Niklas Scharnewski, Laura Fink, Marc Stamminger *Erlangen, 06.05.2025*

Interactive Computer Graphics (exercise sheet 1)

Assignment 1 [10 Points] Temporal Anti-Aliasing

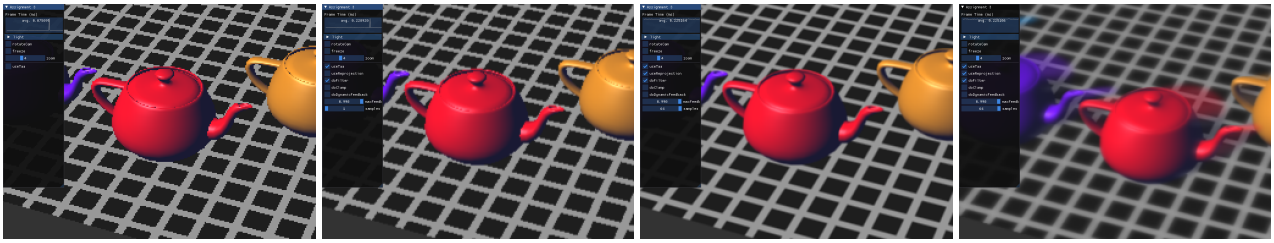

Figure 1: Left to right, the original aliased input, only filtering (with 1 spp), the temporal anti-aliased static scene, and ghosting despite correct reprojection with a moving camera. (notice the camera options `freeze`, `zoom` and `rotateCam`).

a) Temporal Supersampling [3 Points] The first task of this assignment is implementing temporal anti-aliasing for static input. To enable temporal supersampling a small jitter has to be applied to the projection matrix in every frame. Individually jittered frames are recombined over time utilizing exponential smoothing, which yields an anti-aliased result. The feedback represents the ratio that is used to mix the accumulated preceding frames with the newly generated one. With a feedback value of `0.99` for the exponential smoothing the jitter should barely be noticeable. You should see a nicely anti-aliased result for a still camera and extreme ghosting if the camera is moved.

b) Weighted Sampling [2 Points] In the next step, 9 weighted samples from the new frame are combined to additionally smooth the color and counteract jittering. Prepare the required weights in `a01.cpp` depending on their euclidean distance to the center pixel's jittered midpoint and use these weights in `taa.cs.glsl`. Apply Blackmann-Harris 3.3 ($W(x) = e^{-2.29x^2}$, see Slide 23 [1] accompanying the template code) to prepare your filter weights and don't forget to normalize them.

c) Reprojection [4 Points] Temporal supersampling depends on the pixel's preceding color. Acquiring that data in a static camera setup is quite easy. However, a moving camera requires reprojection to get the texture coordinates of the current pixel in the previous frame. With the reprojected texture coordinates the pixel's previous color can be obtained from the history buffer. The slides [2] accompanying the template code demonstrate this approach very nicely. Note however that, their approach cannot be copied 1:1 for our setup! Also note that, reprojection cannot remove all types of ghosting (e.g. ghosting caused by disocclusion).

d) Color Clamping [1 Points] Since the retrieved history color is not guaranteed to match the actually required pixel color, the history color must be constrained. This is done by clamping the history color to the rgb-min/max-box of the 3x3 pixel neighborhood. Constraining the history sample should eliminate almost all ghosting effects. But, do you find the case where color clamping leads to an obviously wrong result in our scene?

Bonus A fixed feedback value of e.g. `0.99` is not globally suitable for every scenario. For high velocities (large pixel distances between frames) it might be better to use a lower feedback value while a high feedback is better when no movement occurs. In addition, for high variations in luma the mapping to previous values is more likely to fail or lead to poor results. Adaptively mixing the current frame with the history buffer regarding velocity and luma can improve the result.

Additionally, you can improve the quality of the history sample by clipping instead of clamping it to the min/max color of the neighborhood.

Files to submit: a01.cpp, shader/taa.cs.glsl
The assignment is due until Wednesday, 14.05.2025, 11:55pm. Happy Hacking! :)

References

[1] Brian Karis. High quality temporal supersampling.

[2] Lasse Jon Fuglsang Pedersen. Temporal reprojection anti-aliasing in inside.