Quentin Farrell
WR 227 Section 401
October 29th 2023

Blur: Expanded Definition

Low Context: General Public

Blur is a widely used image effect which softens edges and covers up fine details. Figure 1 shows blur in use.

Figure 1: The blur effect removes details and makes edges less defined.

Images and Colors on a Computer

To describe how blur affects images, it is essential to understand the basics of digital images. Images on computers are a very fine grid of colored squares called ‘pixels.’ Every color consists of 3 separate numbers: red, green and blue (RGB) [1]. Each number or channel controls how much light of the channel’s specific color is emitted from the screen. Mixing of different amounts of these three lights results in every perceivable color from a computer screen. When a computer program (such as blur) considers an image, all it really sees is a massive grid of triplets of numbers. Any program that will modify an image will need to read and change these numbers.

Mixing colors involves selecting a value for each RGB channels that is between that of the input colors of that channel. There is an example in figure 3.

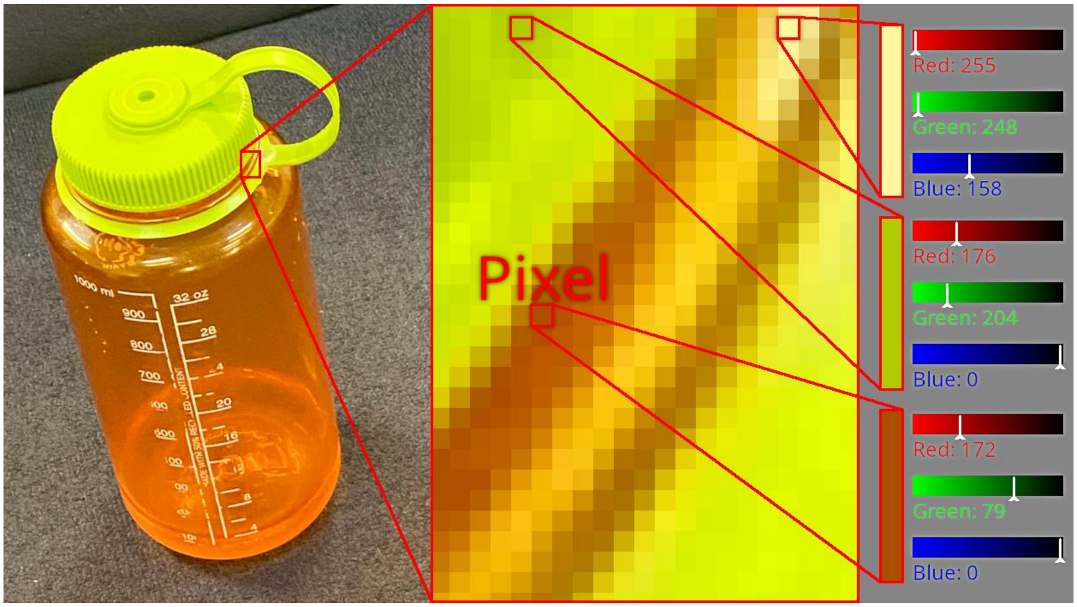

Figure 2: On computers images are stored as a grid of a thousand to a billion colored squares called pixels. The colors of the pixels are stored as 3 numbers defining how the red, green, and blue content. [1]

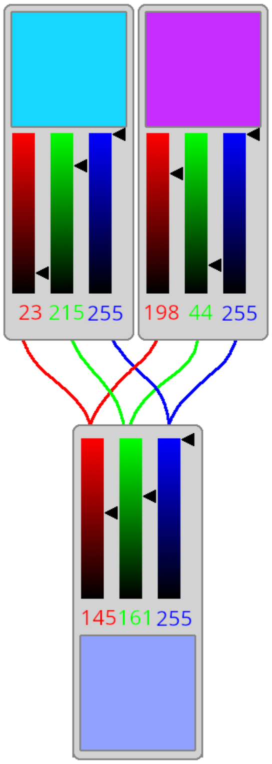

Figure 3: A cyan color with a lot of green and blue can be mixed with a magenta color with a lot of red and blue and the result will have a lot of blue, and less of the red and green which is a light blue.

The Blurring Algorithm

The purpose of blur is to reduce the sharpness of edges – or to smear the impact of a sudden change out (so it is now less sudden). For blur, pixels now become affected by the pixels adjacent to them – specifically become more similar.

Mixing pixels together with their neighbors will result in a color that is affected by all the colors around it. Making a new image where every pixel is a mix of neighboring pixels would result in all pixels becoming more similar to their neighbors. Thus, it blurs the image. [2]

In short, blur makes every pixel in the resulting image be a mix of the colors of neighboring pixels from the input image.

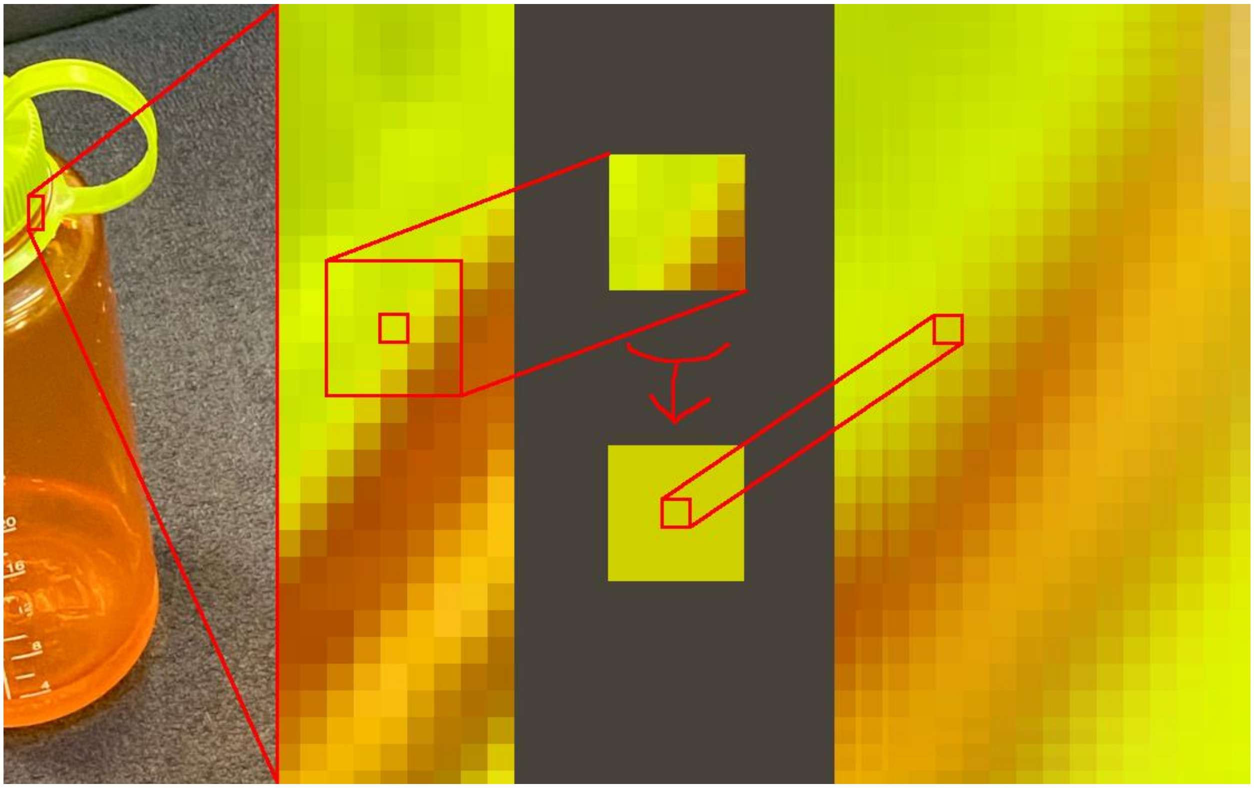

Figure 4: For every pixel in an image, a section around it is mixed together then the result is placed where the pixel was in the result image. After being applied to every pixel, the result image is a blur version of the original.

There are multiple different types of blurs all which follow the same basic principle using different definitions for how colors are mixed and which pixels are considered neighbors.

High Context: Computer Science Students

Blur is a widely used image effect which softens edges and reduces high frequency details. Figure 1 shows blur in use. This document will focus on blurring bitmap images in the sRGB color space.

Figure 5:This gaussian blur effect removes high frequency details and makes edges less defined.

Mixing Colors

A key part of most types of blurring algorithms involves mix colors together. A naïve attempt to mix sRGB colors is to average each color channel. However, this does not work (see figure 2), as your monitor does not output the pure RGB values, but instead the square of each channel. This is done to give a larger selection range of dark colors. [3] Given that, when mixing our sRGB colors, we must be mindful of this (see figure 3).

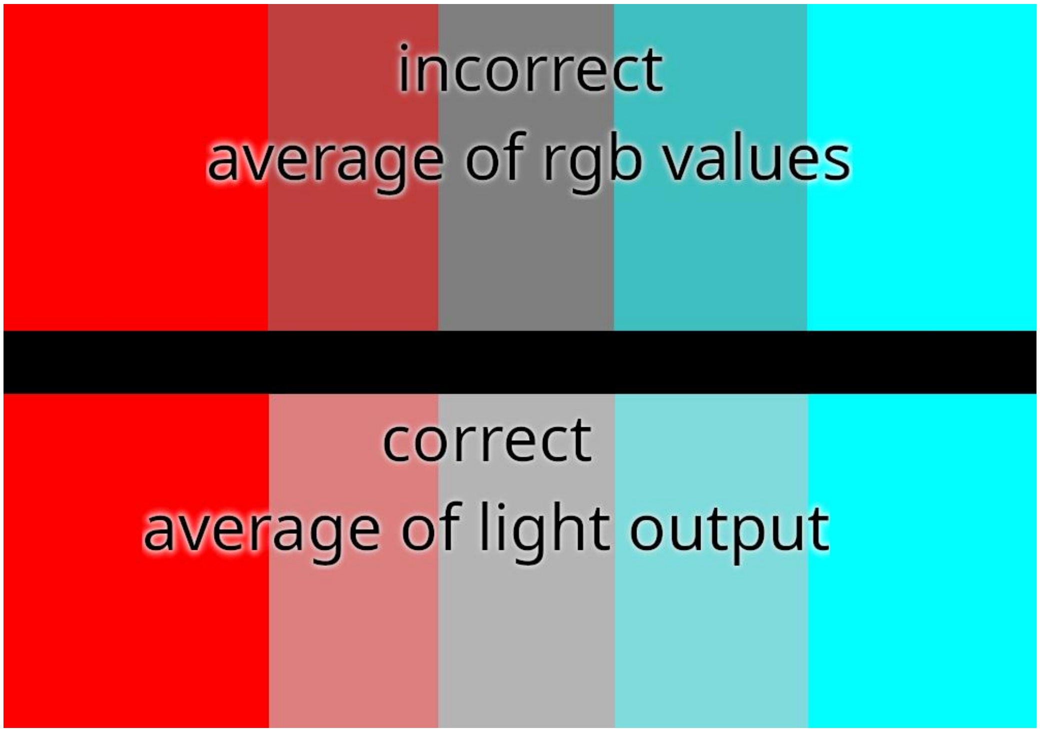

Figure 6: Averaging the RGB values instead of the light values is a common color mixing mistake that has existed in many softwares in the past from Photoshop to iPhone lock screen. [3]

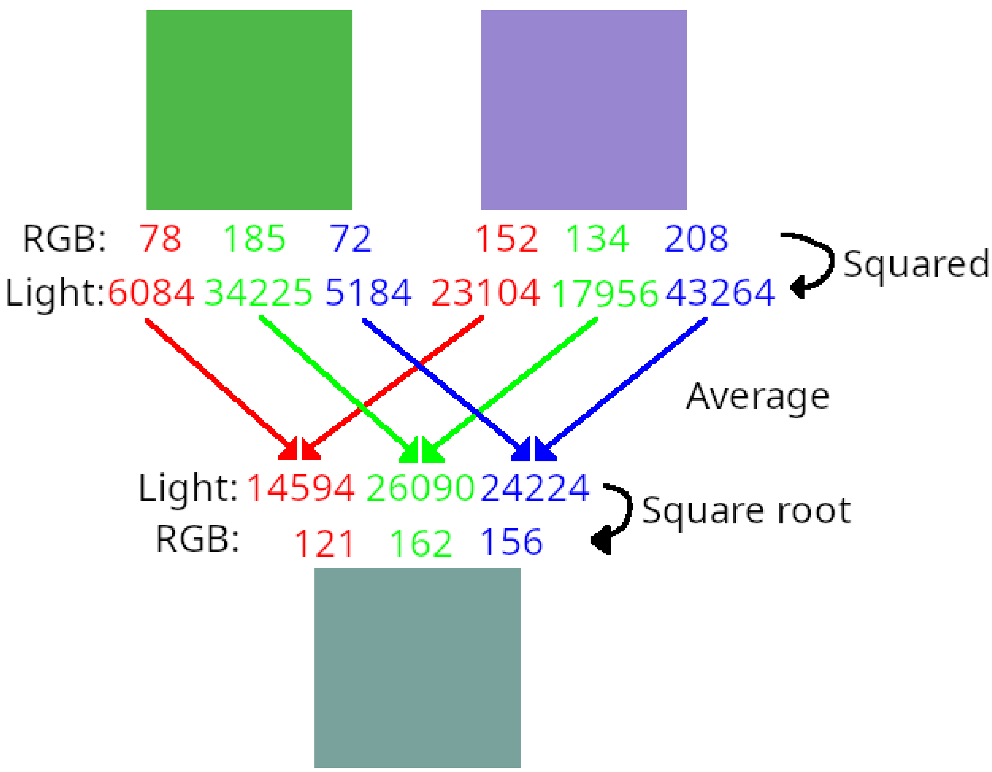

Figure 7: To mix sRGB colors, the red, green and blues are first squared then averaged then square rooted to produce a final result.

The Blurring Convolution

The purpose of blur is to smooth the image so changes become less harsh. Pixels need to affect their neighbors to become more similar.

The technique used is called a discrete convolution, where every pixel is a mixture of other pixels around it.

Convolving over the images mixes pixels together with their neighbors will result in a color that is affected by all the colors around it. This makes a new image where every pixel is more similar to their neighbors. Thus, it blurs the image. [2]

Which pixels are considered neighbors is determined by the convolution's kernel. Different blurs have different kernels. For now, a square kernel will be used resulting in a box blur.

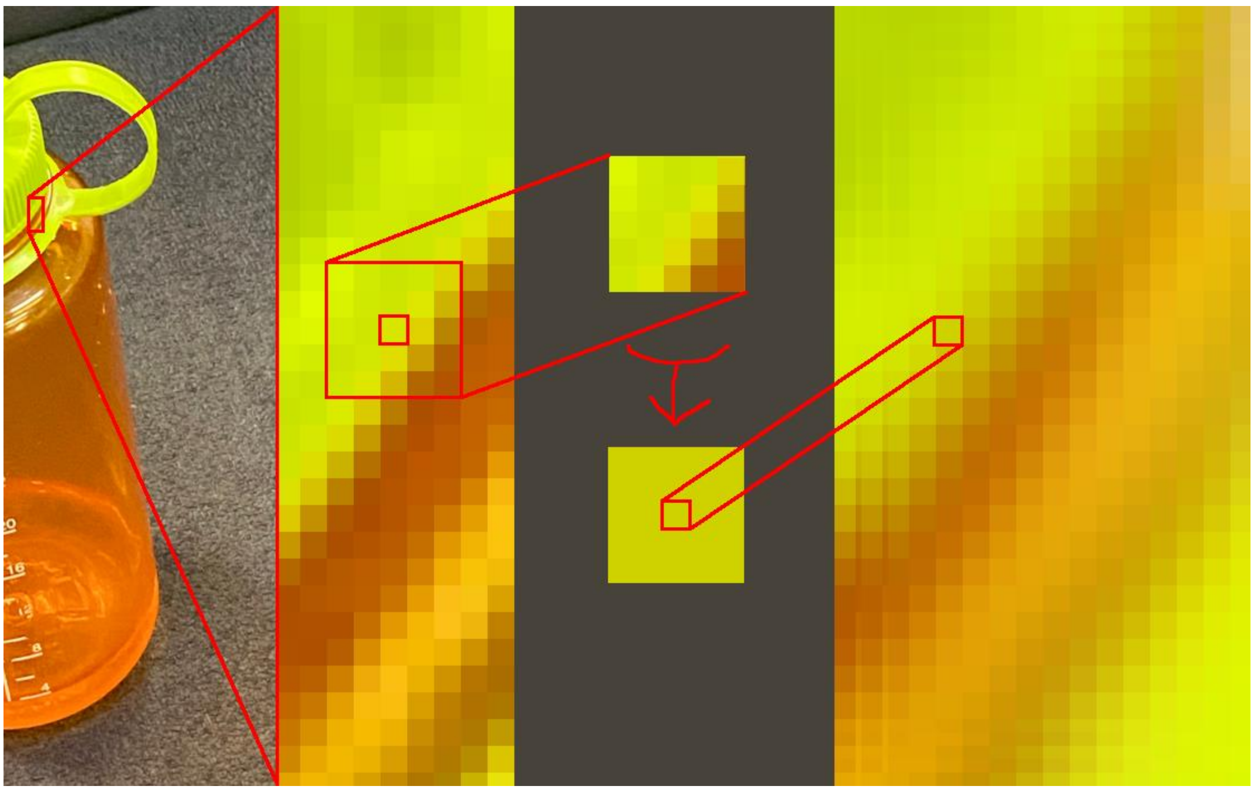

Figure 8: For every pixel in a image, a section around (know as the kernel) it is mixed together then the result is placed where the pixel was in the result image. After being applied to every pixel, the result image is a blur version of the original.

Optimization

As the size of a desired blur grows, the size of the kernel must grow. A 3x3 kernel will require every pixel to mix 9 colors, however a 23x23 kernel will require to mix 529 colors per pixel. Often, we want very blur images very fast and an algorithm that takes quadratic time, $O(n^2)$, will not cut it.

If a vertical kernel for example 1x23 then it will only mix 23 colors, however it yields a result know as vertical linear motion blur – not box blur. However, if a horizontal kernel (23x1) follows that then the result will be the same, but it will only require considering 46 colors, and in fact now takes only linear time, O(n).

This process is shown in figure 5.

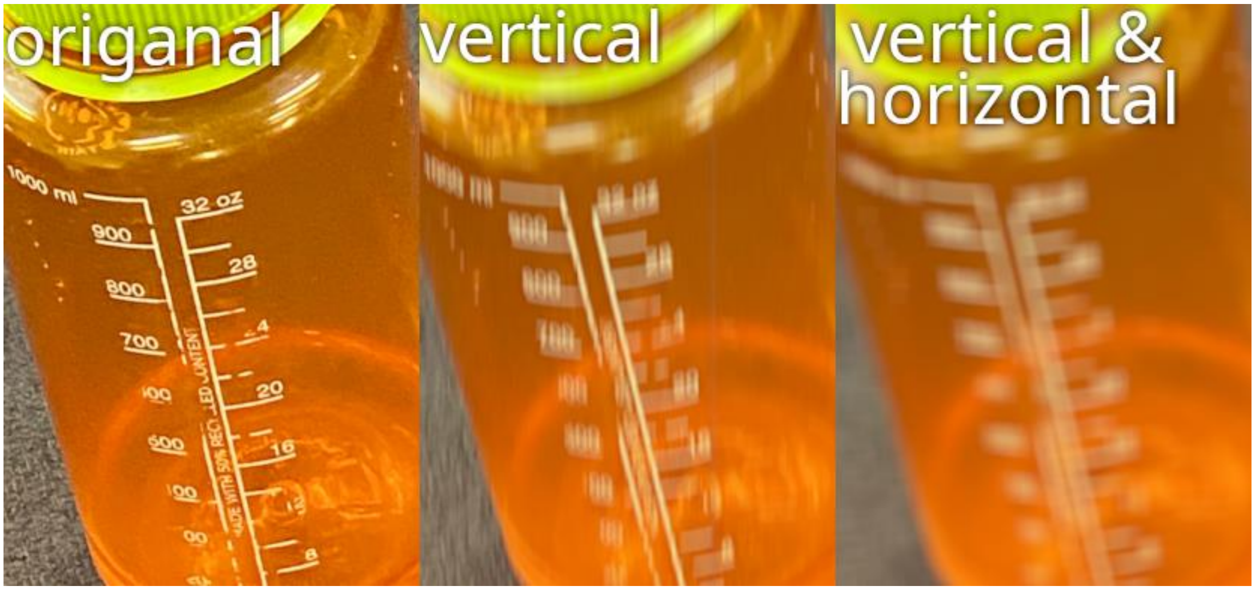

Figure 9: Blurring vertically then horizontally yields the same results as it would in one go with a 2-dimensional kernel. However, it is more performant because the two kernels grow linearly with the blur radius opposed to a single kernel growing quadratically with the blur radius.

Gaussian Blur

Box blur has square shaped artifacts – notice the last section of figure 9, which may not be desired. Applying box blur again will blur the blurring artifacts somewhat. Repeated box blur will result in an artifact-less blur [4]; however, can become computationally expensive. Instead, the kernel can be altered to reflect these changes so only one pass (or two for the vertical/horizontal optimization) are required. See figure 6. This is called a gaussian blur. See figure 1 and figure 5 as examples of gaussian vs box blur respectively.

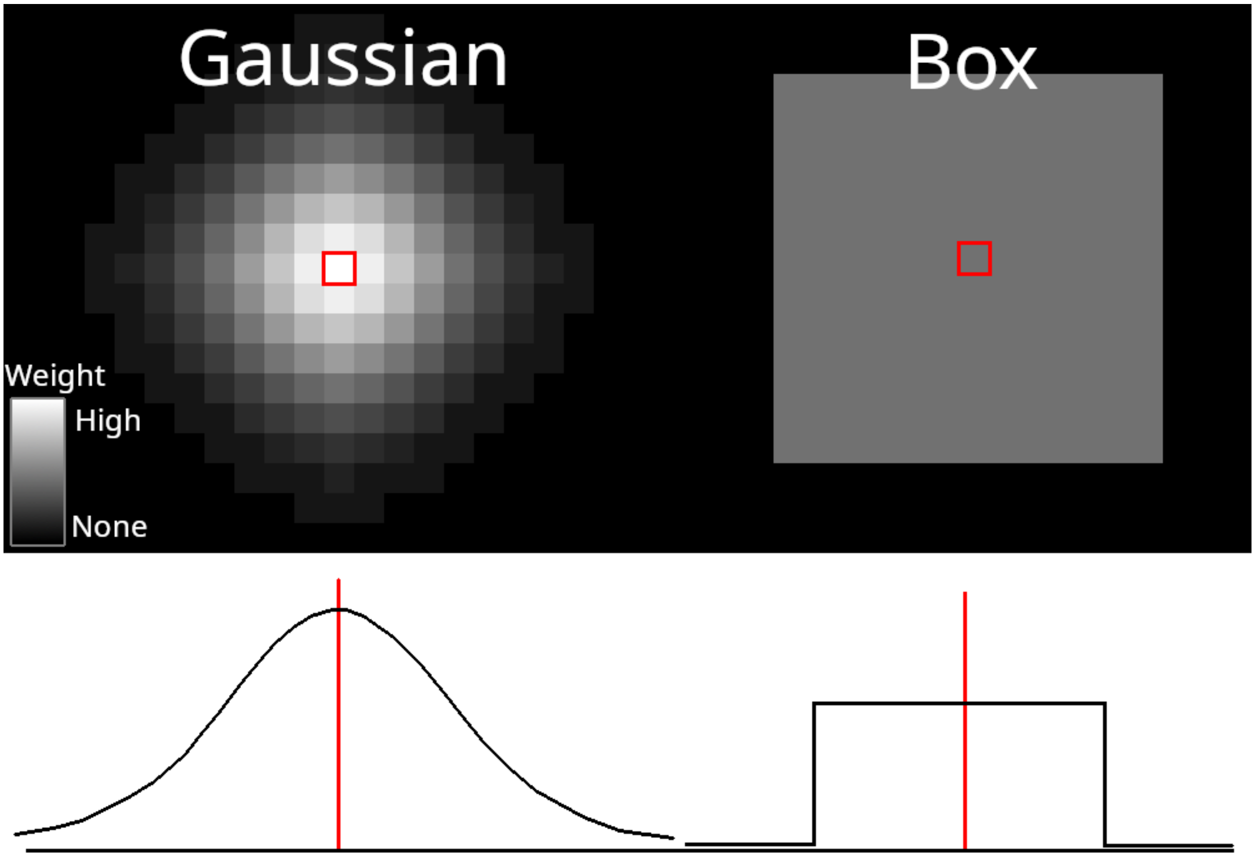

Figure 10: The difference between gaussian and box blur is the kernel. Box blur uses a simplistic box where all adjacent pixels have equal weight. Gaussian blur uses a weighted kernel where closer pixels have a larger weight then further pixels. The intensity of the weight is based on the distance applied to a gaussian distribution, hence the name.

Other blurs
Different blurs are used for different reasons.

Gaussian blur is the most common in digital applications because it simulates frosted glass and is largely unoffensive.

Another common blur is circular blur, similar to box blur, except the kernel is a circle instead of a square. The biggest downside is that there are not many possible optimizations. Circular blur simulates something out of focus in a camera. [5]

Box blur is used because it is the fastest because other optimizations specific to it can bring it to constant time O(1). [6]

There are a multitude of blurs to emphasize movement in photo editing from such as linear, zoom and rotational blur. [7]

References:

[1] Taylor Ashley, "Introduction to Digital Images," *Stanford*. [Online]. Available: https://web.stanford.edu/class/cs101/image-1-introduction.html. [Accessed: Oct. 19, 2023].

[2] Ludwig Jamie, "Image Convolution," *Satellite Digital Image Analysis, 581 Portland State University*. 2007. [Online]. Available: https://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Ludwig_ImageConvolution.pdf. [Accessed: Oct. 19, 2023]

[3] Minute Physics, "Computer Color is Broken," *YouTube*. March, 2015. [Online]. Available: https://youtu.be/LKnqECcg6Gw. [Accessed: Oct. 19, 2023]

[4] Pascal Gwosdek et al., "Theoretical Foundations of Gaussian Convolution by Extended Box Filtering," *Saarland University.* Sept, 2011. [Online] Available: https://www.mia.uni-saarland.de/Publications/gwosdek-ssvm11.pdf. [Accessed: Oct. 19, 2023]

[5] Kragen Javier Sitaker. "Real-time bokeh algorithms, and other convolution tricks," August, 2019. [Online] Available: https://dercuano.github.io/notes/convolution-bokeh.html. [Accessed: Oct. 19, 2023]

[6] Mazumdar Amrita, "Understanding Box Blur," Feb, 2013. [Online] http://amritamaz.net/blog/understanding-box-blur. [Accessed: Oct. 19, 2023]

[7] "Blur Filters," *GIMP.* [Online] https://docs.gimp.org/en/filters-blur.html. [Accessed: Oct. 19, 2023]