs (a) using our method under various experimental configuration. The 30 character 8-bit ASCII message 'This is a longer test message!' has been embedded in all paintings. See section 4 for further discussion.*

ference (EGUK) (Leicester, June 2002), Eurographics, pp. 122–128. 1, 2

[CH05] Collomosse J., Hall P.: Genetic paint: A search for salient paintings. In *Proceedings of EvoMUSART (LNCS)* (March 2005), vol. 3449, Springer, pp. 437–447. 1, 2

[CYKB04] Chu F., You X., Kong X., Ba X.: A DCT-based image steganographic method resisting statistical attacks. In *Proc. ICASSP* (2004), IEEE, pp. 953–956. 2

[DGPP10] Dhobale D., Ghorpade V., Patil B., Patil S.: Steganography by hiding data in tcp/ip headers. In *Intl. Conf. on Advanced Computer Theory and Engineering (ICACTE)* (2010), vol. 4, pp. 61–65. 1

[DS02] DeCarlo D., Santella A.: Stylization and abstraction of photographs. In *Proc. SIGGRAPH* (2002), pp. 769–776. 2

[DSMC03] Dabeer O., Sullivan K., Madhow U., Chandrasekharan S.: Detection of hiding in the least significant bit. 2

[FvDFH90] Foley J. D., van Dam A., Feiner S. K., Hughes J. F.: *Computer graphics: principles and practice (2nd ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990. 2

[Hae90] Haeberli P.: Paint by numbers: Abstract image representations. In *Proc. SIGGRAPH* (1990), pp. 207–214. 1, 2

[HE04] Hays J., Essa I.: Image and video based painterly animation. In *Proc. NPAR* (2004), pp. 113–120. 2

[Her98] Hertzmann A.: Painterly rendering with curved brush strokes of multiple sizes. In *Proc. SIGGRAPH* (1998), pp. 453–460. 1, 2

[Her02] Hertzmann A.: Fast paint texture. In *Proc. NPAR* (2002), pp. 91–96. 2, 4

[JK00] Johnson N., Katzenbeisser S.: A survey of steganographic techniques. *Information Hiding* (2000), 43–78. 2

[KCWI12] Kyprianidis J.-E., Collomosse J., Wang T., Isenberg T.: State of the ́art ́: A taxonomy of artistic stylization techniques for images and video. *IEEE Transactions on Visualization and Computer Graphics* (2012). 1, 2

[Kwa13] Kwan M.: The SNOW homepage. http://www.darkside.com.au/snow/, 2013. [Online; accessed May 2013]. 1

[Lit97] Litwinowicz P.: Processing images and video for an impressionist effect. In *Proc. SIGGRAPH* (1997), pp. 407–414. 1, 2

[MJF07] Mabry F., James J., Ferguson A.: Unicode steganographic exploits: Maintaining enterprise border security. *Security Privacy, IEEE 5*, 5 (2007), 32–39. 1

[Pet13] Petitcolas F.: The stirmark benchmark. http://www.petitcolas.net/fabien/watermarking/stirmark/, 2013. [Online; accessed May 2013]. 2

[Sch96] Schneier B.: *Applied cryptography: protocols, algorithms, and source code in C*, 2nd ed. Wiley, New York, 1996. 2

[SHRC08] Song Y. Z., Hall P. M., Rosin P. L., Collomosse J. P.: Arty shapes. In *Proceedings of Computational Aesthetics* (June 2008), pp. 65–73. 2

[SY00] Shiraishi M., Yamaguchi Y.: An algorithm for automatic painterly rendering based on local source image approximation. In *Proc. NPAR* (2000), pp. 53–58. 2

[TC97] Treavett S. M. F., Chen M.: Statistical techniques for the automated synthesis of non-photorealistic images. In *Proc. EGUK* (1997), pp. 201–210. 2

[ZZ10] Zhao M., Zhu S.-C.: Sisley the abstract painter. In *NPAR '10: Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering* (2010), pp. 99–107. 2