Vision, Modeling, and Visualization (2013)
Michael Bronstein, Jean Favre, and Kai Hormann (Eds.)

*Poster*

# Sampled and Analytic Rasterization

Thomas Auzinger† and Michael Wimmer

Institute of Computer Graphics and Algorithms
Vienna University of Technology, Austria

**Abstract**

*In this poster we present an overview of exact anti-aliasing (AA) methods in rasterization. In contrast to the common supersampling approaches for visibility AA (e.g. MSAA) or both visibility and shading AA (e.g. SSAA, decoupled sampling), prefiltering provides the mathematically exact solution to the aliasing problem. Instead of averaging a set a supersamples, the input data is convolved with a suitable low-pass filter before sampling is applied. Recent work showed that for both visibility signals and simple shading models, a closed-form solution to the convolution integrals can be found. As our main contribution, we present a classification of both sample-based and analytic AA approaches for rasterization and analyse their strengths and weaknesses.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

## 1. Introduction

The term rasterization is used for two different, but related concepts in computer graphics. Generally, it is defined as the process of converting vector information to a raster format. It also refers to a 3D rendering technique, where intermediate buffers throughout the graphics pipeline have the same raster format as the final output image. In this work we take both aspects into account and regard the second definition as the main application of the first.

For many fundamental tasks in image processing, such as acquisition, manipulation and display, raster formats in the form of regular grids of color values are prevalent due to their ease of use and their resemblance to sensor arrays in cameras and displays. A wide range of synthetic scene data is stored in vector format, however, such as the graphical representation of fonts with splines or the geometrical description of surfaces with meshes. Rasterization serves as a conversion process between these formats and constitutes a central element in visual computing.

From a signal-theoretical standpoint, rasterization can be seen as a regular sampling of the input data. Both the 'precise' vector input and densely sampled raster input (e.g. textures) are (re)sampled at the pixel locations of the output image. Generally, this causes a loss of information and the *Nyquist-Shannon-Theorem* describes the theoretical limitation of a given sampling pattern. It relates the spacing of the sample locations to the level of detail that can be preserved. This is described by an upper threshold – the *Nyquist frequency* – of the signal frequencies that can be faithfully retained by the sampling.

If the input signal exhibits frequencies above the Nyquist frequency, not all details can be preserved during the rasterization process. Furthermore, these frequencies do not simply vanish but result in distortions of the sampling values and cause artifacts in the output, called *aliasing*, such as jaggies or Moiré patterns.

## 2. Anti-Aliasing

Input data of typical rasterization tasks exhibits infinitely high frequencies in the form of discontinuities such as object or text silhouettes. Hence, rasterization usually suffers from aliasing and methods that combat this behavior are subsumed under the term *anti-aliasing* (AA).

The two main methods to eliminate frequencies beyond the Nyquist frequency are *supersampling* and *prefiltering*. The former samples the input signal at a higher sampling rate and averages these values with a suitable filter. This method

† thomas.auzinger@cg.tuwien.ac.at



DOI: 10.2312/PE.VMV.VMV13.223-224

| | | Visibility | | | |
|---|---|---|---|---|---|
| | | Trivial | Sampled | Supersampled | Prefiltered |
| Shading | Sampled | Standard | Standard | MSAA, CSAA | NSAA [AMPW13] |
| | Supersampled | Font rasterization [BBD*00] | - | SSAA<br>Decoupled [RKLC*11] | **Future work**<br>Best quality |
| | Prefiltered | Analytic 2D [AGJ12] | - | - | Analytic 3D [AWJ13] |

**Table 1:** *Classification of anti-aliasing methods for rasterization. See Section 3 for details.*

does not completely remove aliasing but merely increases the Nyquist frequency. However, supersampling is the dominant method to perform AA due to its low runtime complexity, simple implementation and trivial parallelization. Prefiltering, on the other hand, applies a low-pass filter to the input data before the actual sampling. While yielding mathematically perfect AA, this method is computationally demanding, as the input data has to be convolved analytically.

## 3. Classification of Anti-Aliasing Methods

In rendering tasks, usually two main components of the input are identified: The *visibility* component describes the visible parts of the input scene while the *shading* component represents the interaction between light and the surface materials. AA of visibility is more demanding due to its binary nature and view direction dependence whereas shading AA can be partly achieved by preprocessing, e.g. texture mipmapping.

In Table 1, an overview of both supersampling and prefiltering based AA methods is given and classified according to their treatment of the visibility and shading component. Three AA methods are differentiated: *Sampled* denotes a single sample per output pixel, whereas several samples are used for *supersampling*. *Prefiltering* employs a filter convolution before taking one sample per pixel. An additional column for rasterization applications with *trivial* visibility denote input scenes that do not require visibility computations (e.g. font rasterization). Methods that put more effort into shading AA than visibility AA are uncommon and marked with '-'. Note that this table merges several design dimensions of general rasterization settings such as primitive complexity (lines, curves, triangles, curved surfaces), camera effects (motion blur, depth of field), hardware implementation (CPU, GPU), etc., into the two presented components.

## 4. Analysis of Analytic Anti-Aliasing

In the last years, we contributed various methods for prefiltering-based AA of both visibility and shading signals for common rasterization tasks. One key observation is that prefiltering of shading signals is only possible for very simple light-material interaction models. So far, it was shown that only prefiltering of polynomial functions can be computed analytically, which includes flat shading and linear color gradients in image space. The filter convolution with non-linear shading models such as perspective-correct Gouraud shading or Phong shading cannot be computed in closed-form. Visibility signals, however, are binary and can be prefiltered exactly, which is shown in our paper at VMV 2013 [AMPW13]. The 'future work' entry of the classification in Table 1 is a combination of visibility prefiltering and shading supersampling. This will yield the highest possible quality for non-linear shading models but requires a significantly improved supersample placement, as current MSAA approaches use only a crude approximation.

Prefilting requires a preservation of the vector format of the input data throughout the analytic graphics pipeline. Traditional acceleration methods, such as depth-buffering, cannot be used and exact hidden surface elimination has to be performed. This requires robust geometric intersection computations and tile-based rasterization has to be used due to large intermediate buffers. Although the traditional graphics APIs do not map to this design, our experience showed that analytic prefiltering is still embarrassingly parallel and we succeeded in efficiently implementing our algorithms on massively parallel hardware using the GPGPU programming language CUDA.

Performance evaluations showed that prefiltering is 2-3 orders of magnitude slower than massive supersampling despite these efforts and the fact that rasterization hardware cannot be used. While we do not expect that analytic rasterization will become prevalent for real-time application, promising areas of application are high-quality rasterization of static content, ground truth computations for sampling-based effects and the repurposing of the new pipeline design to output intermediate results in vector format.

## References

[AGJ12] AUZINGER T., GUTHE M., JESCHKE S.: Analytic anti-aliasing of linear functions on polytopes. *Comput. Graph. Forum 31*, 2pt1 (2012). 2

[AMPW13] AUZINGER T., MUSIALSKI P., PREINER R., WIMMER M.: Non-sampled anti-aliasing. In *Proc. VMV* (2013). 2

[AWJ13] AUZINGER T., WIMMER M., JESCKE S.: Analytic visibility on the GPU. *Comput. Graph. Forum 32*, 2pt4 (2013). 2

[BBD*00] BETRISEY C., BLINN J., DRESEVIC B., HILL B., HITCHCOCK G., KEELY B., MITCHELL D., PLATT J., WHITTED T.: Displaced filtering for patterned displays. In *Proc. Soc. Inf. Disp. Symp.* (2000), pp. 296–299. 2

[RKLC*11] RAGAN-KELLEY J., LEHTINEN J., CHEN J., DOGGETT M., DURAND F.: Decoupled sampling for graphics pipelines. *ACM Trans. Graph. 30*, 3 (May 2011), 17:1–17:17. 2