2013 13th International Conference on Computer-Aided Design and Computer Graphics

Stego-Marbling-Texture

Jiayi Xu
Hangzhou Dianzi University, China
Email: xujiay@hdu.edu.cn

Xiaoyang Mao (Corresponding Author)
University of Yamanashi, Japan
Email: mao@yamanashi.ac.jp

Xiaogang Jin
State Key Lab of CAD&CG, Zhejiang University, China
Email: jin@cad.zju.edu.cn

Aubrey Jaffer
Digilant, USA
Email: agj@alum.mit.edu

Shufang Lu
State Key Lab of CAD&CG, Zhejiang University, China
Email: lushufang@cad.zju.edu.cn

Li Li
Hangzhou Dianzi University, China
Email: lili2008@hdu.edu.cn

Masahiro Toyoura
University of Yamanashi, Japan
Email: mtoyoura@yamanashi.ac.jp

***Abstract*—We present *stego-marbling-texture*, a new and unique texture design method which allows users to deliver personalized messages with beautiful marbling textures. Our approach is inspired by the success of the recent work on modeling traditional marbling operations as mathematical functions. The encrypter transforms an input image or a text message into an intricate marbling pattern using marbling operations defined as reversible functions, and the decrypter recovers the input image or message through reversing the process of marbling operations. When applying marbling operations, the parameters of operations are automatically recorded, encrypted, and then invisibly embedded into the marbling pattern to create a stego-marbling-texture. In this way, the decrypter can be implemented as a stand along software, enabling the receiver to extract the hidden message from the stego-marbling-texture without requiring any extra information from the sender. To ensure that the message is unnoticeably and beautifully covered by the marbling texture, we propose a new technique for automatically creating a background which is harmonious with the input message based on a set of visual perception cues.**

***Keywords*-Texture design; Marbling texture; Steganography; Information hiding; Visual perception;**

I. Introduction

Marbling is a traditional art of making patterns by manipulating the colors floating on liquid. Through centuries, marbling patterns have been widely used for decorating various products such as book cover, stationery, wallpaper, wrapping paper, fabric, picture frames, and so on. This traditional art is revitalized recently with modern computational technologies. A series of research works [1] [2] have been conducted to simulate the pattern formation process on computers and several digital marbling systems have been developed allowing users to create fascinating marbling textures interactively. Inspired by the latest research work on mathematical marbling [2], this paper proposes a novel technique for creating value-added marbling textures which can convey additional information. The technique is comprised of an encrypter and a decrypter. The encrypter is realized as an interactive marbling texture design system which allows a user to take a message or an image as the input and turn it into a desired marbling pattern by successively applying reversible mathematical marbling functions. The parameters of the functions are automatically recorded, encrypted, and invisibly embedded into the resulting marbling pattern to create a *stego-marbling-texture*. The decrypter extracts the marbling parameters from the stego-marbling-texture and then reveals the hidden message by applying the reverse of the marbling functions.

One large advantage of marbling is that only a few basic operations are sufficient to design a beautiful piece of art and even novice users can easily create an intricate and mysterious pattern by compounding the basic operation steps. By representing each operation as the parameters of the mathematical formula, our technique succeeded in encoding the sequence of marbling operations as a compact bit string which can be embedded into the marbling pattern itself. In this way, any marbling textures, freely created by a user can be used for conveying private messages. Since the stego-marbling-textures consist of all the information required for retrieving the hidden messages, the decrypter can be implemented and distributed independently of the interactive marbling texture design system. For example, we can make the decrypter as a part of an image browser or a cell phone application.

Figure 1(a) shows an example of using stego-marbling-texture for decorating an e-card. One can artistically transform a photograph or a private message into a stego-marbling-texture and share it with intimate friends only. The receiver of the e-card can enjoy watching the animation in which the private message appears gradually when the decrypter executes the reverse marbling operations successively. Fig-

978-1-4799-2576-6/13 $31.00 © 2013 IEEE
DOI 10.1109/CADGraphics.2013.38

Figure 1: Stego-marbling-textures and the process to reveal the hidden messages. (a) An e-card; (b) Decorated virtual objects.

ure 1(b) shows an example of using stego-marbling-texture for decorating virtual objects in 3D rendered scenes. By making the decrypter as a part of a renderer, we can produce fascinating animations from a single stego-marbling-texture. The proposed technique can also be viewed as a unique data hiding technique with which messages are invisibly inwrought in beautiful marbling patterns. Unlike the majority of the information embedding algorithms that may pose various limits on the hidden message or may require the use of visually unpleasing cover images, our technique can embed any image or hand writing message in its original form into a beautiful marbling texture.

To be more specific, our work offers the following contributions:

- We present the novel idea of stego-marbling-texture, which is useful not only as an unique texture synthesis technique but also an effective information embedding technique.
- We refine and extend Lu's work [2] by defining a set of marbling operations as reversible mathematical functions. Even novice users can generate intricate patterns freely by combining these very basic operations in different order. The resulting marbling texture can be used for various decorating purposes.
- We propose an automatic background generation method based on perceptual measures to make the embedded message less noticeable and retain harmonious in the resulting texture.
- We propose a new data encoding and embedding technique to invisibly hide the operation parameters into the resulting marbling texture, covering any message with his/own preferred marbling patterns and making it visible only to authenticated receivers when necessary.

II. Related Work

Traditional marbling design involves tedious manual steps. Being able to relieve users from physical constraints and tough manual steps, computer marbling systems have been attracting large attentions in the past decade. Typical computer marbling texture design systems follow the fluid dynamic theory. Mao et al. [1] first succeeded in modeling marbling texture design process as a 2D computational fluid dynamic problem. Akgun [3] presented a design tool for creating the Turkish style marbling and improved the speed problem by reducing the image size. Acar et al. [4] proposed to use a multi-scale fluid solver to model high turbulent marbling patterns. Jin et al. [5] extended Mao's work and first succeeded in achieving real-time performance by implementing the solver on GPU. Xu et al. [6] succeeded in generating more realistic marbling textures by using complex high-order advection schemes to alleviate dissipation. Zhao et al. [7] employed an accurate yet fast third-order unsplit semi-Lagragian constrained interpolation profile method to further reduce numerical dissipation with improved stability. Recently, instead of physical simulation, Lu et al. [2] used mathematical formulae to efficiently describe the marbling operations. In addition to the advantage of being computationally efficient and completely dissipation free, the mathematical-formula-based approach also implied the possibility of defining reversible marbling operations. Our technique refines and extends Lu et al's formulae to define a set of reversible marbling operations. A user can turn a private message into a beautiful marbling texture by combining those operations in different order, and the message can be recovered when necessary by reversing the operations.

Our technique can also be used as an effective steganography technology, which aims to conceal a message inside some media that will not arouse suspicion [8]. The use of steganography technologies can be dated back to thousands years ago. One famous method of the traditional steganography technique is to write a message with invisible secret ink, such as lemon juice. When the ink is heated, the message appears magically. Our technique mimics such effects digitally. By processing a stego-marbling-texture with a decrypter installed computer, the embedded private message appears gradually when the reverse of marbling operations is applied. The major components of a steganography system include the cover message, secret message, stego key, and embedding algorithm. Although many digital media, such as text and video can be used as cover massages in recent technologies, image remains to be most popularly used [8]. Watermarking can also be viewed as a kind of steganography technology with the image to be protected being the cover image and the watermark being the message to be embedded. Compared to existing image based steganography techniques, our technique has two advantages: first, it can directly hide an arbitrary image, such as the image of a hand written message or photograph in its original form, while the majority of the existing steganography technologies need to encode a secret message as bit strings or impose some restraint on it to embed an image directly. For example, a classic technique to embed an image into another one is to substitute the lower significant bits of the cover image with the secret image [9] [10], which may largely suppress the dynamic range of both images. Second, our technique uses a multi-layered encryption scheme which makes it more difficult to detect the embedded message. The secret message is first hidden by invisibly in-wrought into a marbling pattern. Then the parameters of marbling operations are encoded into an encrypted bit string and embedded into the marbling pattern to create a stego-marbling-texture using a steganography technique. We use RSA algorithm [11] to encrypt the parameters and extended FPcode [12] for embedding the marbling parameters. Typical steganography algorithms, such as those based on LSB [9] [10], usually cannot resist from the image degradation caused by printing or capturing. Since we want to apply the stego-marbling-texture for decorating real products found in our daily life in the future, such defection has to be taken into consideration. FPcode uses the relative average intensity of adjacent blocks in the cover image to embed a bit string, therefore it is less sensitive to the change of illumination conditions and the degradation caused by printing.

Information embedding is an attention attracting topic also for computer graphics community. Several works have been published on creating artistic work with hidden information. Yoon et al. [13] presented a hidden-picture puzzle which contains hidden objects in a line drawing background image. Mitra et al. [14] proposed a technique to generate images consisting of 3D objects which are difficult to be detected by computer vision techniques but can be recognized by humans. Papas et al. [15] designed real-world magic lenses to warp source images into several hidden images viewed from prescribed orientations. Chu et al. [16] created camouflage images by replacing the texture details of hidden figures with the surrounding background and made sure an appropriate amount of visual clues are remained. Tong et al. [17] proposed a method to generate hidden images by first detecting edge features of a hidden object, and then finding a suitable embedding place in the background scene using an energy optimizing process. Du et al. [18] utilized the contour completion property of human perception to generate digital camouflage images. All the above mentioned techniques aim to hide objects in background images by fooling the human recognition process. Although the hidden objects are less noticeable, they are still recognizable by any observers. In case of stego-marbling-texture, however, we can make the hidden message retrievable only to authenticated receivers when necessary.

III. System overview

The proposed technique consists of 2 components: a real-time marbling design system for creating a stego-marbling-texture from a given message, and a decrypter which can be distributed separately for the retrieval of the hidden message. Figure 2 and Figure 3 depict the overall procedure for the stego-marbling-texture generation and hidden message retrieval, respectively. As shown in Figure 2, to create a stego-marbling-texture, first a background harmonious with the input message in color and texture is automatically created. Then, the user is allowed to perform the marbling operations interactively until a desired pattern has been obtained. The operations together with their parameters are automatically recorded and encoded into a bit string. Finally, the bit string is embedded into the marbling pattern to create the final stego-marbling-texture. To retrieve the hidden message, as shown in Figure 3, the embedded bit string is first extracted from the stego-marbling-texture and decoded to obtain the parameters of marbling operations. Then the reverse of those operations are applied to the stego-marbling-texture to reveal the hidden message.

We introduce perceptual measures based on color, curviness, orientation, and frequency to guide the generation of a background with which the message becomes less noticeable in the resulting stego-marbling-texture. Although as the default, the system can automatically create such a background, the user is also allowed to change the background as he/she likes. In addition to a text message, the input message can also be a full size image, such as a memorial photograph. In such a case, marbling operations can be applied to the photograph directly without using an additional background.

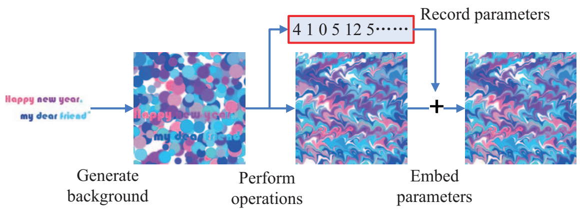


Figure 2: Stego-marbling-texture generating steps.

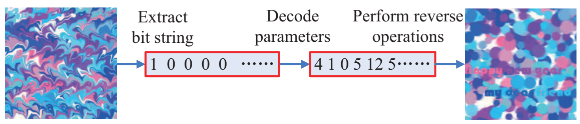


Figure 3: Hidden message retrieval steps.

Figure 4: Embed data with a random chosen background.

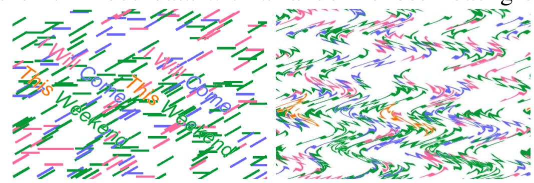


Figure 5: Embed data using our method.

The details about the definition of reversible marbling operations, the generation of background, and the encryption and decryption algorithms will be given in Sect. IV, Sect. V, and Sect. VI, respectively.

IV. Reversible marbling operations

A. Generating background

A stego-marbling-texture should satisfy the following two requirements:

- It should be as beautiful as conventional marbling textures.
- The hidden message should be as unnoticeable as possible.

The first step of marbling texture design is to paint the background with dye colors. In the traditional way, droplets and stripes are usually used. To create a stego-marbling-texture satisfying the above requirements, special considerations are required for the background. For example, if we simply use a randomly chosen stripe pattern as the background, the hidden message becomes noticeable as an artifact which ruins the marbling pattern itself (Figure 4). Figure 5 shows a result using a background automatically created with the proposed technique, where the message has been artfully hidden into the marbling pattern. Comparing Figure 4 to Figure 5, we can find that a key point to create a successful stego-marbling-texture is to make the message harmonious to the background. Based on such consideration, we propose an automatic background generating method based on several perceptual measures of visual cues. Currently, four visual cues are considered: color, curviness, orientation, and element size.

More specifically, dye colors are determined by the main colors from the input message. The type and size of background elements are determined by the strokes detected in the input message. Then, droplets are randomly scattered on the background. The background generation process is finished when the overall percentage of colorful pixels is roughly the same as that in the input message. The diagram of design process is shown in Figure 6.

1) Determining the dominant colors: To achieve the harmony in color, we detect i most dominant colors in the input message and use them for designing the background. We construct a color histogram by recording the per-pixel colors of the input message and select i tallest bins. i is the smaller one of p and c, where p is a user given constant and c is the total number of colors in the input message.

2) Measuring the curviness: We employ Hough line detector to measure the curviness of visual elements in the input message. We compute the percentage of straight lines in the detected lines. If straight lines are in the majority, elements like sticks and stripes are used for background. Otherwise, elements like cycles, rings, ellipses, and arcs are chosen.

3) Measuring the orientation: We use the following Gabor filter to measure the orientation.

$$G(u,v) = e^{\frac{u^2+v^2}{2\delta^2}} \cos(2\pi \frac{u}{2\lambda} + \frac{\pi}{2}), \tag{1}$$

$$\begin{aligned} u &= x\cos(\theta) + y\sin(\theta), \\ v &= x\sin(-\theta) + y\cos(\theta). \end{aligned} \tag{2}$$

Here λ and θ are the parameters for controlling the frequency and orientation, respectively. We compute the power of the filter response for k different frequencies and n orientations (k and n are given by the user). For each frequency, the orientation with the largest power is selected. As a result, we have k dominant orientations. When generating the background, dye elements are rotated to these orientations.

4) Measuring the element size: We adopt two different methods to determine the size of droplets. When using stick, arc, and ring elements, we use the Gabor filter to measure the element size in the input message. We test k frequencies and n evenly spaced orientations in the same way as measuring the orientation. For each frequency, we add power of all directions together. The inverse of the frequency with maximum power should be close to half of the dominant element size. Therefore, we adopt λ of Equation 1 as the expected average size of elements for the background. When round shaped elements should be

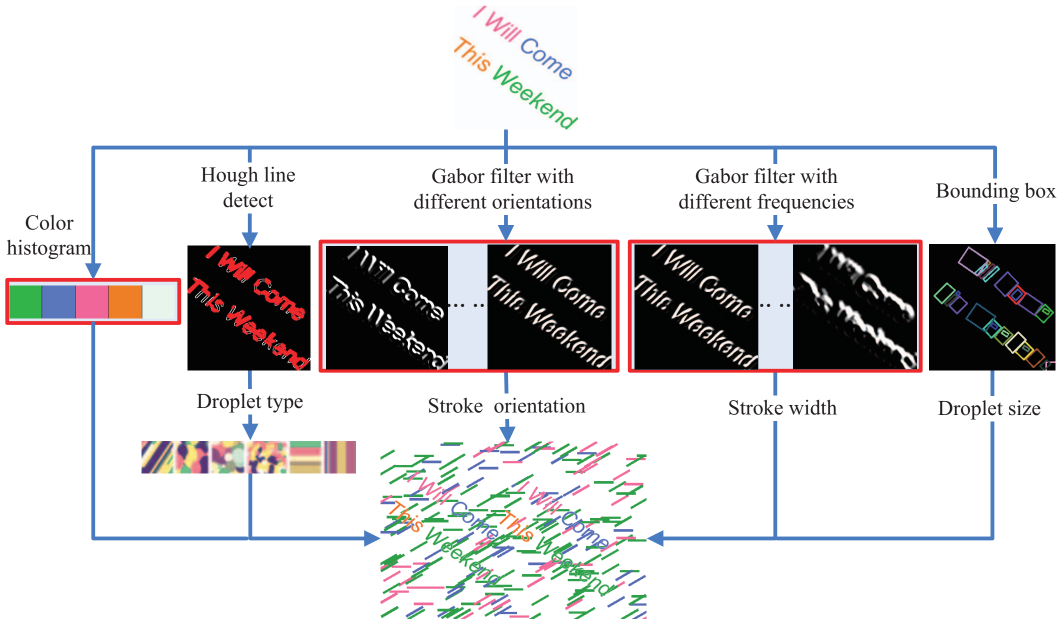


Figure 6: Diagram of perception measure based background generation algorithm.

(a) Tine-line pattern (b) Sinusoidal pattern (c) Tangent pattern (d) Skew pattern (e) Circular pattern

Figure 7: Basic operations in our current implementation.

used, we compute the bounding box of each element in the message and use them to define the size of droplets.

B. Reversible marbling operations

We refine the definitions of mathematical formulations used by Lu [2], and ensure tine-line, sinusoidal wavy, and circular marbling patterns to be reversible. In addition, we propose new skew, and tangent-like wavy patterns. Users may compose several manipulation steps to create a complex and beautiful marbling result of their own style. The effects of currently provided basic operations are demonstrated in Figure 7.

1) Tine-line pattern: This function mimics the effect a comb running straight across the image in x or y direction. According to [2]'s definition (Figure 8(a)), point $P(x,y)$ is mapped to point $P'(x',y')$ according to equation:

$$P' = P + \frac{\alpha\lambda}{d+\lambda}M, \tag{3}$$

where $d = |P - (A + fmod(d/s)s)N|$ calculates the distance between point P and L ($fmod$ is the modulo function), and s defines the space between two tines.

We define a new reverse function as follows: if $M_y = 0$, Equation 4 is used; otherwise, Equation 5 is used.

$$\begin{cases} x = x' - \frac{M_x}{M_y} = x' \\ y = y' - \frac{\alpha\lambda M_y}{|x'N_x - N_x\frac{M_x}{M_y}y' - A_xN_x - A_yN_y| + \lambda} \end{cases}, \tag{4}$$

$$\begin{cases} x = x' - \frac{\alpha\lambda M_x}{|y'N_y - N_y\frac{M_y}{M_x}x' - A_xN_x - A_yN_y| + \lambda} \\ y = y' - \frac{M_y}{M_x} = y' \end{cases}. \tag{5}$$

2) Sinusoidal wavy pattern: In order to get a reversible mapping, this pattern is restricted to be sinusoidal shaped wavy in x or y direction. Readers may refer [2] to get implementation details.

3) Tangent wavy pattern: This function transforms part of the image into tangent curves. Only deformation in x or y direction is allowed. In case of deformation in x direction, x coordinate is mapped to x' while keeping y coordinate unchanged:

$$x' = x + min(\beta\tan(\alpha y), t), \tag{6}$$

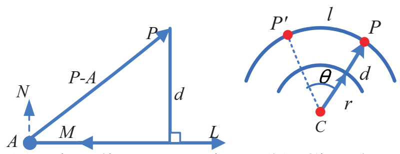


Figure 8: (a) Tine-line operation. (b) Circular operation.

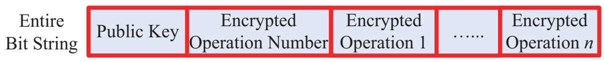


Figure 9: Structure of encrypted bit string.

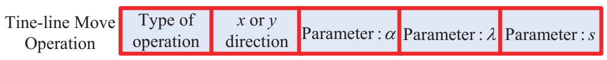


Figure 10: Code structure for tine-line operations.

where α is the wavelength, and the maximum amount of displacement is clamped to t.

4) Skew pattern: Combining this pattern with the sinusoidal wavy pattern or the tangent wavy pattern can realize wavy curves at any direction. In case of deformation in x direction, x coordinate is mapped to x' while keeping y coordinate unchanged:

$$x' = x + \alpha y \tag{7}$$

where α controls the amount of movement.

5) Circular pattern: We modify circular pattern proposed by [2] by introducing γ and new definition of d. To mimic concentric cycles (Figure 8(b)), point P is mapped to P' according to equation:

$$P' = C + (P - C)\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \tag{8}$$

$$\begin{aligned} \theta &= \tfrac{l}{|P-C|^{\gamma}}, \\ d &= |fmod(|P-C|/s) - r|, \end{aligned} \tag{9}$$

where γ controls the attenuation, and s defines the space between two tines.
By replacing $-\theta$ with θ, we get the backward transformation of point P from point P'.

V. Embedding and Extraction Algorithms

When creating a marbling pattern, the parameters of the operations applied are automatically recorded and encoded into a bit string. Then, a steganography technique is employed to embed the bit string into the marbling pattern to create a stego-marbling-texture. To retrieve the hidden message, the bit string is first extracted from the stego-marbling-texture and decoded to get the operation parameters with which inverse operations can be performed to reveal the hidden message. In this session, we describe the details on the encoding of operations, embedding of bit strings and the extraction of embedded bit strings.

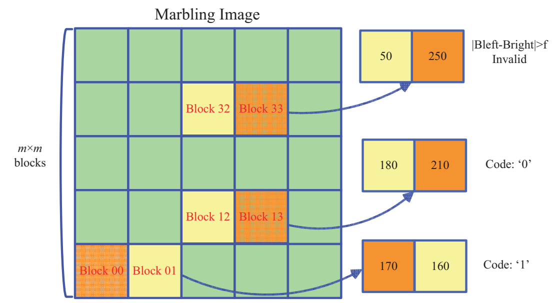


Figure 11: Embedding algorithm.

(a) Original image (b) FPcode method (c) Our method

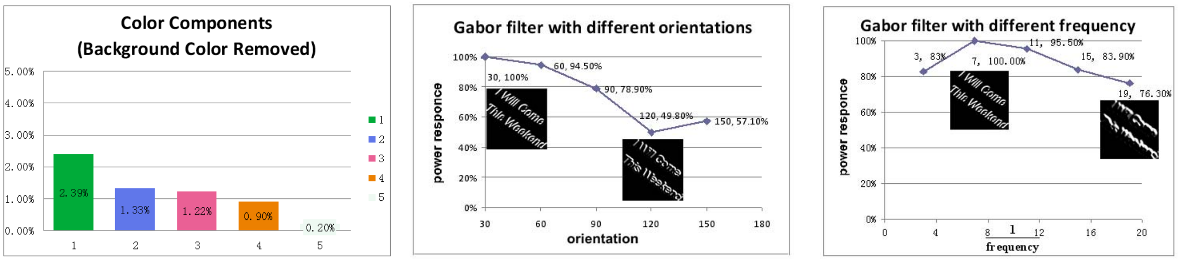
Figure 12: Comparison of visual quality between existing method and ours.

A. Encoding of operations

The encoding step is realized using RSA algorithm [11]. As shown in Figure 9, information to be embedded into marbling textures is encoded as a bit string composed of 2 parts: the public key, and the ciphertext. The ciphertext is composed of marbling operations that encrypted by the public key. More specifically, each parameter is encrypted according to public key and then concatenated to form the ciphertext. Since each operation has different parameters, a non-fixed length coding structure is used to keep the code as compact as possible. The coding of marbling operations starts with the number of operations, followed by the parameters of each operation. To balance the security and capacity consumed, 32 bits are used to represent the public key in current implementation. In Figure 10, we show the structure of the tine-line operation.

B. Embedding of bit strings

To embed the bit string into a marbling pattern, we employ FPcode proposed by Genta et al. [12]. FPcode aims to provide an easy link between offline materials and online information by embedding the URL information into the offline materials. A user can access the online information simply by capturing the offline materials with a decrypter installed cellphone. Represented by QR-code, these kinds of technology, called media bridge [19], have been attracting more and more attentions recently with the popularization of the mobile-device-based Internet access. While the main concern of conventional steganography techniques is how to make the algorithm robust against various attacks, a main technical issue of media bridge techniques is how to raise the success rate of decoding under various capturing environments. Recently more media bridge technologies [20]



(a) Color histogram. (b) Gabor filter with different orientations. (c) Gabor filter with different frequencies.

Figure 13: Feature detection of text message.

based on steganography or watermarking algorithms have been developed. We adopt Genta et al.'s technique for taking its advantage of having a relatively large embedding capacity and being robust against color distortion caused by scanning, printing or the illumination conditions of capturing.

To ensure a high success rate of decoding from a printed media, Genta et al.'s technique embeds one bit using the relative average intensity of two adjacent blocks. As shown in Figure 11, the original image is divided into $m \times m$ blocks. The pixels of two adjacent blocks are modified so that the left one has a lower average value than the right one for embedding $'0'$, and higher than the right one for embedding $'1'$. They suggest using the yellow channel of the image for the embedding since human eyes are less sensitive to the change in the yellow channel. We found, however, a naive implementation of Genta et al.'s technique can introduce noticeable artifacts in the resulting stego-marbling-texture. An example of such artifacts is indicated in Figure 12(b). It is caused by over changing the relative average value between two blocks across edges. We solve this problem by limiting the amount of color change from the original image and skipping those block pairs which exceed the limit. Such block pairs can be identified by computing the difference of average values between the two blocks.

We actually compare the sum of yellow values between the block pairs. Constants m and f are user-controllable parameters. Increasing m can enlarge the embedding capacity, but small blocks would need larger changes for individual pixels to produce a difference of sum between block pairs, which is significant enough for embedding a bit. f is also a parameter which trades image quality with embedding capacity. A small f identifies more block pairs as not suitable for embedding. The best values of m and f should be application dependent.

C. *Retrieval of hidden message*

Taking Figure 11 as an example, the extraction of the bit string from the stego-marbling-texture can be realized with the following 3 steps:

1) Extract the yellow channel from the stego-marbling-texture and divide it into $m * m$ blocks.
2) Compare the sum of yellow values between the adjacent block pairs (such as Block00 and Block01, Block12 and Block13).
3) If the difference is smaller than f, then
   a) if the left block is smaller than the right block (such as Block12 and Block13), we get $'0'$;
   b) if the left block is larger than the right block (such as Block00 and Block01), we get $'1'$.

From the first 32 bits of extracted bit string, we get the public key. Using both the transmitted public key and one's own private key in hand, we decrypt the type of operations and their parameters. Finally, the reverse of those operations are applied in a reverse order to reveal the hidden message.

VI. Results and Discussion

We have implemented our stego-marbling-texture design system and decrypter on a computer with 2.93GHz Intel Core 2 Due CPU and NVIDIA Geforce GTS 250 GPU. The background generation algorithm is implemented using OpenCV. The remaining part of our system is implemented using GLSL shaders on GPU.

In Figure 13, we present a detailed explanation of the feature detection process by taking Figure 4 as an example. Firstly, we analyze the main colors components (Figure 13(a)). Then, we find 82% of the detected stroke is straight using the Hough line detector, so sticks are used as the painting element. To decide the width and orientation of sticks, we use a Gabor filter with 6 orientations and 5 different frequencies. In Figure 13(b), we fixed the orientation, and tested 5 different frequencies. The response reaches strongest when λ is 3. So we define the width of sticks to be $3 \pm \sigma$, where σ is a randomly defined small scalar. In Figure 13(c), we fixed λ as 3, and tested 6 orientations. The the main orientation is 30 degrees. Figure 1(a) shows an e-card of stego-marbling-texture style. In Figure 1(b), stego-marbling-texture is used to decorate a virtual cup in 3D rendered scene.

VII. Conclusions and Future Work

We have proposed a novel technique for creating marbling textures with the additional value of being able to convey private messages. Users may enjoy the fun of interactive marbling texture creation. The resulting stego-marbling-texture can be used for sharing private messages among authenticated friends. Under the beautiful marbling pattern, large capacity of personalized messages can be hidden and recovered in a fascinating way.

In our current version, hidden messages can only be retrieved when the stego-marbling-texture is delivered as a digital image. For the decoding from a printed stego-marbling-texture, the captured images need to be pre-processed for correcting the geometric distortions. Recently, many rotation and scaling resistant techniques have been developed for embedding information in printed materials without using visible registration markers [21] [22]. We are now working on incorporating these technologies into stegano-marbling-texture.

ACKNOWLEDGMENT

This research was supported by the Natural Science Foundation of Zhejiang Province, China (No. Q12F020007), the Zhejiang Provincial Education Department, China (No. Y201121352), the Major Science and Technology Innovation Team of Zhejiang Province, China (No. 2010R50040), the National Key Technology Research and Development Program of the Ministry of Science and Technology of China (No. 2012BAH91F03), the National Natural Science Foundation of China (No. 61370218).

The authors like to thank Mr. Dongliang Chen for his help in creating Figure 1.

REFERENCES

[1] X. Mao, T. Suzuki, A. Imamiya, Atelierm — a physically based interactive system for creating traditional marbling textures, in: Proc. of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, 2003, pp. 79–86.

[2] S. Lu, A. Jaffer, X. Jin, H. Zhao, X. Mao, Mathematical marbling, IEEE Computer Graphics and Applications 32 (6) (2012) 26–35.

[3] B. T. Akgun, The digital art of marbled paper, Leonardo 37 (1) (2006) 49–52.

[4] R. Acar, P. Boulanger, Digital marbling: A multiscale fluid model, IEEE Transactions on Visualization and Computer Graphics 12 (4) (2006) 600–614.

[5] X. Jin, S. Chen, M. Xiaoyang, Computer-generated marbling textures: A gpu-based design system, IEEE Computer Graphics and Applications 27 (2) (2007) 78–84.

[6] J. Xu, X. Mao, X. Jin, Nondissipative marbling, IEEE Computer Graphics and Applications 28 (2) (2008) 35–43.

[7] H. Zhao, X. Jin, S. Lu, X. Mao, S. Jianbin, Atelierm++: A fast and accurate marbling system, Multimedia Tools and Applications 44 (2) (2009) 187–203.

[8] B. Li, J. He, J. Huang, Y. Shi, A survey on image steganography and steganalysis, Journal of Information Hiding and Multimedia Signal Processing 2 (2) (2011) 142–172.

[9] V. K. Sharma, V. Shrivastava, A steganography algorithm for hiding image in image by improved lsb substitution by minimize detection, Journal of Theoretical and Applied Information Technology 36 (1) (2012) 1–8.

[10] D. E. Walia, P. Jain, N. Navdeep, An analysis of lsb and dct based steganography, Global Journal of Computer Science and Technology 10 (1) (2010) 4–8.

[11] R. L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM 21 (2) (1978) 120–126.

[12] G. Suzuki, N. Yamaguchi, S. Nakamura, H. Chiba, Mobile interaction using steganographic image on mobile display, in: Proc. of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services, ACM, 2008, pp. 507–510.

[13] J.-C. Yoon, I.-K. Lee, H. Kang, A hidden-picture puzzles generator, Comput. Graph. Forum 27 (7) (2008) 1869–1877.

[14] N. J. Mitra, H.-K. Chu, T.-Y. Lee, L. Wolf, H. Yeshurun, D. Cohen-Or, Emerging images, ACM Trans. Graph. 28 (5) (2009) 1–8.

[15] M. Papas, T. Houit, D. Nowrouzezahrai, M. Gross, W. Jarosz, The magic lens: Refractive steganography, ACM Trans. Graph. 31 (6) (2012) 186.

[16] H.-K. Chu, W.-H. Hsu, N. J. Mitra, D. Cohen-Or, T.-T. Wong, T.-Y. Lee, Camouflage images, ACM Trans. Graph. 29 (4) (2010) 1–8.

[17] Q. Tong, S.-H. Zhang, S.-M. Hu, R. R. Martin, Hidden images, in: Proc. of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering, ACM, 2011, pp. 27–34.

[18] H. Du, X. Jin, X. Mao, Digital camouflage images using two-scale decomposition, Comput. Graph. Forum 31 (7) (2012) 2203–2212.

[19] A. M. Alattar, Bridging printed media and the internet via digimarc's watermarking technology, in: Proc. of the ACM workshop on multimedia and security, ACM, 2000.

[20] A. Mohan, G. Woo, S. Hiura, Q. Smithwick, R. Raskar, Bokode: Imperceptible visual tags for camera based tnteraction from a distance, ACM Trans. Graph. 28 (3) (2009) 98:1–8.

[21] T. Shono, M. Muneyasu, Y. Hanada, Implementation of data embedding to printing images for information retrieving by cellular phones considering lens distortion, in: International Symposium on Intelligent Signal Processing and Communication Systems, IEEE, 2009, pp. 558–561.

[22] H. Kudo, K. Furuta, M. Muneyasu, Y. Hanada, Automatic information retrieval from data embedded printing images using correction of rotational angles based on reference marks, in: International Symposium on Communications and Information Technologies, IEEE, 2010, pp. 626–629.