

## A Coherent Noise Control Using Templates

Heekyung Yang<sup>1</sup> and Kyungha Min<sup>2\*</sup>

<sup>1</sup>*Dept. of Computer Science, Graduate School, Sangmyung Univ., Seoul, Korea*

<sup>2</sup>*Dept. of Media Software, Sangmyung Univ., Seoul, Korea*

*minkh@smu.ac.kr*

### **Abstract**

*We present a noise control algorithm that produces temporally coherent artistic styles on movie files. Our algorithm is devised using a template to preserve the temporal coherence of the artistic effects produced on the frames composing a video. In the first step, we produce a set of triangular meshes embedding the contents of a frame. We, then, generate noise in the coordinate system defined by the templates. We finally apply a texture coverage scheme to preserve the coherence of the noise distribution inside the templates. For the noise outside the templates, we propose local recursive filters that interpolate the noise between frames.*

**Keywords:** *noise control, non-photorealistic rendering, artistic style, video, template tracking*

### **1. Introduction**

The progress of photography including digital camera and smart phone increases the amount of contents we produce and share in an exploding scale. This trend raises the needs for producing various customized styles on the contents and, consequently, those techniques that allow users to produce or apply various effects of their own style are available on diverse platform including computers, internet and mobile devices.

In the diverse styles customized to users' aesthetic viewpoint, those styles following artistic techniques are pursued by many researchers. Till now, many studies have been proposed the schemes that apply various artistic effects on images. Recently, some researchers try to apply the artistic effects on video. The major obstacle in applying the artistic effects on video is to avoid some artifact such as flickering, show door effect. In other words, preserving temporal coherence of the effects on the frames composing video is a key challenge in developing an algorithm that produces artistic effects on video.

To our analysis, noise is one of the most widely used background components in producing artistic effects. Noise itself plays a role of stipple, which is an interesting style pursued by many researchers. Pencil and pastel drawing effects are other styles produced by generating and integrating noise. In producing artistic effects on video, the intensity consistency as well as the temporal coherence is a key challenge in expressing visually pleasing styles.

We present a scheme that preserves the temporal coherence and intensity consistency in producing artistic effects using noise on video. Our key approach is to use templates. Template presents coherent and consistent shape information among the frames of a video.

---

\* Corresponding Author

The motion of the objects in the scene is segmented and tracked using user-initiated templates. We present a scheme that preserves the temporal coherence of the styles using dart throwing and texture coverage algorithms. The intensity consistency is resolved using local recursive filter scheme.

This paper is organized as follows. We survey related works in Section 2 and explain the proposed algorithm in Section 3, respectively. In Section 4, we present our results. Finally, we draw conclusions and propose future work in Section 5.

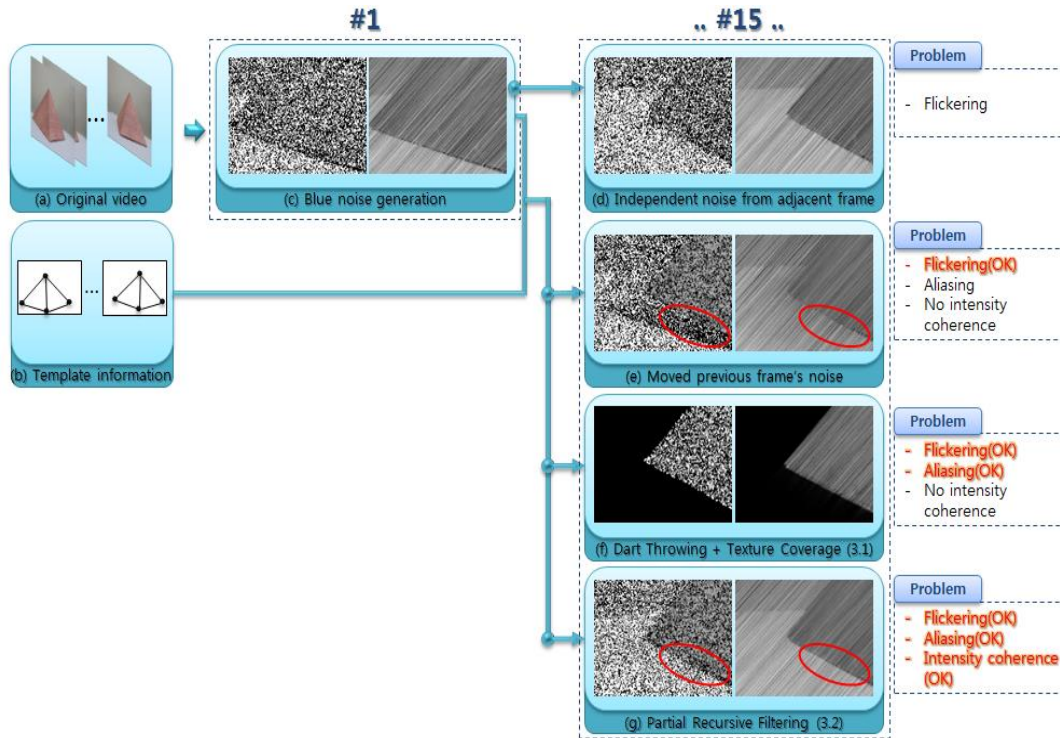


Figure 1. The Overview of our Algorithm

## 2. Related Work

Unfortunately, the studies that attack the intensity consistency and temporal coherence simultaneously are not frequently noticed. In this Section, we briefly survey the conventional noise generation schemes and noise generation schemes that consider the intensity of an image. We also survey the optical flow-based schemes and 3D model-based schemes for preserving temporal coherence. Finally, we summary other existing schemes that produce artistic effects on video.

### 2.1 Procedural Noise

The early works on producing noise based on a procedure concentrate on creating visually convincing noise pattern such as marble, granite or sand. Perlin [1] and Peach *et al.* [2] pioneered the procedural noise generation schemes and Cook and Lagae [3, 4] improved the schemes by controlling their spatial properties. Their purpose is to develop a scheme that mimics the textures on natural objects. Therefore, their schemes do not include the solutions for temporal coherence and intensity consistency.

## 2.2 Blue Noise

Blue noise generation is one of the most frequently schemes used in non-photorealistic rendering (NPR) field for producing artistic effects. The position, density, magnitude of blue noise depends on the properties of an image such as intensity, color and features. The blue noise itself is recognized as a kind of representation scheme used in NPR. Furthermore, several schemes [Yang and Min] use the blue noise as a preprocessing step for producing pencil effects.

In early days, researchers [5, 6, 7] have presented a scheme that controls the density of noise and the size of noise according to the intensity of image. Kopf *et al.* [8] proposed a scheme that maintains the density of noise when an image is zoomed in and out. Kim *et al.* [9] and Li *et al.* [10] arranged the distribution of noise according to the intensity of image. Yang *et al.* [11, 12, 13, 14] generated blue noise according to the color and intensity of image and expressed the progress of details of artistic effects. These schemes, however, have been devised to produce noise on image. They have limitations in producing noise in video with temporal coherence.

## 2.3 Optical Flow

Optical flow is used to extract the motion information embedded in the frames. The motion information is used to preserve the temporal coherence of artistic effects in video. The brush-based artistic effects [15, 16, 17] generate noise to correspond the positions of strokes using optical flow. However, the effects from [15, 16] still possess flickering of the effects and those from [17] suffer from blurring instead of flickering. These artifacts are due to the unstable optical flow extraction.

Neyet *et al.* [18] developed a texture advection scheme that expresses the temporal flow using texture instead of noise. This scheme, however, suffers from the blurring on the area where the texture is enlarged and the compression on the area where the texture is reduced. Bousseau *et al.* [19] extended the texture advection scheme to flow the texture according to the motion of object and to control the noise following the texture. Their target effect is water-coloring. They achieved stable texture control by applying the advection of texture in both forward and reverse directions. Even though they produce very temporally coherent effects, they have limitations in applying real-time processing. Note that the advection of texture in both directions requires the information of frames in the future.

### 2.4.3 D Model

We can compute the correct motions of 3D models. Benard *et al.* [20] generated Gabor noise on 3D models to produce artistic effects. Kass *et al.* [21] used the depth and velocity information on 3D animation to preserve the temporal coherence of noise. Since they generate white noise instead of blue noise, it can be applied in producing artistic effects on video.

## 2.5 Other Works

Yoon *et al.* [22] presented 3D Sobel operation to compute spatially and temporally coherent flow in video. The flow is applied to a texture whose initial shape is from white noise in order to mimic painterly rendering effects. Their scheme produces a successful result for temporal coherence. However, it has a limitation in mimicking the painterly rendering effect. Sundquist [23] visualizes the magnetic flow by applying line integral convolution (LIC) scheme for noise. He reduced the aliasing artifact when the noise in real space is embedded in pixel space.

### 3. Coherent Noise Generation

We describe problems that occur when we generate noise on the frames of video and suggest the solutions. Figure 1 illustrates the overview of our system. The input to our system is composed of frames captured from an input video (see Figure 1(a)) and the templates embedded on the frame (see Figure 1(b)). Using the template, we specify the positions and motions of the objects in the frame. Additionally, we can distinguish the inside and the outside of an object in the frame using templates. We generate blue noise in the first frame using the scheme proposed in [13] (see Figure 1(c)). We consider temporal and spatial coherence in generating noise from the second frame to the last frame of a video. We illustrate our scheme in the fifteenth frame from a video, which is composed of thirty-nine frames. Figure 1(d) ~ (g) illustrate our approaches.

Approach 1: This approach generates noise on a frame without considering the noise in the previous frame. The noise generated by the approach 1 and the resulting artistic effects seriously suffer from flickering artifacts (see Figure 1(d)).

Approach2: This approach translates the noise in the previous frame according to the motions between the frames (see Figure 1(e)). These motions are computed using a barycentric averaging on the motions of the vertices of a template. This scheme removes the flickering artifacts that can be produced by the approach 1. This scheme, however, produces aliasing artifact that comes from the mismatch between the real-valued motion vector and integer-valued noise pace. Another artifact is the inconsistent intensity.

Approach3: To resolve the temporal coherence problem, we present a novel scheme constructed by combining dart throwing and texture coverage scheme. We explain details of this approach in Section 3.1.

Approach4: The approach 3 can resolve the temporal coherence problem, but it still suffers from inconsistent intensity. We propose a local recursive filter scheme to present intensity consistency, which is described in Section 3.2.

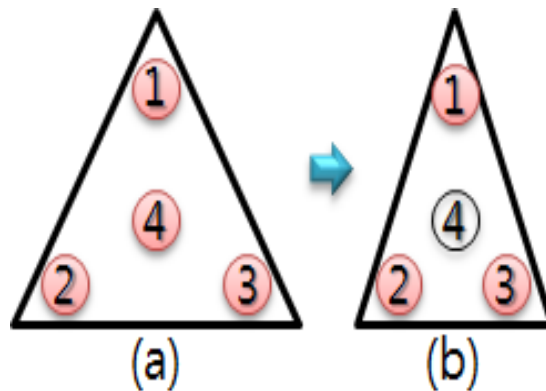


Figure 2. Modified Dart Throwing Algorithm

#### 3.1 Solution to Temporal Coherence

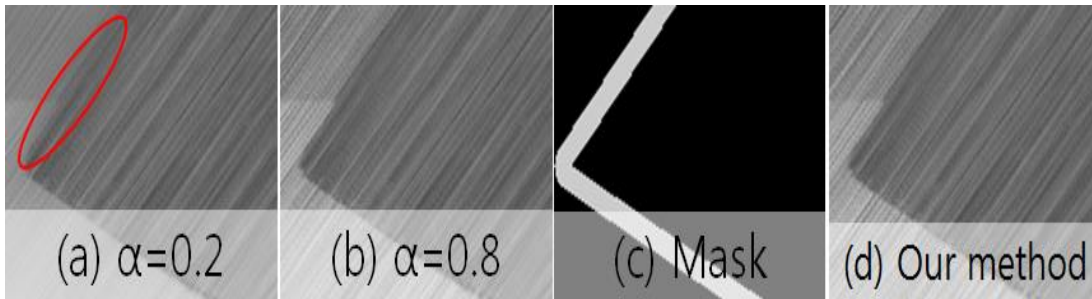
The aliasing from breaking temporal coherence is illustrated in the red circle in Figure 1(e).

To produce evenly distributed noise over frames we use dart throwing algorithm [24], which controls the number of noise per area. Our scheme works as follows. In the first frame, we generate noise using dart throwing algorithm and store the sequence of generated noise (see Figure 2(a)). The number of generated noise is proportional to the number of the pixels in the frame. The noise per pixel is designated to one. For this constraint, we deactivate the noise when more than two noise is generated on a pixel. The density of a noise is computed as follows:

$$D_N = \frac{N_N}{N_P},$$

where  $D_N$  denotes the number of noise per pixel (density),  $N_N$ , the total number of noise and  $N_P$ , the number of pixels inside a template.

In the next frame, the generated noise is translated according to the motion of the template. The noise is activated or deactivated according to the change of the area of the template (see Figure 2(b)). Even though this scheme distributes noise evenly over the frame, the aliasing artifact that comes from the mismatch of real-valued motion and integer-valued noise still remains. To resolve this artifact, we use texture coverage scheme proposed in [23] to map the noise in the pixel space (see Figure 1(f)).



**Figure 3. Local Recursive Filtering**

### 3.2 Solution to Consistent Intensity

As illustrated in Figure 1(f), the above scheme preserves temporal coherence by removing flickering and aliasing artifacts. This scheme, however, still has limitation in resolving intensity consistency as illustrated in the red circle in Figure 1(g). This artifact is observed when the boundary of a template is overlapped with the shadow of objects. As illustrated in Figure 1, motions of an object may produce or remove the shadow of the object. In this case, the noise with black value generated on the shadow may remain outside of the template. This black noise reduces the intensity consistency.

We propose a local recursive filter on the boundary of the templates. The conventional recursive filter [25] is formulated by the following formula:

$$T(t) = (1 - \alpha)T(t - 1) + \alpha T(t)$$

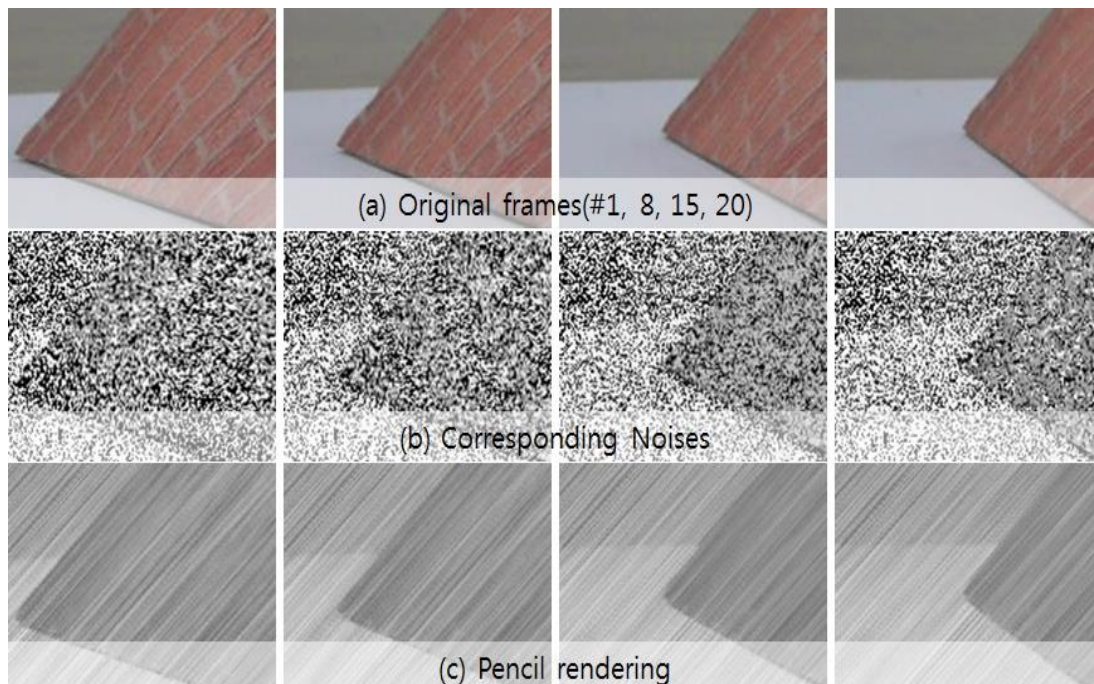
$T(t)$  denotes a new noise texture generated at time  $t$ , and  $T(t-1)$  is the noise texture of the previous frame.  $\alpha$  is a control parameter where larger value of  $\alpha$  increases the effects of noise generated at the current frame. When we set  $\alpha$  as 0.2 as illustrated in Figure 3(a), the intensity in the shadow loses the consistency. In Figure 3(b) illustrates the effects when we set  $\alpha$  as 0.8. It preserves the intensity consistency, but suffers from flickering artifact. In our approach, we

set  $\alpha$  as 0.8 for the boundary of templates and  $\alpha$  as 0.2 for the other area. This approach resolves the flickering artifact and intensity consistency simultaneously.

## 4. Results

We applied our scheme for a video with pyramids, which is composed of twenty frames. In Figure 4, we sample four frames (1st, 8th, 15th and 20th) and produced two images: noise image and pencil drawing effects from the noise.

Even though the noise shows some blurring due to texture coverage computation, the pencil drawing effects from the noise shows convincing effects with temporal coherence and intensity consistency.



**Figure 4. Result of our Method. (a) Original Frames, (b) the Corresponding Noises, (c) the Corresponding Pencil Rendering.**

## 5. Conclusion and Future Work

We have presented a frame work that controls noise distribution for temporal coherency and intensity consistency of the distribution. For temporal coherence, we track the motion of the objects inside the frame to remove flickering artifact and propose a scheme based on dart throwing and texture coverage for resolving aliasing. This scheme is extended for preserving intensity consistency.

This scheme can be further applied to a stereoscopic image which is composed of two images. We can further extend our scheme by automatic template matching scheme on video.

## Acknowledgements

This research was supported by a 2013 Research Grant from Sangmyung University.

## References

- [1] K. Perlin, "An image synthesizer", In Proceedings of ACM SIGGRAPH, (1985), pp. 287–296.
- [2] D. Peachey, "Solid texturing of complex surfaces", In Proceedings of ACM SIGGRAPH, (1985), pp. 279–286.
- [3] R. Cook and T. DeRose, "Wavelet noise", ACM Transactions on Graphics, vol. 24, no. 3, (2005), pp. 803–811.
- [4] A. Lagae, S. Lefebvre, G. Drettakis and P. Dutre, "Procedural noise using sparse Gabor convolution", In Proceedings of ACM SIGGRAPH, (2009), pp. 1–10.
- [5] O. Deussen, and S. Hiller, K. Van Overveld and T. Strothotte, "Floating points: A method for computing stipple drawings", Computer Graphics Forum, vol. 19, no. 3, (2000), pp. 40–51.
- [6] A. Secord, "Weighted voronoi stippling", In Proceedings of Non-Photorealistic Animation and Rendering, (2002), pp. 37–43.
- [7] A. Lagae and P. Dutre, "A procedural object distribution function", ACM Transactions on Graphics, vol. 24, no. 4, (2005), pp. 1442–1461.
- [8] J. Kopf, D. Cohen-Or, O. Deussen and D. Lischinski, "Recursive Wang tiles for real-time blue noise", In Proceedings of ACM SIGGRAPH, (2006), pp. 509-518.
- [9] D. Kim, M. Son, Y. Lee, H. Kang and S. Lee, "Feature-guided image stippling", Computer Graphics Forum, vol. 27, no. 4, (2008), pp. 1209-1216.
- [10] H. Li, L. Wei, P. Sander and C. Fu, "Anisotropic Blue Noise Sampling", In Proceedings of ACM SIGGRAPH Asia, article no. 167, (2010).
- [11] H. Yang and K. Min, "A Realistic Pencil Rendering Technique for Game Characters", Journal of Korean Society for Computer Game, vol. 2, no. 23, (2010), pp. 49-59.
- [12] H. Yang and K. Min, "An Accelerated Color Pencil Rendering Technique for Game", Journal of Korean Society for Computer Game, vol. 24, no. 4, (2011), pp. 127-136.
- [13] H. Yang and K. Min, "Color Pencil Drawing from Photographs", Journal of Korean Society for Computer Game, vol. 25, no. 3, (2012), pp. 55-64.
- [14] H. Yang, Y. Kwon and K. Min, "A Stylized Approach for Pencil Drawing from Photographs", Computer Graphics Forum, vol. 31, no. 4, (2012), pp. 1471-1480.
- [15] P. Litwinowicz, "Processing Images and Video for an Impressionist Effect", In Proceedings of ACM SIGGRAPH, (1997), pp. 407-414.
- [16] A. Hertzmann and K. Perlin, "Painterly rendering for video and interaction", In Proc. NPAR, (2000), pp. 7-12.
- [17] J. Hays and I. Essa, "Image and Video-Based Painterly Animation", In Proceedings of Non-Photorealistic Animation and Rendering, (2004), pp. 113-120.
- [18] F. Neyret, "Advection textures", In Proceedings of ACM-SIGGRAPH/EG Symposium on Computer Animation, (2003), pp. 147-153.
- [19] A. Bousseau, F. Neyret, J. Thollot and D. Salesin, "Video Watercolorization using Bidirectional Texture Advection", ACM Transactions on Graphics, vol. 26, no. 3, no. 104, (2007).
- [20] P. Benard, A. Lagae, P. Vangorp, S. Lefebvre, G. Drettakis and J. Thollot, "A dynamic noise primitive for coherent stylization", Computer Graphics Forum, vol. 29, no. 4, (2010), pp. 1497-1506.
- [21] M. Kass and D. Pesare, "Coherent noise for non-photorealistic rendering", ACM Transactions on Graphics, vol. 30, no. 4, article no. 14, (2011).
- [22] J. Yoon, I. Lee and H. Kang, "Video Painting Based on a Stabilized Time-Varying Flow Field", IEEE Transactions on Visualization and Computer Graphics, vol. 18, no. 1, (2012), pp. 58-67.
- [23] A. Sundquist, "Dynamic line integral convolution for visualizing streamline evolution", IEEE Transactions on Visualization and Computer Graphics, vol. 9, no. 3, (2003), pp. 273-282.
- [24] M. McCool and E. Fiume, "Hierarchical poisson disk sampling distributions", In Proceedings of Graphics Interface, (1992), pp. 94-105.
- [25] J. Wijk, "Image Based Flow Visualization", In Proceedings of ACM SIGGRAPH, (2002), pp. 745-754.

## Authors



### **Heekyung Yang**

She got her M.S. at Dept. of Computer Science of Sangmyung Univ. at 2012. Currently, she is a Ph.D. candidate at the same department. Her research interest includes computer graphics and image processing.



### **Kyungha Min**

He got his Ph. D. at Dept. of Computer Science and Engineering of POSTECH at 2000. He is a professor at Dept. of Media Software at Sangmyung Univ. since 2006. His research interest includes computer graphics and image processing.