The Painting Camera: Abstract Rendering with Camera-Control-Images

ERGUN AKLEMAN AND SCOTT MEADOWS [*]
Visualization Laboratory
College of Architecture

Texas A&M University

Abstract

In this paper, we present a simple and intuitive rendering technique to create abstract paintings. This technique allows the users to render a given animated scene with multiple cameras, using a method inspired by concepts used by cubist and futurist abstract painters. In our method, the number of cameras is unlimited and the camera parameters are continuous functions. We define the camera parameters as parametric functions that are described by images, which we call *camera-control-images*. The spatial distribution color in these camera-control-images determines the camera parameters such as position and orientation.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.6 [Computer Graphics]: Methodology and Techniques

Keywords: Non-photorealistic rendering, Abstract rendering

1 Introduction

The 20th century has seen a large number of abstract art movements such as Cubism , Futurism [5], Surrealism, Dadaism Cercle et Carre and Abstraction-Creating Suprematism [6] and artists such as Picasso, Braque, Leger, Delauney, Duchamp, Dali, Max Ernst, Kupka, Klee, Kandisky, Mondrian, Malevich, Marinetti, Boccioni, Russolo, Marey, Balla, Severini and Carra [3]. The most important common theme among these art movements is that they view painting in a way similar to music: an abstract entity to be enjoyed by using one of our senses. Some of these movements such as suprematism even claimed that to be really original there should not be constant recourse to reality.

In computer graphics, although there has been an increasing interest in artistic (or non-photorealistic) approaches, only a few abstract rendering approaches [8] have been developed to create abstract paintings.

In this paper, we present a simple and intuitive abstract rendering approach for creating abstract paintings. Our abstract rendering approach is inspired by the abstract art movements we mentioned above. We observed that several of these movements especially cubism and futurism can be best grasped by viewing abstract painters as high dimensional beings. These high dimensional painters can simultaneously view any animated 3D scene from every possible time, from all points of view, using any conceivable camera type. Since theses abstract painters have this supernatural ability, they can choose to paint at each point of the canvas what they see from any specific choice of time, point of view and camera. It is possible to represent this painting process as a parametric function from the canvas space to the space of camera parameters including position and time. We call this parametric function the *camera function*.

In order to implement such an abstract rendering approach in a simple and intuitive way, we describe camera functions by using sets of images, since painters are comfortable in dealing with images. These images can be used as a control net for tensor product B-splines. We call the images that are used for representing camera functions *camera-control-images*. In a camera-control-image, the values of red, green, and blue color channels provide information about camera parameters.

The number of camera-control-images to be used depends on the number of camera parameters. To simplify the presentation for this paper, we have considered here only the position of the camera. Based on this choice, we only need one camera-control-image to describe camera parameters (the position of a camera requires only 3 real numbers). To render the abstract images, we use a simple ray-tracer. Again for the purpose of simplicity, the 3D scenes used in this paper include only texture mapped spheres and planes. Figure 1 shows an abstract painting created using our approach.

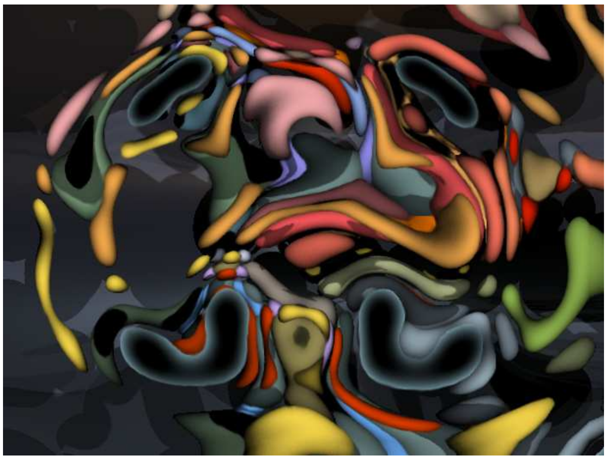

Figure 1: An abstract portrait of Mr. Bean.

Let us consider how artists use the camera-control-images to create abstract paintings from 3D scenes. Given a 3D scene, artists need first experiment with the effects of using single color camera-control-images. After repeated experiments, they eventually develop an intuition of how to control the images. Using this intuition, artists can create interesting abstract paintings by incrementally changing camera-control-images.

We have also investigated another artistic method for creating abstract paintings by incrementally building 3D scenes for a given camera-control-image. This method is especially convenient by using existing images such as paintings and photographs.

Artistic rendering, or as it is commonly known as non-photorealistic rendering, has been becoming an increasingly important subfield of computer graphics. Most non-photorealistic render-

[*] Address: 216 Langford Center, College Station, Texas 77843-3137. email: ergun@viz.tamu.edu. Supported in part by the Texas A&M, Scholarly & Creative Activities Program.

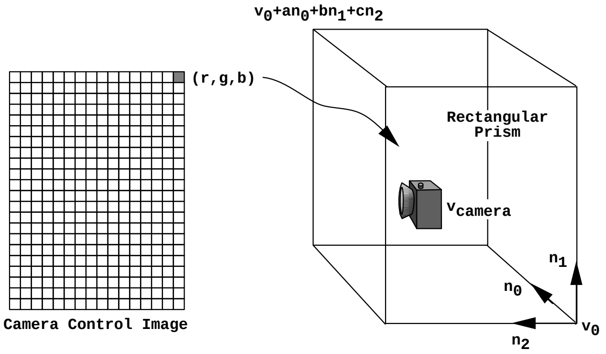

Figure 2: Rectangular prism that defines the camera position in 3D.

ing approaches try to formalize the effects of material such as pen and ink drawing [9], watercolor [2] or mimicking artistic effects such as impressionist [7] or fur, grass and trees [4].

Approaches to doing purely abstract painting have not been fully explored with only a few exceptions such as Karl Simms' pioneering work on artificially evoluated images [8].

Our work can be classified between these two approaches. Our concept is based on the cubist paradigm of rendering scenes by using more than one distinct camera. Our approach differentiates from the cubist approach in that 1) we can use an unlimited number of cameras, and 2) the parameters of our cameras can be C^n continuous.

2 Methodology

For rendering abstract paintings, we use a simple ray-tracer. Our 3D scenes include only texture mapped spheres and planes. We also use a simple pinhole camera. Let the orientation of this pinhole camera be given by three orthonormal vectors n_0, n_1 and n_2, and let the distance from the pinhole to the picture plane be d. We assume that these camera parameters are fixed. Any point on camera's picture plane

$$v_{picture}(s_0, s_1) = v_{camera}(s_0, s_1) + dn_2 + s_0 n_0 + s_1 n_1$$

is given by two real numbers s_0, s_1 where $v_{camera}(s_0, s_1)$ is the camera position to be used for computing the color of $v_{picture}(s_0, s_1)$. The equation of the ray passing through this point is

$$v(s_0, s_1, t) = v_{camera}(s_0, s_1) + t(dn_2 + s_0 n_0 + s_1 n_1).$$

Note that the only difference between this and normal ray tracing is that in camera painting the position of the camera $v_{camera}(s_0, s_1)$ varies with s_0, s_1 as defined by the user using a camera control image.

Let a pixel in a camera-control-image be given by two integers (i, j) also let the color of pixel (i, j) be given by a triple $c_{i,j} = (r_{i,j}, g_{i,j}, b_{i,j})$ where $r_{i,j}, g_{i,j}$ and $b_{i,j}$ are real numbers between 0 and 1. To compute the positions of the camera control points $v_{i,j}$ we use

$$v_{i,j} = v_0 + ar_{i,j} n_0 + bg_{i,j} n_1 + cb_{i,j} n_2$$

where a, b and c are user defined real numbers.

Based on these control points, the position of the camera on any point of picture, $v_{camera}(s_0, s_1)$, is computed by a tensor product B-spline surface [1]. In this way, the camera-control-image and the abstract painting can be of different sizes. The B-spline representation also helps to eliminate the aliasing problem. According to our goals, we use linear, cubic or even quintic B-splines [1]. Based on convexity property of B-splines, it is easy to see that the camera stays in a *rectangular prism* shown in Figure 2.

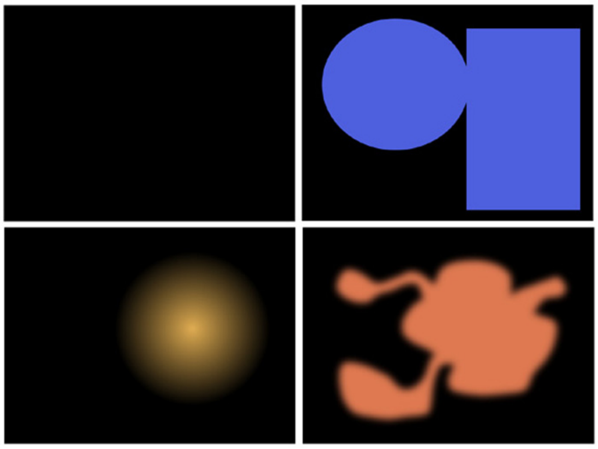

Figure 3: Examples of camera-control-images.

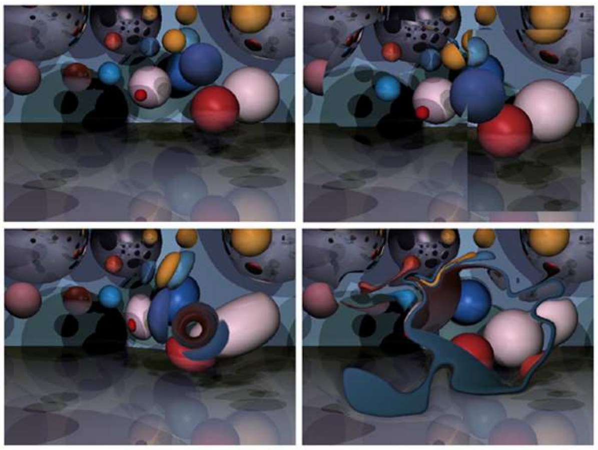

Figure 4: Abstract paintings created by using camera-control-images in Figures 3.

Figures 3 and 4 illustrate some effects that can be achieved by camera painting. The top-left black image in Figure 3 describes a constant camera position at v_0. Therefore, the corresponding top-left image in Figure 4 is a classical ray-traced image. The top-right image in Figure 3 shows how to add a new camera position by painting a solid blue color. Since blue moves the camera in the n_2 direction, the new camera is at the position of $v_0 + cn_2$. The effect can be clearly seen in the top-right image of Figure 4. Warping effects can be created by using a ramp map such as the one that is shown in the bottom-left image of Figure 3. This circular ramp image warps the objects around the center creating toroidal looking shapes as shown in the bottom-left image of Figure 4. The bottom-right image of Figure 3 shows the effect of low-pass filtering. A low-pass filter, by smoothing the edges, creates a boundary region that transitions from one camera position to the other. This smooth transition creates the interesting painted look shown in the bottom-right image of Figure 4.

3 Abstract Paintings

For the creation of abstract paintings, we have developed two artistic techniques. In the first technique, artists incrementally paint a complicated camera-control-image for a given 3D scene. The second technique is the opposite of the first. In this technique, for a given camera-control-image artists incrementally built a 3D scene to create an interesting abstract painting. The second technique is helpful when using existing images such as photographs and paintings.

3.1 Painting Camera-Control-Images for a 3D Scene

In order to efficiently paint useful camera-control-images, artists first have to go through a learning process for any given 3D scene, lighting model and rectangular prism that defines camera positions in 3D.

Images that are painted with only one color are extremely useful for this learning process. We generally use 8 fundamental colors (white, black, red, green, blue, magenta, yellow and cyan), which correspond to the 8 corners of the rectangular prism that defines the camera position in 3D. The resulting ray traced images give an intuition about what can be seen from any given point inside the rectangular prism. Based on this intuition, artists can incrementally create camera-control-images and related abstract paintings.

We used a paint program to create a series of images that document the steps involved in creating a set of abstract paintings for a given 3D scene. Each of these steps was run through the renderer in order to make decisions on how the painting should progress. In this example, the 3D scene includes spheres with shiny surface material and a light source that is directly in front of the scene. The images in Figure 5 show the progress in camera-control-images and Figure 6 shows the related abstract paintings.

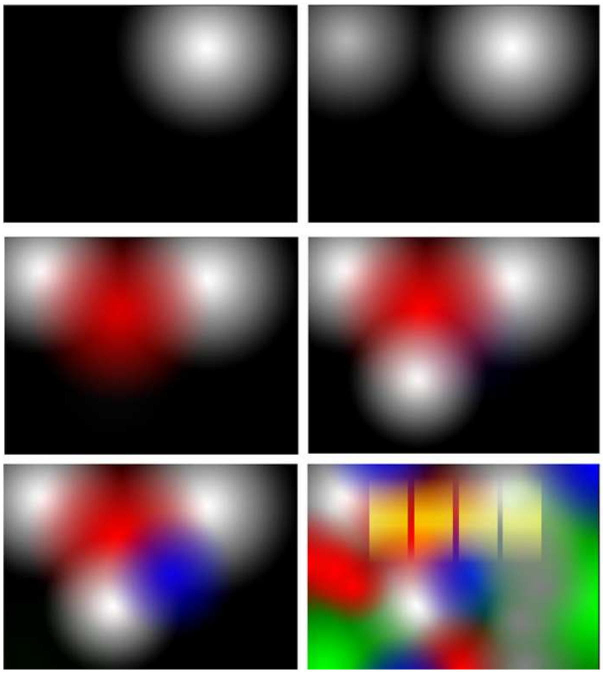

Figure 5: An example of incremental progress in camera-control-images.

The top-left image in Figure 5 shows the initial camera-control image. We began the painting by filling the background with black in order to set the camera at a constant position for all pixels. We then added a white circle in the upper right hand corner in order to begin warping. As can be seen in the corresponding top-left abstract painting in Figure 6, this simple white circle has a very powerful effect on the composition.

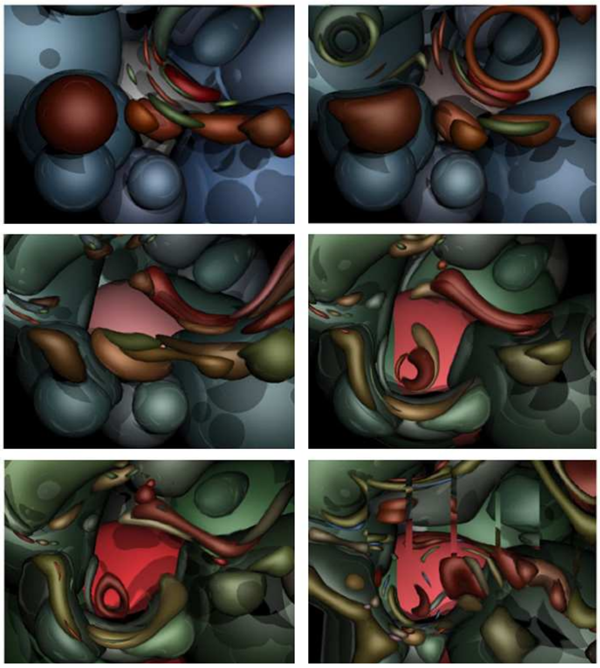

Figure 6: Abstract paintings created by using camera-control-images in Figure 5.

The top-right image in Figure 5 shows the second camera-control image. Here we added another white circle to balance out the scene. In the top-right abstract painting in Figure 6, we see that the additional white circle continues to warp the 3D scene. Notice how the second white circle breaks up the 3D scene geometry into rings.

The middle-left image in Figure 5 shows the third camera-control image in the series. In this image, we began to add some red so that the camera would begin moving along the x-axis. This warps the output image even further. However, as a result of this warp the center of the image has become empty.

The middle-right image in Figure 5 shows the fourth camera-control image in the series. In this image, we added another white circle and a bit of blue to the painted scene to fill the center portion of the image. As can be seen in Figure 6, the middle-right abstract painting appears to be more balanced than the middle-left one. However, there is still too much black in the lower left hand corner.

Blue has been added to create the bottom-left image in Figure 5. Blue moves the camera in the z direction. This move pushes the camera into the scene where the pixels are blue and fills the previously dark empty spaces. As a result of this addition, the composition is further improved as shown in the bottom-left image in Figure 6. However, when we continued to add more color to the camera-control-image, related abstract painting became overworked and lost its appeal. In order to solve this problem, we needed to break up the flow of the image. We accomplish this task by adding the crisp yellow rectangles, shown in the bottom-right abstract painting in Figure 6, this addition gave more depth and complexity to the composition.

The Figures 7 and 8 show two abstract paintings created by this method. We used the same 3D scene to create these images. Note that since the same 3D scene is used for creating both images, they look similar as if they were painted by the same artist.

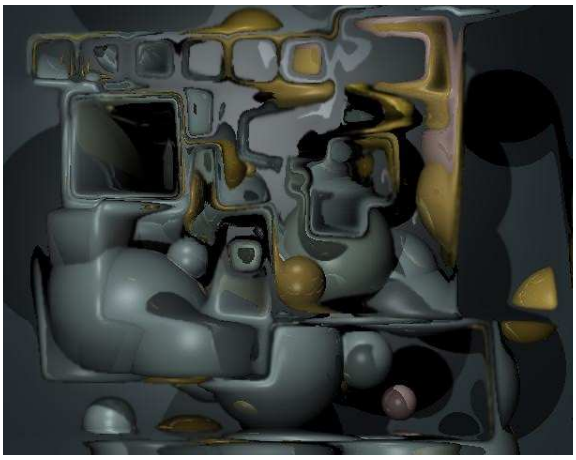

Figure 7: Rectangles: an example of abstract paintings that are created by painting camera-control-images.

3.2 Building 3D Scenes for a Camera-Control-Image

Existing images such as photographs and paintings are resources to use as camera-control-images. The photographs that have a lot of texture and detail are especially useful for creating interesting abstract paintings. For such photographs, instead of changing the camera-control-image, it is better to change 3D scene. In order to control the resulting abstract image for a given photograph, artists can start with a simple 3D scene and update the scene until achieving a satisfactory result.

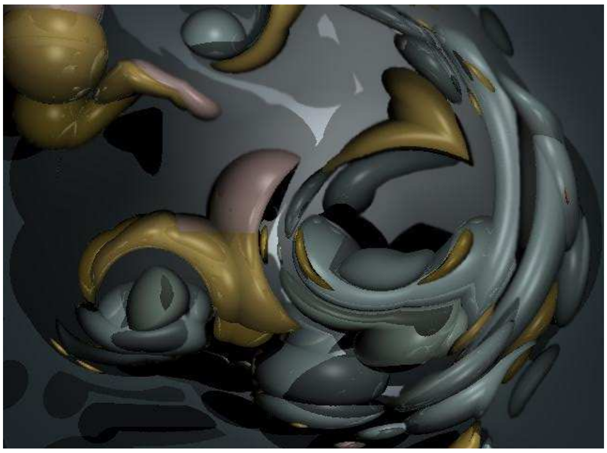

Figure 8: Circles: an example of abstract paintings that are created by painting camera-control-images.

We have developed a simple user interface to create and update 3D scenes. We used this user interface to create a series of 3D scenes that document the steps involved in creating a set of abstract paintings for a given camera-control-image. As we have done in previous section, each of these steps was run through the renderer in order to make decisions on how to update 3D scene. In this example, we chose the top-left image in Figure 9 as a camera-control-image.

The top-center image in Figure 9 shows the first abstract painting. We developed initial 3D scene by adding several spheres with the larger ones to serve as the back drop. After inspecting the results, as it can be seen that the flower petals produce some very soft and beautiful impressions on the background spheres. We set the sphere material to be shiny in order to appear smooth and metallic. The light source consists of three point lights that are positioned directly in front of the entire scene.

The top-right image in Figure 9 shows the second abstract painting. In order to create this image, we added a few more spheres, and positioned them closer to the camera. We also assigned a bright orange color to each sphere so that some contrast could be made between the foreground and background of the image. As you can see in the output image, the orange and blue work very well together. By adding more spheres, we also begin to get more dark regions due to shadows.

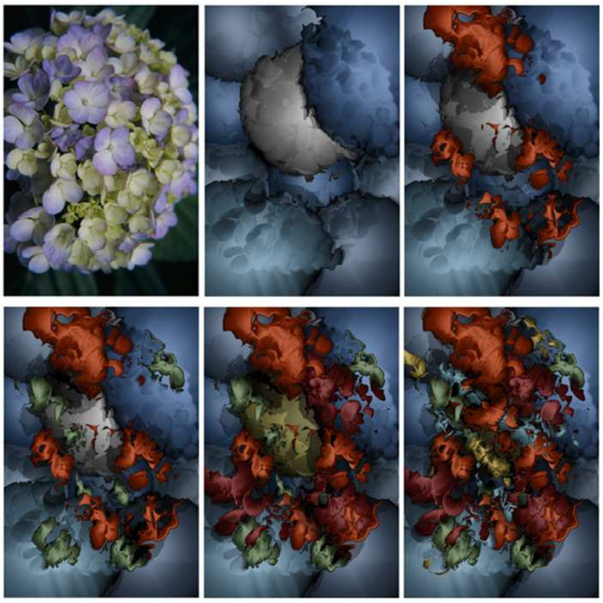

Figure 9: Example of a photograph with a lot of texture and detail and the process of creating an abstract painting from a given photograph.

We added more spheres and assigned a soft green color to them to fill out the scene to create the third image that is shown in the bottom-left in Figure 9. With the addition of more spheres, the scene gains more blacks and texture, due to the shadows cast by the new spheres.

The bottom-center image in Figure 9 is the fourth abstract painting in the series. For this step, we were not comfortable with one of the white background sphere, so we changed it to a golden color. We then added a few red spheres to add some red color to the final image. This really helped to fill out the composition.

The bottom-right image in Figure 9 is the final abstract painting in the series. In this final stage, we added five more spheres and assigned a light blue color to them. We positioned them within the scene so that the color would be sprinkled through out and act as a highlight to the other colors. As it can be seen in the final image, the building of different colored spheres allows the scene to appear much more complex and gives it a great deal of depth.

The Figures 10, 11 and 12 show abstract paintings created by this method. The Figures 13, 14, 15 show abstract paintings created by using 3D scenes that consist of regular grids of spheres.

4 Conclusion and Future Work

In this paper, we presented a simple and intuitive abstract rendering approach to create abstract paintings. This approach allows painters

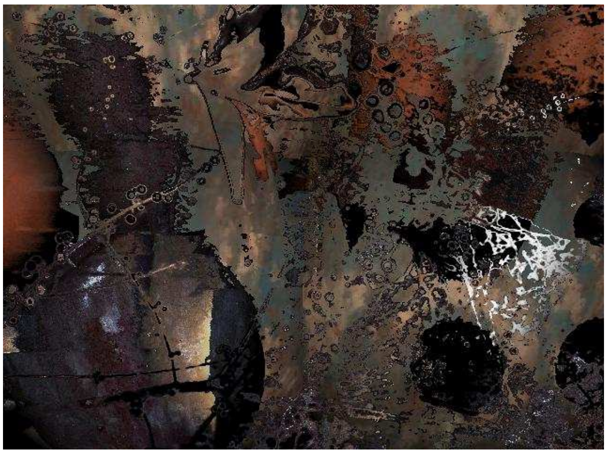

Figure 10: Binary trees: an example of abstract paintings that are created by building 3D scenes.

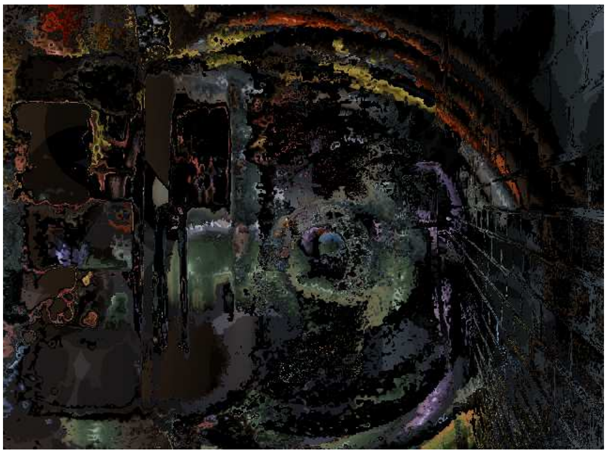

Figure 11: Brick swirl: an example of abstract paintings that are created by building 3D scenes.

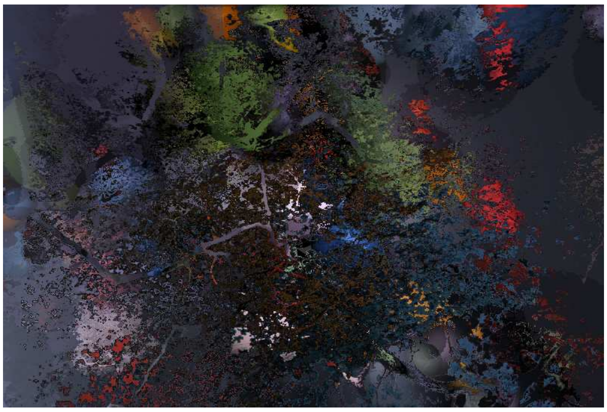

Figure 12: Maples: an example of abstract paintings that are created by building 3D scenes.

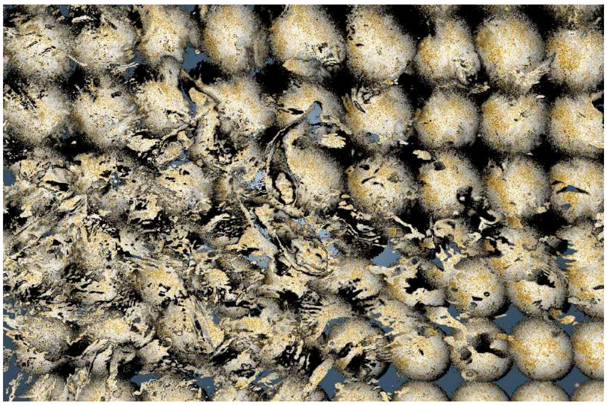

Figure 13: Hurricane: an example of abstract paintings that are created by building 3D scenes.

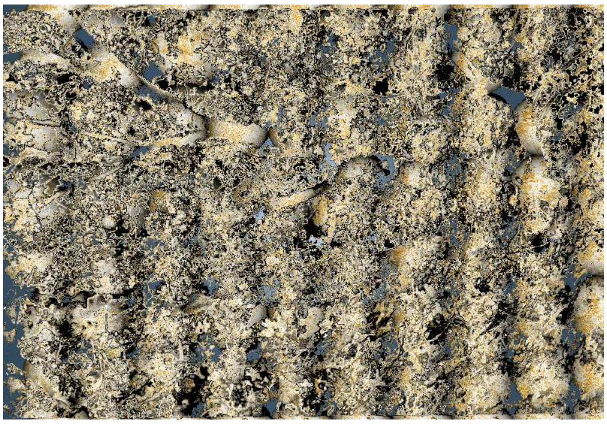

Figure 14: Flowers: an example of abstract paintings that are created by building 3D scenes.

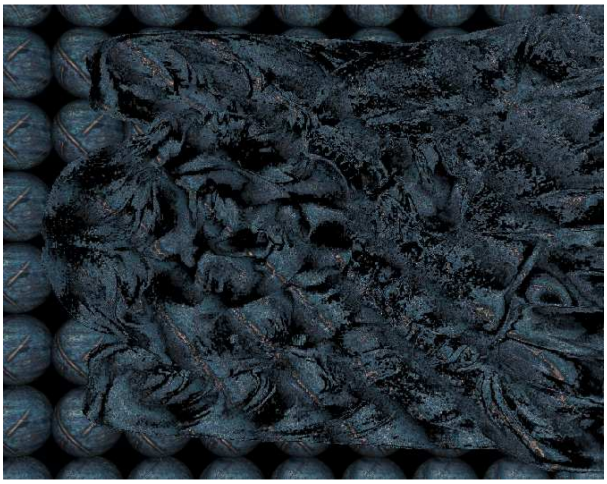

Figure 15: Angel: an example of abstract paintings that are created by building 3D scenes.

to create abstract paintings by using a set of images. Implementation of the camera painting is also simple. Any ray-tracer can easily be expanded to include the camera painting. Camera painting can even be implemented as an homework in an undergraduate graphics class.

As it can be seen from example images, a wide range of painting styles can be obtained by using our approach. Our work shows that the cubist approach of using more than one camera for painting a single image is a useful paradigm to create interesting images. The concept of multiple cameras is so useful that for most camera-control-images and 3D scenes we obtained interesting abstract paintings.

References

[1] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*, (Morgan Kaufmann Publishers, Los Altos, CA, 1987).

[2] J. C. Curtis, S. E. Anderson, J. E. Seins, K. W. Fleisher and D. H. Salesin "Computer Generated Watercolor", *Computer Graphics*, **4** (1997) 421-430.

[3] J. L. Ferrier and Y. L. Pichon, Editors, *Art of the 20th Century: The History of Art Year by Year from 1900 to 1999*, (Chene-Hachette, ISBN: 284.277221.0, 1999).

[4] M. A. Kowalski, L. Markosian, J. D. Northrup, L. Bourdev, R. Barzel, L. S. Holden and J. F. Hughes "Art Based Rendering of Fur, Grass and Trees", *Computer Graphics*, **4** (1999) 433-438.

[5] F. T. Marinetti "Futurist Manifesto", *Le Figaro*, (February 20 1909).

[6] K. Malevich "From Cubism and Futurism to Suprematism", *Petrograd Manifesto*, (1915).

[7] B. J. Meier "Painterly Rendering for Animation", *Computer Graphics*, **4** (1996) 477-484.

[8] K. Sims "Artificial Evaluation for Computer Graphics", *Computer Graphics*, **4** (1991) 319-328.

[9] G. Winkenbach and D. H. Salesin "Rendering Parametric Surfaces in Pen and Ink", *Computer Graphics*, **4** (1996) 469-476.