Color Mixing using Colliding Particles

Gary R. Greenfield
Department of Mathematics & Computer Science
University of Richmond
Richmond, VA 23173, USA
ggreenfi@richmond.edu

Abstract

The use of simulation to apply virtual paint to virtual paper can take many forms ranging from Pollock-style drip paintings to simulated watercolor techniques to paint program airbrush tools. We use physically based modeling to simulate streams of encapsulated paint particles. When set in motion, particles are subject to an artificial gravity, and melt when they contact a virtual canvas. By incorporating algorithmic art methods, and by taking advantage of the scattering effects that result from particle collisions, we explain how we created a series of color studies using colliding particles organized into carefully orchestrated streams of simulated paint droplets.

1 Introduction

Interest in the authentication of Jackson Pollock drip paintings [9][6][8][10] has led to the development of imitation drip paintings in the style of Pollock by Lee et al. [7]. Their simulation relies on sophisticated models of fluid flow [2]. If we view simulation of paint drop formation as lying at one end of a spectrum, paint program air brush patterning simulation [3] as lying at the other end of the spectrum, and computer graphics techniques for simulated watercolors [1][11] as lying somewhere in the middle, we realize that there is a broad range of methodologies for simulating paint drops being deposited on a virtual canvas. Much of the work in this area can be traced back to Haeberli's seminal paper [5] introducing interactive techniques for stylized renderings of digital images.

In this paper we simulate the depositing of virtual paint on virtual canvas using streams of interacting particles, and we examine the use of such streams, or *fountains*, for artistic purposes. Our individual particles move under the influence of artificial gravity and are subject to a simple collision model. Our particles thus model encapsulated spheres of paint that burst when they contact the canvas surface. Our goal is to develop background textures and color studies by relying on algorithms to orchestrate the placement and sequencing of particle streams in such a way that collisions and scattering effects are harnessed for aesthetic purposes.

This paper is organized as follows. In Section 2 we present our model that supports our streams of colliding particles. In Section 3 we detail the testing and development phase of our work. In Section 4 we explain how algorithms were used to program multiple streams of particles and give examples of our resulting color studies. In Section 5 we give our conclusions and discuss future work.

2 Colliding particles

In this section we describe particles, their motion, their interaction, and their organization into particle streams. The attributes for a (spherical) particle are: the position of its center $s = (s_x, s_y, s_z)$, its velocity $v = (v_x, v_y, v_z)$, its RGB color $c = (c_r, c_g, c_b)$ where $0 \leq c_r, c_g, c_b \leq 1$, and its radius r. For convenience, throughout we use fixed radius $r = 0.4$. We view our canvas as lying horizontally in the xz-plane. The vertical axis is therefore the y-axis. By letting $0 \leq x, z < M$, we assume our $M \times M$ pixel canvas is positioned so that it lies in the first quadrant of the xz-plane.

When particles are in motion, an artificial gravity constant $g < 0$ influences the y-component of their position according to the standard Newtonian equation. This means that at time $t \geq 0$,

$$\begin{aligned} s_{x,t} &= s_{x,0} + v_{x,0}t, \\ s_{y,t} &= s_{y,0} + v_{y,0}t + \frac{1}{2}gt^2, \\ s_{z,t} &= s_{z,0} + v_{z,0}t, \end{aligned}$$

where $(s_{x,0}, s_{y,0}, s_{z,0})$ and $(v_{x,0}, v_{y,0}, v_{z,0})$ are the particle's initial position and initial velocity at time $t = 0$. Here we will always let $g = -0.5$.

When two particles with positions s_1 and s_2, masses m_1 and m_2, and radii r_1 and r_2 collide, deflection occurs. In general, if we let we let n be the unit vector from s_1 to s_2, and denote the respective velocities just prior to impact by v_1^- and v_2^-, and just after impact by v_1^+ and v_2^+, then by considering the change in momentum

$$\Delta p = \frac{(1+\varepsilon)w}{\frac{1}{m_1} + \frac{1}{m_2}}$$

where

$$w = v_1^- \cdot n + v_2^- \cdot n$$

and $0 \leq \varepsilon \leq 1$ is the coefficient of *restitution*, in order to conserve momentum we let

$$\begin{aligned} v_1^+ &= v_1^- - \frac{1}{m_1}\Delta p, \\ v_2^+ &= v_2^- + \frac{1}{m_2}\Delta p. \end{aligned}$$

By taking the coefficient of restitution to be one so that collisions are *elastic*, and taking the particle masses to be identically one, we obtain the simpler equations:

$$\begin{aligned} v_1^+ &= v_1^- - wn, \\ v_2^+ &= v_2^- + wn. \end{aligned}$$

Thus, using a forward integration scheme, as time advances from t to $t + \Delta t$ we have

$$s_{t+\Delta t} = s_t + \Delta s$$

where $\Delta s = (\Delta s_x, \Delta s_y, \Delta s_z)$ satisfies

$$\begin{aligned} \Delta s_x &= v_{x,0}\Delta t, \\ \Delta s_y &= v_{y,0}\Delta t + \frac{1}{2}g[2t\Delta t + (\Delta t)^2], \\ \Delta s_z &= v_{z,0}\Delta t. \end{aligned}$$

This shows how implementing collisions can be reduced to simply modifying the values of each particle's initial velocity vector in the forward integration scheme.

Particles are organized into streams, or fountains, as follows. From the stream attributes position $S = (S_x, S_y, S_z)$, velocity $V = (V_x, V_y, V_z)$, and color $C = (C_r, C_g, C_b)$, all particles belonging to a stream are instantiated by initializing their attributes to be "jittered" values of these stream attributes. By jittered, we mean if the stream attribute value is A, then the particle attribute value will be $A + \delta D_A$ where $D_A > 0$ is the maximum displacement allowed and δ is randomly chosen from the interval $[-1, 1]$. To ensure that fountains are positioned above the canvas, we always set $S_y = 1$. In order to simulate the collective motion of the particles within the stream, particle position updates are calculated with reference to a global time variable t which steps from time $t = 0$ to time $t = T$ in increments of 0.25. In addition, the global times for the first position updates of particles i.e., the particle release times, are staggered so that they occur uniformly over the time interval $[0, T_I]$ where $T_I << T$. In this way, particles within a stream are clustered in both space and time. When a particle eventually impacts the canvas, it “splats”. This is modeled by diffusing its color over a five pixel neighborhood realized using either a horizontal shape or an L-shape.

3 Testing and development

A testing and development phase was undertaken using 200×200 pixel canvases i.e., $M = 200$. We set $T = 60$ so that almost all particles would impact. Figure 1a shows the rudimentary collision testing phase using two streams of 100 particles. The red stream emanates from the lower left — (0,0) in the xz-plane — and is aimed toward the upper right — (200,200) in the xz-plane — while the green stream emanates from the upper left — (0,200) in the xz-plane — and is aimed toward the lower right — (200,0) in the xz-plane. Thus collisions take place near the center of the canvas at (100,100) in the xz-plane. Here the splat used is the horizontal splat and all particles are set in motion simultaneously at time $t = 0$ by setting $T_I = 0$. Figure 1b shows our preferred ”four-up” configuration of streams where the four streams are located at the four endpoints of an X and aimed so that they collide at the center of the X. In Figure 1b, the blue stream emanates from the lower right at (0,200) in xz coordinates and the yellow from upper right at (200,200) in xz coordinates. Figure 1c increases the size of each stream to 200 particles, uses what we call the L-splat, and increases the amount of jittering applied to the stream attributes during particle instantiation so that the particles are in general spaced farther apart and their initial velocities are less uniform.

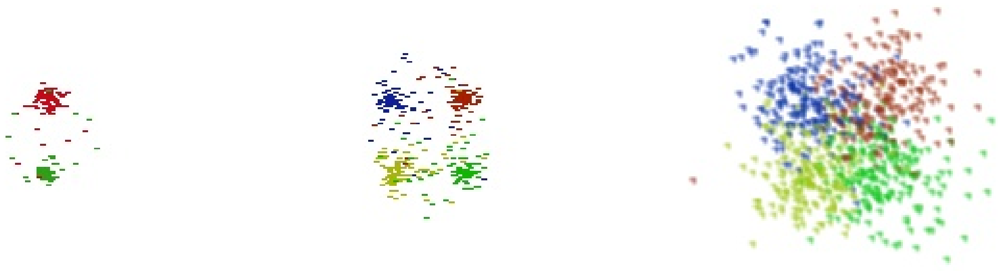

Figure 1: a) Collisions occurring near the canvas center between a red stream with $S = (0, 1, 0)$ and $V = (3, 10, 3)$ and a green stream with $S = (0, 1, 200)$ and $V = (3, 10, -3)$ implemented using 200 particles. b) The test results when additional blue and yellow streams were positioned at $(200, 1, 0)$ and $(200, 1, 200)$ with velocity vectors $(-3, 10, 3)$ and $(-3, 10, -3)$, respectively. c) Another test where the L-splat is introduced and the particles within the stream are initially spread further apart from one another.

Henceforth, we set $T_I = 3$ so that instead of all particles being released in a burst at time $t = 0$, they are released more or less uniformly over the time interval $[0, 3]$. For Figure 2, we also increased the size of each stream to 400 particles in order to intensify the coloring. Figure 2a is a test showing streams that were created by increasing the amount of jitter allowed in each stream’s y-component. Since this is the component that is subject to artificial gravity, the additional variation in particle’s initial y-values causes the distances the particles will travel to have a greater range. Figure 2b shows the effect of orchestrating particle streams. It was achieved by compositing our basic four-up configuration with a second four-up configuration that was obtained from the original one by translating each of the four stream positions by the vector $(20, 0, 20)$. The net result is as if one had turned on four streams and then subsequently turned on four more. Figure 2c shows the results of ten consecutive four-up translations invoking a total of forty streams.

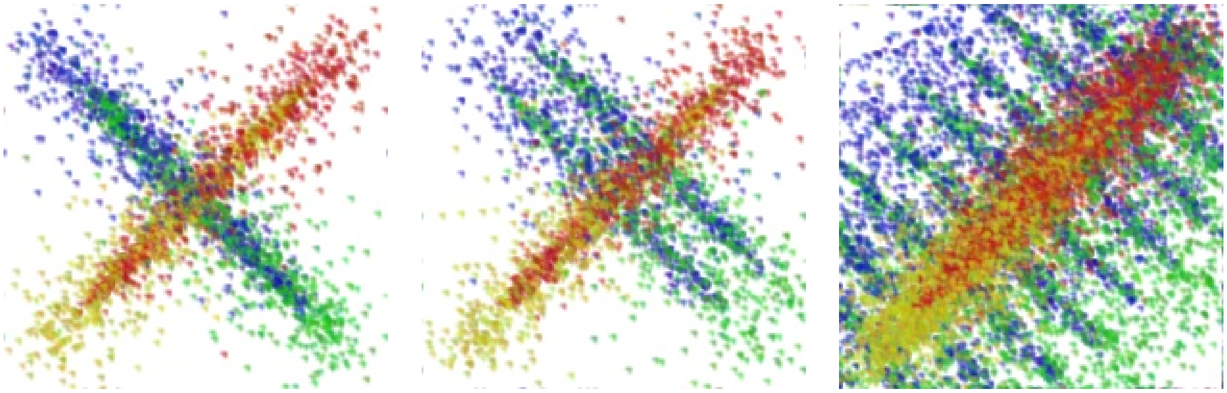

Figure 2: a) The number of particles is increased to give more intense coloration; the streams produce long narrow strips because the x and z initial velocity components vary minimally among stream particles, while the y components of the initial velocity vary considerably among particles. b) The basic four-up scheme is repeated using translation so that eight streams are activated. c) Four-up translation mimics the orchestration of forty streams of particles.

4 Algorithmic color studies

Testing and development yielded an artistic/aesthetic texturing and coloring effect that I chose to exploit for my algorithmic color studies. The color intensity and scattering effects matched my aesthetic sensibilities. I therefore re-adjusted the displacement parameters that controlled the amount of noise that was added to the stream attributes when instantiating particles in order to compensate for a larger 600×600 pixel canvas size. I also modified the basic four-up configuration so that particles from the two streams determining each of the diagonals of the X collided at different spots. That is, rather than place the four streams at the corners of an X in such a way that particles collided in the center of the X, I shifted the diagonal for the green and blue streams so that those streams would be positioned in the xz-plane at $(0, d)$, $(d, 0)$ and then shifted the diagonal for the red and yellow streams so that those streams would be positioned at (d, d), $(d + f, d + f)$. I translated this new base configuration $2 \times 3 \times 10 = 60$ times so that there were $4 \times 60 = 240$ streams yielding a total of $240 \times 400 = 960,000$ colliding particles that were simulated. It should be pointed out, however, that in order to arrange for the amount of canvas coverage that met my aesthetic criteria, not all the stream particles necessarily fell onto the canvas! By choosing $d = 40$ and $f = 30$ so that my new four-up stream configuration anchored the stream position initializations at (40,0,40), (0,0,40), (40,0,0), and (70,0,70) for the red, green, blue, and yellow streams respectively, and by using the following algorithmic scheme twice, the second time imposing a flip about the vertical axis of the canvas (see Figure 3a) or the horizontal axis of the canvas (see Figure 3b) when particles splat, I produced my color studies. In the algorithm S_a denotes the $d = 40$, $f = 30$ anchor position associated to a stream. The algorithm is similar to laying out strips of dough in a crisscross pattern to form the top of a pie crust. Without the flip in the second pass, the images would not have the color symmetry they do. For example, in Figure 3a the strip from top left to bottom right that goes from predominantly purple to predominantly green would not have a matching top right to bottom left strip.

```
for (i = -1; i <= 1; i++)
    for each of the four streams
        $S_0$ = $S_a$ + (80,0,80) + i(100,0,-100)
    for (j = 0; j < 10; j++)
        simulate the four streams
        translate green-blue diagonal by (60,0,60)
        translate red-yellow diagonal by (30,0,30)
```

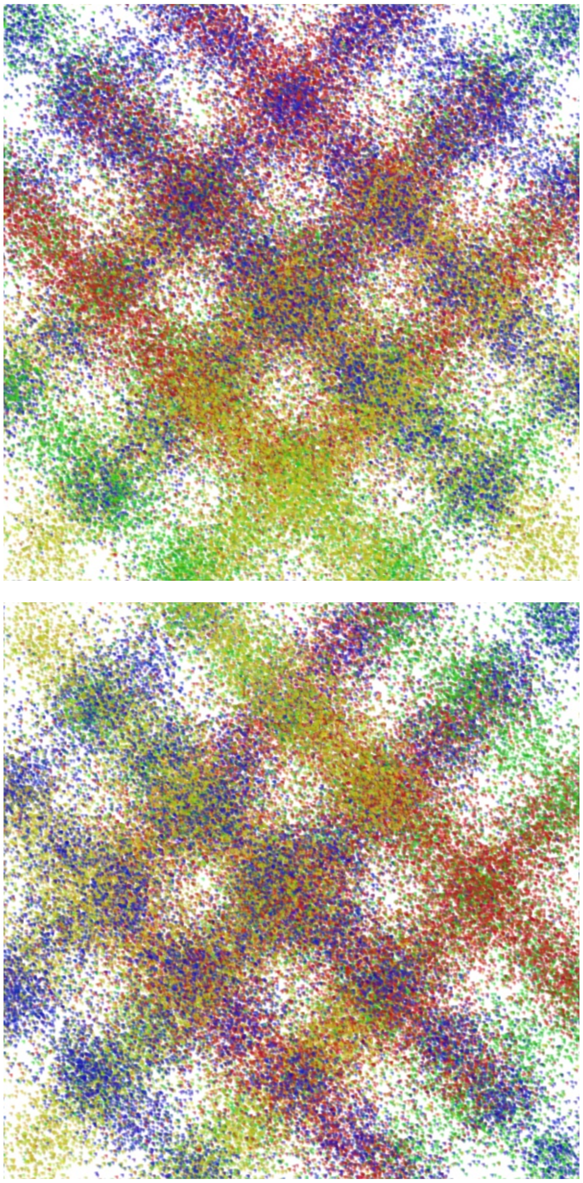

Figure 3: Up to particle distribution, color studies with vertical axis color symmetry (top) and horizontal axis color symmetry (bottom). Each was obtained using 240 streams activated four at a time.

5 Conclusions and future work

We have developed a relatively elementary physically based interacting particle simulation for use in combination with algorithmic art techniques to create color mixing studies. Since these algorithmic works relied only on the use of vector translation in the plane, there are many further avenues that could be pursued for arranging or orchestrating particle streams with potentially interesting aesthetic consequences. In particular, it would be interesting to explore interacting streams that are activated in accordance with curvilinear initial position settings. Recently, we have also considered the problem of evolving stream attributes to satisfy certain well-defined aesthetic criteria. Preliminary results suggest this is a very hard problem [4].

6 Acknowledgment

I would like to thank the anonymous referee for many helpful suggestions. This work was supported in part by a University of Richmond 2007 Summer Fellowship.

References

[1] C. Curtis, S. Anderson, K. Fleischer, and D. Salesin, Computer-generated watercolor, ACM SIGGRAPH 97 Conference Proceedings, 421–430, 1997.

[2] J. Eggers and T. Dupont, Drop formation in a one dimensional approximation of the Navier-Stokes equation, *Journal of Fluid Mechanics*, **262**, 205–221, 1994.

[3] Electronic airbrushing - US Patent 5131056, 1992.

[4] G. Greenfield, Evolutionary computation for aesthetic purposes involving interacting particle simulation, *WCCI 2008 Conference Prooceedings*, to appear.

[5] P. Haeberli, Paint by numbers: abstract image representations, *Computer Graphics*, **24**(4), 207–214, August 1990.

[6] K. Jones-Smith and H. Mathur, Revisiting Pollock's drip paintings, *Nature*, **444**(7119), 9–10, November, 2006. Available on-line at http://www.nature.com/nature/journal/v444/n7119/full/nature05398.html (accessed November 2007).

[7] S. Lee et al., Simulating and analyzing Jackson Pollock's paintings, *Journal of Mathematics and the Arts*, **1**(2), 73–83, 2007.

[8] J. Rehmeyer, Fractal of fake? Novel art-authentication method is challenged, *Science News*, **171**(171), 122–122, February 24, 2007. Available on-line at http://www.sciencenews.org/articles/20070224/bob9.asp (accessed November, 2007).

[9] R. Taylor et al. , Authenticating Pollock paintings using fractal geometry, *Pattern Recognition Letters*, **28**(6), 695–702, 2007. Available on-line at http://materialscience.uoregon.edu/taylor/art/Taylor Submission.pdf (accessed November 2007).

[10] R. Taylor et al. , Fractal Analysis: Revisiting Pollock's drip paintings (Reply), *Nature*, **444**(7119), 10-11, November, 2006.

[11] Watercolor simulation in computer graphics - US Patent 5432896, 1995.