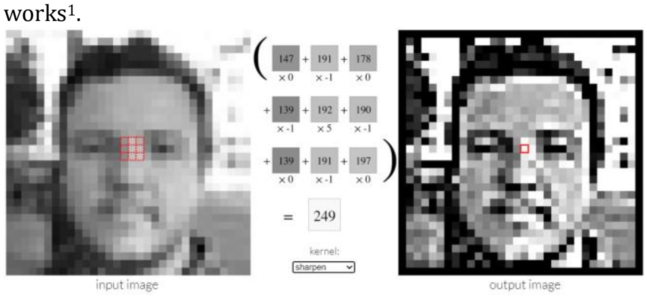
Austin Jetrin Maddison 6481268

PROJECT PROPOSAL
Circular Separable Convolution Depth of Field

CONVOLUTIONAL IMAGE PROCESSING

CONVOLUTIONAL Image Processing is abundantly used technique used in many domains to achieve a variety of effects due to speed, versatility and simplicity. Common effects such as blurring and sharpening are also produced by this technique. This technique lays the foundation for many modern image processing algorithms such as Circular Seperable Convolution Depth of Field (CSCDoF). The core of convolutional image processing is a convolution matrix better known as a kernel. Every pixel in the image gets processed through the kernel to create the processed image. The kernel contains weights that determine the contribution neibouring pixels in the creation of a new processed pixel. Meaning to achieve different effects such as bluring or sharpening all that needs to be done is the change of weights in the kernel.

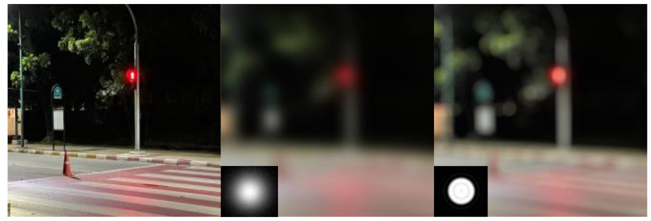
Here is a intutive illustration made by Victor Powell on how it works[1].



The red component on the left indicates the 3x3 kernel on the input image, highighting the group of neibouring pixels used to generate the final red outlined pixel on the right. In the middle section, it shows the kernel's weights and they combine with the neibouring input image pixels summation that result in the final output pixel. This procedure is called a convolution hence the name of the method.

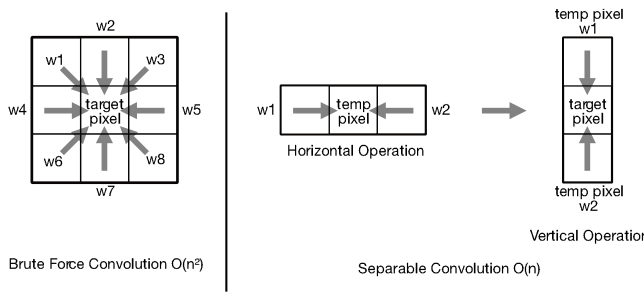
CIRCULAR SEPARABLE CONVOLUTION DEPTH OF FIELD (CSCDoF)

CSCDoF, is an algorithm developed at Eletronic Arts Inc. (EA) that aims to simulate physically correct looking camera lens blur on an input image for realtime applications[2]. Lens blur has a distinctive disc shape (bokeh) which is not reproducible with naïve approaches such as box or guassian blur algorithms. The problem with the naïve approaches of simulating lens blur is that the falloff the the gaussian and box functions defined in the kernel do not model the disc shape bokeh manifested by the camera apeture accuratley. This can be solved by describing the convolution as circle instead of a gaussian or box.

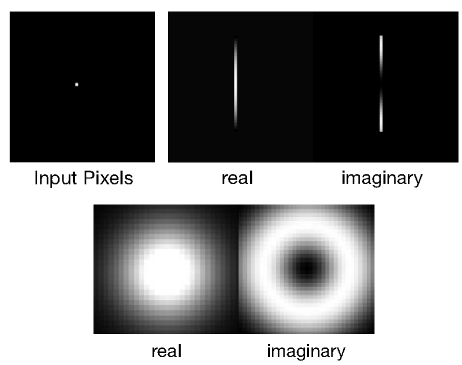


Guassian convolution in middle, Circular convolution on the right.

However changing the convolution to model a circle makes the time complexity non-linear which is expensive and undesirable for realtime applications, specifically video games. This is because when the convolution is modeled as a circle it isn't obvious how to make the blur seperable into 2 operations in the domain of real numbers: vertical blur followed by a horizontal blur. The seperability of these operations is important to performance as it requires more bandwidth and work to compute brute force convolution.



However no fear, smart people have figured out how to reduce the time complexity by making circular convolution seperable by using a precomputed imaginary component to the kernel to construct the circular convolution that is able to produce a good approximation of the brute force circular convolution[3].

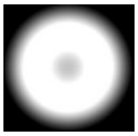


$$F(x) = e^{-ax^2}(\cos(bx^2) + i\sin(bx^2))$$

[1]Image Kernels explained visually
https://setosa.io/ev/image-kernels/ setosa.io

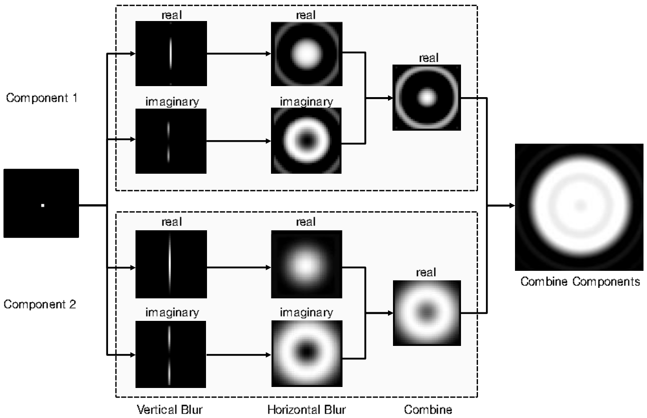
[2] Kleber Garcia. 2017. Circular Separable Convolution Depth of Field. In Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017, 2 pages. https://doi.org/10.1145/3084363.3085022

[3] Circularly symmetric convolution and lens blur
http://yehar.com/blog/?p=1495

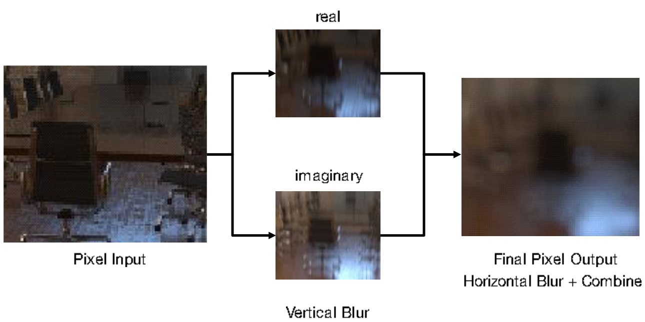


real + imaginary

$$Color(x) = A * F_{real}(x) + B * F_{imaginary}(x)$$



Example of convolution with 2 components, more components the more accurate the circular shape.

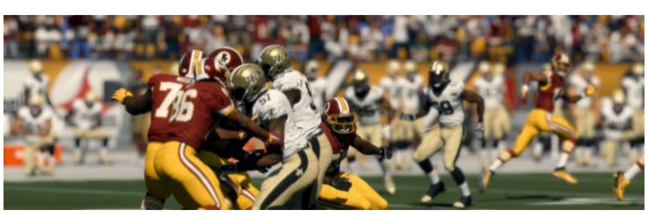


Example of convolution on a input image image.

This makes the CSCDoF distinctivley fast compared to more expensive methods of renddering bokeh depth of field such as sprite based bokeh.

PUBLICATIONS

The research by Olli Niemitalo in 2010 showed that seperability of the circular convolution operations was possible. This research inspired further research in EA which resulted in the development of CSCDoF.



EA's Madden was shipped with this CSCDoF.

Now the CSCDoF has been adapted in modern realtime render engines as the defacto technique for rendering high quality, fast and accurate bokeh depth of field.

Epic's Unreal Engine 5 ships with CSCDoF as the default depth of field post processing algorithm.

PROJECT PROPOSAL SPECIFICATION

I propose to implement Circular Separable Convolution Depth of Field into Unity game engine's render pipeline. This will be done by creating a custom render pipeline inside Unity using HLSL and C# to define buffers and shaders to facilitate the procedures described in EA's research paper.

The depth of field effect parameters (shader uniforms):

⟶ *Circle of confusion radius (Bokeh radius)*

If time permits a plane of focus can be defined using near and far plane parameters:
⟶ *Near plane z-position.*
⟶ *Far plane z-position.*

Convolutional image processing is embarassingly parallel because each final output pixel can be computed independently from each other which makes it an ideal candidate for parallelism on the GPU using shaders.

To test if my implmentation of the algorithm is successful I will compare the output from my implementation of the algorithm and the ground truth images of bokeh depth of field. Furhtermore I will perform benchmarks of my GPU implementation against somebody else's CPU implementation of the algorithm and compare its running time. We would expect the GPU to perform better than the CPU by a large margin as the GPU has more cores than the CPU.

I chose image processing as my final project because the nature of the algorithms in this domain lend themselves well to parralism. Moreover, this project offers a valuable addition to my personal portfolio. Given I aspire to work in the domain of computer graphics, I see it as an opportunity to elegantly merge this aspect in my final project.