Report of Simulation Assignment

Di Yang
s5212191@bournemouth.ac.uk
Bournemouth University

ABSTRACT

The main purpose of the report is to explain the principle of Volumetric retiming utilising the idea of interpolation and how to implement it in Houdini. the procedure of how to improve it by using the Back and Forth Error Compensation and Correction (BFECC) method will also be introduced.

1 INTRODUCTION

Fluid simulations such as fire, smoke, clouds, explosions, and water are ubiquitous in the production of visual effects. These simulations are often computationally expensive, and there are difficult to edit, like slowing down and speeding up, to fit some artistical shots like the slow-motion shot.

So this dilemma often appears in the special effects production process: the artist finally reached the ideally "look and feel" of fluid simulation, however, the VFX supervisor asked him to slow down to create a larger illusion, or just change the shooting rhythm. This is why volume retiming method should be studied and implemented so that the artist can more easily solve the similar problems mentioned above.

2 BACKGROUND RESEARCH

Volume Advection

The simulation of incompressible fluids involves several calculation steps, including diffusion, advection and pressure projection etc.(Fedkiw *et al.* 2001), showing in the (Figure 1):

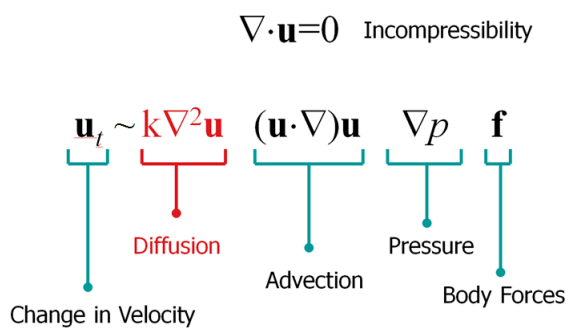

Figure 1: Navier-Stokes Equations: Incompressible Flow

from Lagrangian view of Navier-Stokes Equations we can get:

$$\frac{d\rho}{dt} + div(\rho \mathbf{v}) = 0 \tag{1}$$

in which the $\frac{d\rho}{dt}$ could be consided as $\rho_{current}$ - ρ_{back}, Combining with Semi-Lagrangian Advection:

$$\frac{\partial v}{\partial t} = -v \cdot \nabla v \tag{2}$$

wet can get a Simple Forward Euler estimation:

$$\rho_{current} = \rho_{back} - v_{back} \Delta t \tag{3}$$

So we can understand that Advection means transport quantity from one region to another along the fluid's velocity field(Fedkiw *et al.* 2001).

Volume Interpolation

the key issue for the fluid retiming is that when we slow down the origin simulation we need to insert some frames between two simulated frames in order to let fluid smoother and more natural.

the method of making this Interpolation is simple: as the values from forward and backward frames have already known, we could advect the values from two frames to the current position and then mix or blend these values as the current value(Syuhei Sato 2018), showing like Figure 2:

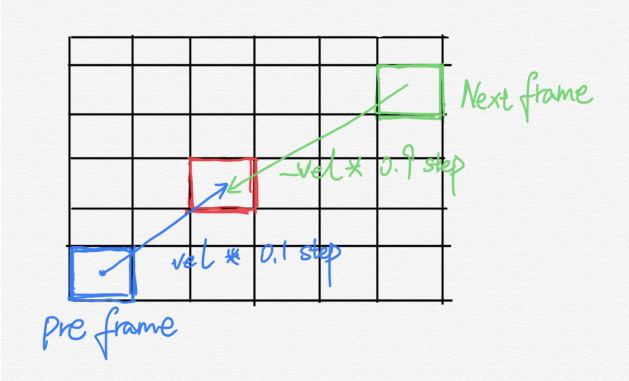

Figure 2: Original Interpolation Method

the algorithm is like:

- step1: preframe = value.advect(backward,v*0.1);
- step2: nextframe = value.advect(forward,v*-0.9);
- step3: value.get(blend(preframe,nextframe,0.1));

However, if the key volume is not sampled close enough, this method may cause blurring or temporary popping.

Back and Forth Error Compensation and Correction(BFECC)

Therefore, the Back and Forth Error Compensation and Correction (BFECC) was introduced to improve the accuracy of the Volumetric retiming.

the initial idea using BFEEC to improve the result of retiming is inspired by Su(2014). BFECC was recently proposed by Dupont and Liu(2003) as a level set interface computation method. they combined BFECC with their simple redistancing technique and applied it to the Zalesak's problem, showing significantly reduced the volume loss(ByungMoon Kim 2005).

Since we want to apply it to different advections, we use φ to denote a quantity that is advected.This φ can be every component in volume like velocity, smoke density, satisfying:

$$\varphi_t + \mathbf{u} \cdot \nabla_\varphi = 0 \tag{4}$$

Let $\boldsymbol{L}$ be the integration step(4) of the first-order upwind or semi-Lagrangian, such that:

$$\varphi^{n+1} = \mathbf{L}(\mathbf{u}, \varphi^n) \tag{5}$$

the BFECC can be written as the following three L-calling process:

$$\varphi^{n+1} = \mathbf{L}(\mathbf{u}, \varphi^n + \frac{1}{2}(\varphi^n - \Delta\varphi)) \tag{6}$$

and in which:

$$\Delta\varphi = \mathbf{L}(-\mathbf{u}, \mathbf{L}(\mathbf{u}, \varphi^n)) \tag{7}$$

In summary, the core idea of BFECC is to find the error,then correct the error in advance(Hu *et al.* 2016), showing like the list below:

- step1: error = value.advect(orign,v).advect(orign,v) - value(orign);
- step2: orign = orign - $\frac{1}{2}$(error);
- step3: value.advect(orign,v);

3 IMPLEMENTATION

In this section, we will introduce how to implement this Volumetric retiming node in Houdini and also will illustrate how to improve it by using BFECC method.

Simple Interpolation Version

Since the core idea formula has been given above, seeing the algorithm steps(2) in Volume Interpolation part, the remaining question is how to implement it in houdini.

volumesample function. volumesample is a vex function which will return the volume's sampled value at the given position, so it will perform like advect function.

lerp function. the lerp function performs a bilinear interpolation between the two given values.

Therefore, the achieved code of a simple interpolation version could be seen in Figure 3:

```
float substep = ch("substep");
float blendratio = ch("blend_ratio");

//vel
vector bvel,fvel;
bvel = volumesamplev(@OpInput1,"vel",@P);
fvel = volumesamplev(@OpInput2,"vel",@P+bvel*substep);
v@vel = bvel*(1-blendratio) + fvel * blendratio ;

//density
float bdensity = volumesample(@OpInput1,"density",@P - bvel*blendratio*substep);
float fdensity = volumesample(@OpInput2,"density",@P + fvel*(1-blendratio)*substep);
@density = lerp(bdensity,fdensity,blendratio);
```

Figure 3: The code of a simple interpolation version

Improvement through BFECC

When talking about the improvements through BFECC theory, the process becomes a little more complicated. In other words, we cannot directly use the method given above but need to make some changes, because the known forward and backward frames are produced through a similar simulation mode, so we can use it to achieve more optimized results.

The 3 steps we mentioned in the BFECC section will have a tiny improvement:

- step1: error = value.advect(backward,v) - value.advect(forward,-v);
- step2: orign = value.get(blend(preframe, nextframe,0.1));
- step3: current = orign - $\frac{1}{2}$(error);

Therefore, the achieved code of BFECC version could be seen in Figure 4:

```
float substep = ch("substep");
float blendratio = ch("blend_ratio");

function float BFEECredress(float backvalue;float frountvalue;float blendratio){
    float currentvalue;
    currentvalue = lerp(backvalue,frountvalue,blendratio) - 0.5*(backvalue - frountvalue)
    return currentvalue;
}
function vector BFEECredress(vector backvalue;vector frountvalue;float blendratio){
    vector currentvalue;
    currentvalue = lerp(backvalue,frountvalue,blendratio) - 0.5*(backvalue - frountvalue)
    return currentvalue;
}

//vel
vector bvel,fvel;
bvel = volumesamplev(@OpInput1,"vel",@P);
fvel = volumesamplev(@OpInput2,"vel",@P+bvel*substep);
v@vel = BFEECredress(bvel,fvel,blendratio);

//density

float backdensity = volumesample(@OpInput1,"density",@P - @vel*blendratio*substep);
float frountdensity = volumesample(@OpInput2,"density",@P + @vel*(1-blendratio)*substep);
@density = BFEECredress(backdensity,frountdensity,blendratio);
```

Figure 4: The code of BFECC version version

4 OUTCOMES

I made an HDA in Houdini. According to the above concept, it can steadily and effectively complete the volume retiming task and can make some more interesting effects, and I

recorded a **Video** or go to the next url: https://vimeo.com/420666889 to see its effect.

we could find that using BFECC method can significantly improve the results of 0.5X and 0.25X retiming and it is suitable for various situations, however, there is still plenty of things can be improved.

5 CONCLUSION

This project helped me have a deeper understanding of the fluid simulation, Especially the algorithm and the realization method of each function such as advect, diffusion, and dispassion etc.

Although the Volumetric retiming node has successfully worked, some improvements are still needed. At present, the node cannot deal with the effect of playing in reverse, and the methods for adding and deleting elements are very complicated. and some people like carrier(2020) have started research on using Neural Network to do the frame Interpolation of Smoke Simulation,

REFERENCES

[1] ByungMoon Kim I. L. J. R., Yingjie Liu, 2005. Flowfixer: Using bfecc for fluid simulation. *Eurographics Workshop on Natural Phenomena*.

[2] Carrier S. C. G. G., 2020. Retiming smoke simulation using machine learning.

[3] Dupont T. F. and Liu Y., 2003. Back and forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function. *Journal of Computational Physics*, **190**(1), 311–324.

[4] Fedkiw R., Stam J. and Jensen H. W., 2001. Visual simulation of smoke. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 15–22.

[5] Hu L., Li Y. and Liu Y., 2016. A limiting strategy for the back and forth error compensation and correction method for solving advection equations. *Mathematics of Computation*, **85**(299), 1263–1280.

[6] Su J. Houdini pyro's fluid interpolation [Online]. 2014. Available From: https://blog.csdn.net/cuckon/article/details/39813787 [Accessed 10 April 2020].

[7] Syuhei Sato T. N., Yoshinori Dobashi, 2018. Editing fluid animation using flow interpolation. *ACM Transactions on Graphics*, **37**(5), 1–12.