ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/4247935

Natural Textures for Weather Data Visualization

Conference Paper · August 2006

DOI: 10.1109/IV.2006.77 · Source: IEEE Xplore

CITATIONS
18

READS
194

4 authors, including:

Yingcai Wu
Microsoft
120 PUBLICATIONS **2,764** CITATIONS
SEE PROFILE

Hong Zhou
Shenzhen University
24 PUBLICATIONS **1,067** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Computational Sports Science View project

All content following this page was uploaded by Hong Zhou on 24 July 2014.

The user has requested enhancement of the downloaded file.

Natural Textures for Weather Data Visualization

Ying Tang *
Software College
Zhejiang University of Technology
Hangzhou, China
ytang@cad.zju.edu.cn

Huamin Qu Yingcai Wu Hong Zhou
Computer Science Department
Hong Kong University of Science and Technology
Kowloon, Hong Kong
{huamin | wuyc | zhouhong}@cs.ust.hk

Abstract

In this paper we present a novel method to visualize weather data with multi-layer controllable texture synthesis. Texture possesses multiple principal perceptual channels, which makes it good at encoding multiple data attributes contained in weather data. The natural textures existed in the real world especially provide plenty of choices to encode the data with visually pleasing images. A controllable texture synthesis method is developed to generate a large amount of textures which change the appearances of their individual perceptual dimensions according to the underlying distribution of data attributes. In order to encode more data attributes we further propose multi-layer texture synthesis. The background and foreground textures are separately synthesized and then combined together for display. In the end, we apply our method to some real-world weather data and demonstrate its effectiveness with a user study.

1. Introduction

With the explosion of simulated and acquired data in many areas ranging from scientific communities to industrial regions, visualization is employed to help users explore and gain insight into the data with effective graphical representations. Recently, the need to effectively visualize multi-dimensional data arises in various fields such as environmental studies, climatology and geology. In this paper, we focus on weather data visualization which arises as a typical problem in multi-variate data visualization. For multi-variate data display, it is necessary to design the methods to depict these data in a singe display to facilitate users to develop an integrated understanding of the whole data distributions and find out the possible correlations between different attributes.

*This work was done when Ying Tang was a visiting scholar at Hong Kong University of Science and Technology.

Textures are ubiquitous visual phenomena in our life. The observation of textures usually only involves low-level visual system, which means we can differentiate textures very rapidly and accurately without the need for focused attention. Ware and Knight did a pioneering work in using visual textures for information display [17]. According to the results of vision research, he identified three fundamental visual dimensions of textures for human perceptions, which are orientation, size and contrast. In order to incorporate richer natural textures in data visualization, Interrante proposed to harness natural textures for multivariate visualization [8]. Restrained by the available tools on hand, the author only illustrated the desired results via Adobe Photoshop. With the development of texture synthesis techniques during recent years, it becomes possible to encode multivariate data via real natural textures. Colorful photographs or digital images can be used as the input samples to the texture synthesis algorithms.

In this paper we present a novel controllable multi-layer texture synthesis method from which the synthesized results can be used to encode the underlying changes of different data attributes in a single output image. In controllable texture synthesis, we can synthesize the results not only varying in scale, but also in orientation and regularity. However, to synthesize a texture by varying too many visual channels will overwhelm an observer's viewing ability. Besides, it has been a general agreement that there are a small number (about three) of characteristic dimensions. In order to make our method capable to encode more data attributes, we further propose the multi-layer synthesis in which we separately synthesize the background and foreground textures. By taking advantages of the human vision system's ability to consistently complete the background and unambiguously differentiate between the foreground and background, this method is a favorable solution for multi-variate data visualization.

This paper is organized as follows: After briefly reviewing previous work in Section 2, we introduce the guidelines for selecting natural textures in Section 3. The controllable multi-layer texture synthesis is introduced in Section 4. The results are presented in Section 5 where a user study is also conducted. We conclude and describe future directions of our work in Section 6.

2 Related Work

Textures have been extensively studied and applied in many research fields, such as computer vision, computer graphics and cognitive psychology. Although different groups concentrate on different tasks, it is advantageous to consider interdisciplinary integration of these research efforts and apply it in new areas, e.g., data visualization. In this section, we briefly review some papers highly related to our work.

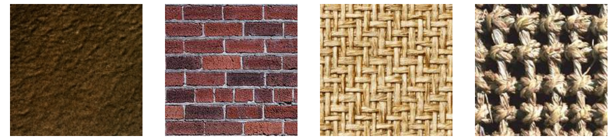
Information Encoding via Textures: In the visualization field, people have studied methods for using texture patterns to display information. Ware and Knight [17] identified three principal visual dimensions of textures according to vision research and employed Gabor filters to construct texture patterns that can be modulated along the three dimensions. Interrante *et al.* have done a lot of research about how to use textures to enhance the 3D shape perception [8] [9] [10]. She proposed in [8] to harness natural textures for multi-variate visualization. Healey *et al.* have been investigating visualization methods to explore and analyze large, complex, and multidimensional datasets by exploiting the power of the low-level human visual system [2] [3] [4] [5]. They proposed an engaging nonphotorealistic visualization system using perceptually-based brush strokes in [5].

Texture Synthesis: Texture synthesis has been a hot topic in recent years in computer graphics. Non-parameterization methods allow us to generate arbitrary-sized similar textures of high quality from a small input sample [1] [19]. In order to incorporate variances in the synthesis results, some schemes were developed to offer some forms of user guidance. Zhang *et al.* synthesized locally deformed texture elements of the transition between two homogeneous textures [21]. Lefebvre *et al.* [12] presented parallel controllable texture synthesis which control the amount of texture regularity by multiresolution jittering of exemplar coordinates.

3. Texture Selection

It is obvious that not all natural textures are suitable for our application. For example, it is difficult for us to decern the useful information if the sand texture with stochastic features is used to encode the data (as shown in Figure 1(a)).

In order to introduce universal variances in the feature dimensions of textures in the synthesized image to encode the data attributes, we need to start from a regular texture sample. This regular sample provides the anchor point from which the feature dimensions are varied and measured. However, due to different viewpoints or lighting conditions, most of the existing textures are near-regular instead of being regular or homogeneous [6] [20]. Directly applying such near-regular textures to our controllable texture synthesis will not produce satisfying data encoding results. The users may become confused about whether the feature variations are inherent in the sample or purposely synthesized, thus unable to understand the underlying data distributions. In our algorithm, we use the near-regular texture analysis method in [13] to obtain a regular texture sample. However, we need not reconstruct the irregularity of the original sample, instead we artificially generate non-homogeneous synthesis results according to the data distributions.



(a) (b) (c) (d)

Figure 1. Texture samples: (a) Sand texture; (b)-(d) Near-regular textures.

4 Multi-Layer Controllable Texture S̃ynthesis

In this section, we first introduce controllable texture synthesis which produces the background textures. Multi-layer texture synthesis for the foreground textures is described later.

4.1 Texture Synthesis with Variances in Multiple Dimensions

Our texture synthesis method belongs to patch-based methods where the synthesis units are patches instead of pixels. Before synthesizing texture, a preprocessing is needed to generate the patch units to be synthesized.

4.1.1 Preprocessing

In our application for weather data visualization, each data at the same location have multiple different data attributes. In order to generate descriptive textures for such data, we first partition the plane according to data distributions and assign each data a region or a patch where customized textures are synthesized.

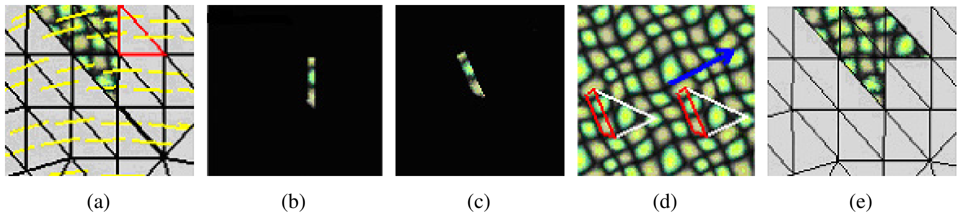


Figure 2. Controllable texture synthesis: (a) The triangle in red is to be textured with the left neighbor triangle already been textured. The yellow lines indicate the direction of each triangle; (b) The mask extracted from the textured neighbors for the current triangle (in red); (c) The mask rotated according to the direction difference between the current triangle and the sample texture (The direction of sample texture is indicated in blue arrowed line in (d)); (d) Looking over the sample for candidate locations that match this rotated mask by translating the mask over sample texture; (e) Selecting the one that best fits the mask and rotate it back to paste on the current triangle.

Guaranteed-Quality Triangular Mesh Generation: In our algorithm, we use the triangle patches as the building blocks for the texture synthesis method since it is flexible and convenient to obtain them. We first invoke Voronoi diagram to partition the plane into convex polygons such that each polygon contains exactly one data point. The particular data attributes of a data point are assigned to the corresponding polygon and the textures synthesized on each polygon should have the uniform appearances indicating underlying data values. We further divide each polygon to a set of triangles which are the units to be synthesized. The sets of triangles belonging to the same polygon share the same data attributes associated with that polygon.

Control Field Generation: The data attributes associated with each triangle determine the control field of texture synthesis. The vectors in the control field have three attributes which correspond to the scale, orientation and brightness values of the textures respectively. We map different data attributes to vector sub-values with some mapping functions. Usually, such mapping functions are linear since we assume the perceived differences of orientations, scales and brightness are in linear relation with the differences in values. Our principle of data-feature mapping is to combine human preferences with feature hierarchy. If there are no visual interferences among data features, we need to consider the preferences of users.

4.1.2 Texture Synthesis With Multiple-Dimensional Variances

Here we introduce how to synthesize triangles with varied scales, orientations, and brightness values. We adopt the traversal method used in [16] to visit all triangles in the output region. During the traversal process, texture synthesis is grown from one triangle to its neighboring ones. The algorithm is stated as follows:

1. Take a triangle from the queue;
2. Extract the mask from the already textured neighbors and rotate the mask to comply with the current orientation.
3. Look in the scaled sample for a good patch that matches the mask and paste it on the triangle after the proper rotation with modified brightness.
4. Add the non-textured neighbors to the current queue.

Mask Extraction and Rotation: The mask is the region used to constrain the possible matches for the current patch. We use the textures over the narrow bands extracted from textured neighboring regions as a constrained region. This is a simple way to take into account local statistics of the texture across regions' borders. The textured neighbor triangles provide constrained masks for current triangle to be synthesized. After extracting the masks, we need to rotate the mask to align with the orientation of the sample texture to search for the best fit in the space of sample texture. In our algorithm, we rotate the masks according to the angle difference between the current patch and the sample texture. Figure 2 shows the process of extracting and rotating masks.

Scale Variance Control: The scaled sample texture instead of the original sample is used as the searching space for the best matching of masks. The pixels in the scaled texture sample are resampled from the original texture sample by bilinear interpolation of the four nearest pixels in the original sample image.

Brightness Variance Control: The color values of the original image are transformed from RGB color space to CIE Lab color models to separate the luminance and the chromatic values. The three parameters L, a and b respectively represent the luminance of the color, its position between red and green, and its position between yellow and blue. We leave the values of a and b intact, while linearly change the L values according to the underlying data value.

4.2 Multi-Layer Texture Synthesis

In order to synthesize foreground textures, we use the simple glyph-like textures and distribute them across the whole image according to the data values. The density of the foreground's textures is used to encode the data.

When the foreground texture is overlaid upon the background texture, it is important that the foreground texture should have a reasonable luminance difference from the background. Otherwise, it is difficult for us to differentiate the foreground from the background. According to Mullen [15], pure chromatic differences are not suitable for displaying any kind of fine detail. The International Standards Organization (ISO 9241, part 3) recommends a minimum 3:1 luminance ratio of text and background and 10:1 is preferred. In our application, we adaptively adjust the illuminance of the foreground according to that of the background to ensure there is a reasonable luminance difference between them.

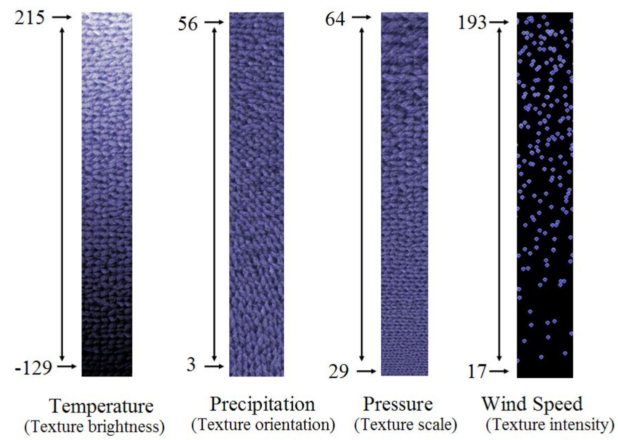


Figure 3. Encoding scheme: Texture brightness used to encode temperature; texture orientation to encode precipitation; texture scale to encode pressure; and foreground texture density to encode wind speed.

5 Results and Discussion

In this section, we first present the results of applying our algorithm to visualize the real weather data sets. Next we conduct a user study to test the effectiveness of our method compared with other visualization methods.

5.1 Results of Real Applications

We have tested our method with the collection of monthly weather data downloaded from the website of ipcc (intergovenmental panel on climate change). Healey et al. also used such data sets as their testbeds in [5]. Regarding people's preferences, we use the following mappings to encode weather data attributes:

1. Temperature - Brightness of textures. Dark for low temperature and bright for high temperature.

2. Precipitation - Scale of textures. Small scale for light precipitation and large scale for heavy precipitation.

3. Wind speed - Orientation of textures. Vertical principal orientation for high wind speed and horizontal principal orientation for low wind speed.

4. Vapor pressure - Density of foreground texture. Dense region for high vapor pressure and sparse region for low vapor pressure.

In Figure 5 we show the visualization result of the climate condition for Feburary according to the above mapping rules over a large part of China and some regions to the south of Himalayas, which is indicated by the red square in Figure 5(a). The black regions in the result are places such las ocean or lake, where no weather data have been recorded. The combined visualization result of three variables except pressure is shown in Fig. 5. We did not include foreground textures in this result. The changes of the brightness of textures vividly show the temperature's variation pattern across the whole region running northwest to southeast from low to high. The amount of precipitation is represented by the scale of textures. Large scales in the south-east part of Chinese mainland as well as Taiwan indicate that there is abundant precipitation there. The wind speed modifies the principal orientation of texture: weak wind speed corresponds to near-horizontal orientation (e.g. Sichuan Province), while strong wind corresponds to near-vertical orientation (e.g. the Jiaodong Peninsula).

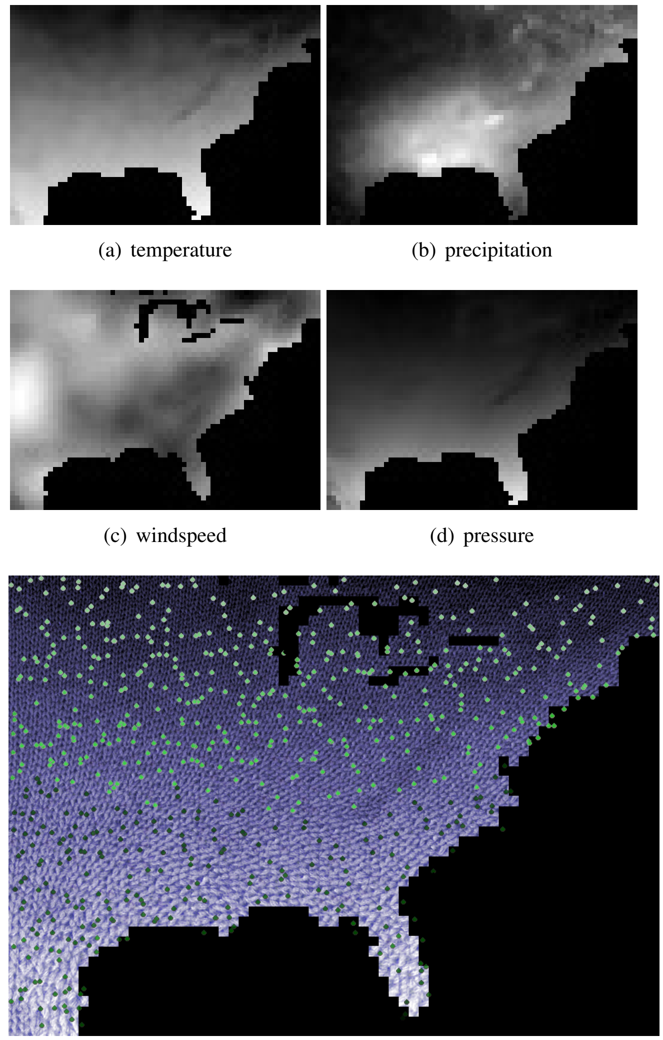
In Figure 6 we show two texture synthesis results of the same texture sample over large regions in China (the same as the region in Figure 5) and the eastern U.S. for the same period of February. We can compare these two visualization results to get many interesting insights on the differences of the weather conditions in these two countries. For example, according to the brightness variation it is obvious that the temperature's distribution is much more diverse in China than that in the U.S. This is due to the more complex topography in China. Furthermore, there are a lot of near-vertical orientations in the U.S. compared with many near-horizontal orientations in China, which suggests that the U.S. has overall stronger wind than China for the same period.

5.2 User Study and Discussion

Our work of employing texture synthesis to encode multi-variate data is in part inspired by the work of non-photorealistic visualization by perceptually-based brush strokes in [5]. In order to validate the encoding ability of our algorithm, we design a basic user study experiment to study the effectiveness of our method. The questions have been designed to ask users to identify areas in the images which have the following properties: the highest temperature, the largerst precipitation, the weakest wind, and highest pressure. We conduct this user study for our algorithm as well as the non-photorealistic visualization algorithm in [5] for comparison.

Twenty people with normal visual systems participated in this test and they were equally divided into two groups. In order to be in accord with mapping rules used in [5], we use texture brightness to encode temperature, texture orientation to encode the precipitation, texture scale to encode pressure, and foreground texture density to encode wind speed (See Figure 3). We limit each group to the results from one algorithm to avoid the impressions from both algorithms interfering with each other. The test image produced by our algorithm is shown in Figure 4. We ask the users to answer the questions for this test image and the non-photorealistic visualization image (Figure 9 in [5]). In Table I the responses from users are listed, from which we can see that as a whole the performance of our algorithm is comparable or even better than that of non-photorealistic algorithm. Specifically, users are better at identifying high temperature and high pressure in our visualization results and are better at identifying low wind speed for non-photorealistic results.

We try to give some explanations to the comparison results concerning the rules in human perception. As we have learned from the users, they are more sensitive to the changes of intensities than colors. Furthermore, it is shown in [18] that the changes of intensities are much better at encoding details than color patterns. By taking advantage of the intensities' changes, our algorithm is more accurate in encoding temperatures and more vivid in representing detailed change patterns. For example, the darker band at the east coast of the U.S. (as indicated by the yellow square) suggests the mountain of Appalachians which is difficult to discern from Figure 9 in [5]. The amount of precipitation is encoded by the size of strokes in [5] which would be difficult to discern when the color becomes too dark. In this case the perception of color masks the detection of stroke size. In our algorithm, we adjust the intensities within a range to ensure that too light or too dark regions will not appear and such range is large enough to encode the variance of data. The wind speed is encoded as coverage of strokes in [5] which is easy to detect and makes the algorithm in [5] outperform ours in the ability to encode wind. During our user study, many subjects felt that our visualization results with real-world natural textures are more attractive than the results in [5]. In addition to being effective, our method produces results more engaging or aesthetic.



(a) temperature (b) precipitation

(c) windspeed (d) pressure

(e) result

Figure 4. The visualization result generated by our method. The above four gray images of ((a)-(d)) are distributions of four climate attributes.

Table 1. User Responses

	Texture		Stroke	
	Right	Wrong	Right	Wrong
Highest Temperature	10	0	8	2
Largest Precipitation	9	1	9	1
Highest Pressure	10	0	8	2
Weakest Wind	8	2	9	1

6 Conclusion and Future Work

In this paper we present a novel method to encode weather data by multi-layer controllable texture synthesis. The principal visual dimensions of textures are variantly synthesized according to the changes of the underlying data. In order to encode more variables in our framework, we further propose multi-layer texture synthesis where background textures and foreground textures are separately synthesized.

There remains a lot of interesting future work to investigate for the topic of information visualization using texture synthesis. The potential of foreground texture synthesis can be further explored to effectively encode more information. Another interesting topic is whether there is more perceptual dimensions of textures available for controllable texture synthesis apart from the three used in this paper.

Acknowledgment

This work is partially supported by HK RGC grant CERG 618705 and HKUST grant DAG 04/05 EG02.

References

[1] A. Efros and W. Freeman. *Image Quilting for Texture Synthesis and Transfer*, ACM SIGGRAPH 2001, pp. 35-42, 2001.

[2] C. Healey and J. Enns. *Building Perceptual Textures to Visualize Multidimensional Datasets*, IEEE Visualization 1998, pp. 111-118, 1998.

[3] C. Healey and J. Enns. *Large Datasets at a Glance: Combining Textures and Colors in Scientific Visualization*, IEEE Transactions on Visualization and Computer Graphics, Vol. 5, No. 2, pp. 145-167, 1999.

[4] C. Weigle, W. Emigh, G. Liu, R. Taylor, J. Enns, and C. Healey. *Oriented Texture Slivers: A Technique for Local Value Estimation of Multiple Scalar Fields*, Graphics Interface 2000, pp. 163-170, 2000.

[5] C. Healey, L. Tateosian, J. Enns and M. Remple. *Perceptually-Based Brush Strokes for Nonphotorealistic Visualization*, ACM Transactions on Graphics, Vol. 23, No. 1, pp. 64-96, 2004.

[6] I. Hargittai and M. Hargittai. *In Our Own Image: Personal Symmetry in Discovery.* Kluwer Academic Publishers.

[7] D. Hubel and T. Wiesel. Receptive Fields and Functional Architecture of Monkey Striate Cortex. *J. Physiol.*, pp. 215-243, 1968.

[8] V. Interrante. *Harnessing Natural Textures for Multivariate Visualization*, IEEE Computer Graphics and Applications, Vol. 20, No. 6, pp. 6-11, 2000.

[9] G. Gorla, V. Interrante and G. Sapiro. *Texture Synthesis for 3D Shape Representation*, IEEE Transactions on Visualization and Computer Graphics, Vol. 9, No. 4, pp. 512-524, 2003.

[10] Sunghee Kim, Haleh Hagh-Shenas and Victoria Interrante. C*onveying Shape with Texture: experimental investigations of texture's effects on shape categorization judgments*, IEEE Transactions on Visualization and Computer Graphics, Vol. 10, No. 4, pp. 471-483, 2004.

[11] B. Julesz. *A theory of preattentive texture discrimination based on first-order statistics of textons*. Biological Cybernetics 41, pp. 131138, 1981.

[12] S. Lefebvre, H. Hoppe. *Parallel controllable texture synthesis*, ACM SIGGRAPH 2005, pp. 777-786, 2005.

[13] Y. Liu, W. Lin and J. Hays. *Near-Regular Texture Analysis and Manipulation*, ACM SIGGRAPH 2004, pp. 368-376, 2004.

[14] S. Marcelja. Mathematical Descriptions of the responses of simple cortical cells. *Journal of the Optical Society of America*, pp. 1297-1300, 1980.

[15] K. Mullen. *The Contrast Sensitivity of Human Color Vision to Red-Green and Blue-Yellow Chromatic Gratings.* American Journal of Optometry and Physiological Optics 359, pp. 381-400, 1985.

[16] C. Soler, M. Cani, and A. Angelidis. *Hierarchical pattern mapping*. ACM Transactions on Graphics, Vol. 21, No. 3, pp. 673-680, 2002

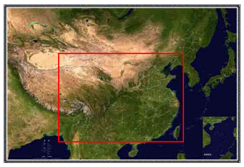
[17] C. Ware and W. Knight. *Using Visual Texture for Information Display, ACM Transactions on Graphics*, Vol. 14, No. 1, pp.3-20, 1995.

[18] C. ware. *Information Visualization: Perception for Design.* 2004.

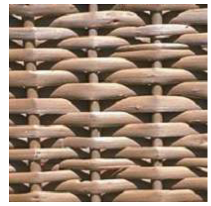
[19] L. Wei, and M. Levoy. *Fast texture synthesis using tree-structured vector quantization*. ACM SIGGRAPH 2000, pp. 479-488, 2000.

[20] A. Zee. *Fearful Symmetry.* Princeton University Press.

[21] J. Zhang, K. Zhou, L. Velho, B. Guo, and H. Shum. *Synthesis of progressively-variant textures on arbitrary surfaces*. ACM SIGGRAPH 2003, 295-302, 2003.



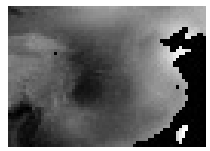
(a) china map



(b) texture sample

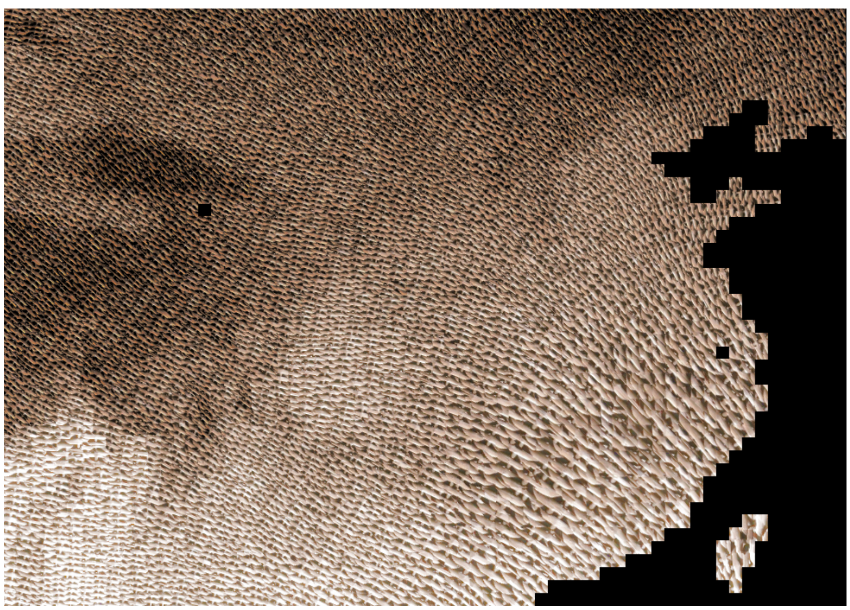
(c) temperature

(d) precipitation



(e) wind speed

(f) pressure



(g) texture synthesis result of four climate attributes

Figure 5. The four gray scale images for four climate attributes ((c)-(f)) and the visualization result in (g) with texture sample in (b) for the region shown in (a). The high intensities of the gray scale images correspond to high values and the low intensities for small values. The texture synthesis result shows the combined visualization of three climate attributes including temperature (from low to high brightness for cold to hot), precipitation (from small to large scale for light to heavy), wind speed (from horizontal to vertical orientation for weak to strong).

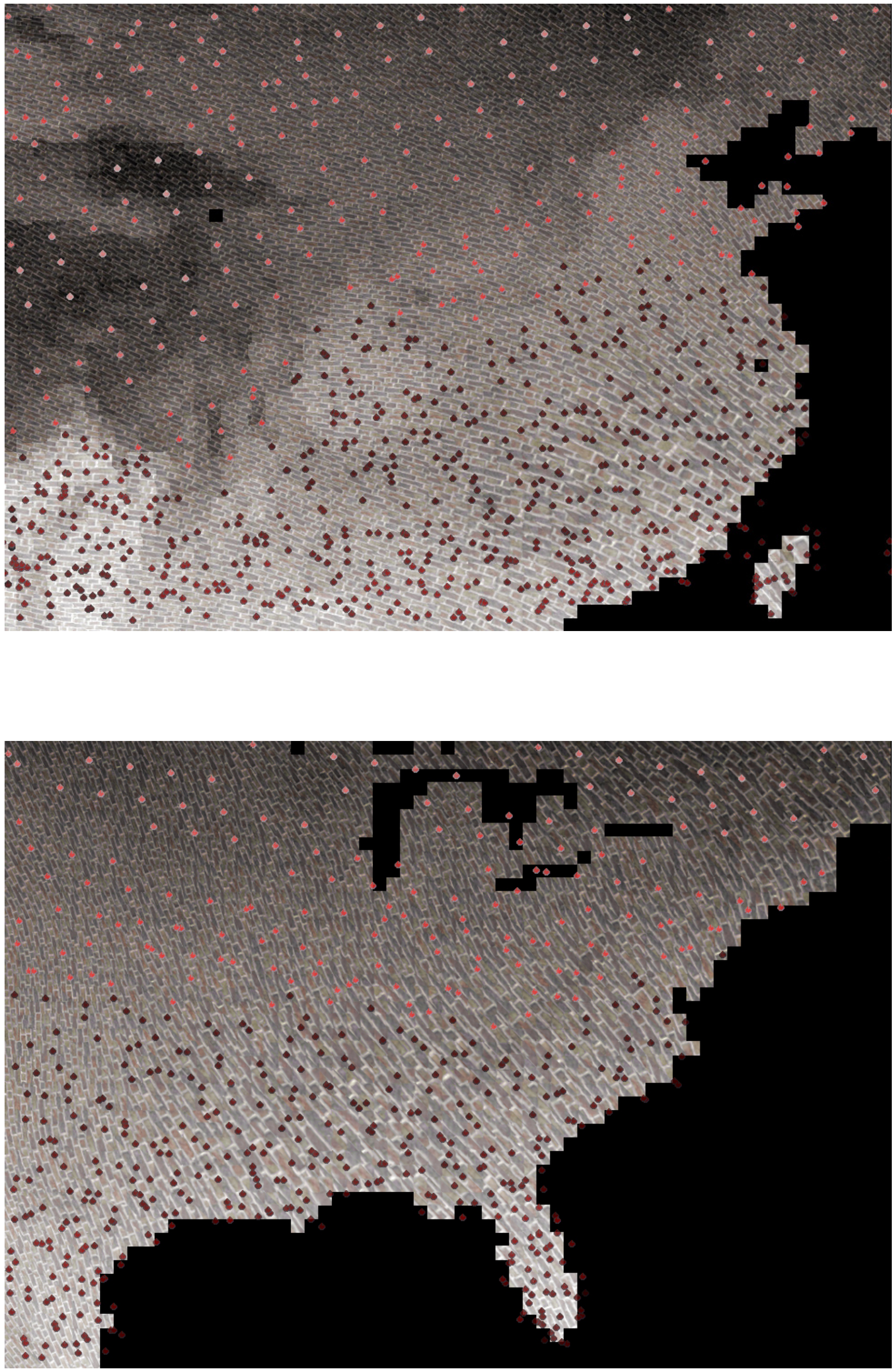


Figure 6. Texture synthesis results with the same texture sample for large regions in China (top image) and the U.S. (bottom image).

View publication stats