

# Dynamic Resolution Rendering Updated

---

Doug Binks

## Introduction

This update to the Dynamic Resolution Rendering sample optimizes the temporal anti-aliasing (TAA) upscale pass and adds a few new features for improved qualitative comparisons focused on TAA.

Readers new to Dynamic Resolution Rendering should read the original article here:

<http://software.intel.com/en-us/articles/dynamic-resolution-rendering-article/>

## Detailed Changes

The new sample includes the following changes, which are detailed in the sections that follow.

1. Added optimization for temporal anti-aliasing (TAA) when using motion blur, which decreases the time this pass takes by approximately 40%.
2. Corrected the velocity scale factor so that if the pixel in the last frame was 1 pixel away it is attenuated by  $\frac{1}{2}$ . Previously a constant scale factor was used, which was chosen for the default resolution.
3. Added a zoom box for viewing scaled portions of the screen.
4. Added a basic TAA mode that does no velocity scaling so any ghosting issues can be found.
5. Improved pause mode that correctly renders the previous and current frames to show TAA artifacts clearly in basic mode.
6. Added a motion blur toggle so that the effect of motion blur and TAA can be properly considered.
7. Added a symmetric TAA toggle.
8. Changed pause key from Space to P.

## Temporal Anti-Aliasing Optimization

The original algorithm for TAA combined the color outputs from the current and previous frames based on the per pixel velocity encoded in a 2 channel 16-bit floating point texture. This led to reading two 32-bit color values along with two 32-bit velocity values for a total of 128-bit read per pixel, and 32-bit output.

Since the previous motion blur pass read both color and velocity values for the current pixel, we are able to encode the required velocity information into the alpha channel at this stage, so that the TAA pass only needs to read the color buffers, reducing the read from 128 bits to 64 bits per pixel. This is a total bandwidth reduction of 40%, and we see a corresponding improvement in performance for that pass. Since the bandwidth for the motion blur pass hasn't changed, this gives a net performance boost, and a further reason to consider using velocity aware TAA!

## Velocity Scale Factor Corrections

This is a simple correction that adds a new constant to the pixel shader for TAA to scale the velocity to pixel units using the actual current resolution in use, instead of a constant based on the default resolution.

## **Zoom Box**

The zoom box, a feature of the [morphological anti-aliasing \(MLAA\) sample](#), has been added so that viewers can observe an enlarged view of a portion of the scene in order to get a better qualitative assessment of the algorithm.

## **Basic Temporal Anti-Aliasing Filter**

This filter shows what would happen if we naively composite the current and previous frames without using the velocity information to scale the amount of the previous frame used. Ghosting artifacts from motion are notable, though motion blur helps to reduce these.

## **Motion Blur Toggle**

Motion blur helps to reduce temporal aliasing between frames, and somewhat reduces the need for geometric anti-aliasing. This toggle helps to show the benefits of using motion blur. Note that when motion blur is disabled, the TAA filter cannot use the new optimization, so falls back to reading per pixel velocities.

## **Symmetric Temporal Anti-Aliasing Toggle**

This toggle switches the jitter offsets from [0,0] on odd frames and [0.5,0.5] on even frames to [-0.25,-0.25] and [0.25,0.25] respectively.

## **Changed Pause Key from Space to P**

DXUT binds the space key to switching the last control used, so we moved to using P for pause.

## **Conclusion**

Overall, we hope these changes help developers to better see the benefits of TAA, especially when using either fixed up scaling or dynamic resolution rendering.