Circular Separable Convolution Depth of Field

Kleber Garcia
Electronic Arts Inc.
1950 Summit Park Dr
Orlando, Florida 32810
kgarcia@ea.com

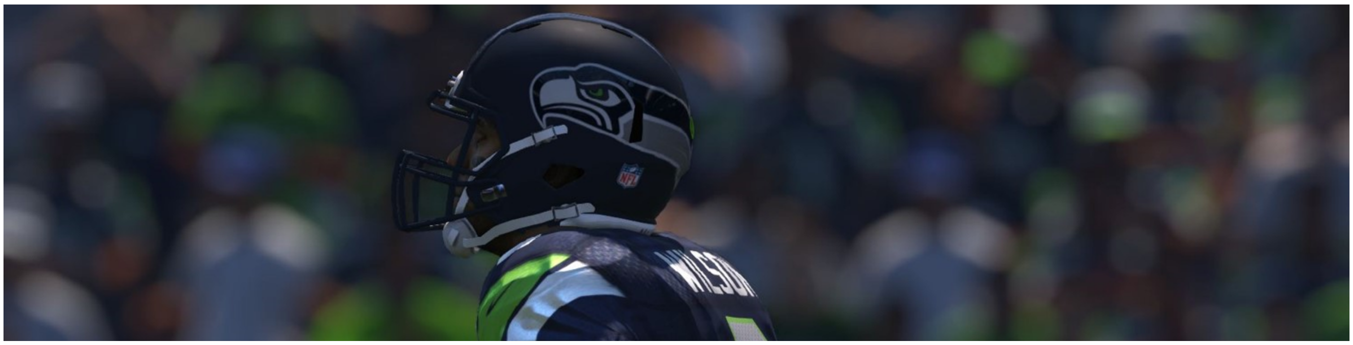


Figure 1: Real time render technique in Madden NFL 16

ABSTRACT

Circular Separable Convolution Depth of Field (CSC DoF) is a mathematical adaptation and implementation of a separable circular filter, which utilizes complex plane phasors to create very accurate and fast bokeh. At its core, this technique convolves a circular pattern blur in the frequency domain using a horizontal and a vertical pass, representing the frame buffer using complex numbers. This technique renders at magnitudes faster than brute-force and sprite-based approaches, since it is a separable convolution. Important properties of this technique include convolution separability, low memory bandwidth and large radii circles. The technique has been shipped on Madden NFL 15, Madden NFL 16, Madden NFL 17, Fifa 17 and PGA Tour Rory McIlroy. The implementation includes an offline shader code generation step containing pre-computed frequency domain filters, multiple weighted passes for imaginary and real number processing. We will present the mathematical derivation and some caveats to achieve the required precision for intermediate frequency domain frame buffers.

CCS CONCEPTS

• **Computing methodologies → Computational photography**; *Rendering*; *Rasterization*;

KEYWORDS

depth of field, bokeh, separability, complex phasors, post effects

SIGGRAPH '17 Talks, July 30 - August 03, 2017, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5008-2/17/07.
https://doi.org/10.1145/3084363.3085022

ACM Reference format:
Kleber Garcia. 2017. Circular Separable Convolution Depth of Field. In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
https://doi.org/10.1145/3084363.3085022

1 INTRODUCTION

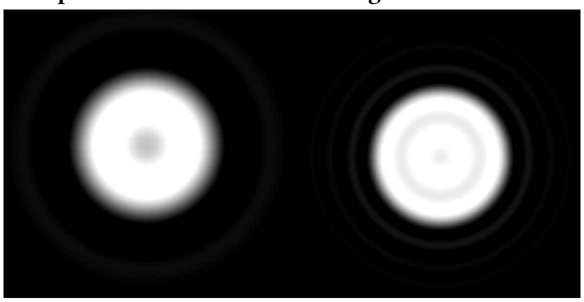
The idea is to achieve a reduction of time complexity when applying a gather convolution. A basic Gaussian convolution can achieve linear time complexity by performing a horizontal and vertical pass. This is not possible when the target shape of the convolution is a circle. Previous research [Niemitalo 2010] proves that performing a separable blur in the frequency domain will be able to achieve separability, thus reducing time complexity and providing nice properties of a standard gather convolution. This research provides us with the complex phasor described in Equation 1. For simplicity sake, it is decided to create a one component low quality filter and a two component high quality filter. Each filter generates a blurred image by adding the results of each one of its respective components. The filter values can be found in Table 1 with visual results in Figure 2.

$$\begin{aligned}
F_c(x) &= |F_c(x)|[cos(argF_c(x)) + isin(argF_c(x))] \\
F_c(x) &= e^{-a_c x^2}[cos(b_c x^2) + isin(b_c x^2)] \\
\text{Color}(x) &= \sum_{c=0}^{C}[A_c F_c(x)_r + B_c F_c(x)_i] \\
&\text{where} \\
C &: \text{component parameter tuple}(a_c, b_c, A_c, B_c) \\
x &: \text{current sample radius} \\
F_c(x)_r, F_c(x)_i &: \text{real and imaginary parts respectively.}
\end{aligned} \tag{1}$$

Table 1: Low and high quality complex filters with their respective coefficients.

Filter component C	(a,b,A,B)
Low quality filter c_0	(0.862325,1.624835,0.767583,1.862321)
High quality filter c_0	(0.886528,5.268909,0.411259,-0.548794)
High quality filter c_1	(1.960518,1.558213,0.513282,4.561110)

Figure 2: Low (left) and high quality (right) complex filter visual representation. The more components, the closer circular representation and less banding.



2 COMPLEX FILTERS IN SHADERS

An offline pass generates an HLSL header, which contains precomputed normalized complex weight values. This helps reducing register usage and guarantees constant folding. These array weights $G_c(x)$ contain a complex number represented as a real and imaginary pair for every component C and for every texel location X. The reason why we express this as a new function G_c is because is normalized, unlike F_c. Equation 2 shows how these array values are computed.

$$G_c(x) = \frac{1}{\alpha} F_c(x) \tag{2}$$

Equation 3 shows how to compute the normalization coefficient α^2. Notice it is squared because we are adding all the complex number weights on a grid of R^2, where R is the max size radius keeping in mind that $G_c x$ is a one dimensional function for separability.

$$\begin{aligned}\alpha^2 = \sum_{c}^{C} \sum_{x=-R}^{R} \sum_{y=-R}^{R} [&A_c(F_c(x)_r F_c(y)_r - F_c(x)_i F_c(y)_i) \\ &+B_c(F_c(x)_r F_c(y)_i + F_c(x)_i F_c(y)_r)]\end{aligned} \tag{3}$$

The result of this pass will be a header file included in the main shader post effect passes.

2.1 Bracketing the Filter for Precision

Precision bit issues could occur due to large and negative values from the kernel G_c. In order to efficiently compress these numbers, the domain $[0, 1]$ is used. This version of the filter is only required for the intermediate render passes that must store large negative unbounded values. Assuming we have a 1 dimensional kernel G_c, we can transform to a bracketed kernel G'_c using Equation 4 where O_c is an offset and S_c is a scale, such that the domain of G'_c is $[0, 1]$. The real and imaginary components will have their own offset and scale tuples respectively.

$$G'_c(x) = \frac{G_c(x) - O_c}{S_c} \tag{4}$$

We can then convolve an HDR image I into an 8 or 10 bit storage I'_{blur} where $I'_{\text{blur}} = \sum_{x=-R}^{R} G'_c(x) I_x$. Our goal is to recover$I_{\text{blur}}$ which will contain our non-bracketed convolution result. If we know the value of I'_{blur} we can apply some basic algebra and recover I_{blur} as shown in Equation 5.

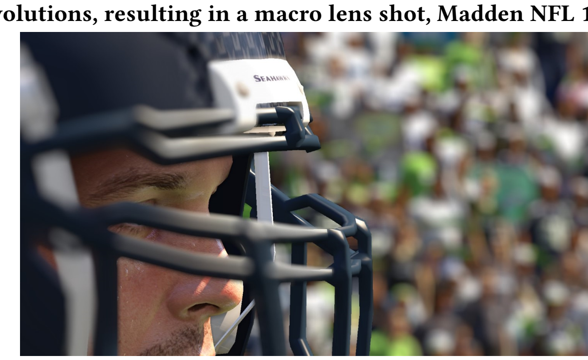
$$I_{\text{blur}} = I'_{\text{blur}} S_c + \sum_{x=-R}^{R} I_x O_c \tag{5}$$

This results in packing unbounded numbers inside render targets without precision loss. Circle of confusion (coc) values are stored in sRGB targets in order to keep precision and avoid banding during blending. Alternatively, coc values could be stored using a power of 2, to allocate most of the bit precisions at low values of coc (which is where precision is important due to blending).

2.2 Composite of Blurs

The multiple passes of the post effect generate two images. One is the near blur, which uses the low quality filter. The reason for this is that the near blur appears less frequently. The near blur utilizes a lower resolution tiled coc buffer, using min coc values. This is so there is edge bleeding when bluring from the front. The far blur utilizes the high quality filter. Figure 3 shows the three images that get composited together (near, far and focus).

Figure 3: Composited result of Near, Far, and In-Focus convolutions, resulting in a macro lens shot, Madden NFL 17.



3 ACKNOWLEDGEMENTS

Arthur Homs, Ishaan Singh, Dan McCallum and Karolyn Boulanger for their contributions.

REFERENCES

Ollie Niemitalo. 2010. Circularly symmetric convolution and lens blur. (2010). http://yehar.com/blog/?p=1495