Vis Comput
DOI 10.1007/s00371-014-0985-7

ORIGINAL ARTICLE

Layered modeling and generation of Pollock's drip style

Yan Zheng · Xuecheng Nie · Zhaopeng Meng · Wei Feng · Kang Zhang

Abstract In this paper, we propose a layered approach to model Jackson Pollock's dripping style of paintings. Having analyzed fractal-based algorithms and observed the details of Pollock's paintings, we designed a layered modeling approach that divides Pollock's artwork into four layers: from bottom up are background layer, irregular shape layer, line layer and water drop layer. The layers are drawn sequentially and independent, forming the desired Pollock style. We have developed a program using Processing to generate artworks of the dripping style. The parameters of our program can be randomly generated or tuned by the user, supporting high flexibility and effectiveness. Experimental results show that our layered modeling approach can systematically generate images resembling Pollock's dripping style.

Keywords Jackson Pollock · Abstract painting · Layered modeling · Processing · Generative art

Y. Zheng · X. Nie · Z. Meng · K. Zhang (✉)
School of Computer Software, Tianjin University, Tianjin, China
e-mail: kzhang@utdallas.edu

Y. Zheng
e-mail: yanzheng@tju.edu.cn

X. Nie
e-mail: xcnie@tju.edu.cn

Z. Meng
e-mail: mengzp@tju.edu.cn

W. Feng
School of Computer Science and Technology,
Tianjin University, Tianjin, China
e-mail: wfeng@tju.edu.cn

K. Zhang
Department of Computer Science, The University
of Texas at Dallas, Richardson, USA

1 Introduction

Well-known for the unique style of drip painting, Jackson Pollock was an influential American painter and a major figure in the decade of remarkable abstract expressionist movement from 1943 to 1952 [25]. In contrast to traditional painting approaches by conventional brush contact with the canvas surface, Pollock generated many artworks by dripping and pouring household paint on canvases, which are rolled across the floor of his windswept barn [27]. Though opinions on this technique were initially polarized, art historians consider Pollock's patterns as a revolutionary approach to aesthetics, not only tangled, messy and complex [32]. In addition, analysis of Pollock's painting style has drawn more and more attention in recent years.

A significant discovery toward understanding Pollock's aesthetics was made by Taylor et al. [30] in 1999, who defined the visual character of Pollock's painting style in fractals. Fractals [1,22–24] bear the "fingerprint" of nature's patterns [12,20], which leads people to label Pollock's work as "Fractal Expressionism" [31]. Since fractal analysis holds great promise for both the academic and artistic communities, most of recent research work on analyzing Pollock's paintings has focused on this technique. Taylor et al. proposed several techniques to quantify the fractal contents of Pollock's drip paintings [26,27,32], to make fractal-based analysis available in practice.

Mureika [14] proposed a statistical method for generating abstract paintings based on multi-fractal analysis to understand the creative process of visual artists. Irfan and Stork [7] proposed a fusion method that combined multiple features and classifiers which could improve the performance of validation on authentication of Pollock's drip paintings. Most of previous investigations have shown that fractal-based analysis can explicitly specify the component shapes in drip style

Published online: 20 May 2014

of paintings. Fractal-based analysis can also simulate the generation process of the component shapes. As the development of modern computer graphics [13] and printing technology, automatic generation of abstract paintings [10,18,33,34], especially Pollock's drip paintings, becomes possible and draws more and more attention.

Generating abstract paintings has now become an interesting subject for not only artists but also computer scientists [2,35]. For efficiently automatic generation of Pollock's style, one important issue to be addressed is the painting process adopted by the artist. Despite the significant impact on aesthetic [5,26], fractal patterns modeling style produced by specific pouring techniques adopted by the artist has its limitations for being flat and non-versatile. Moreover, fractal analysis is a mathematical method for analyzing completed paintings. It does not attempt to mimic the generation process of drip paintings practiced by the artist.

In this paper, we propose a layered approach to generate Pollock-like images. According to the basic idea of fractal-based analysis and detailed observation of the artist's paintings, we design our algorithms to simulate the visual effects of different painting objects. We divide Pollock's drip paintings into four layers, from bottom-up: background layer, irregular shape layer, line layer, and water drop layer. Each layer is composed of basic components which are randomly located and colored. The layers are independent from each other, and are drawn sequentially bottom up. Finally, the complete artwork can be considered the combination of all the layers. We have implemented generation of Pollock's painting in Processing [19]. Processing is a programming language developed by MIT specifically for artists and designers to easily create drawings, animation, and interactive graphics. Because of the simplicity of Processing, it has drawn increasing attention of researchers to use it in prototype design and scientific experiments. Experimental results show that layered modeling provides an efficient approach to simulate Pollock's drawing process, which highly mimics the original artworks of Pollock.

The paper is organized as follows: a brief review of previous research works on analysis of drip paintings is discussed in Sect. 2. The layered approach is discussed in detail in Sect. 3. Section 4 presents the algorithm design and implementation. Experimental results are shown in Sect. 5, including the generated artworks using our layered modeling approach. Finally, Sect. 6 concludes the paper, mentioning contributions and future work.

2 Related work

The rapid development of modern computer graphics makes automatic generation of photo-realistic images of synthetic objects possible. Algorithmic methods now allow plastic elements and their juxtaposition to be expressed to a computer [10]. Compared to the describing style in a natural language, the computer methods offer the advantage of being testable for validity. The tests are accomplished by using the computer to generate new compositions in the style described. Haeberli [6] has used an ordered collection of brush strokes to create abstract images. These abstract images filter and refine visual information before being presented to the viewer. By controlling the color, shape, size, and orientation of individual brush strokes, impressionistic paintings of computer generated or photographic images can easily be created.

Zhao and Zhu [36] presented an interactive abstract painting system named Sisley, which works upon the psychological principle that abstract arts are often characterized by their greater perceptual ambiguities than photographs, which tend to invoke moderate mental efforts of the audience for interpretation, accompanied with subtle aesthetic pleasures. Given an input photograph, Sisley decomposes it into a hierarchy tree of its constituent image components with interactive guidance from the user. It then automatically generates corresponding abstract painting images, with increased ambiguities of both the scene and individual objects at desired levels. Recently, Fogleman [4] has developed an algorithm to generate images in the style of Piet Mondrian procedurally. This inspires us to develop algorithms to generate images in Pollock's drip style of paintings.

Fractal-based analysis of Pollock's paintings was first proposed by Taylor [30] in 1999. Taylor identified the visual characters of Pollock's painting style as fractals, which bear the "fingerprint" of nature's patterns, leading people to label Pollock's work as "Fractal Expressionism". Most of recent research on analyzing Pollock's paintings has focused on fractal techniques. In order to quantify the fractal content of Pollock's drip paintings, Taylor used the well-established 'box-counting' method to calculate the fractal dimension, and his analysis showed that the fractal dimensions increased steadily through the years by Pollock's refinement on the dripping technique and can be used as an objective technique both to validate and date Pollock's drip paintings [9,11]. But recent studies suggest that fractal dimension by approaches based on box-counting algorithms could not yield highly accurate authentication [3,29].

To improve the performance of validation on authentication of Pollock's paintings, Irfan and Stork [7] proposed a fusion method that combined multiple features and classifiers. Despite the uninformative characteristics of a visual feature, Irfan and Stork proved that such a feature could enhance discrimination when combined with other features using a classifier even if these other features are themselves also individually uninformative. Results of Irfan and Stork showed that multiple features in fact outperform a single feature and the fractal feature improves classification accuracy.

To identify the"finger-print" for characterizing the work of a single artist. Mureika et al. [15] adopted the multi-fractal spectrum as an alternative tool for artwork analysis. According to their experimental results, the spectrum can indeed be used to isolate a construction paradigm or art style. The authors have also applied multifractal analysis to two-dimensional, non-representational images to verify whether similar statements may be made about the analysis [16].

Considering scientific experiments to be unusual tools for judging art, Taylor et al. [28] spent 10 years on scientific investigation of human response to fractals, using eye tracking, visual preference, skin conductance, and EEG measurement techniques. Their preliminary experiments provide a fascinating insight into the impact that art might have on the perceptual, physiological and neurological condition of the observer.

Researchers have shown that mathematics and painting are interrelated in many ways [8]. Mathematics can be used to enhance one's appreciation of paintings at a technical level, while paintings can express one's comprehension about reality with mathematics at a conscious level. Mureika has proposed a statistical method of abstract paintings based on multi-fractal analysis to understand the creative process of visual artists [14]. Mureika used a phase space plot to highlight a common depth, which not only quantifies the common visual impact of fractal paintings, but also provides some intriguing insights into the artistic process that generated these paintings.

Despite the significant impact on aesthetics [5,26], fractal patterns are not an inevitable consequence of pouring paint, which can be produced by specific pouring techniques adopted by the artist. It is therefore insufficient to simply generate fractals by highly systematic process to replicate the visual characteristics of Pollock's paintings. In order to identify and quantify this set of characteristics, Taylor [29] has developed a fractal analysis procedure known as Dimensional Interplay Analysis (DIA), which combines the specific physiology of artist and the scientific parameters that quantify the resulting patterns. DIA is based on the hypothesis that Pollock's poured abstract paintings consist of two different sets of fractal patterns generated by physical motions and pouring process.

Fractal analysis is an important move toward understanding Pollock's aesthetics, it can specify what the basic components are and how they could be generated, but the concrete production process is still untouched. Our layered modeling approach aims at understanding and simulating the production process. We follow Taylor's fractal analysis to identify the characteristics of the basic components in Pollock's paintings and divide some of his artwork into several layers, each composed of a number of different shapes. Our layered modeling approach also provides a general framework for generating Pollock-like images.

3 Layered modeling

As discussed above, although fractal-based is a well-known, widely used analysis, which facilitates the understanding of Pollock's aesthetic, it cannot model the creating process with good control. In order for art learners to understand the generation process in simulating drip paintings and for computers to generate the styled images, we propose a layered approach to model Pollock's drip paintings and specify the creating process. We divide some of Pollock's drip paintings four layers, as shown in Fig. 1, i.e. background layer, irregular shape layer, line layer and point layer, all consisting of component shapes of different graphical properties. These layers are independent of each other and drawn bottom up sequentially. The shapes of components in each layer are consistent, and randomly distributed in each layer. Finally, the complete artwork can be considered the combination of these four layers. The layered modeling also provides an alternative way of generating Pollock-like images.

Using the layered modeling approach, we can generate images resembling the artist's style. Moreover, once essential parameters are pre-defined and input to the program, the algorithm will generate styled images without any human

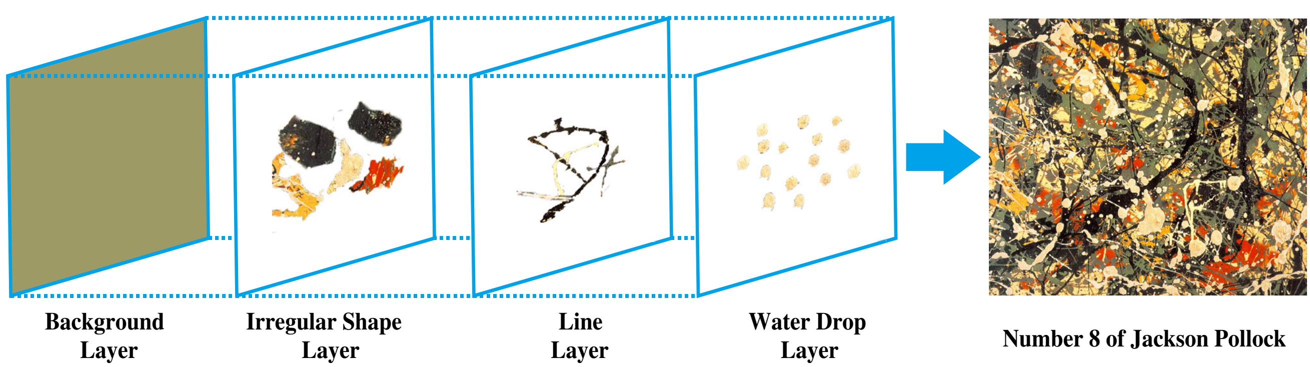


Fig. 1 Layered approach to modeling Jackson Pollock's drip style

intervention. The more precise parameters are chosen, the more similar the result will be to the original style. The approach sets the general foundation for generating abstract paintings of different styles, by different artists, since the individual algorithms presented could be easily adapted to model different styles. In the following, we will present the details of each layer modeling the artist's dripping style.

3.1 Background layer

In the layered modeling approach, the bottom layer is the background, which does not contain any basic shapes but is simply colored with gradient. The background layer covers the entire canvas and can be considered poured paints on the canvas, and provides the fundamental tone of the artwork. Other shapes are distributed above the background layer. The background layer usually contains only one color tone, which may reflect the consistency of Pollock's drip paintings. To show the physical process of paint flowing on the canvas, the background color is displayed with gradient.

3.2 Irregular shape layer

The second layer consists of irregular shapes, that are assumed to be generated by moving the brush on the canvas. Some of the irregular shapes are ellipses, whose radii are randomly generated within a pre-determined given range (a quarter to a half of width/height of the generated image in the current implementation). Some are polygons, whose sizes and vertex numbers are also generated randomly (same as the range of ellipse's sizes above, and the vertex number ranges from 4 to 8). Other irregular shapes are the combination of ellipses and polygons.

To make the boundaries of the generated shapes irregular, we add noises to the shape boundaries to mimic the physically generated ones in natural artworks. The colors of irregular shapes vary greatly, mostly close to the background color. By our observation of Pollock's drip paintings, there are usually a few irregular shapes distributed in an artwork. In the current implementation, the number of irregular shapes is within a range of 5–10. The positions of these irregular shapes on canvas are also randomly generated. All the irregular shapes are extracted from the original artworks of Pollock based on the previous fractal-based studies, with some shown in the second layer in Fig. 1. The shapes in other layers are generated in a similar fashion. The irregular shape layer is above the background layer, and can provide the basic layout of the final image.

3.3 Line layer

The third layer is composed of lines. From Pollock's drip paintings, one could observe many lines distributed throughout the entire canvas, making the main style of Pollock's drip paintings. The artist produced such lines by dragging a stick with his body movement on the canvas, dropping paints on canvas. Some lines extracted from a drip painting are shown in the third layer in Fig. 1. By our observation, the lengths of lines vary greatly, and some are very short while others cross nearly the entire image. Locuses of these lines also vary greatly. We use several mathematical functions to generate the locuses of lines, such as trigonometric functions, quadratic functions, linear functions or combinations of these functions. The colors of lines vary as well, depending on the color of the paint used by the artist. In addition, the change of color gradient of a line corresponds to the speed of the artist's movement. Above the irregular layer, the line layer forms the main layout of Pollock's drip paintings.

3.4 Point layer/paint drop layer

At the top is the point layer. We observe that paint drops are the key technique in generating the artist's abstract paintings. The height from which the paint dropped determines the size of the drop point. Some of paint drops are as small as a point, while others are as big as a rugged oval. The basic shape of a paint drop can be considered a filled circle with rough edges. The paint drops are filled with varied colors, dependent on the paints used by the artist. The positions of these points are randomly generated and uniformly distributed on the entire picture. We set the number of points in the range 300–500. The point layer/paint drop layer represents one of the most significant and natural features of Pollock's drip paintings. Examples of the paint drops extracted from Pollock's original artworks are shown in the fourth layer in Fig. 1.

4 Implementation in processing

We have implemented the layered approach using the programming language Processing to simulate the process of generating Pollock's dripping style. This section describes the generation algorithms in details.

4.1 Rendering four layers

We implement the background layer in the following steps. First, set the canvas with a specific background color predefined by the user or generated randomly; Second, add shapes of different scales, e.g. rectangles circles and ellipses, filled with gradient colors close to the background color, positioned and sized randomly; Last, add a number of tiny points all over the canvas to generate an ash effect to the background, using

Algorithm 1: Add Ash Effect to Background

Input:
width: width of canvas.
height: height of canvas.
transparency: transparency for noise point.

```
xCoord = 0.0;     //x-coordinate in noise space
yCoord = 0.0;     //y-coordinate in noise space
inc = 0.04;       // increment of coordinate
for y = 1 to height {
  for x = 1 to width {
      float gray = noise(xCoord, yCoord) *255;
      //set color and transparency
      stroke(gray, transparency);
      //draw a point
      point(x, y);
      xCoord = xCoord + inc;
  }
  xCoord = 0;
  yCoord = yCoord + inc;
}
```

Algorithm 2: Draw irregular shape

Input:
centerX: x-pos of shape.
centerY: y-pos of shape.
smallArcNum: number of small arcs.
minRadius: minimum radius.
maxRadius: maximum radius.

```
//initialize center point
centerPoint = (centerX, centerY);
angle = 0;          //initizlize the angle
beginShape();      //begin to draw
for i = 1 to smallArcNum
  angle = 360 * i / smallArcNum;   //update angle
  //update xcoord, ycoord and radius.
  radius = random(minRadius, maxRadius);
  x = centerX + radius * cos(angle);
  y = centerY + radius * sin(angle);
  draw curve vertex (x, y);
end
endShape();
```

the function noise() in Processing. The algorithm is outlined in Algorithm 1, whose time complexity is $O(H \times W)$, where H and W denote respectively the height and width of the generated image.

As shown in Fig. 2, to implement an irregular shape, we first randomly generate a center point, then draw the outline composed of many small arcs. These small arcs have different radii in different directions, around the center point. To make the irregular shapes within different size ranges, we add a flag called mutant in our algorithm. When the flag is true, we reassign the minRadius and maxRadius parameters. The algorithm is outlined in Algorithm 2, whose time complexity is $O(N)$, where N is the number of small arcs around the center point (smallArcNum in the pseudocode).

Figure 3 shows that each line moves in a random direction and the weights of different sections also vary. The line-generation algorithm is as follows. First, we use mathematical functions to generate the locuses of the lines, as described before. Second, we divide the locuses into many small independent segments. Then we define the width, direction and color gradient of each small segment, so that different segments may have different characteristics. Last, we combine these small segments together to form the complete lines. To smooth all the connecting points, we assign gradual change in direction, color and weight to each small segments. The detailed algorithm is presented in Algorithm 3, whose time complexity is $O(N)$, where N is the number of small segments (smallLineNum in the pseudocode).

The approach for generating paint drops is based on the generation process of irregular shapes, as shown in Fig. 4. First, we randomly select a center point, then draw small arcs around the center point. The radii of the small arcs vary within a small range so that the drops look roundish and smooth. Noise is also added to the outlines of paint drops. The color of each paint drop is randomly determined as well. The algorithm for producing paint drops is similar to Algorithm 2 with the same time complexity. To distribute paint drops uniformly in the image, we first divide the image into grids of equal size (4×4 in the current implementation), and then locate paint drops in each grid independently. We call such

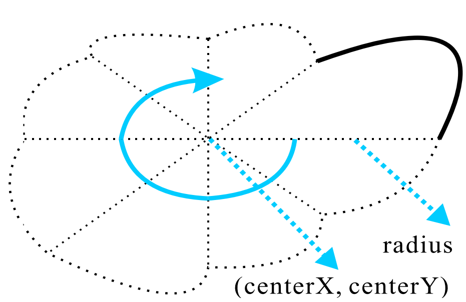


Fig. 2 Process of generating irregular shapes

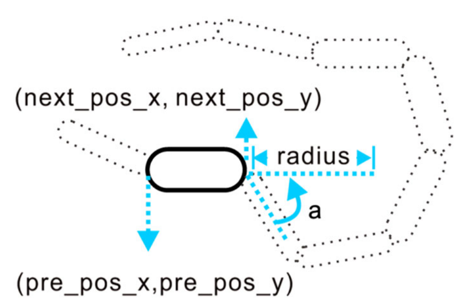


Fig. 3 Process of generating lines

Algorithm 3: Draw line

Input:
centerX: x-pos of start position.
centerY: y-pos of start position.
smallLineNum: number of small segments.
min_weight:maximum weight of segments.
max_weight:minimum weight of segments.
radian: radian of the line.

```
//initialize the start point of the line
pre_pos_x = centerX; pre_pos_y = centerY;
//initialize the angle for direction calculation.
a = random(0, 2π)
//draw each segment
for  i = 1 to smallLineNum
   //calculate position of next point
   next_pos_x = pre_pos_x + cos(a) * radius;
   next_pos_y = pre_pos_y + sin(a) * radius;
   //calculate weight of the next segment.
   strokeWeight = random(min_weight, max_weight);
   line(pre_pos_x, pre_pos_x, next_pos_x, next_pos_x);
   //update variables
   pre_pos_x = next_pos_x , pre_pos_y = next_pos_y;
   a = a + random(-radian, radian);
end
```

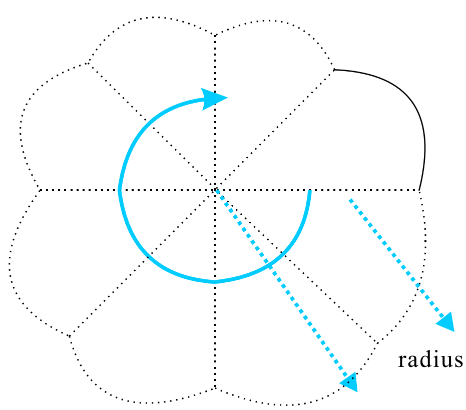


Fig. 4 Process of generating paint drops

a grid "Drop Zone", that includes a number of randomly distributed paint drops. The algorithm for generating drop zones is presented in Algorithm 4.

4.2 Size, quantity and color

By inspecting Pollock's artworks, we find that the sizes of various shapes vary greatly. In our implementation, shapes are generated in a range of random sizes. To avoid generating outliers, each type of basic shapes is limited in a predetermined range. In addition, we provide an interface for users to define the size of each shape. Fig. 5a shows the effect of size to the basic shapes.

Algorithm 4: Generate Drop Zones

Input:
centerX: x-pos of drop zone's center.
centerY: y-pos of drop zone's center.
radius: radius of drop zone.
insX: increment of △X.
insY: increment of △Y.
minNoise: minimum noise.
maxNoise: maximum noise.

```
//Calculate the dropNumber for each zone based on its
//area computed by function "dropNumInZone".
dropNum = dropNumInZone(centerX, centerY, radius);
//initialize variables.
noiseX = random(minNoise, maxNoise);
noiseY = random(minNoise, maxNoise);
△X = 0; △Y = 0;
for i = 1 to dropNum
   X_POS = (centerX + △X + noiseX);
   Y_POS = (centerY + △Y + noiseY);
   drawDrop(X_POS, Y_POS);
   //update intermediate variables
   △X = random(△X,  △X+ insX);
   noiseX = random(minNoise, maxNoise);
   △Y = random(△Y,  △Y+ insY);
   noiseY = random(minNoise, maxNoise);
end
```

Color is one of the key issues in generating Pollock styled images. We use HSB color scheme since it is closer to nature and easier to use than other schemes. There is a main color tone in Number 8. The colors of shapes are also randomly set, and usually with gradient. Shapes in each layer can be set in any color. To generate images resembling to the original, we restrict the color within a range dependent on the main color tone. We also provide an interface for users to set the color of each shape. The color effects in the basic shapes are shown in Fig. 5b.

The number of each type of basic shapes in a layer also varies greatly. We carefully analyzed the Pollock's paintings to estimate the number of objects in each layer. For example, we estimate that in the irregular shapes layer, there are only about 5–10 irregular shapes distributed above the background layer. In the line layer, there are about 200–300 lines, making up the main ingredients in Pollock's drip paintings. In the top layer, there are about 300–500 paint drops, more than irregular shapes, with the effects shown in Fig. 5c.

4.3 Layout

Another key issue in generating Pollock styled images is the layout, i.e. how the shapes in each layer are spatially distributed. By our analysis, there is no rule on how to distribute shapes in the image. To generate images resembling the original artwork, we adopt the stochastic strategy, such

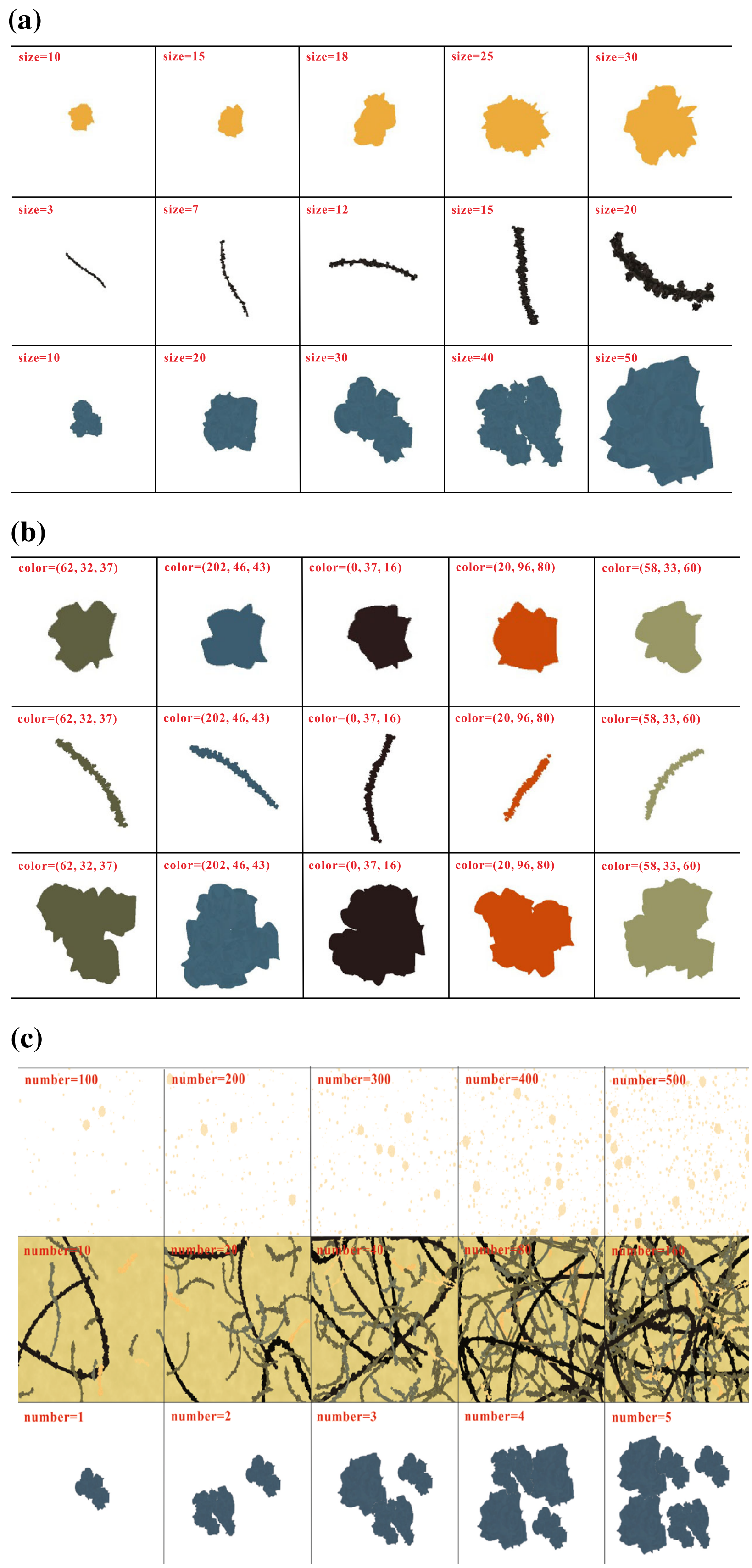


Fig. 5 Experimental Results. **a** Effects on various shapes of different sizes. **b** Effects on various shapes of different colors. **c** Effects on the layers with different quantities of objects

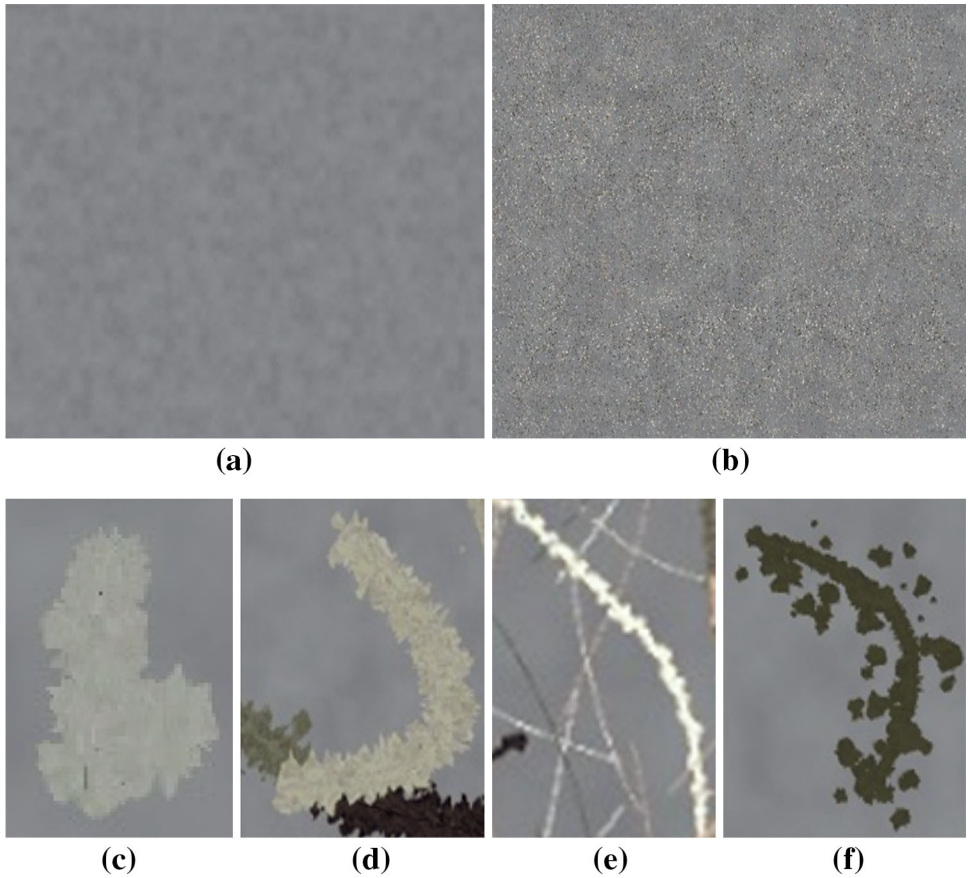


Fig. 6 Further Improvements on the basic objects. **a** Color variations added to the background. **b** Tiny points scattered on the background; **c** Color gradients added to blocks **d** Boundary noises added to lines. **e** Noise injected into line edges. **f** Ink dripped simulated

that the shapes are distributed almost randomly on canvas. Based on our observation, the layout of a layer also depends on the number of shapes in that layer. Too many shapes could possibly cover the entire image, but too few shapes may be scattered in part of the image. We therefore restrict the number of shapes in each layer, and divide the canvas into grids of equal size, and distribute the shapes randomly inside each grid.

4.4 Further improvement

To generate images more natural looking and better match Pollock's original paintings, we have performed further improvement to the background, irregular shapes, lines, and so on. As shown in Fig. 6a, we add color variation to the background, mimicking the effect of pouring paints on the canvas. Figure 6b shows the effect of many tiny points with noise scattered on the background. For the irregular shapes, we add more noise on the boundaries to simulate the penetration of paints into the canvas. We also add the color gradient onto irregular shapes to simulate varied strengths of the brush movement. The effect is shown in Fig. 6c and d.

Line is one of the key shapes in Pollock's drip paintings. The outlines of lines are injected with more noise to make it more realistic looking. We also add a few points around lines randomly, simulating the ink dripped on the canvas, as demonstrated in Fig. 6e and f.

The running speed of the algorithms for generating the three basic shapes is not affected by the scale and number of the shapes.

4.5 Top-level algorithm

The algorithm including all the above algorithms generates an image layer by layer from bottom to up. Each layer is independent from other layers. By combining all the layers together in Algorithm 5, we can generate the complete drip paintings. In the algorithm, all the parameters have their default values, e.g. the size of the generated image defaults to $1,024 \times 768$. Moreover, the user can define these parameters.

Algorithm 5: Auto-Generation of Drip Painting

1. Parameters setting;
2. Draw background layer;
3. Draw irregular shape layer;
4. Draw line layer;
5. Draw water drop layer;
6. Output the generated image;

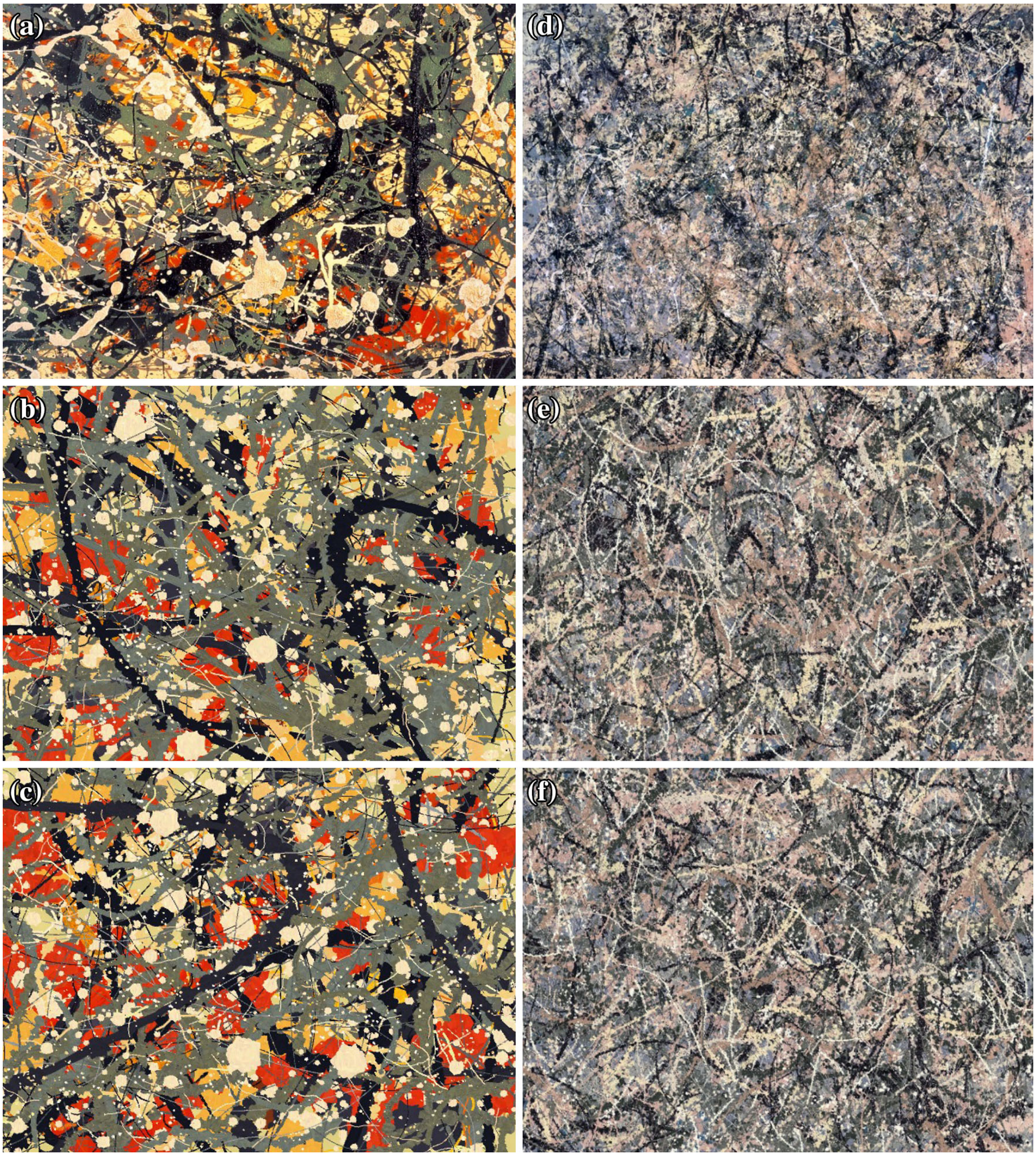


Fig. 7 Images mimicking two of Pollock's drip paintings. **a** Pollock's original "Number 8"; **b** and **c** Program generated images modeling "Number 8"; **d** Original "Number 1"; **e** and **f** Program generated images modeling "Number 1"

5 Experiments

We have used our approach to generate images mimicking two of Pollock's well-known drip paintings. One is the "Number 8" painted in 1948, and the other is the "Number 1" in 1950, as shown in Fig. 7a and d. "Number 8" is one of the earliest drip paintings, and has drawn much attention of researchers for understanding and analyzing Pollock's drip paintings in the last decades. Our idea of using layered modeling approach was also inspired by "Number 8". Basic shapes can be identified clearly in "Number 8". One can easily observe irregular shapes, lines and paint drops, based on which we can divide "Number 8" into the four layers as previously defined. We have used our algorithms to generate several "Number 8" style of images, as shown in Fig. 7b and c.

In the experiments, some of the parameters are defined by users, others are random. To make the generated images look more realistic, we set the colors of shapes the same as those of the original but add gradients. The quantity of shapes in each layer has been estimated empirically. The number of irregular

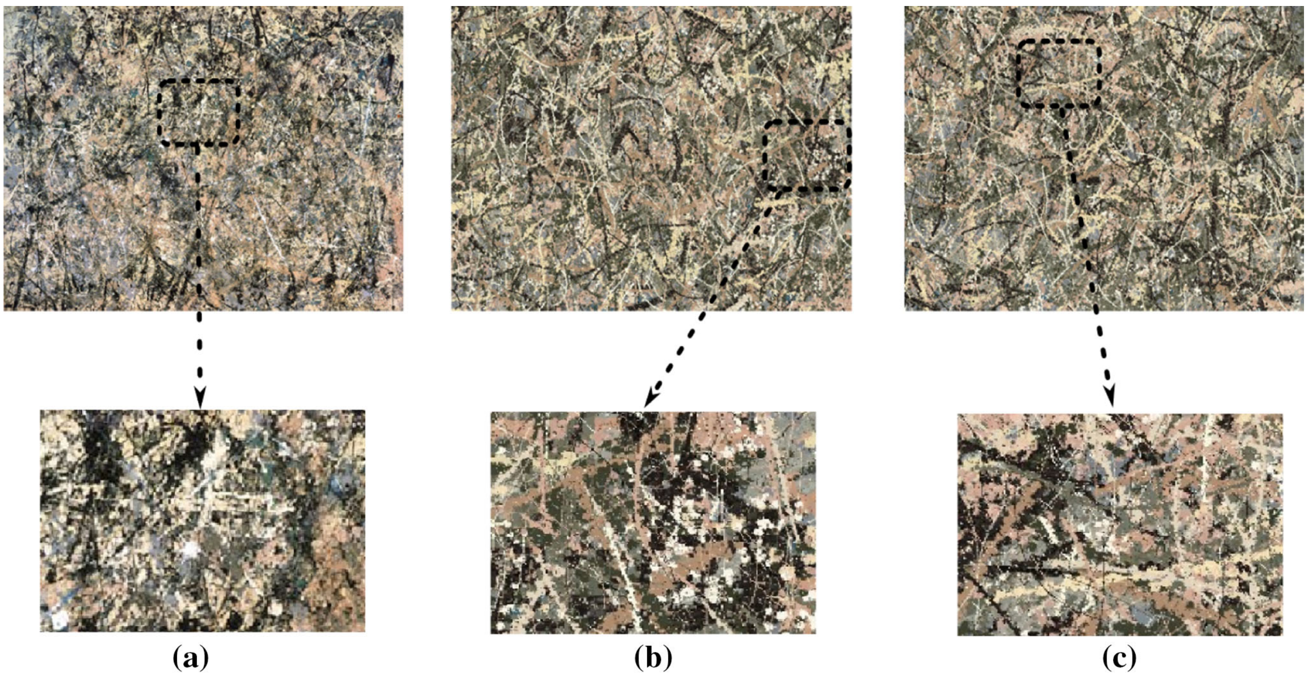


Fig. 8 Magnification of the computer generated and original images. **a** Magnified original Number 1; **b** and **c** Partially magnified images generated using our approach

shapes ranges from 5 to 10, the number of lines ranges from 200 to 300, and the number of water drops ranges from 300 to 500. The sizes of shapes are randomly generated in a range. In our implementation, the size of irregular shapes ranges from a quarter to a half of width/height of the generated image, and the sizes of points range from 3 to 5 (distances between pixels). The position of a shape is completely random. The generated images resemble the original one. The layered modeling approach is efficient in generating drip paintings. The parameters used in the algorithms are easy to adjust and thus suitable for modeling similarly styled paintings with drastically different color and visual compositions.

"Number 1" is another well-known artwork in the later period of Pollock, also known as "Lavender Mist". The layers in "Number 1" are not as clearly identifiable as in "Number 8". We could, however, also analyze "Number 1" using our layered modeling approach. Interestingly, there are more subtleties in "Number 1". For example, shapes in "Number 1" are more complex, especially lines, whose locuses and strokes vary greatly. The layout of "Number 1" is much more stochastic. The parameter setting is similar to that in the generation process of "Number 8", both combining the stochastic strategy and user-defined. Noises are also added to make the generated paintings to look more natural. The results are shown in Fig. 7e and f, demonstrating that our algorithm generates subtle details naturally and efficiently. Since "Number 1" (Lavender Mist) is a large painting, we make partially magnified comparisons between the original (Fig. 8a) and computer generated images using our approach (Fig. 8b, c). Figure 8 shows that low level details of the generated images are extremely similar to the original Pollock style.

We have conducted a simple empirical experiment, with 42 subjects, including 21 Masters students and 21 junior undergraduate students, all majoring in software engineering. By showing the subjects six images in one display, including a copy of the original "Number 8" and five generated images modeling "Number 8", we asked them to identify which image was the original within one minute; 12 out of 42 subjects were able to correctly identify. This simple experiment does not prove the validity of our approach but gives us tremendous encouragement and confidence for further work in this direction.

Our experimental platform is a PC, installed with Windows 7 64 bit, with an i7-CPU, Nvidia GT530-GPU, and a 8 G DDR3 memory. Table 1 reports the speed in generating different sized images, when the number of shapes is fixed. It shows that the time used to generate different sized images is almost the same. The only factor that affects the generation speed is the number of the shapes. The largest image of dripping style that we have generated is $8,192 \times 6,114$ pixels.

6 Conclusions

This paper has presented a layered modeling approach to generate Pollock dripping styled images. According to the analysis of fractal-based algorithms and observation of Pollock's painting, we divide the artworks into four indepen-

Table 1 Generation speeds and sizes of generated images

Size	Time
800 × 600	3 m 42 s
1,024 × 768	3 m 31 s
2,048 × 1,536	3 m 32 s
4,096 × 3,072	3 m 45 s

dent layers: background layer, irregular shape layer, line layer and water drop layer. Using simple mathematics and physics based algorithms to produce components for each layer, and we then combine all the layers together to generate the drip style of artworks.

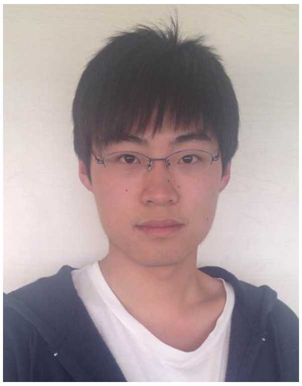
Our algorithms are customizable for either automatic or user-defined generation. Moreover, our algorithms can not only naturally layout the shapes in paintings, but also handle details effectively and efficiently. We have implemented the algorithms in Processing for generating Pollock styled images. Experiments show that our layered modeling approach provides an alternative and feasible approach to generating Pollock style.

As the future work, we will attempt to use our layered modeling approach to analyze different artists' abstract paintings. Getty Museum once used hyperspectral imaging to isolate colors and pigments in a major painting by Jackson Pollock. They found that Pollock used colors with a specific pattern [21]. Through detailed analysis of Pollock's six paintings [14], Mureika et al. speculate that Pollock may have deliberately "tuned" his paintings to contain colored visual structures, based on an intuitive understanding of the visual arts and aesthetics. Following this line of work, we also plan to perform systematic color analysis on the artist's paintings, such as color gradients and frequencies, which may help to generate abstract paintings in a more natural flavor.

References

1. Barnsley, M.F.: Fractals everywhere. Morgan Kaufmann, Massachusetts (1993)
2. Borgeat, L., Godin, G., Massicotte, P., Poirier, G., Blais, F., Beraldin, J.-A.: Visualizing and analyzing the Mona Lisa. IEEE Comput. Graph. Appl. **27**(6), 60–68 (2007)
3. Coddington, J., Elton, J., Rockmore, D., Wang, Y.: Multifractal analysis and authentication of Jackson Pollock's paintings. In: Proceedings of Computer image analysis in the study of art (2008)
4. Fogleman, M.: Procedurally generating images in the style of Piet Mondrian (2011)
5. Fairbanks, M., Mureika, J.R., Taylor, R.P.: Multifractal analysis of the painting techniques of adults and children. In: Proceedings of Electronic Imaging (SPIE'10), vol. 7531, pp. 1–6 (2010)
6. Haeberli, P.: Paint by numbers: abstract image representations. In: SIGGRAPH (1990)
7. Irfan, M., Stork, D.G.: Multiple visual features for the computer authentication of Jackson Pollock's drip paintings: beyond box-counting and fractals. In: Proceedings of IS&SPIE Electronic Imaging (2009)
8. Jensen, H.J.: Mathematics and painting. Interdiscip. Sci. Rev. **27**(1), 45–49 (2002)
9. Johnson, C.R., Hendriks, E., Berezhnoy, I.J., Brevdo, E., Hughes, S.M., Daubechies, I., Li, J., Postma, E., Wang, J.Z.: Image processing for artist identification. IEEE Signal Process. Mag. **25**(4), 37–48 (2008)
10. Kirsch, J.L., Kirsch, R.A.: The anatomy of painting style: description with computer rules. Leonardo **21**(4), 437–444 (1988)
11. Lyu, S., Rockmore, D., Farid, H.: A digital technique for art authentication. Proc. Natl. Acad. Sci. **101**(49), 17006–17010 (2004)
12. Mandelbrot, B.: The fractal geometry of nature. New Sci. **127**(1734), 38–43 (1990)
13. Marcos, A.F.: Digital art: when artistic and cultural muse merges with computer technology. IEEE Comput. Graph. Appl. **27**(5), 98–103 (2007)
14. Mureika, J.R.: Fractal dimensions in perceptual color space: a comparison study using Jackson Pollock's art. Chaos **15**, 043702 (2005)
15. Mureika, J.R., Cupchik, G.C., Dyer, C.C.: Multifractal Fingerprints in the Visual Arts. Leonardo **37**(1), 53–56 (2004)
16. Mureika, J.R., Dyer, C.C.: G. C Cupchik, "On Multifractal Structure in Non-Representational Art". Phys. Rev. E **72**, 046101 (2005)
17. Mureika, J.R., Taylor, R.P.: The abstract expressionists and Les Automatistes: a shared multi-fractal depth? IEEE Signal Process. **93**(3), 573–578 (2012)
18. Noll, A.M.: Human or machine: a subjective comparison of Piet Mondrian's 'composition with lines' and a computer-generated picture. Psychol. Rec. **16**(1), 1–10 (1966)
19. Reas, C., Fry, B.: A programming handbook for visual designers and artists. MIT Press, Massachusetts (2007)
20. Redies, C., Hasenstein, J., Denzler, J.: Fractal-like image statistics in visual art: similarity to natural scenes. Spat. Vis. **21**(2), 137–148 (2007)
21. Restoring Jackson Pollock's 'Mral': http://online.wsj.com/news/articles/SB10001424127887323585604579008323626960360
22. Singh, G.: Stringing the fractals. IEEE Comput. Graph. Appl. **25**(5), 4–5 (2005)
23. Singh, G.: Fractal fracas. IEEE Comput. Graph. Appl. **25**(3), 4–5 (2005)
24. Singh, G.: Fun with fractal art. IEEE Comput. Graph. Appl. **29**(1), 4–5 (2009)
25. Steven, N., Smith, G.W.: Jackson Pollock: an American saga. CN Potter, New York (1989)
26. Taylor, R.P.: Order in Pollock's chaos. Sci. Am. **287**, 116–121 (2002)
27. Taylor, R.P.: Personal reflections on Jackson Pollock's fractal paintings. História Ciências Saúde-Manguinhos **13**, 109–123 (2006)
28. Taylor, R., Spehar, B., Hagerhall, C., Van Donkelaar, P.: Perceptual and physiological responses to Jackson Pollock's fractals. Front. Hum. Neurosci. **5**, 60 (2011). doi:10.3389/fnhum.2011.00060
29. Taylor, R.P., Guzman, R., Martin, T.P., Hall, G.D.R., Micolish, A.P., Jonas, D., Scannell, B.C., Fairbanks, M.S., Marlow, C.A.: Authenticating Pollock paintings using fractal geometry. Pattern Recognit. Lett. **28**(6), 695–702 (2007)
30. Taylor, R.P., Micolich, A.P., Jonas, D.: Fractal analysis of Pollock's drip paintings. Nature **399**(6735), 422–422 (1999)
31. Taylor, R.P., Micolich, A.P., Jonas, D.: Fractal expressionism. Phys. World **12**(10), 25–28 (1999)
32. Taylor, R.P., Micolich, A.P., Jonas, D.: The construction of Jackson Pollock's fractal drip paintings. Leonardo **35**(2), 203–207 (2002)
33. Zhang, K.: From abstract painting to information visualization. IEEE Comput. Graph. Appl. **27**(3), 12–16 (2007)
34. Zhang, K., Harrell, S., Ji, X.: Computational aesthetics—on complexity of computer-generated Paintings. Leonardo, vol 45, no. 3. MIT Press, pp. 243–248 (2012)

35. Zhang, K., Yu, J.H.: Generation of Kandinsky art. Leonardo. MIT Press (2014) (accepted to appear)
36. Zhao, M., Zhu, S.-C.: Sisley the abstract painter. In: Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering, pp. 99–107. ACM, New York (2010)

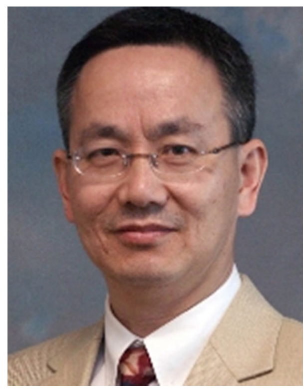


Yan Zheng received the B.S. degree from School of Computer Software, Tianjin University, Tianjin, China, in 2012. He is currently pursuing the M.S. degree in the School of Computer Software in Tianjin University. His major research interest is media computing, distributed computation, natural language processing and data mining.

Xuecheng Nie received the B.S. degree from School of Computer Software, Tianjin University, Tianjin, China, in 2012. He is currently pursuing the M.S. degree in the School of Computer Software, Tianjin University. His research interests include computer vision, such as image segmentation, pattern recognition, and natural language processing, such as story segmentation, and semantic similarity learning.

Zhaopeng Meng is Professor of Software Engineering and Dean of Software College, Tianjin University, and is also a member of Software Engineering Steering Committee, Ministry Of Education. He is also a leading director of Tianjin Academy of Computers. Dr. Meng had hosted and participated in more than 20 research projects including National Natural Science Fund, Major National Science and Technology and Economic and Information Technology Commission of Tianjin. He had published more than 40 journal papers and conference papers, and also 5 patents as the first inventor. His current research interests are in the Internet Of Things, Intelligence Transport System and Distributed Multimedia.

Wei Feng received the B.S. and M.Phil. degrees in Computer Science from Northwestern Polytechnical University, China, in 2000 and 2003 respectively, and the Ph.D. degree in Computer Science from City University of Hong Kong in 2008. From 2008 to 2010, he worked as research fellow at the Chinese University of Hong Kong and City University of Hong Kong, respectively. He is currently an associate professor in School of Computer Science and Technology, Tianjin University. His major research interest is media computing, specifically including general Markov Random Fields modeling, discrete/continuous energy minimization, image segmentation, semi-supervised clustering, structural authentication, and generic pattern recognition. He got the support of the Program for New Century Excellent Talents in University, China, in 2011.



Kang Zhang is Professor and the Director of Visual Computing Lab, Department of Computer Science, University of Texas at Dallas, Richardson, Texas, USA, and an Adjunct Professor at the School of Software Engineering, Tianjin University, China. His research interests include visual languages, aesthetic computing and information visualization, and software engineering. He has authored and edited six books, and published over 180 papers in journals and conference proceedings.