What Time Is It?

Efficient And Robust FX Retiming Workflow For Spies In Disguise

Ravindra Dwivedi
Blue Sky Studios
ravindrad@blueskystudios.com

Jon Barry
Blue Sky Studios
jonb@blueskystudios.com

Sean C McDuffee
Blue Sky Studios
scmcduff@blueskystudios.com

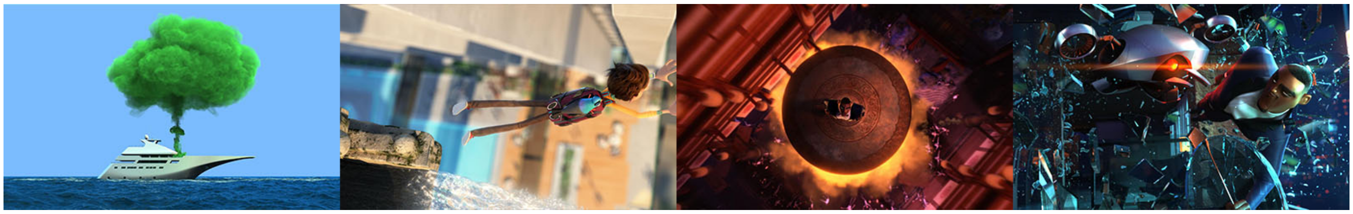

Figure 1: Variety of Retimed FX, Spies In Disguise ©2019 Twentieth Century Fox Film Corporation. All Rights Reserved

ABSTRACT

We present our FX retiming workflow developed on Blue Sky Studios' latest feature, Spies in Disguise. Retiming refers to the slowing down and speeding up of FX assets in a shot. These include point particles, rigid bodies, volumetric elements like smoke and fire, and fluids. Our solution is shown to be robust and efficient, even for the challenging cases of retiming heavily dynamic events, such as fire and explosions.

CCS CONCEPTS

• **Computing methodologies** → **Physical simulation**; *Procedural animation*;

KEYWORDS

fluid simulation, retiming, fire, explosion, flip, particles, rigid bodies

ACM Reference Format:
Ravindra Dwivedi, Jon Barry, and Sean C McDuffee. 2019. **What Time Is It?: Efficient And Robust FX Retiming Workflow For Spies In Disguise**. In *Proceedings of SIGGRAPH '19 Talks.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3306307.3328152

1 INTRODUCTION

Early in production for Spies in Disguise, the FX team was tasked with handling in-shot time manipulation of large scale effects, such as fire, explosions, particles, rigid bodies, and fluids. Manipulating the time step in simulation directly had a number of issues. One being the extra computational effort of small time steps and another being the non-linear effect of the time step parameter on the final look. Both of these issues contributed to longer iteration cycles. In addition to the known retiming aspect from animation, after an effect is simulated, it's common to receive notes to keep the same look but to modify it's timing. As stated, re-simulation with a different time step would modify the look and waste computational resources and artist time. Retiming post sim helped to reduce these inefficiencies.

We categorized effects as either point-based or voxel-based to better tackle their unique technical problems and to provide separate artist-friendly tools. A particularly challenging task was the retiming of fire and explosions. When interpolating data, these effects were prone to flickering artifacts which our new system helps to mitigate. To our knowledge, a solution to this problem has not been presented before.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGGRAPH '19 Talks, July 28 - August 01, 2019, Los Angeles, CA, USA
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6317-4/19/07.
https://doi.org/10.1145/3306307.3328152

2 OUR SYSTEM

2.1 Workflow

Effects typically depend on the animation of characters, props, and cameras. How those departments managed the time warping of assets across a shot influenced our options for simulation and retiming. There was constant communication between FX and Animation to find a standardized workflow that maximized control and flexibility for both departments in order to achieve the artistic goals of a shot. In the end, animators would apply a global time warp curve to the characters and camera that we could then import and apply to our simulations (which were solved with a uniform timescale).

In addition, we developed a comprehensive set of tools that allowed us to adjust the timescales from animation, as needed, to meet the artistic needs of a particular element and interpolate subframes of any simulation data to correctly represent the effect for that frame.

2.2 Point Retiming

Several improvements were made to existing toolsets to increase accuracy and flexibility. For generalized point information, we implemented a system that could handle arbitrary attributes and interpolate samples correctly based on data type as well as exposing smoother interpolation methods, such as Catmull-Rom or Cubic B-Spline. Additionally, rigid body sims and instancing also required using spherical linear interpolation of orientation information. This eliminated the flipping that could occur when interpolating Euler rotations.

Because particles can be born and killed frame to frame, we created a way of detecting these cases and procedurally jittering when these events occurred. By exposing an attribute describing the particle behavior, artists were able to scale the size or opacity of the particle. This prevented large groups of particles from simply popping on or off, offering a more appealing look.

2.3 Volumetric Retiming

Our algorithm is based on the level set advection scheme from [Selle et al. 2008]. In their approach, they use the nearest two frames and advect the level set both forward and backward to the interpolated time using an averaged velocity. While this approach works well enough for collisions, it posed issues for volumetric FX. Advection causes a diffusion of the volumetric quantity that maximizes half way between the original frames. Figure 2 shows the fluctuation of luminance values caused by interpolating the data. In our previous talk on retiming in [Ecker et al. 2016], we proposed jittering the interpolation fraction uniformly to hide the problem. This worked well for diffuse smoke/mist FX but still failed when retiming explosions and fire. With these emissive FX, any slight diffusion of the volume results in a jarring flickering artifact. We extended our system with several improvements to overcome this issue.

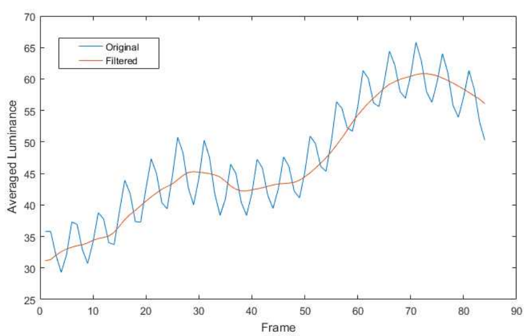

Figure 2: Comparison of filtered and unfiltered volumetric data.

Instead of averaging the velocities of the nearby frames, $\boldsymbol{v}^n$ and $\boldsymbol{v}^{n+1}$, we found that using the end of step velocity, $\boldsymbol{v}^{n+1}$, for both forward and backward advection gave a steadier look to the resulting volume via

$$\phi(\boldsymbol{x}, \alpha) = (1 - \alpha)\phi^n(A(\boldsymbol{x}, -\Delta t, \boldsymbol{v}_1, \alpha)) + \alpha\phi^{n+1}(A(\boldsymbol{x}, \Delta t, \boldsymbol{v}_2, 1 - \alpha))$$

where $A()$ is the advection function, $\alpha \in [0, 1]$, $\boldsymbol{v}_1 = \boldsymbol{v}_2 = \boldsymbol{v}^{n+1}$, if $\alpha \neq 0$. $\boldsymbol{v}_1 = \boldsymbol{v}^n, \boldsymbol{v}_2 = \boldsymbol{v}^{n+1}$, if $\alpha = 0$.

For the advection step, we required a higher order integration than the original Euler and settled on RK4. We also implemented temporal filtering of the volumetric data. This resulted in slight diffusion of the original data but sharpened the worst inbetweens (see Figure 2). The smoothing improvements to the emissive fields offset any issues with density degradation. We found that the best results arose from using a Mitchell-Netravali filter [Mitchell and Netravali 1988]. The wider the filter, the smoother the result gets with loss of high frequency details. The filter radius was kept constant for all frames, but was exposed as a parameter to maximize user control.

We also implemented jittering of interpolation fraction randomly for each voxel. This was exposed as a parameter to the user for use in extreme cases where dense volumetric data was slowed down to very low speeds. The local jittering, when combined with Mitchell filter further helped reduce the global flicker.

After combining all of these steps, we were successfully able to retime explosions and fire without any significant flickering which was not possible using our previous system. The new approach was easy to integrate into our existing pipeline and was efficient to run.

2.4 FLIP Retiming

For FLIP simulations, we wanted to preserve the advantages of SideFX Houdini's recently introduced compressed fluid workflow. This provided both a decrease in disk usage and increased speed in processing the data post-simulation. The fluid is stored as a narrow band of particles and an interior surface SDF, both of which are necessary for accurate meshing. With our developed point and volume retiming tools, we were able to keep both sets of data in sync and merge them together for a meshed result that preserved the surface detail as well as the correct fluid interior.

For both points and volumes, in cases where greater accuracy was needed for sub-frame collisions or to maintain high levels of detail, users could cache substep data from the simulation and interpolate between them instead of only using whole frame data as samples.

3 LIMITATIONS

Our volumetric retiming method relies on accurate velocities and other fields in the simulation which, when advected, would result in the next frame of the simulation. Due to this, our interpolation is discontinuous at sources because new information is introduced which has no correspondence to the previous frames. As a result, our method is not effective with moving sources.

4 CONCLUSIONS

We developed new techniques for retiming various effects and expanded the retiming pipeline used for our current show. In addition to giving us better, more art directable control over the process, the new pipeline also allowed us to meet our production demands smoothly. We would like to thank Maurice van Swaaij and Trevor Thomson for many helpful discussions and suggestions.

REFERENCES

Gregory Ecker, Ravindra Dwivedi, and Ilan Gabai. 2016. Directed volcano getting the most out of your simulations *(ACM SIGGRAPH 2016 Talks(SIGGRAPH '16))*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA. https://doi.org/10.1145/2897839.2927454

Don P. Mitchell and Arun N. Netravali. 1988. Reconstruction filters in computer-graphics. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques (SIGGRAPH '88)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 221–228. https://doi.org/10.1145/54852.378514

Andrew Selle, Michael Lentine, and Ronald Fedkiw. 2008. A mass spring model for hair simulation. In *ACM SIGGRAPH 2008 Papers (SIGGRAPH '08)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA. https://doi.org/10.1145/1360612.1360663