Image Processing by Bilateral Filtering Method

**Khyati Aman***
Under Graduate Student
Gandhi Institute of Engineering & Technology
Gunupur, India

**Susmita Suman**
Under Graduate Student
Gandhi Institute of Engineering & Technology
Gunupur, India

**Sonali Panda**
Under Graduate Student
Gandhi Institute of Engineering & Technology
Gunupur, India

**Nalikanta Barpanda**
Head of Department
Gandhi Institute of Engineering & Technology
Gunupur, India

**Abstract**

In this paper, we propose a new method to construct an edge-preserving filter which has very similar response to the Gaussian filter which is known as bilateral filter. Bilateral filter smoothenes the image while preserving edges, by means of a nonlinear combination of nearby image values. It combines gray levels or colours based on both their geometric closeness and their photometric similarity and prefers near values to distant values in both domain and range. In contrast with filters that operate on the three bands of a colour image separately, a bilateral filter can enforce the perceptual metric underlying the CIE-Lab colour space and smooth colours and preserve edges in a way that is tuned to human perception. Also, in contrast with standard filtering, bilateral filtering produces no phantom colours along edges in colour images and reduces phantom colours where they appear in the original image. In this paper, we also introduced the image cartooning which is an application of bilateral filtering. Cartooning is one of the most extensively used procedures, be it a gaming world, animation or the entertainment industry.

**Keywords:** Filtering, Gaussian filter disadvantages, Bilateral filter, MATLAB software, CIE-LAB color space.

**Author for correspondence** khyatiaman@gmail.com

1. **Introduction**

Image processing pertains to the alteration and analysis of pictorial information. Common case of image processing is the adjustment of brightness and contrast controls on a television set by doing this we enhance the image until its subjective appearing to us is most appealing in [1] and [2]. The biological system (eye, brain) receives, enhances, analyzes and stores images at enormous rates of speed. Basically there are two-methods for processing pictorial information as referred in [3] and [4]. They are: Optical processing, and Electronic processing.
Optical processing uses an arrangement of optics or lenses to carry out the process. An important form of optical image processing is found in the photographic dark room. Electronic image processing is further classified as: Analog processing, and Digital processing.

2. Filtering
When an image is acquired by a camera or other imaging system, often the vision system for which it is intended is unable to use it directly. The image may be corrupted by random variations in intensity, variations in illumination or poor contrast that must be dealt with in the early stages of vision processing. Filtering is perhaps the most fundamental operation of image processing and computer vision. In the broadest sense of the term filtering the value of the filtered image at a given location is a function of the values of the input image in a small neighbourhood of the same location. As mentioned earlier, images are often corrupted by random variations in intensity values called noise. Some common types of noise are salt and pepper noise, impulse noise and Gaussian noise. Salt and pepper noise contains random occurrences of both black and white intensity values. Unlike these, Gaussian noise contains variations in intensity that is drawn from a Gaussian or normal distribution and is a very good model for many kinds of sensor noise, such as the noise due to camera electronics. Linear smoothing filters are good filters for removing Gaussian noise and in most cases, the other types of noise as well. A linear filter is implemented using the weighted sum of the pixels in successive windows. Typically, the same pattern of weights is used in each window, which means that the linear filter is spatially invariant and can be implemented using a convolution mask. If different filter weights are used for different parts of the image, but the filter is still implemented as a weighted sum, then the linear filter is spatially varying. Any filter that is not a weighted sum of pixels is a nonlinear filter. Nonlinear filters can be spatially invariant, meaning that the same calculation is performed regardless of the position in the image or spatially varying.

a) Gaussian Filtering
Gaussian filters are a class of linear smoothing filters with the weights chosen according to the shape of a Gaussian function. The Gaussian smoothing filter is a very good filter for removing noise drawn from a normal distribution. The zero-mean Gaussian function in one dimension is

\[ g(x) = e^{-x^2/2\sigma^2} \]

Large Gaussian filters can be implemented very efficiently because Gaussian functions are separable. Two-dimensional Gaussian convolution can be performed by convolving the image with a one-dimensional Gaussian and then convolving the result with the same one dimensional filter oriented orthogonal to the Gaussian used in the first stage. Thus, the amount of computation required for a 2D Gaussian filter grows linearly in the width of the filter mask instead of growing quadratically.

Disadvantages:
   a. Slow process, and
   b. Does not preserve edges.
In particular, Gaussian low-pass filtering computes a weighted average of pixel values in the neighbourhood, in which, the weights decrease with distance from the neighbourhood center. The noise values that corrupt these nearby pixels are mutually less correlated than the signal.

![Image of an eye with focus on the pupil](image1.png) ![Image of an eye with focus on the iris](image2.png)

**b) Bilateral Filtering**

The idea underlying bilateral filtering is to do in the range of an image what traditional filters do in its domain. Two pixels can be close to one another, that is, occupy nearby spatial location or they can be similar to one another, that is, have nearby values, possibly in a perceptually meaningful fashion. Closeness refers to vicinity in the domain, similarity to vicinity in the range.

Advantages: simple, and preserves edges.
Applications: used in cartooning, denoising, and tone management.

3. **MATLAB**

The paper is implemented in the version MATLAB R2009a includes new features in MATLAB® and Simulink®, 2 new products and updates and bug fixes to 91 other products. Subscribers to Math Works™ Software Maintenance Service can download product updates. Since R2008a, the MATLAB and Simulink product families require activation. R2009a also includes enhancements to the License Center, the online tool for activating software and managing license and user information. Many of the toolbox functions are MATLAB M-files, a series of MATLAB statements that implement specialized image processing algorithms. You can view the MATLAB code for these functions using the statement type function name. You can extend the capabilities of the toolbox by writing your own M-files or by using the toolbox in combination with other toolboxes, such as the Signal Processing Toolbox™ software and the Wavelet Toolbox™ software. The Image Processing Toolbox software is a collection of functions that extend the capability of the MATLAB numeric computing environment.

Some of the functions used in the code:

A = imread (filename, fmt) reads a grayscale or colour image from the file specified by the string filename. If the file is not in the current directory, or in a directory on the MATLAB path, specify the full pathname.
r = randn(n) returns an n-by-n matrix containing pseudorandom values drawn from the standard normal distribution. randn(m,n) or randn([m,n]) returns an m-by-n matrix. randn(m,n,p,...) or randn([m,n,p,...]) returns an m-by-n-by-p-by-... array. randn returns a scalar. randn(size(A)) returns an array the same size as A.

length (X) returns the size of the longest dimension of X. If X is a vector, this is the same as its length.

reshape(A,m,n) returns the m by n matrix B whose elements are taken column-wise from A. An error results if A does not have m*n elements.

applycform(A,C) converts the colour values in A to the colour space specified in the colour transformation structure C. The colour transformation structure specifies various parameters of the transformation.

Waitbar (x,'message') displays a wait bar of fractional length x. The wait bar figure displays until the code that controls it closes it or the use clicks its Close Window button. Its (figure) handle is returned in h. The argument x must be between 0 and 1.

makecform(type) creates the colour transformation structure C that defines the colour space conversion specified by type. To perform the transformation, pass the colour transformation structure as an argument to the apply form function.

zeros(n) returns an n-by-n matrix of zeros. An error message appears if n is not a scalar.

**Image Create and display image object:**
- a) imcropCrop image
- b) imresizeResize image
- c) imrotateRotate image
- d) imcloseClose image
- e) colormapSet or get colour lookup table.
- f) rgbplotPlot RGB colourmap components

**4. CIELAB Colour Space**
A colour space is an abstract mathematical model to describe the way colours can be represented. A colour space with no mapping function associated to absolute colour space is an arbitrary colour system with no connection to any of the globally known colour interpretation. The different types of colour space are adobe RGB and sRGB both are based on RGB values. RGB colour space is an objective specification of quality of the additive primary colour red, green, blue and can produce chromaticity defined by the triangle of those primary colours. sRGB stands for standard RGB colour space. CIELAB is another type of colour space which gives most complete colour space. It describes the colour visible to human eye and is used as a reference as it is a device-independent model. Here we convert from sRGB image to CIELAB as it gives most accurate colour balance corrections. As VGA monitors use sRGB, we convert CIELAB image back to sRGB.
Gradient is used for gradual blend of colour. We can extract information from an image using image gradient. It is a directional change of intensity or colour of an image. Gradient images are created from the original image by convolving with a filter. Each pixel of a gradient image measures the change in intensity of that same point in the original image, in a given direction. To get the full range of direction, gradient images in the x and y directions are computed. After gradient images have been computed, pixels with large gradient values become possible edge pixels. The pixels with the largest gradient values in the direction of the gradient become edge pixels and edges may be traced in the direction perpendicular to the gradient direction.

Using gradient magnitudes create a simple edge map of the image. Determine the per pixel sharpening with the help of gradient magnitude and sharpness levels. Since the image is transformed into CIELAB colour space earlier, the luminance levels are quantized. Then the image is transformed back to sRGB colour space and the gradient edges are added to the bilateral filtered image to obtain the cartooned image.

5. Image Abstraction using Bilateral Filter
Abstraction is one of the most extensively used procedures, be it a gaming world, animation or the entertainment industry. The aim is to perform image abstraction for a given image. Bilateral filter is used for abstraction the given image because it is a noise reducing (smoothening) as well as edge preserving, non-iterative filter. The process of abstraction starts with an image abstraction. The image must be a colour image or a gray scale image with double precision matrix NxMx3 in a closed interval \([0, 1]\). Here we are using the bilateral filter. Bilateral filter is a smoothening and edge preserving filter. Since the edges are preserved, it can be highlighted and used in the process of abstraction. The method is non-iterative, local and simple. It combines gray levels or colours based on both their geometric closeness and their photometric similarity and prefers near values to distant values in both domain and range. In contrast with filters those operate on the three bands of a colour image separately, a bilateral filter can enforce the perceptual metric underlying the CIE-Lab colour space and smooth colours and preserve edges in a way that is tuned to human perception.

6. Results
Bilateral filtering of gray scale image: The output of the bilateral filter is as shown in figure for gray scale image. Here in the output image each pixel is replaced by a weighted average of its neighbours and averaging is done both in spatial and range technique.
Bilateral filtering of colour image: The output of the bilateral filter is as shown in figure for colour image. Here in the output image each pixel is replaced by a weighted average of its neighbours and averaging is done both in spatial and range technique. So the output is the smoothened image and noise is reduced.

Image Cartooning:

7. Conclusion
We have presented the bilateral filter, its applications and its variants, reviewed our current theoretical understanding of it and explained fast algorithms to evaluate it. We believe that the success of the bilateral filter lies in its combination of simplicity, good results and efficient algorithms. Although alternatives exist for each of these points, few, if any, combine all these advantages. The filter is very flexible because the range weight can be adapted to accommodate any notion of pixel value difference, including arbitrary colour spaces.

References