

# Generating Abstract Paintings in Kandinsky Style

Kang Zhang<sup>1</sup>  
Department of Computer Science  
University of Texas at Dallas  
Richardson, TX 75083-0688, USA

Jinhui Yu<sup>2</sup>  
State Key Laboratory of CAD & CG  
Zhejiang University  
Hangzhou, 310027, P. R. China

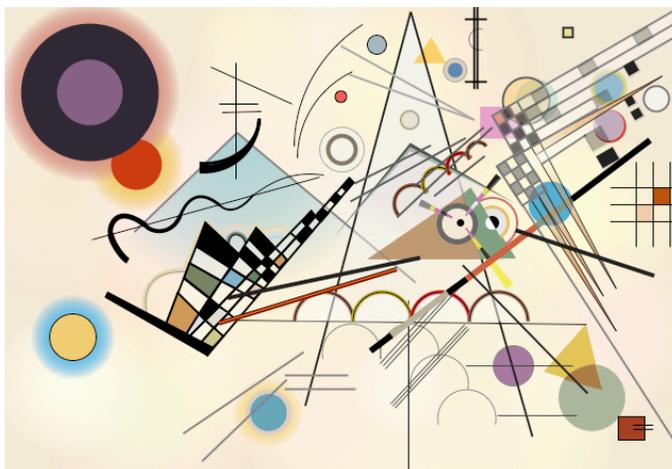


Figure 1: Program generated *Composition VIII*

## Abstract

This paper presents a recent project on automatic generation of Kandinsky style of abstract paintings using the programming language Processing. It first offers an analysis of Kandinsky's paintings based on his art theories and the author's own understanding and observation. The generation process is described in details and sample generated images styled on four of Kandinsky's paintings are also demonstrated and discussed. Our approach is highly scalable, limited only by the memory space set in Processing. Using random generation, every styled image generated can be unique. A selection of the images generated in the required resolution is also submitted and 70 images are made into a video companion.

**CR Categories:** J.5 ARTS AND HUMANITIES – Fine Arts

**Keywords:** abstract painting, processing, generative art

<sup>1</sup>Email: [kzhang@utdallas.edu](mailto:kzhang@utdallas.edu)

<sup>2</sup>Email: [jhyu@cad.zju.edu.cn](mailto:jhyu@cad.zju.edu.cn)

## 1. Introduction

As modern computer graphics and printing technology become increasingly advanced, automatic generation of sophisticated abstract paintings becomes possible. One of the most successful computer generated art forms has been the fractal art [Barnsley 1993]. According to a recently proposed 4-level classification in terms of computational power being utilized [Zhang *et al.* 2012], fractal art is considered Level 2. The classification measures computer generated abstract paintings by their computational complexity, i.e. to what extent machine intelligence is utilized in generating the paintings, rather than by the visual complexity as in Taylor's analysis [2004]. Limited attempts have been made at Level 3, i.e. using heuristic rules and patterns to encode an artist's styles and then generating artist-specific styles of paintings, as summarized in the next section.

This paper presents a work on a scalable and parameterized approach to automatic generation of Kandinsky style of abstract paintings (Figure 1) using the Processing language [Reas and Fry 2007]. Processing is a programming language built on top of the Java language, designed for artists and designers with the purpose of teaching the fundamentals of computer programming in a visual context.

The rest of this paper is organized as follows. Section 2 reviews prior work in computer generated abstract paintings. Section 3 discusses Kandinsky's styles associated with his art theory and

practices, and analyzes his styles and patterns. Section 4 describes our generation process and steps. Section 5 presents a concrete example with generation details, followed by Section 6 on the scalability issue and significance. Section 7 concludes the paper.

## 2. Related Work

Artificial intelligence researchers have long been making creative computers for art generation, such as AARON, (<http://www.kurzweilcyberart.com/>). On the other hand, computer graphics researchers and digital artists have used the computer to generate abstract art works based on fractals [Ammeraal and Zhang 2007; Barnsley 1993]. There has, however, been limited effort on automatic generation of abstract paintings based on specific artists' styles. The notable example is that of Taylor [2002], who used of fractals to model and analyze Jackson Pollock's dripping style of paintings and generated remarkable results. There have also been a few attempts in automatically generating Mondrian-style of abstract paintings [Fogleman 2011], which could trace back to the pioneering work of Noll [1966].

Kirsch and Kirsch [1988] first proposed to model Miro's style by storing typical Miro shapes into a database and then manually analyze the target composition using the stored shapes. They had not encoded the analysis any grammar or program but offered some preliminary discussion.

Transforming existing images to abstract paintings using the non-photorealistic rendering (NPR) techniques have been attempted for a couple of decades. The techniques include mimicking brush strokes and texture patterns [Haerberli 1990; Zhao and Zhu 2010]. The composition and style of the original image determine very much the composition and style of the resulting painting.

Barnett and Barnett [1989] analyzed Kandinsky's compositions, while Price [2007] applied Kandinsky's aesthetics to Java programming.

Zhang *et al* [2012] made an initial attempt in automatic generation of Kandinsky style of abstraction paintings, that was briefly reported in the aforementioned classification. That preliminary work included very few styled patterns and generated uniform backgrounds and no sophisticated combination of geometric objects. This paper reports the results of an in-depth and comprehensive experiment on generating several styles of Kandinsky's paintings.

## 3. Kandinsky Style

Wassily Kandinsky is widely considered the founder of abstract painting and has influenced generations of abstract artists. His styles went through several stages of changes, from representational to non-representational in his early career; then from non-objective to geometry, and finally to what he called "biomorphic" abstraction. The authors of this paper are particularly fond of the geometry works during his Bauhaus years from 1922 to 1933. It was also during this period when Kandinsky was interested in geometrical forms and their composition and interaction, and also published his well-known theoretical art book [Kandinsky 1926]. He emphasized the

importance of form, color, rhythm and the artist's inner need. The most representative work during the Bauhaus years is *Composition VIII* [Kandinsky 1923], illustrated in Figure 1, which is a computer-generated version using Processing. *Composition VIII* is regarded by the artist himself as one of the most important works of his post-war years and marks his first systematic application of his own theory on the interactions of colors and forms [Backs-Malorny 1994].

We will call typical structures composed of geometrical forms as *styled patterns*. The process starts with the observation and analysis, then the construction of individual styled and aesthetic patterns. It finally assembles the patterns in a pseudo-random fashion into a structure until harmony is achieved.

### 3.1 Style Analysis

Kandinsky [1926] interpreted the three primary forms as being constructed of angular lines:

- Acute angles lead to a triangle and have the yellow color within;
- Right angles lead to a square that is of a plan-like nature and parallel to the red color; and
- The obtuse angle is passive and related to the light blue tone.

According to Kandinsky [1914], the effect of yellow is emphasized well when combined with a sharp form such as a triangle. In contrast, the effect of a deeper color (e.g. blue) is reinforced by a rounded form such as a circle [Becks-Malorny 1994].

Establishing theoretical relationships between lines, planes and colors, he also proposed three primary contrasting pairs of elements:

- Straight line vs. curved line;
- Triangle vs. circle; and
- Yellow vs. blue.

Most of his paintings during the Bauhaus years are composed of various forms of these three pairs of elements and also the interaction between the pairs. Forms "grow outwards from many points to envelop the whole (in a manner similar to the way new skin grows over a wound)" [Barnett and Barnett 1989] rather than being developed from a single center. Bearing the above theories in mind, we have made the following observations from some of Kandinsky's paintings during the Bauhaus years:

- Semi-circles are mostly upward.
- Equilateral triangles are upward with only two sides and open bottom.
- Horizontal and vertical lines (thin) set the base, augmented with angled lines.
- Overlapped forms reflect the overlapping using different colors.
- Dominating full circles are often surrounded by gradual changing shades of contrasting colors, mostly yellow and blue.
- Semi-grid forms with interleaving color grids, grid lines parallel, forming an angle, or meeting at the end making sharp triangles.
- Non-circle curve lines are rare, but when used, in dominant and contrast positions.
- Dark boundaries, with light colors for filling, except black which is occasionally used to fill some forms to emphasize.
- Red combined with black highlights the entire space.
- Background usually with non-uniform colors.

- Typical neighboring colors: green-blue, red-yellow-black, and red-blue.

### 3.2 Styled Patterns

Based on the above analysis, we have encoded 32 styled patterns. Using and Kandinsky's theoretical relationships between lines, planes and colors, we categorize these styled patterns into the following three groups:

1. **Straight line vs. curved line:** one or more straight lines intersecting with a curved line, and individual lines and curves. Some lines are filled with segmented colors.
2. **Triangle vs. circle:** one or more triangles overlapping on a circle or concentric circles, one or more half circles in a row, and individual filled triangles and circles. Some triangles are filled with segmented colors.
3. **Yellow vs. blue:** circles filled fill with yellow or blue, having an outer circle in blue or yellow gradually fading outward. Straight lines intersect at the right angles to form one or more filled rectangles.

When the randomly overlapping effect needs to be emphasized, we use colors with transparency to fill the overlapping objects by setting their gradient levels. When pre-determined overlapping needs to be demonstrated, we simply hardcode the overlapping area filled with a color different from the involved objects.

### 4. Generation Process

The generation process starts with the analysis and modeling of a small set of Kandinsky paintings, including *Composition VIII* (1923, Figure 1), *Black and Violet* (1923), *On White II* (1923), *Several Circles* (1926), that are all from the Bauhaus period.

*Black and Violet* is characterized by groups of lines and triangles and their interactions with curved forms. Its non-uniform dark background colors are harmoniously embedded with similarly color-shaded foreground objects. *One White II* has a notable characteristic of two large intersecting black diagonals, with various relatively large triangles and quads within the central area. The whitish background provides a natural contrast with the foreground dark and bright-colored objects. *Several Circles* emphasizes the importance of circles in their dominant role, culminating in a cosmic and harmonious image.

Except *Composition VIII* in Figure 1 that was generated with hard-coding of colors, sizes, and positions of all the objects according to the original, all other images (e.g. Figure 4) were fully automatically generated by randomizing all the attributes (e.g. size, color, location, etc), as described in the remaining part of this section.

#### 4.1 Steps

As depicted in Figure 2, the generation process consists of the following major steps:

1. The setup step prepares all the necessary variables or parameters, their sizes and ranges;
2. Drawing of the background colors based on the painting characteristics;
3. Selection, parameterization and rendering of styled patterns pre-coded (described in Section 3.2), which is the key step in the generation process; and

4. When no more patterns to render, the program saves the entire image rendered on the canvas.

The following subsections describe these steps in more details.

#### 4.2 Setup and Background Generation

The setup step includes the definition of canvas size, color ranges, variables and their possible ranges, and setting of their initial values; and generation of a pseudorandom set of coordinates on the canvas and a pseudorandom set of colors.

We divide the entire screen into two regions: the central region and peripheral region surrounding the central region. This simple division serves two purposes. First, some objects such as groups of lines are typically located in the peripheral region while others such as large triangles and quads located in the central region. Second, some Kandinsky styles, such as *Yellow, Red, Blue* (1925), have slightly lighter background colors in the central region than in the peripheral region.

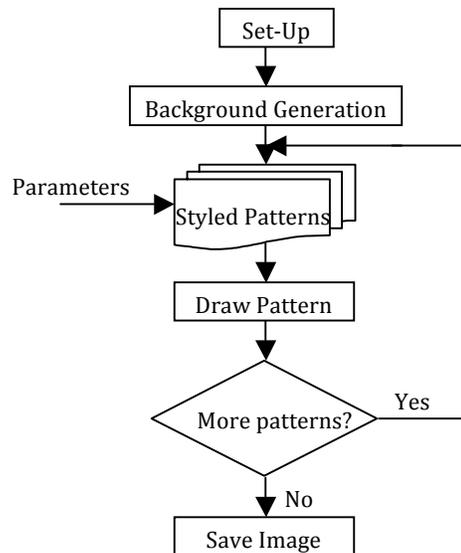


Figure 2: Steps of automatic generation

A set of X and Y coordinates within a pre-defined margin are randomly selected from the above two regions. Random sets of other parameters, such as length and radius, are also generated within predefined ranges during the setup phase.

Typical Kandinsky paintings have a non-uniform background color mix, dominated by one color shade. The background color shade at a specific location is sometimes influenced by the foreground object shape at that location. Based on this observation, we first fill the canvas background with a uniform color, that is randomly determined within a pre-defined range according to the painting style. Then gradient color shades within a given range of variations from the uniform color are applied to different locations of the canvas. The peripheral region may be filled with slightly darker shades if necessary (for paintings such as *Yellow, Red, Blue* (1925)).

### 4.3 Parameterization and Randomization

Using Kandinsky’s terminology, the canvas on which abstract paintings are drawn is considered a basic plane (BP). The four sides of the BP and the interpretation of their tensions of movement and “literary” are illustrated in Table 1.

Table 1: Four boundaries of the basic plane (BP)

Side	Tension	“Literary”
Above	Toward	Heaven
Left	Into	The distance
Right	Toward	Home
Below	Toward	The Earth

Guided by the artist’s interpretation of the four boundaries, the division of the BP into the central and peripheral regions facilitates the positioning of the styled patterns. Each pattern can be parameterized by its BP position, rotation angle, size, fill color, with or without boundary and boundary color. There are also pattern-specific parameters.

Figure 3 illustrates two example patterns that are parameterized with random numbers with their predetermined ranges. Figure 3(a) shows a styled pattern formed by striped arcs, that includes parameters of the position on the BP, radius of the arcs, rotation angle. The double circles in Figure 3(b) includes many more parameters such as radius, fill-color, border color, and border thickness of each of the inner and outer circles, whether the outer circle has a gradient and fading color, as well as the center position of the circles on the BP. In general, the more parameters are defined, the more variations and fine-tuned a pattern can be generated with and thus potentially more resembling to the Kandinsky style.

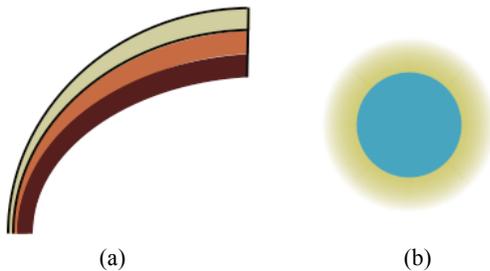


Figure 3: Two example patterns

The aesthetic appeal of patterns and randomness in Kandinsky’s paintings was described by Schlegel as “structured, artistic chaos”: the arabesque [Thompson 1993]. We therefore used pseudorandom process to generate the aforementioned parameters, each within a desirable range that was obtained through experiments.

## 5. A Generation Example

Let us now use *Composition VIII* as an example to demonstrate the details of the generation process. As mentioned earlier, the bottom image in Figure 1 was generated by hard-coding the

graphical objects, their sizes, colors, orientations, and exact positions to mimic the original painting. The canvas is divided into 50 small areas. The entire canvas background is first filled with the uniform color of (R, G, B) = (241, 226, 192). Each small area is drawn with a circle that is randomly sized between 120 to 250 pixels and filled with a color of a small variation from the above RGB values, gradually fading outward. The area size and circle size are chosen so that the circles have enough overlaps with to yield smooth yet non-uniform color variations. The iterative process of fine-tuning of all these parameters and random ranges, and then comparing the results with the original painting is indeed time-consuming and tedious. This process is necessary to ensure that the program-generated image could be close enough to the original painting. Then it is rewarding to see the images generated of the same style after setting all the above parameters within some carefully tuned random ranges.

Figure 4(a) is one such image picked from hundreds of automatically and uniquely generated images following the style of *Composition VIII*. The background is first filled with the uniform color, set by randomly generating the color code of (R, G, B) within the ranges of ([221, 241], [205, 225], [162, 192]). Each of the 50 small areas is drawn with a vary-sized circle in a similar fashion as for Figure 1(bottom) with a gradient color of a small variation from the above uniform RGB values. For the background of *Composition VIII* style, we do not differentiate central and peripheral regions.

The generation program then selects zero to five different variations of each styled pattern. The variations are realized by parameterization and randomization as described in Section 4.3. The styled patterns are parameterized according to the artist’s own interpretation described in Section 3.1. For example, some patterns such as the large black curve (top-left in Figure 4(a), implemented by connecting eight cubic Bézier curves) are set with the rotation angles and sizes corresponding to their positions facing the four boundaries. Other patterns such as open-bottom equilateral triangles (gradient light green and gradient light blue patterns in the central area) are positioned upward and grown out of the “earth” like mountains. As another example, a row of half-circles may be defined by the number of such half-circles and increasing or decreasing in size in the row, in addition to the aforementioned common parameters.

The entire generated image is finally saved as an image file, whose size is determined by the canvas size set in the set-up phase. The limitation on the size, i.e. the scalability, is discussed in the next Section.

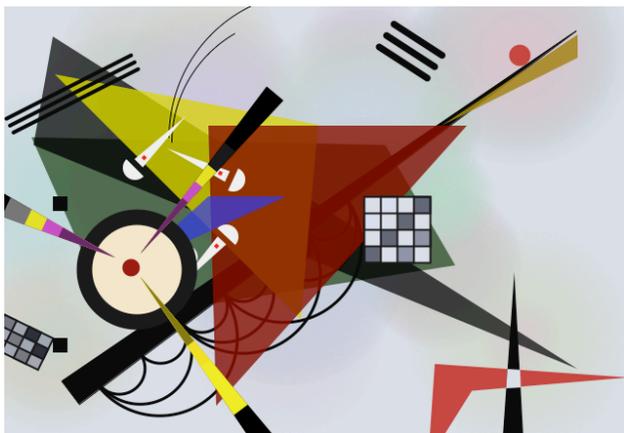
The same generation program is used to generate the other images in Figure 4 (b, c, and d) and in the submitted video companion, that model Kandinsky’s three other different paintings. Their background color mixes and styled patterns are different from those for modeling *Composition VIII*.



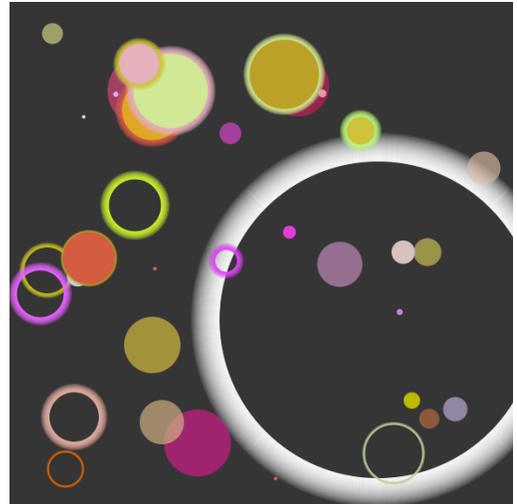
(a) Modeling *Composition VIII*



(b) Modeling *Black and Violet*



(c) Modeling *On White II*



(d) Modeling *Several Circles*

Figure 4: Samples of automatically generated four series of images modeling (a) *Composition VIII*, (b) *Black and Violet*, (c), *On White II*, and (d) *Several Circles*

## 6. Significance and Scalability

Although the patterns encoded are all styled on Kandinsky's paintings during his Bauhaus years, our approach is equally applicable to generating other styles of paintings. It demonstrates the possible impact of computer-simulated human aesthetics. With more details and styled patterns added, and aesthetic rules introduced, programmed generative processes could mimic or reproduce many types of art. Our approach has also a great potential in its applications. For example, Kandinsky's paintings expose strong inner dynamics of geometrical objects. The ability of automatic generation of his style of abstract paintings would apparently enable abstract representation and animation of dynamic systems, such as music and human movement [Ione 2004; Snibbe and Levin 2000].

The program is highly scalable, not only with the size of the generated image, using Processing's "scale" statement, but also with the number and size of any individual styled pattern, set by a set of parameters as described in Section 4.3. The only limitation to the image size is the machine's memory space used by the Processing language. Using an Apple iMac with 4 GB memory, Intel Core 2 Duo (1 processor) at 3.06 GHz, while setting Processing language required memory as 1 GB, we have been able to generate the images of  $10880 \times 7514$  pixels. This means that such an image requires a storage space of 245.3 MB. The generation process for an image of this size takes about 4-5 minutes time, while the program has not been optimized for either speed or memory concerns.

## 7. Conclusion

This paper has introduced a recent project in automatic generation of Kandinsky style of abstract paintings. The approach used is scalable and generic in generating pseudorandom images resembling the artist's style. As with any automated approach, our approach has its limitations in generating detailed artistic

styles that match the artist's interpretation. Automatically generating non-geometrical shapes resembling the artist's style is much harder but possible, requiring much more detailed modeling and encoding.

There are almost endless opportunities to further refine and extend the work, and apply it to commercial and design arenas. An immediate future extension would be to include other styled forms, and to model Kandinsky's paintings in other periods (with non-geometric shapes) and other abstract artists' styles. The approach could also be extended into the graphic design field by encoding design rules and generating desired designs. Such an approach would assist and extend designers' creativity through machine's intelligence and automation. The analysis of Kandinsky aesthetics would certainly benefit the future design of information visualization [Zhang 2007].

## 8. Acknowledgment

This work is partially supported by the State Key Program of National Natural Science Foundation of China (No. 60933007).

## 9. References

- [1] L. Ammeraal and K. Zhang, *Computer Graphics for Java Programmers*, Second Edition, John-Wiley & Sons, 2007.
- [2] M.F. Barnsley, *Fractals Everywhere*, Morgan Kaufmann, 1993.
- [3] V.E. Barnett and P.H. Barnett, The Originality of Kandinsky's Compositions, *The Visual Computer*, Vol.5, No.4, July, 1989, 203-213.
- [4] U. Becks-Malorny, *Wassily Kandinsky – The Journey to Abstraction*, Benedikt Taschen, 1994.
- [5] M. Fogleman, Procedurally Generating Images in the Style of Piet Mondrian, <http://fogleman.tumblr.com/post/11959143268/procedurally-generating-images-in-the-style-of-piet>, 2011.
- [6] P. Haeberli, Paint by Numbers: Abstract Image Representations, *SIGGRAPH'90*, Dallas, 6-10 August 1990.
- [7] A. Ione, Kandinsky and Klee: Chromatic Chords, Polyphonic Painting and Synesthesia, *Journal of Consciousness Studies*, Vol.11, No. 3-4, 2004, 148-58.
- [8] W. Kandinsky, *Composición VIII*. 1923. Solomon R. Guggenheim Museum, New York.
- [9] W. Kandinsky, *Concerning the Spiritual in Art*, (originally published in 1914) translated with an introduction by M.T.H. Sadler, Dover Publications Inc., New York, 1977.
- [10] W. Kandinsky, *Point and Line to Plane*, (originally published in 1926) translated by H. Dearstyne and H. Rebay, Dover Publications Inc., New York, 1979.
- [11] J.L. Kirsch and R.A. Kirsch, The Anatomy of Painting Style: Description with Computer Rules, *Leonardo*, Vol. 21, No. 4, 1988, 437-444.
- [12] A.M. Noll, Human or Machine: A Subjective Comparison of Piet Mondrian's 'Composition with Lines' and a Computer-Generated Picture, *The Psychological Record*, Vol. 16. No. 1, January 1966, 1-10.
- [13] C.B. Price, From Kandinsky to Java (The Use of 20th Century Abstract Art in Learning Programming), *ITALICS*, Vol.6, No.4, October 2007, 35-50.
- [14] C. Reas and B. Fry, *Processing: A Programming Handbook for Visual Designers and Artists*, 2007, The MIT Press.
- [15] S.S. Snibbe and G. Levin, Interactive Dynamic Abstraction, *Proc.1st Int. Symp. on Non-photorealistic Animation and Rendering (NPAR'00)*, Annecy France, 2000, 21-29.
- [16] R.P. Taylor, Order in Pollock's Chaos, *Scientific American*, December 2002, 116-121.
- [17] R.P. Taylor, A.P. Micolich, and D. Jones, The Construction of Pollock's Fractal Drip Paintings, *Leonardo* Vol. 35, 2002, 203.
- [18] R.P. Taylor, Pollock, Mondrian and Nature: Recent Scientific Investigation, *Chaos and Complexity Letters* 1, 2004, 29.
- [19] G.R. Thompson, *The Art of Authorial Presence: Hawthorne's Provincial Tales*, Duke University Press, 1993, p.21.
- [20] K. Zhang, From Abstract Painting to Information Visualization, *IEEE Computer Graphics and Applications*, May/June 2007, 12-16.
- [21] K. Zhang, S. Harrell, and X. Ji, Computational Aesthetics - On Complexity of Computer-Generated Paintings, *Leonardo Journal*, MIT Press, June 2012, Vol. 45, No. 3, 2012, 243-248.
- [22] M. Zhao and S-C. Zhu, Sisley the Abstract Painter, *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering (NPAR'2010)*, Annecy, France, 2010, 99-107.