GPU Based Bilateral Filtering for Images

Paul Logas

Computer Engineering Department

University of Florida

Gainesville, FL

jacobpaullogas@outlook.com

*Abstract*—A Bilateral filter smooths images while preserving edges, this is accomplished through nonlinear combination of a neighborhood of image values. The method combines colors or intensities based on geometric closeness and photometric similarities. Bilateral filters can enforce the perception metric of the CIE-Lab color space to smooth colors and preserve edges tuned to human perception. Bilateral filtering has the added benefit of an absence of phantom colors along the edges of a color image. A naïve, iterative solution to Bilateral filtering will iterates over all the input image’s pixels. The iterative and local nature of this algorithm has tremendous potential for parallel GPU implementations. (LENGTHEN)

Keywords—bilateral; parallel; filtering; GPU

# Introduction

One of the most fundamental operations of computer vision and image processing is filtering. Broadly defined, a value of a filtered image at some location is a function of color or intensity values in a neighborhood about the location. For example, a Gaussian low-pass filter, one of the most commonly used filters, computes a weighted average value for the location in question. The weights are computed to have a higher weight for values nearby the location and lower weights for values further away. A more formal explanation of the weight fall-off can be found [1], intuitively an image varies little over a small space, thus nearby pixels are going to be similar. By averaging these pixels together, the image function is smoothed and noise is removed.

The assumption of a slow spatial variation fails at edge cases, a pure Gaussian filter will blur image edges as it does not consider legitimate variations in pixel intensities or color values. This is a major issue for computer vision applications, cameras inherently introduce noise which could negatively alter feature matching techniques however if a standard Gaussian filter is applied to the image edge information can be completely lost. Edge based segmentation for example look for abrupt changes in intensity. However, if an image is not smoothed than false edges can be detected. There have been attempts to reduce the effect of smoothing as well as do without smoothing altogether [2] [3].

One popular answer to this problem is anisotropic diffusion [4]. In this approach, the local image variation is measured at every point with pixel values being averaged from neighborhoods with size and shape depending on local variation. Diffusion methods average over an extended region through solving partial differential equations and are thus iterative inherently. In addition to possible efficiency issues, iteration can raise issues of stability.

A scheme introduced by C. Tomasi and R. Manduchi [5] provides a non-iterative and simple scheme for smoothing that preserves edges. The basic idea of this approach is to do what traditional filters do in the image domain, in the image range. Two pixels can either be spatially near each other or have values near each other, in a perceptually meaningful way. Closeness is specific to the vicinity in the domain, while similarity refers to the vicinity in the range. A traditional filter is filters by domain, enforcing closeness by weighing pixel values with coefficients which fall off over a distance. Range filtering on the other hand, averages image values with weights decaying by similarity. A range filter is nonlinear as weights depend on intensity or color of the image.

In the scheme, spatial locality is not altogether disregarded. Range filtering alone will only distort the image’s color map. Thus, the scheme combines range and domain filtering. This is what is referred to as *bilateral* filtering [5].

As this *bilateral* filters have an explicit definition for distance in an image function’s domain and range, they can be applied to any function in which these distances can be defined. Applying this filter to a color image is just as simple as applying to black-and-white images. The CIE-Lab color space gives the color space a meaningful measure of color similarity which in short distances correlate with human color discrimination performance. A bilateral filter often uses this metric as it provides a smooth image with edges that are preserved in a way that is tuned to human performance. In other words, colors that are perceptually similar are averaged together and perceptually visible edges are preserved.

This scheme introduced by C. Tomasi and R. Manduchi [5] is non-iterative and so each pixel must only be evaluated once. This is a perfect candidate for implementation in a massively multi-threaded environment such as a GPU which is made up of hundreds if not thousands of computation units providing excellent throughput [6]. With no one output pixel depending on the output of another pixel and a pixel’s output depending upon a single set of calculations, a GPU’s high throughput can be used to great effect.

# Overview

## The Premise

A low-pass filter applied to an image produces an output defined as:

|  |  |  |
| --- | --- | --- |
|  |  | 1 |

where is the measure of the closeness in geometric terms between the center and a point nearby . Both the output function **h** and input function **f** can be multiband. If low-pass filtering is to preserve the dc component of low-pass signals

|  |  |  |
| --- | --- | --- |
|  |  | 2 |

If filter is shift-invariant than is purely a function of the vector difference and will be constant.

Like domain filtering, range filtering is defined as:

|  |  |  |
| --- | --- | --- |
|  |  | 3 |

where is the measure of the photometric similarity between the pixel value at the center and some nearby point . The similarity function therefore deals with the range of the image function, while closeness function operates in the domain of the image function. The constant for normalization ( 2 ) is replaced in the range domain by

|  |  |  |
| --- | --- | --- |
|  |  | 4 |

Instead of depending on the distance from center as the function does, the normalization for the similarity function is dependent upon the image function. The similarity function is said to be *unbiased* if it depends on the difference alone.

In these equations, the spatial distribution of the image intensities has no role in range filtering alone. However, a combination of intensities from the entire image does not make much sense as values a large distance from should have no bearing on the final value at location . An appropriate solution to this shortcoming in range filtering is to combine both range and domain filtering into one filter. This achieves both geometric and photometric locality. Combination of the filters is simply:

**Figure 1**: (a) A level step image with Gaussian noise. (b) The surface plot of the unfiltered image. (c) Surface plot after Gaussian filter applied. (d) Surface plot after bilateral filter applied. (e) The output image after bilateral filter applied.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (a) | (b) | (c) | (d) | (e) |

|  |  |  |
| --- | --- | --- |
|  |  | 5 |

with the normalization as a simple combination of domain and range normalization

|  |  |  |
| --- | --- | --- |
|  |  | 6 |

The combination of the two filters is termed *bilateral filtering*. This filter replaces the pixel value at a point with an average of similar and nearby pixel values. Regions without edges have pixel values that are similar to each other which results in ( 4 ) being close to one. Thus, the filter in these regions is essentially a domain filter, averaging away weakly correlated pixel differences that are caused by noise.

Now considering an image function such as the one in Figure 1(a), which has a sharp boundary between a dark and a light region. When the similarity function is centered on a pixel on or near the edge, weights for the pixel values on the same side as the center will be near one. Conversely, weights on the other side of the image edge will be near zero. Thus, the desired effect is achieved, as shown in Figure 1(d), when the range filtering is combined with the domain filtering.

## Gaussian Case

The shift-invariant Gaussian filter, in which both the domain, , and range, , closeness functions are Gaussian functions of Euclidean distance between the arguments. The function is radially symmetric

where the Euclidean distance between point and is

This equation is recognizable as the Gaussian point-spread function without the normalization coefficient [1]. Now we wish to define in a way that will provide a point-spread of the range. The similarity function as it turns out is analogous to the function so it is defined as

where

Instead of the measure of the Euclidean distance as it was for the domain function, is the measure of distance between two intensity or color values and .

There are two values, and , in the bilateral filter. The value for is the for the domain part of the filter, thus it acts exactly as the in a standard Gaussian blurring filter would. A large blurs more, as it combines pixel values from a greater distance in the image. The must be changed if one desires the same results from an image that is scaled up or down. The other in a bilateral filter, , is termed the photometric spread. This value is set in order to achieve the desired combination of pixel values. Pixels with values much closer to each other than are mixed while values more distant than are not. Similar to when the image is edited the value must be changed to achieve similar results. In this case, must be changed if an image is amplified or attenuated. [5]

As a Gaussian blurring filter is shift-invariant, the range filter is insensitive to the additive changes of image intensity, making it unbiased. Thus, an image which when a range filter is applied has an output .

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

# References

|  |  |
| --- | --- |
| [1] | B. Klaus and P. Horn, Robot Vision, Cambridge, MA: MIT Press, 1986. |
| [2] | M. Tabb and N. Ahuja, "Multiscale Image Segmentation by Integrated Edge and Region Detection," *IEEE Transactions on Image Processing,* vol. 6, no. 5, 1997. |
| [3] | R. T. Chin and C. L. Yeh, "Quantitative evaluation of some edge-preserving noise-smoothing techniques," *CVGIP,* no. 23, pp. 67-91, 1983. |
| [4] | P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans.,* vol. 12, no. 7, pp. 629-639, 1990. |
| [5] | C. Tomasi and R. Manduchi, Bilateral Filtering for Gray and Color Images, Bombay: IEEE International, 1998. |
| [6] | D. Luebk, "GPU Architecture: Implications & Trends," in *Siggraph*, 2008. |