k-DOP Clipping: Robust Ghosting Mitigation in Temporal Antialiasing

Julius Ikkala
Tampere University
Finland
julius.ikkala@tuni.fi

Tuomas Lauttia
Tampere University
Finland
tuomas.lauttia@tuni.fi

Pekka Jääskeläinen
Tampere University
Finland
pekka.jaaskelainen@tuni.fi

Markku Mäkitalo
Tampere University
Finland
markku.makitalo@tuni.fi

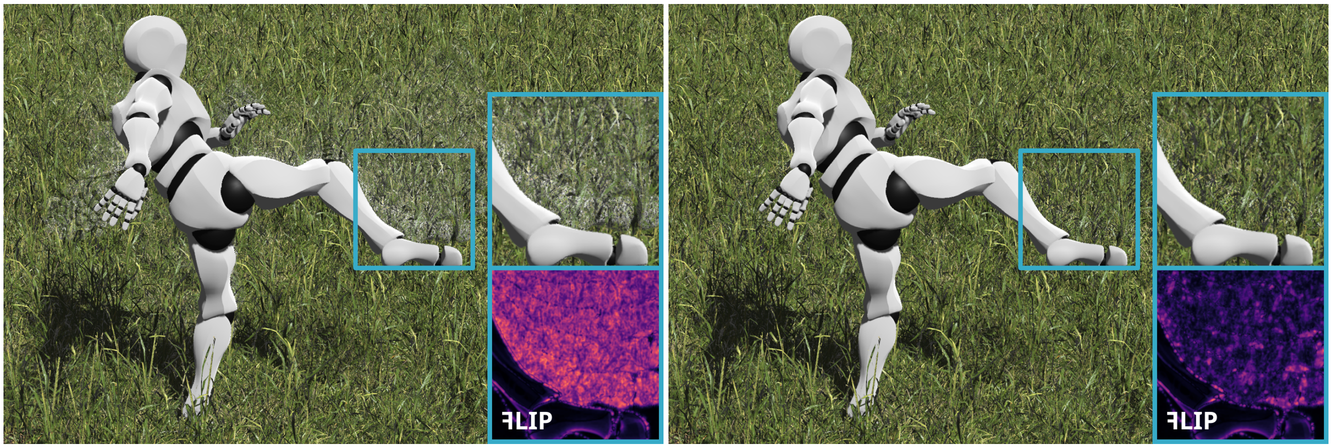


Figure 1: Left: neighborhood clipping with an AABB often allows colors which don't fit in the color neighborhood; this results in "ghosting" behind moving objects. Right: our k-DOP clipping more accurately rectifies invalid colors, thus mitigating ghosting.

Abstract

Temporal antialiasing is one of the most common methods for removing aliasing artifacts in contemporary real-time rendering, based on utilizing reprojected color data from previous frames. One typical issue with the method is ghosting: moving objects leaving a wake of visual trails. To mitigate this, a common technique is to validate and rectify reused history colors by comparing and clipping them to the color neighborhoods of the current frame's pixels. Previous, bounding box based methods are only situationally effective and cause significant ghosting in less favorable circumstances. We propose using k-Discrete Oriented Polytopes ("k-DOPs") for more robust neighborhood clipping. For a 0.2 ms performance overhead, our method more reliably mitigates ghosting across scenes where previous methods have inconsistent results.

SA Technical Communications '24, December 03–06, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1140-4/24/12
https://doi.org/10.1145/3681758.3697996

CCS Concepts

- **Computing methodologies → Antialiasing**.

Keywords

supersampling, antialiasing

ACM Reference Format:
Julius Ikkala, Tuomas Lauttia, Pekka Jääskeläinen, and Markku Mäkitalo. 2024. k-DOP Clipping: Robust Ghosting Mitigation in Temporal Antialiasing. In *SIGGRAPH Asia 2024 Technical Communications (SA Technical Communications '24), December 03–06, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3681758.3697996

1 Introduction

While Temporal Antialiasing (TAA) has become a mainstay in real-time graphics, its various issues have also become a common point of contention. TAA works by reprojecting pixel data from the previous frame to the current one, while applying small offsets ("jitter") to the image plane on each frame. This process supersamples the image over a few frames, but also causes a major issue: ghosting, where visual trails are left behind disocclusions. These artifacts cannot be easily removed with regular disocclusion detection algorithms, as they can simply re-introduce aliasing [Yang et al. 2020].

One common way to reduce this ghosting is by rectifying reprojected colors to fit within the neighborhood of the current frame's pixel. To do this, one constructs a bounding volume (typically an Axis-Aligned Bounding Box, AABB) in color space, over the colors of the current pixel's neighborhood. Then, reprojected colors are either *clamped* inside the bounding volume's range, or *clipped* to the bounding volume's shell with a ray casting operation between the reprojected color and current pixel color. In this paper, we discuss the latter approach, but the method should be applicable to clamping as well.

We propose using Discrete Oriented Polytopes (DOPs) [Klosowski et al. 1998] in neighborhood clipping to form tighter approximations around the convex hull of the color neighborhood. This greatly diminishes many visually obvious ghosting artifacts with only minor added complexity over AABB tests, and doesn't have major negative effects on performance or temporal stability.

2 Related work

The basis for TAA, "Amortized supersampling", was introduced in [Yang et al. 2009]. Its difficulties with spatio-temporal blurring were later improved by rectifying reprojected colors with neighborhood clamping in [Lottes 2011], and later with neighborhood clipping and using the YCoCg color space [Karis 2014]. These methods implement clamping/clipping with AABB bounding volumes, and are able to reduce ghosting for a very low performance cost. As temporal upsampling can be implemented as an extension of TAA [Yang et al. 2020], neighborhood clamping based on colors has been found useful in that context as well [Riley and Arcila 2022].

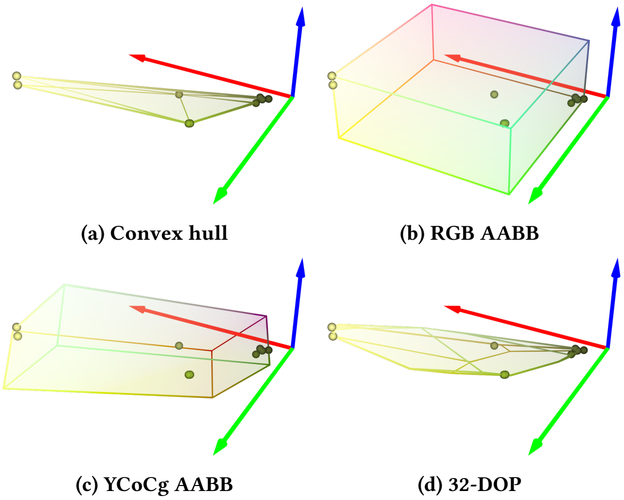
Variance clipping was introduced as an alternative that creates more compact AABBs based on the color variance of the neighborhood [Salvi 2016]. This method allows tradeoffs between ghosting and temporal stability with an adjustable parameter, γ, which the presentation recommends setting to 1.

There are various other methods for ghosting mitigation, but each has its own issues. Methods based on disocclusion detection are often prone to re-introducing aliasing [Yang et al. 2020]. A method based on ray tracing has been proposed [Marrs et al. 2018], but it has a steep performance cost even with hardware-accelerated ray tracing. Neural networks have also been used for adjusting weighting of temporal reuse [Herveau et al. 2023], but this is at least an order of magnitude slower than color clipping. As TAA is often reached for as a real-time AA solution, such multi-millisecond performance costs may be unacceptable. For further details of TAA, we refer to a survey of TAA techniques [Yang et al. 2020].

In contrast to RGB or YCoCg clipping, our method presents adjustable and less coarse clipping boundaries. This allows for mitigating ghosting in a more robust, less scene-dependent fashion, but also permits tuning for specific scenes as well.

3 Method

Ideally, neighborhood clipping rectifies colors which are not achievable with a weighted average over the neighborhood's colors. In geometric terms, the gamut of acceptable colors is the convex hull around the neighborhood's color. Unfortunately, convex hull construction algorithms are too inefficient to be used in practice [Yang et al. 2020]. Instead, TAA implementations typically use an AABB as the bounding volume, either in RGB or YCoCg color space.


(a) Convex hull **(b) RGB AABB**

(c) YCoCg AABB **(d) 32-DOP**

Figure 2: Visualization of various bounding volumes around a 3x3 pixel neighborhood, as used in TAA neighborhood clipping.

Looser bounding volumes can lead to invalid colors passing as valid. Figure 2 displays various bounding volumes over one 3x3 pixel neighborhood from Figure 1. For example, the RGB AABB approach is accepting some magenta colors in this case, despite the neighborhood not containing any similar tones.

To achieve tighter bounds for the neighborhood's color gamut, we propose to use discrete oriented polytopes. DOPs can be thought of as a generalization of AABBs: k-DOPs define extents over $\frac{k}{2}$ arbitrarily selected axes, instead of just the X, Y and Z axes. A k-DOP with more than three axes can achieve much tighter bounding volumes than AABBs (which is itself a 6-DOP).

The k-DOP construction and ray casting algorithms are similar to their AABB counterparts. For construction, we compute extents covering all points along each axis. For an axis A_i, the extents are $\min_{c \in C} c \cdot A_i$ and $\max_{c \in C} c \cdot A_i$, where C is the set of neighborhood colors. With AABBs, this dot product is implicitly done by accessing the R, G or B component. In practice, we dilate the ranges by $\epsilon = 10^{-5}$ to avoid floating point precision issues. Variance-based bounds can also be implemented similarly, by replacing component accesses with dot products and operating with more axes.

As our rays always start from inside the volume and k-DOPs are convex, the ray casting algorithm is simplified into finding the nearest ray intersection with axis-aligned planes positioned at the min/max extents of each k-DOP axis.

The number of axes has an impact on both clipping quality and performance: more axes allow tighter bounds but cause more computation. In practice, we found that around 16 axes provides a generally satisfying result in most cases. We propose two alternative approaches for selecting the k-DOP axes. As a general-purpose approach, we include the X, Y and Z axes and optimize the rest of the axes to achieve tightest bounds around a unit sphere. This approach should never be looser than an AABB and is not scene-dependent. We also propose a scene-specific approach: axes can be optimized for tightness around neighborhoods in aliased representative screenshots. This allows fewer axes to be used for similar anti-ghosting quality, which results in a slight performance benefit.

4 Experimental setup and results

To compare our method, we evaluate it against several common TAA neighborhood clipping approaches. Methods “RGB” and “YCoCg” are AABB neighborhood clipping in their respective color spaces, while “32-DOP” and “16-DOP” are our proposed k-DOP methods. “+Var” means that variance clipping (with $\gamma = 1$) is used, and “+Opt” means that the k-DOP axes were optimized for that specific scene.

We measure the performance of the TAA pass, amount of temporal instability, and amount of ghosting. In all cases, we let the renderer accumulate a history of 100 frames before starting the measurement, in order to make sure that full temporal history is available. We measure performance over 1000 frames with a fixed animation delta-time between frames on a GTX 1080 Ti. While relative performance differences are similar with an RTX 3090 as well, we highlight an older GPU to show the performance impact on less powerful hardware. The jitter pattern used is 10 entries long and generated with the R2 sequence [Roberts 2018].

Temporal instability or “shimmer” is measured over 100 frames, by averaging the mean ꟻLIP metric [Andersson et al. 2020] between successive frames without animation. This way, differences between frames are caused by the TAA algorithm’s potentially overeager discarding of history, which is affected by neighborhood clipping.

For ghosting measurements, we disable camera jittering. This causes TAA to no longer antialias, but still preserves ghosting caused by motion. We do it this way, because we cannot get a perfectly ghosting-free TAA ground truth sequence, but we can easily compute an aliased set of reference images. Comparing to (non-temporally) supersampled reference images would introduce a constant error by also comparing general TAA quality to supersampling. We compare 100 frames of animation to the aliased ground truth by using the average value of the mean ꟻLIP over all frames.

High-frequency color variation makes color gamuts in the neighborhoods large, thereby emphasizing clipping errors and thereby ghosting. We use two scenes to represent the kinds of color variation that are typical failure cases of TAA:

- **Grass** (Figure 4): a character kicking in front of a plain of grass. The background has high-frequency chromaticity and luminance variation, as well as lots of geometric edges prone to temporal instability. Results are shown in Table 1.
- **Asphalt** (Figure 3): a car driving on asphalt, with relatively few geometric edges. The asphalt has high-frequency luminance variation, causing ghosting. Results are shown in Table 2.

In all cases, the 32-DOP version is roughly 200 µs slower than the AABB methods, while the 16-DOP method is only 60-80 µs slower than the YCoCg variants. On an RTX 3090, the AABB methods take around 160 µs and 32-DOP takes 210 µs. Variance clipping seems to not majorly impact performance, but it always increases temporal instability and reduces ghosting. This is expected, as variance clipping can produce bounding volumes that don’t encompass the neighborhood, thereby rejecting more valid colors as well. While RGB performs decently in one context and YCoCg in another, our methods mitigate ghosting as well or better in both scenes.

Table 1: Measurement results for the “Grass” scene.

Method	Time (ms)	Shimmer (ꟻLIP)	Ghosting (ꟻLIP)
RGB	**0.449**	0.0346	0.0499
YCoCg	0.452	**0.0344**	0.0538
32-DOP	0.662	0.0433	0.0399
16-DOP+Opt	0.538	0.0491	0.0374
RGB+Var	0.465	0.0406	0.0417
YCocCg+Var	0.451	0.0411	0.0448
32-DOP+Var	0.665	0.0568	**0.0336**

Table 2: Measurement results for the “Asphalt” scene.

Method	Time (ms)	Shimmer (ꟻLIP)	Ghosting (ꟻLIP)
RGB	**0.428**	**0.0111**	0.0520
YCoCg	0.465	0.0117	0.0319
32-DOP	0.635	0.0120	0.0321
16-DOP+Opt	0.528	0.0130	0.0300
RGB+Var	0.466	0.0116	0.0439
YCocCg+Var	0.466	0.0134	0.0289
32-DOP+Var	0.653	0.0144	**0.0286**

Temporal stability and minimal ghosting are often contradictory goals [Yang et al. 2020]. Our measurements reflect this, as the least shimmering method ghosts the most and the most shimmering method ghosts the least in both tables. While our method with variance clipping produces the least ghosting in both scenes, it is less temporally stable than the other methods. Where this is unacceptable, we suggest using 32-DOPs without variance clipping, which provides a better tradeoff between anti-ghosting and shimmering.

5 Limitations

Neighborhood clipping on its own is not enough to completely remove all ghosting; even if a convex hull were to be used, the ghosting color may lie inside the neighborhood’s color gamut and go undetected. In practice, this manifests as antialiasing choices that bias towards the ghosting color. As the bias occurs over a large area, it can be noticeable, although less objectionable than colors completely outside of the neighborhood’s gamut. In the test scenes, this occurs with the character’s shadow and the car’s tires; both backgrounds have many dark pixels, which then allow black objects to leave ghosting. Methods that do not rely on the current frame’s color neighborhood may be necessary to mitigate these.

6 Conclusions

We introduce k-DOP clipping, a robust alternative to existing neighborhood clipping methods for mitigating ghosting in temporal antialiasing. We believe that it can have a major practical impact on TAA implementations in the wild. It allows a significant reduction in ghosting for a minor performance cost. To aid in implementations, we provide various sets of precalculated k-DOP axes and

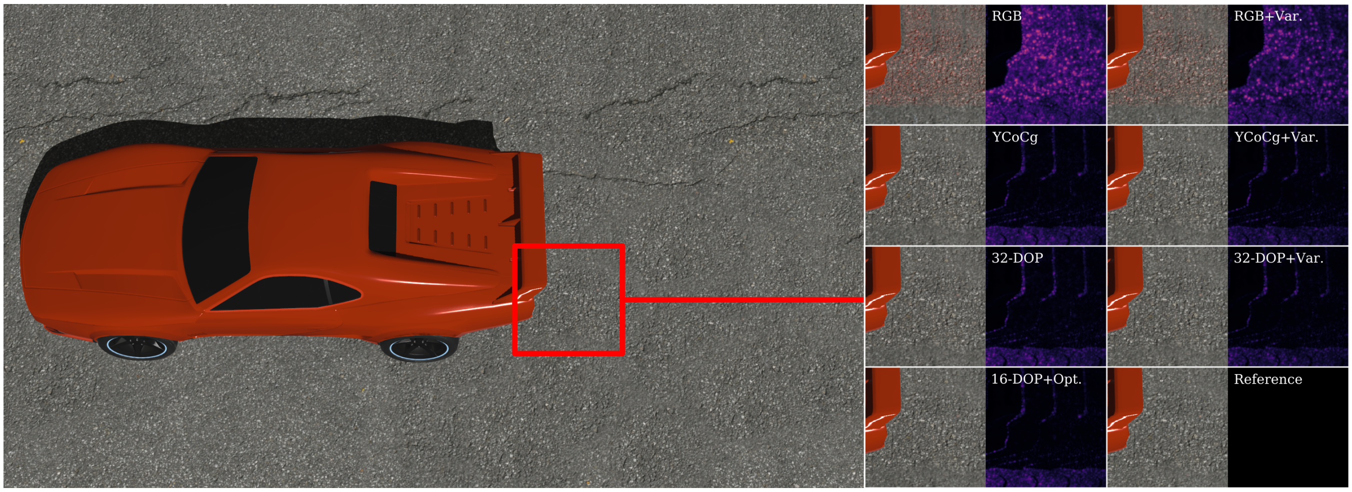


Figure 3: The "Asphalt" scene. The ghosting measurement's output images and ꟻLIP images are shown.

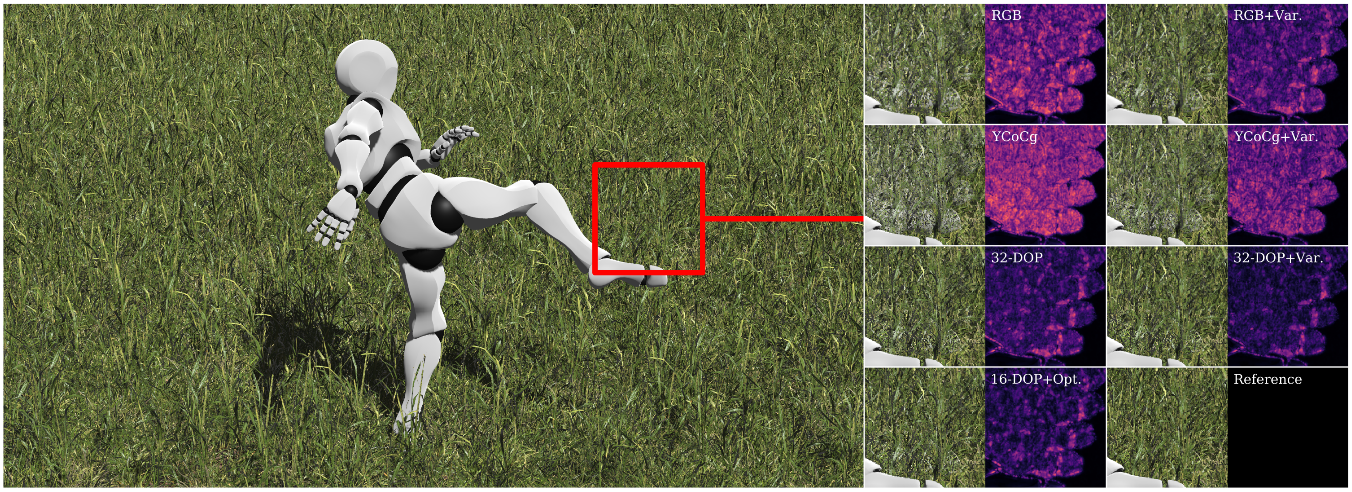


Figure 4: The "Grass" scene. The ghosting measurement's output images and ꟻLIP images are shown.

example shader code in supplemental material. Assuming a baseline TAA implementation with AABB-based neighborhood clipping, our method does not require any additional hardware capabilities or input data from the renderer.

Acknowledgments

This work was supported by the Research Council of Finland under Grant 351623. The car model, recolored to red, is "Cyberpunk car" (https://skfb.ly/6QUAI) by 4d_Bob, licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

References

Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D. Fairchild. 2020. ꟻLIP: A Difference Evaluator for Alternating Images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 3, 2 (2020).

Killian Herveau, Max Piochowiak, and Carsten Dachsbacher. 2023. Minimal Convolutional Neural Networks for Temporal Anti Aliasing. In *High-Performance Graphics - Symposium Papers*. The Eurographics Association.

Brian Karis. 2014. High Quality Temporal Supersampling. In *ACM SIGGRAPH Courses: Advances in Real-Time Rendering in Games*.

J.T. Klosowski, M. Held, J.S.B. Mitchell, H. Sowizral, and K. Zikan. 1998. Efficient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics* 4, 1 (1998).

Timothy Lottes. 2011. TSSAA (Temporal Super-Sampling AA). http://web.archive.org/web/20120120082628/http://timothylottes.blogspot.com/2011_04_01_archive.html. Accessed: 2024-07-25.

Adam Marrs, Josef Spjut, Holger Gruen, Rahul Sathe, and Morgan McGuire. 2018. Adaptive Temporal Antialiasing. In *ACM SIGGRAPH / Eurographics High Performance Graphics*. https://casual-effects.com/research/Marrs2018Antialias/index.html

Colin Riley and Thomas Arcila. 2022. FidelityFX Super Resolution 2.0. In *Game Developers Conference*.

Martin Roberts. 2018. The Unreasonable Effectiveness of Quasirandom Sequences. https://extremelearning.com.au/unreasonable-effectiveness-of-quasirandom-sequences/. Accessed: 2024-07-25.

Marco Salvi. 2016. An Excursion in Temporal Supersampling. In *Game Developers Conference*.

Lei Yang, Shiqiu Liu, and Marco Salvi. 2020. A Survey of Temporal Antialiasing Techniques. *Computer Graphics Forum* 39, 2 (2020).

Lei Yang, Diego Nehab, Pedro V. Sander, Pitchaya Sitthi-amorn, Jason Lawrence, and Hugues Hoppe. 2009. Amortized supersampling. *ACM Trans. Graph.* 28, 5 (2009).