IMPLEMENTING FAST APPROXIMATION OF THE BILATERAL FILTER AND ITS COPARISION WITH OTHER METHODS

Özgür Bağcı

Middle East Technical University

Computer Engineering Department

Fundamentals of Image Processing Class Assignment

Abstract

The Bilateral Filter is a non-linear filter used in Computer Vision and Image Processing which while smoothing image, reduces noise and preserves the edges. But it may be slow and it is non-linear. In order to speed up processing and pack non-linearities into an operation applied at the end, a method is proposed in a paper[1] called "A Fast Approximation of the Bilateral Filter using a Signal Processing Approach" by Paris and Durand [2006]. Instead of just implementing plain Bilateral Filter, this method is, and it compared with other methods.

Introduction

A lot of methods exist to fasten Bilateral Filter while preserving its accuracy but most of them is not proved to be accurate. In the paper[1] of Paris et al. [2006] it is shown that the 'fast approximation' method is an accurate approximation to Bilateral Filter.

In order to compare its accuracy two other implementations are picked. The first one[2] is plain Bilateral Filter Implementation as stated in "Bilateral Filtering for Gray and Color Images" by Tomasi et al. [1998] which is used in Python package scikit-image. The second implementation is the OpenCV implementation which is based on a paper released by The University of Edinburgh School of Informatics[3].

For detailed explanation of code please refer to 'README.md' provided along with The Report and The Code. As stated in README The Code highly inspired by four implementations which you may find online.

The Fast Approximation

The Fast Approximation of Bilateral Filter is uses a method which projects 2D image to a 3D array, down-samples it and then applies Convolutions to that array, right after that up-samples the 3D array and project it back to a 2D image. For more detailed explanation refer to original paper[1].

Tests

Several configurations are tested using the implementation provided with The Report.

In order to apply filtering random sigma values generated then Gaussian Noise added to grey-scale images. The sigma values picked to be range [10, 25] in order to better visualize the results.

It is observed that The Bilateral Filter Spatial Sigma should be divided by a value in range [2.5, 3.0]. As the value goes to 1 the effect caused by Bilateral Filter is becoming more and more uncertain. And if the value goes below 1, the effect may be described as noising rather than denoising.

Sampling sigma values should be derived by dividing their corresponding sigma values by a value in range [2.0, 2.5] since smaller values does not do significant down-sampling and obliterate its speed-up benefits, higher values results in poor results.

A mildly noised image is selected in order to better visualize the results. You may would like to see other images which are provided with The Code and The Report.

In order to fix time to a interval of [15, 45] seconds for each of the filter implementations some configurations adjusted considering also it may cause some improvements in result.

Figure 1: Original Image

Figure 2: Noisy Image

Figure 3: OpenCV Filter

Figure 4: scikit-image Filter

Figure 5: The Code Filter

Conclusion

As shown in the 'Figure 5', scikit-image implementation did a very bad job at denoising the image, removing all the details with the noise.

In 'Figure 3' it may be seen that OpenCV did a good job at denoising the image preserving the detail and edges despite it is not at the level of The Code.

'Figure 4' shows The Code does a very good job at denoising and smoothing the image.

With the time speed-ups of The Code which is observed from the tests and better denoising and smoothing it provides, it can be concluded that this method is a better, faster Bilateral Filter Approximation.

References

[1] Sylvain Paris and Fredo Durand. A Fast Approximation of the Bilateral Filter using a Signal Processing Approach. 2006

[2] C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. 1998

[3] Bilateral Filtering for Gray and Color Images. Retrieved from http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html