

Trabajo final: A Lightweight, Procedural, Vector Watercolor Painting Engine

Franci Suni Lopez

Universidad Católica San Pablo, Arequipa, Perú,
franci.suni@ucsp.edu.pe,

Resumen En el presente informe se plantea reproducir parcialmente el trabajo *A Lightweight, Procedural, Vector Watercolor Painting Engine*, el cual propone un algoritmo para generar comportamientos de pintura dinámica similares a las acuarelas. El objetivo no es exactamente duplicar la pintura de acuarela, sino crear una gama de comportamientos dinámicos que permitan a los usuarios lograr un estilo similar de proceso y resultado, mientras que al mismo tiempo tienen un carácter único propio.

Key words: acuarela, splat, stamp

1. Algoritmo

Durante la entrada de trazo, los *stamps* se colocan en incrementos de longitud de trazado uniforme a lo largo del recorrido del trazo, de modo que los trazos lentos y rápidos que cubren la misma distancia darán como resultado el mismo número de *stamps* colocados. Cada *stamp* es un conjunto de uno o más *splats* (inicialmente círculos), dispuestos de acuerdo con el tipo de pincel actual. Usando el pincel *'wet-on-dry'* como ejemplo, que consiste en siete *splats* de $n = 25$ vértices dispuestos con un uno en el centro del stamp y seis colocados alrededor de su perímetro (ver Figura 1). Cada *splat* almacena vectores de movimiento que incluyen un vector b de polarización de movimiento y un vector de velocidad por vértice v . Además, un *splat* almacena su edad α (en pasos), así como los parámetros del tipo de pincel: rugosidad r (en píxeles) y flujo f (porcentaje).

1.1. Pigment advection

En cada paso de tiempo, el conjunto de *splats* activos se itera, actualizando cada posición de vértice x_t a x_{t+1} , según las siguientes ecuaciones [1]:

$$d = (1 - \alpha)b + \alpha \frac{1}{U(1, 1 + r)}v$$

$$x^* = x_t + fd + g + U(-r, r)$$

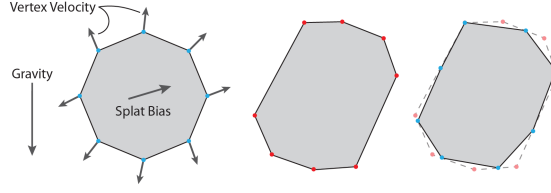


Figura 1. El movimiento de cada splat está dictado por una velocidad por vértice, un sesgo de splat y una gravedad global. De izquierda a derecha: configuración inicial de un splat, después de la advección y después del remuestreo del límite. Figura extraída de [1].

$$x_{t+1} = \begin{cases} x^* & \text{if } w(x^*) > 0 \\ x_t & \text{otherwise} \end{cases}$$

donde g es un vector de gravedad global, α es un parámetro de sintonización que combina el sesgo de movimiento splat con velocidad por vértice, que establecemos en 0,33, $U(a, b)$ es una variable aleatoria uniforme entre a y b , y $w(x)$ es el valor del mapa húmedo en la posición x . Con base en los parámetros splat y los números aleatorios, se calcula una nueva posición candidata x^* , y luego si el lienzo está mojado en esa posición, el vértice se actualiza; de lo contrario, el vértice no se mueve. Vea la figura 1 para una ilustración. Una vez que todos los vértices de splat se han actualizado, su opacidad se vuelve a calcular al conservar la cantidad de pigmento, definida como la opacidad por el área.

1.2. Tipos de pincel

Implementamos cinco tipos de pinceles en nuestro modelo que reproducen una variedad de trazos de acuarela característicos, aunque son posibles muchas más variaciones. Los tipos de pincel son *simple*, *wet-on-dry*, *wet-on-wet*, *blobby*, y *crunchy*. Cada uno usa una disposición diferente de splats por stamp (ver Figura 1.2), y tiene diferentes configuraciones para los parámetros del pincel. Cada trazo en la figura 1.2 tiene un ancho $w = 45$ píxeles. Las celdas del mapa húmedo se establecen en 255 cuando se humedecen, y la simulación se ejecuta a $30Hz$, por lo que tardan 8,5 segundos en secarse. A menos que se especifique lo contrario, los splats tienen vida $l = 30$ (1 segundo), rugosidad de $r = 1$ píxel y flujo de $f = 100\%$. El vector de gravedad es $\langle 0, 0 \rangle$.

El pincel *simple* usa un único splat por stamp, y muestra una distribución de pigmentos no uniforme debido a la naturaleza aleatoria de la evolución de los splats y la mezcla entre trazos debido a la advección en las partes húmedas circundantes del lienzo. El diámetro de splat $d = w$ y su vector de polarización es $b = \langle 0, 0 \rangle$.

Wet-on-dry coloca siete splats por stamp, con seis dispuestos alrededor del perímetro de la séptima. El centro del splat es $b = \langle 0, 0 \rangle$, mientras que el

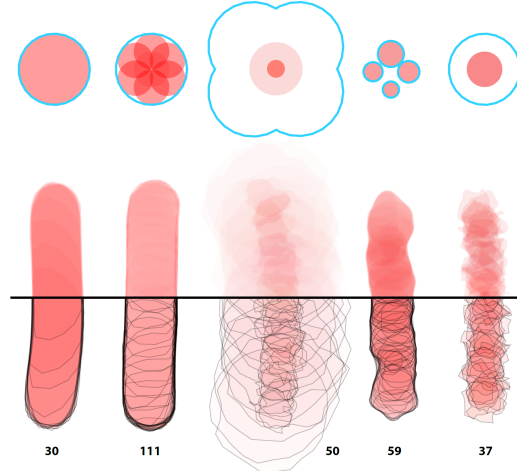


Figura 2. Configuraciones iniciales de splat y trazo resultante para cada tipo de pincel. De izquierda a derecha: *simple*, *wet-on-dry*, *wet-on-wet*, *blobby*, y *crunchy*.

perímetro de los seis splats están en $\theta = 0, \frac{\pi}{6}, \dots, \frac{5\pi}{6}$ tiene $b = \langle \frac{d}{2}\cos\theta, \frac{d}{2}\sin\theta \rangle$, donde $d = \frac{w}{2}$. Esto hace que el perímetro de los splats se acumulen alrededor de los bordes del trazo, agregando pigmento adicional para una apariencia de oscurecimiento del borde.

Wet-on-wet coloca un pequeño splat ($d = \frac{w}{2}$) dentro de un splat más grande ($d = \frac{3w}{2}$) para cada stamp. Esto da como resultado una región progresivamente más oscura hacia el centro del trazo para un aspecto emplumado característico de aplicar pigmento a un lienzo mojado. Ambos símbolos tienen $r = 5$ píxeles, $l = 15$ pasos y $b = \langle 0, 0 \rangle$.

Blobby coloca cuatro símbolos de tamaño aleatorio ($d \in [\frac{w}{3}, w]$) en un patrón cruzado por stamp, con $l = 15$ pasos y $b = \langle 0, 0 \rangle$. Además de agregar algo de ruido al color de los símbolos, esto crea una forma de trazo y densidad de pigmento mucho más uniforme, similar a la pintura de acuarela aplicada a un lienzo más rugoso o con un pigmento más granular.

Finalmente, el pincel *crunchy* coloca un splat por stamp ($d = w$), pero establece $r = 5$ píxeles, $f = 25\%$ y $l = 15$ pasos para reducir la propagación de los vértices mientras aumenta la influencia del componente de caminata aleatoria de la advección. El resultado es una apariencia de trazo que exhibe una ramificación más significativa y una textura de trazo roto, con bordes muy ásperos.

2. Descripción

El algoritmo para implementar el paper tiene las siguientes partes: 1) Paint Initialization-configuración inicial de cada *splat*, 2) Pigment advection-efecto del fluido de la pintura, 3) Lifetime Management-tiempo de fluido y secado y 4) Brush Types-configuración de los diferentes parámetros de cada estilo de brocha.

Figura 2 muestra el resultado de nuestra implementación de las diferentes configuraciones de cada tipo de splat.

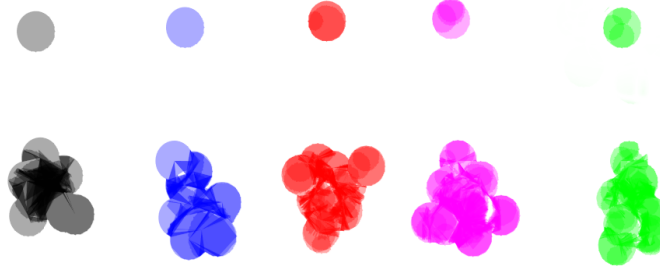


Figura 3. Trazos generados por nuestra implementación. De izquierda a derecha: *simple*, *wet-on-dry*, *wet-on-wet*, *blobby*, y *crunchy*. Figura extraída de [1].

Finalmente en la Figura 2 se muestra un ejemplo de dibujo en base a nuestra implementación de simulación de la acuarela.

3. Conclusiones

- En el paper se muestra la tecnica a grandes razgos con poca información con respecto a los algoritmos usados, debido a que el producto final del paper es comercial. Por lo tanto hemos tratado de inferir que técnicas que han sido usadas.
- En el paper lo único que se indica es que fue implementado en OpenGL y no se da mas detalles con respecto a versiones o otras herramientas usadas. Inicialmente hemos usando las funciones *glBegin(...)* y *glEnd()* para graficar, por tener una forma simple de gestionar la memoria en tiempo de ejecución. El problema es que este metodo no usa *shaders*. Luego hemos decido usar *QTCreator* con OpenGL para poder usar *glDrawArrays()* con *shaders*, adicionalmente usar otras funciones, como por ejemplo *QTime*.
- Finalmente, en base a la exposición inicial del trabajo, concluimos que hemos cumplido con los hitos que se mencionaron al inicio (configuración de cada splat).



Figura 4. Ejemplo de dibujo de nuestra implementación. Trazos de color son los diferentes tipos de pincel.

Referencias

1. S. DiVerdi, A. Krishnaswamy, R. Mech, and D. Ito, “A lightweight, procedural, vector watercolor painting engine,” in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D ’12, (New York, NY, USA), pp. 63–70, ACM, 2012.