MoXi: Real-Time Ink Dispersion in Absorbent Paper

Nelson S.-H. Chu Chiew-Lan Tai

The Hong Kong University of Science and Technology*

Introduction

Our paint system, *MoXi,* allows users to paint in the spontaneous style of Eastern ink painting, on a computer [Chu and Tai 2005]. The simulations of both brush and ink are essential for a successful extension of this traditional art into the digital domain. For real-time performance, we have implemented our ink flow model entirely on the GPU, leaving the CPU for the brush simulation. In this sketch, we provide details of our ink model implementation.

Data Packing

All the data fields (like water density, velocity) needed for the ink simulation are stored as RGBA textures. For the LBE flow simulation, we use 4 textures listed in Table 1 for a total of 16 data fields. We call these *simulation textures.*

Texture	Contents
1. VelDen	[u, v, wf, seep]
2. Misc	[blk, f0, lwf, ws]
3. Dist1	f[N, E, W, S]
4. Dist2	f[NE, SE, NW, SW]

Table 1. Texture data packing. Symbol descriptions are listed in Table 2.

u, v	Water velocity (x, y components)
wf	Water density in flow layer
lwf	Water density in flow layer in the last iteration
ws	Water amount on surface layer
seep	Amount of water seeping from surface layer to flow layer
blk	Blocking factor
f0	Dist. function for stationary particles
f[N, E, W, S]	Dist. functions towards nearest neighbors
f[NE, SE, NW, SW]	Dist. functions towards next nearest neighbors

Table 2. Symbol descriptions.

Texture Updates

Each of the above simulation textures is updated by rendering the paper geometry to a pixel buffer using a fragment program that performs the needed LBE operations. The content of the pixel buffer is then copied to the destination texture. The six texture updates for the LBE simulation are listed in Table 3.

For ink deposition, we apply fragment programs that perform the necessary operations on the brush tuft geometry. During the deposition, a small part of the brush tuft penetrates the virtual paper, giving the brush footprint [Chu and Tai 2004]. The brush tuft geometry is stretched as shown in Figure 1 [Wloka and Zeleznik 1996] to give a continuous stroke (rather than instances of brush geometry at discrete times given by the brush dynamics simulator). Both the penetrating part of the stretched tuft and the paper geometry are rendered to the pixel buffer simultaneously so that deposition does not need extra rendering passes.

Texture Update	Dest. Texture	Operations
1	Misc	Derive f0, blk Deposit or update ws Save wf to lwf
2	Dist1	Collide f[N, E, W, S]
3	Dist2	Collide f[NE, SE, NW, SW]
4	Dist1	Stream f[N, E, W, S]
5	Dist2	Stream f[NE, SE, NW, SW]
6	VelDen	Derive u, v, wf, seep Receive water from surface

Table 3. Texture updates for the LBE simulation.

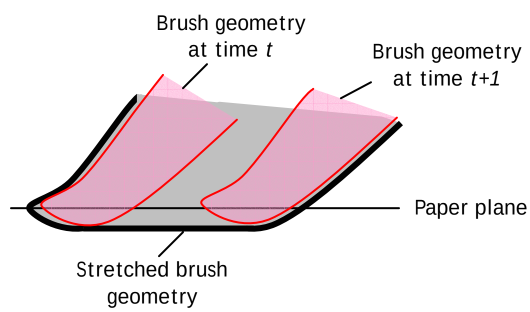

Figure 1. Stretching brush geometry for deposition.

Use of Infinity

We use the fact that any interpolation between a finite value and infinity is infinity to facilitate efficient processing. In our pigment advection scheme, we have to check if the pigment concentrations at the source point $\mathbf{p_f}(\mathbf{y})$ are cross-boundary interpolations (Section 5.4.2 of the main paper [Chu and Tai 2005]). For efficient checking, we set the block factor of all pinning sites equal to infinity (the 'large value' mentioned in Section 5.4.3) when we effect the pinning. The required check is realized by testing if the blocking factor sampled at **y** is infinity. This infinity-interpolation trick is also used in boundary trimming to indicate the regions where trimming should take place.

References

CHU, N. S., AND TAI, C.-L., 2004. Real-Time Painting with an Expressive Virtual Chinese Brush, *IEEE Computer Graphics and Applications 24*, 5, 76-85.

CHU, N. S., AND TAI, C.-L., 2005. MoXi: Real-Time Ink Dispersion in Absorbent Paper. *ACM Transactions on Graphics*. To Appear.

WLOKA, M. M. AND ZELEZNIK, R. C., 1996. Interactive real-time motion blur, *The Visual Computer 12*, 6, 283-295.

*e-mail: {cpegnel, taicl}@ust.hk