Pacific Graphics (2011)
Bing-Yu Chen, Jan Kautz, Tong-Yee Lee, and Ming C. Lin (Editors)

Short Papers

Perceptually-based Stroke Pattern Synthesis

Dongwei Liu, Junsong Zhang and Changle Zhou

Mind, Art & Computation Group, Cognitive Science Department, Xiamen University
Fujian Key Laboratory of the Brain-like Intelligent Systems, Xiamen University

Abstract

We present a novel method to synthesize stroke patterns from a user-specified vector reference pattern. The main idea is inspired by research in visual perception. We first extract the features which have an important impact on visual perception from the reference pattern, and then adopt a three-stage approach to synthesize new patterns. The synthesis starts with a randomly generated pattern whose element density is the same as that of the reference pattern, and then we substitutes the elements iteratively to optimize the proportion and neighborhood features. Finally, we iteratively adjust the elements' position to ensure that the distance feature of the target pattern is close to that of the reference one. The method can synthesize most of the stroke patterns with the element distribution from uniform to non-uniform, and from overlapping to non-overlapping.

Categories and Subject Descriptors (according to ACM CCS): I.3.4 [Computer Graphics]: Graphics Utilities—Paint systems

1. Introduction

The rendering and synthesis of stroke patterns is an important research area of non-photorealistic renderings. Generally, a stroke patterns is made up of some basic elements, and the elements appear repeatedly in the pattern. An intuitive and effective synthesis method can remarkably reduce the burden of artists.

Texture synthesis technologies [WLKT09] can generate pixel-based textures, but have difficulty in keeping the integrality of the elements in stroke patterns. In addition, the existing methods for stroke patterns synthesis can be grouped into two categories: neighborhood based [BBT*06, BBMT06, IRMM08, PWS10] and statistical parameters based methods [HLT*09, JHH10, Wei10]. The former mainly considers the local features like neighborhood relationship, and the latter focuses on the global features like the proportion of different types of elements.

Stroke pattern synthesis by example can be interpreted as the process that firstly extracts a set of style features from the reference pattern, and then create new patterns in accordance with these features. Obviously, the selections of the features are extremely significant for the synthesis. Research findings from the cognitive science, especially human visual perception, can provide good inspiration. According to the feature integration theory of attention [TG80], the global features like the proportion between element types and the density of elements, and the local features like neighborhood relationship and distance between elements are both important to the pattern.

Based on the above discussion, we present a novel method to synthesize stroke patterns, which considers both global and local features of patterns. The method takes a small-size vector pattern as reference, and it includes two parts: the analysis part clusters the elements of the reference pattern into types and extracts the global and local features; then the synthesis part rearranges elements with these features. Figure 1 gives an overview of our method.

1.1. Related works

Texture synthesis methods can generate large-scale texture base on user-specified examples. Some of these methods splice pixels or patches iteratively to extend the textures by neighborhood comparison [EL99, WL01], and some others synthesize new textures by statistical parameters [PS00]. Wei et al. [WLKT09] presented a good overview of these methods. The results of these methods are usually satisfactory for most cases, but they are not applicable for the synthesis of the stroke patterns. Since the basic unit of synthesis is one or several pixels, these methods can hardly keep the

DOI: 10.2312/PE/PG/PG2011short/013-017

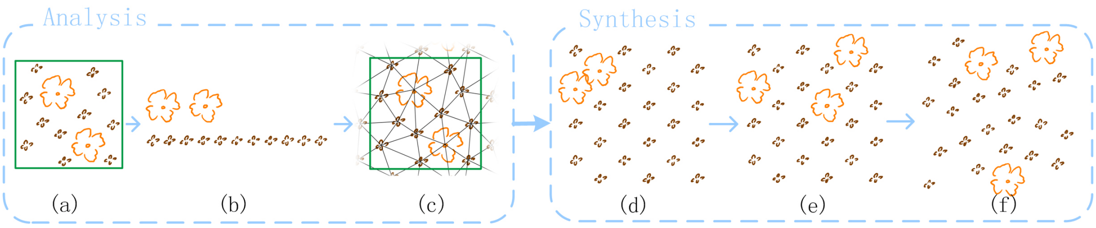

Figure 1: *Method overview: The analysis part takes (a) a reference pattern as input, then (b) clusters the elements of the reference pattern and extracts global features, finally (c) probes the neighborhood relationships between elements and extracts local features. The synthesis part first (d) initializes with a random pattern, then (e) optimizes the proportion and neighborhood features, finally (f) optimizes the distance feature and achieves the final result.*

integrality of the elements and the whole distribution features of the pattern.

In recent years, some researchers began to focus on vector-based pattern synthesis techniques. Barla et al. [BBT*06, BBMT06] first introduced a method that uses the neighborhood relationship between elements to synthesize stroke patterns. Ijiri et al. [IRMM08] and Passos et al. [PWS10] also described similar methods. These methods share the same process that first places some points on the target pattern, called seeds, and then replaces the seeds by elements, however this process is not propitious to the control of the distance between elements and the global features of patterns.

Differing from the neighborhood comparison based methods, Hurtut et al. [HLT*09] firstly cluster the elements in the reference pattern by appearance, then they describe the global features of the reference pattern as a set of statistical parameters, and finally generate new element arrangements via these parameters. Jenny et al. [JHH10] and Wei [Wei10] directly take distribution parameters and elements as input. They do not use reference pattern. The above methods well keep the global feature of the reference pattern, but ignore the neighborhood feature.

1.2. Contributions

The main contributions of our method include the following: 1) Adopt a method that can improve the utilization rate of the elements in the reference pattern. 2) Propose a multi-stage optimization algorithm to optimize the local and global features of the target pattern. 3) Build a dynamic adjustment algorithm when optimizing the distance feature.

The advantage of our method is that it takes into account both the local and global features of the pattern, which meets the rules of visual perception. In addition, it can synthesize most of the stroke patterns from uniform to non-uniform, and from overlapping to non-overlapping.

2. Analysis

The first part of our method is to extract the important features of the reference pattern. These features will be used in the next part to synthesize new patterns.

2.1. Global features extracting

Similar elements can often be found in stroke patterns. Fox's research [Fox78] shows that in human visual information processing elements are clustered by their appearance attributes. Inspired by this finding, we employ a density-based clustering algorithm to group the elements of the reference pattern into types. We select the same characteristic parameters as Hurtut's [HLT*09], including area, principal orientation, elongation, extremities and crossings. These characteristic parameters form a feature space, and each element in the reference pattern can be corresponded to a point. Then we connect any two points in the space which have a Euclidean distance less than an empirical threshold. At last, the elements in each connected component form a type.

After this, we count the number of elements in each type, and the proportion feature vector p^{ref} records the percentage of each type. The density feature is computed as the total number divided by the area of the reference pattern.

2.2. Local features extracting

In order to probe the neighborhood relationships between elements, we use the center of gravity of each element to represent its location, and construct an adjacent graph by the Delaunay triangulation (as shown in Figure 1(c)). It should be noted that the neighborhood of the elements on the edge of the reference pattern is not complete because of the lost of the context beyond the edge. The usual practice [BBT*06, IRMM08, PWS10] to solve this problem is to discard elements on the edges of reference patterns. In order to make full use of the limited elements, our method assumes that the reference pattern is quadrilateral continuous, which means that the reference pattern is not along but encircled with the same patterns, so that the top of the pattern can serve as the missed context of the bottom edge, and so on.

For each triangle j of the adjacent graph, we register the types of the elements on its vertexes, and create a neighborhood relationship vector Ω_i^{ref} (remove duplicate ones).

In the adjacent graph, the element E_n which connects directly with the element E_c is called a neighbor of E_c. Since the reference pattern is assumed quadrilateral continuous, their distance is computed as the Euclidean distance from E_c to the nearest mirror of E_n. For each element E_c, we measure the type and distance of all the neighbors.

3. Synthesis

So far we have got those features which describe the visual characteristics of the reference pattern.In order to take the global features into account and control the distance feature freely, we create a multi-stage optimization algorithm. The algorithm initializes a random pattern which keeps the density feature of the reference pattern, then optimizes the proportion and neighborhood features, and finally optimizes the distance feature and achieves the final result.

3.1. Pattern initialization

Given the size of the target pattern, the elements amount can be determined by the density feature of the reference pattern. We equiprobably select elements from the reference pattern to form a new pattern. It is easy to adjust the distribution of elements from uniform to non-uniform, thus we start with a uniform distribution to fit both cases of the uniform and non-uniform patterns. To this end, we place elements via a uniform triangular mesh as shown in Figure 1(d). The neighborhood relationships of the pattern have been determined with this process. So far the locations of the elements are temporary, and the elements will be moved during the process of distance optimization.

3.2. Proportion and neighborhood optimization

To make the proportion and neighborhood features similar to that of the reference, we iteratively substitute the elements in the target pattern. First we calculate the difference of proportion and neighborhood between the reference and the target pattern, called error ε. Then the elements in the target pattern are iteratively substituted for other ones until ε can not be reduced further. The same as reference pattern, we apply quadrilateral continuity again to repair the context.

The error ε of the target pattern combines the proportion error ε_p and the neighborhood error ε_Ω.

$$\varepsilon = \varepsilon_p + W \cdot \varepsilon_\Omega \tag{1}$$

$$\varepsilon_p = \left| P^{\text{ref}} - P^{\text{tar}} \right| \tag{2}$$

$$\varepsilon_\Omega = \sum_i \min_j \left| \Omega_j^{\text{ref}} - \Omega_i^{\text{tar}} \right| \tag{3}$$

Here P^{tar} and Ω_i^{tar} are the proportion feature vector and the neighborhood relationship vector of the target pattern, and can be computed in the same way of P^{ref} and Ω_j^{ref}. The weight parameter W is assigned to 0.1 by experience.

A substitution replaces an element with another one. Some substitutions may reduce the whole error ε of the target pattern, and our goal here is to reduce ε as much as possible. Greedy algorithm can solve the problem. We iteratively implement the substitution that can brings the greatest decrease of ε, until no substitution can reduce ε anymore. Since ε is always no less than 0 and monotone decreasing, the algorithm is convergence. Though greedy algorithm cannot always find the optimal solution, it is still an efficient method for those problems which have many solutions. The results in Section 4 show that it performs well.

3.3. Distance optimization

An important visual feature of stroke patterns is the distance between elements. In the third stage of synthesis, each element in the pattern is regarded as an individual entity and adjusts its location adaptively in response to the environmental impact. The iteration times can be set by user. Usually more times are needed to synthesize the non-uniform distributed patterns than the uniform ones.

Firstly, we calculate the impact on elements. For each element E_t in the target pattern, we find out its corresponding element E_r in the reference one, and match the neighbors of E_t with those of E_r by type and distance. Then we consider the distance between E_t and one of its neighbors E_n. If it is larger than the distance between E_r and the corresponding neighbor $E_{n'}$, as shown in Figure 2(a), we think that E_t and E_n are too far from each other, thus E_t receives an attractive force from E_n, and in the reverse case, E_t receives a repulsive force. The strength of the force depends on the difference of the two distances. We calculate the resultant force on E_t, and it will determine the action of E_t in the next adjustment. Then, we can adjust the location of

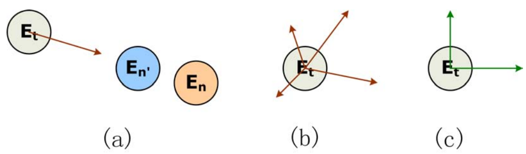

Figure 2: *Impact calculation: (a) E_t receives an attractive force because it is too far from the neighbor E_n; (b) all the neighbors make impacts on E_t; (c) the horizontal and vertical component of the resultant force are calculated.*

elements .During the location adjustment process, each element iteratively moves a small distance along the direction of resultant force pushed on it. The environment of each element in the pattern would change after each adjustment, so

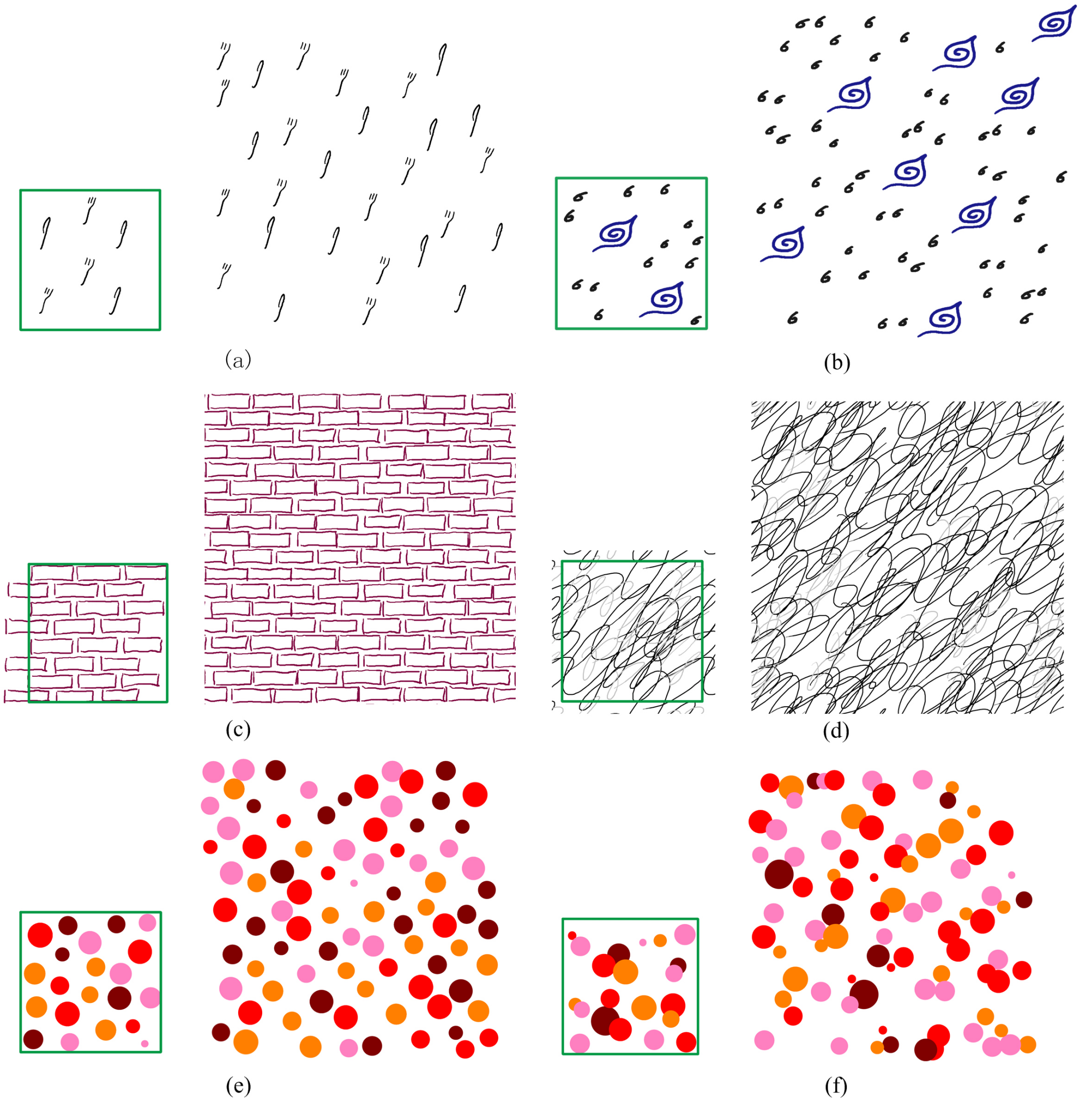

Figure 3: *Results: the pattern in the green box of each sub-figure is the reference pattern, and the large pattern on the right is the synthesized result.*

the resultant force on elements should be calculated every time. As we move elements slightly, this operation will not damage the neighborhood feature optimized in last stage.

4. Results and Discussion

Figure 3 shows the synthesized results in the case the reference patterns with different kinds of element distribution. In Figure 3(b), there are no two big elements adjacent to each other in the reference pattern, and this visual characteristic is kept well in the synthesized pattern, which indicates that the greedy algorithm works well in the optimization of neighborhood feature. This synthesis method starts with a uniform distributed pattern and adjusts the location of elements later, so it can synthesize both the non-uniform patterns like Figure 3(f) and the uniform ones like Figure 3(e).

The element distribution looks sparse in Figure 3(a) and (b), but dense in (c) and (d). The result patterns inherit these characteristics well. And some elements are overlapped in the reference pattern of Figure 3 (d) and (f), but not in the other sub-figures of Figure 3. This method can control the distance feature precisely in the synthesis process, so it can deal with overlapping. Figure 4 gives a comparison of our method and the methods in [HLT*09] and [PWS10]. The former is a typically statistical parameters based synthe-

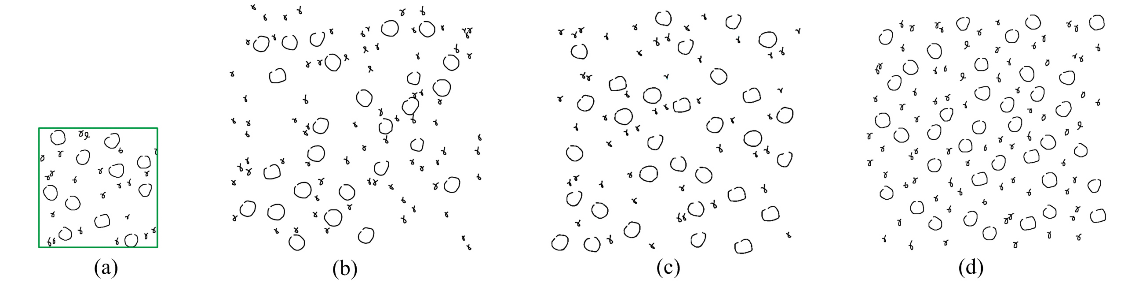

Figure 4: *The comparisons with related works: (a) reference pattern; (b) result by Hurtut et al. [HLT*09]; (c) result by Passos et al. [PWS10]; (d) our result.*

sis method, and the latter is the latest neighborhood based method.

Certainly, our method also has some limitations and shortages. The same as others, our method also simplify elements as points when analyze and control the neighborhood relationship between elements, which is inappropriate in the case that the element shape is narrow. Figure 5 shows a failure result in this case. In addition, this method is not suitable for the patterns with special structure, because the pattern structure is beyond the descriptive power of our method. Besides, the iterative frequency of the distance optimization needs to preset by experience. These problems are expected to be solved in our future work.

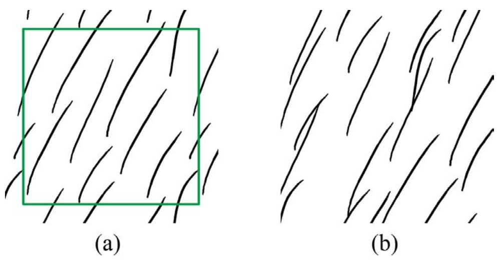

Figure 5: *A failure case of our method: (a) the reference pattern; (b) the synthesis result.*

5. Conclusion

Inspired by the visual perception research, we present a novel method to synthesize stroke patterns from a user-specified reference. The synthesis makes use of the local and global features analyzed from the reference pattern to generate patterns similar to the reference one. To the best of our knowledge, our method is one of the first trials to synthesize stroke patterns integrating local and global features. Our method can well synthesize stroke patterns from uniform to non-uniform, and from overlapping to non-overlapping.

Acknowledgment: We thank all the reviewers for their helpful comments. The work described in this paper is supported by National Natural Science Foundation of China (No. 60903129 and No. 60975076).

References

[BBMT06] BARLA P., BRESLAV S., MARKOSIAN L., THOLLOT J.: Interactive hatching and stippling by example. *Tech. rep. INRIA* (2006). 1, 2

[BBT*06] BARLA P., BRESLAV S., THOLLOT J., SILLION F., MARKOSIAN L.: Stroke pattern analysis and synthesis. *Computer Graphics Forum 25*, 3 (2006), 663–671. 1, 2

[EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by nonparametric sampling. *IEEE Int. Conf. on Computer Vision* (1999), 1033–1038. 1

[Fox78] FOX J.: Continuity, concealment and visual attention. *Strategies of Information Processing* (1978), 23–66. 2

[HLT*09] HURTUT T., LANDES P.-E., THOLLOT J., GOUSSEAU Y., DROUILHET R., COEURJOLLY J.-F.: Appearance-guided synthesis of element arrangements by example. *Proceedings of the 7th International Symposium on Non-photorealistic Animation and Rendering* (2009), 51–60. 1, 2, 4, 5

[IRMM08] IJIRI T., R. MECH T. I., MILLER G.: An example-based procedural system for element arrangement. *Computer Graphics Forum 27*, 2 (2008), 429–436. 1, 2

[JHH10] JENNY B., HUTZLER E., HURNI L.: Point pattern synthesis. *The Cartographic Journal 47*, 3 (2010), 257–261. 1, 2

[PS00] PORTILLA J., SIMONCELLIE P.: A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. J. Comp. 40*, 1 (2000), 49–70. 1

[PWS10] PASSOS V. A., WALTER M., SOUSA M. C.: Sample-based synthesis of illustrative patterns. *18th Pacific Conference on Computer Graphics and Applications* (2010), 109–116. 1, 2, 4, 5

[TG80] TREISMAN A. M., GELADE G.: A feature-integration theory of attention. *Cognitive Psychology 12* (1980), 97–136. 1

[Wei10] WEI L.: Multi-class blue noise sampling. *Proceedings of ACM SIGGRAPH 79* (2010), 1–8. 1, 2

[WL01] WEI L.-Y., LEVOY M.: Texture synthesis over arbitrary manifold surfaces. *Proceedings of ACM SIGGRAPH* (2001), 355–360. 1

[WLKT09] WEI L., LEFEBVRE S., KWATRA V., TURK G.: State of the art in example-based texture synthesis. *Eurographics '09 State of the Art Reports* (2009). 1