Sobel and Scharr 3x3 and 5x5 convolution kernels for image gradient calculations

Guennadi (Henry) Levkine
Email: hlevkin@gmail.com
Vancouver, Canada.

Abstract.
Prewitt, Sobel and Scharr 3x3 gradient operators are very popular for edge detection. This article demonstrates how to get Sobel and Scharr gradient operators analytically. For that was using linear approximation of brightness in windows 3x3 and 5x5. Every pixel in window has its own weight. Special weight matrix selection gives Sobel and Scharr gradient operator. Selection of different weight matrices gives a lot of new smooth and gradient operators.

Key words: image gradient, edge detection, Prewitt operator, Sobel operator, Scharr operator, local polynomial approximation

1 Introduction

In this article we use 2-dimensional Local Polynomial Approximation (LPA) method, which will be shortly described here (more detailed description is in [1]). In this method for every pixel we calculate 2D polynomial, which approximates image intensity in some block around the pixel. Here we consider only linear approximation and use special notation

$$I(x, y) = a + b * x + c * y$$

For a,b,c coefficients we use standard Least Square Fit method for minimization of difference between our approximation A(x, y) of brightness and real brightness in image.
Gradient of function *A* (*x*, *y*) is defined as a vector with coordinates:

$$G_x = \frac{\partial A}{\partial x}, \qquad G_y = \frac{\partial \mathbf{A}}{\partial \mathbf{y}}$$

In polynomial approximation the components of gradient are: $G_x = b$, $G_y = c$

Using different weight matrices we can generate many different convolution kernel matrices for gradient estimation.

In this article we develop kernels for gradient calculations in 3x3 and 5x5 image blocks.

2 Prewitt gradient operators with convolution kernels 3x3 and 5x5.

We have to select weight matrix (1) to get 3x3 Prewitt gradient operators

1	1	1
1	1	1
1	1	1

This weight matrix is uniformed: all pixels around the center have the same weight.
For 3x3 blocks estimation of a, b, c coefficients is obvious. Convolution kernels for their calculation are

1	2	1
2	4	2
1	2	1

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

(Weights equals 1/6, it means that we have to multiply convolution matrix onto this fraction 1/6)

If we process image with 1-byte brightness for pixel, minimal possible X and Y gradient values equal -127, maximum equals 127 (don't forget multiplier 1/6).

5x5 Prewitt weight matrix is (uniform as in the 3x3 case):

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Kernels for gradients are (weights =1/50):

-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2

-2	-2	-2	-2	-2
-1	-1	-1	-1	-1
0	0	0	0	0
1	1	1	1	1
2	2	2	2	2

3 Sobel gradient operators 3x3 and 5x5.

When we calculate kernels for coefficient estimation, all the pixels in the block have the same weight. But it is possible to use different pixel weights: higher for pixels close to block center. We consider here weight matrix (Sobel weight matrix)

1	2	1
2	4	2
1	2	1

This weight matrix is isotropic. It means that weights are dropping as 1/ *r* in any direction, where *r* is the distance between center of matrix and element of it.
Now we calculate coefficients a, b, c using this weight matrix. Kernel weights are 1/16, 1/8, 1/8):

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

These kernels are well known and widely used, kernel for “a” is used for noise filtration. Kernels for “b, c” called Sobel’s kernels and are used for gradient estimation.
This approach (using pixel weight matrices) gives rich opportunities for creating of many different kernels. It is convenient for different tasks in image processing.
Let’s try to build 5x5 weight matrixes, which is generalization of 3x3 Sobel weight matrixes. In this matrix weights are decreasing by 1/ *r* law, where r is the distance between center of matrix and current element of matrix. This weight matrix looks like:

5	8	10	8	5
8	20	40	20	8
10	40	80	40	10
8	20	40	20	8
5	8	10	8	5

Center of it equals to Sobel weight matrix 3x3. We call this matrix as Sobel 5x5 weight matrix.
After LPA calculation we found that coefficients b, c (gradients) are calculating using convolution matrices (weights are 1/240)

-5	-4	0	4	5
-8	-10	0	10	8
-10	-20	0	20	10
-8	-10	0	10	8
-5	-4	0	4	5

-5	-8	-10	-8	-5
-4	-10	-20	-10	-4
0	0	0	0	0
4	10	20	10	4
5	8	10	8	5

These convolution matrices represent Sobel 5x5 operators for x- and y-gradients.

4. Scharr gradient operators using 3x3 and 5x5 masks

One of the popular gradient operators is Scharr operator 3x3. It also can be get from linear LPA using special weight matrix. We call it Scharr weight matrix:

3	10	3
10	25	10
3	10	3

This matrix is not isotropic: weights in diagonal directions drop faster than in vertical and horizontal directions. After using of LPA we get convolution matrices for a, b, c (weights 1/77, 1/32, 1/32):

3	10	3
10	25	10
3	10	3

-3	0	3
-10	0	10
-3	0	3

-3	-10	-3
0	0	0
3	10	3

Here is simplified generalization of Scharr weight matrix to 5x5 case:

5	8	10	8	5
8	20	40	20	8
10	40	80	40	10
8	20	40	20	8
5	8	10	8	5

We found this matrix, based on assumptions:
- In horizontal and vertical directions the weight is proportional to reciprocal distance from the center
- In diagonal directions the weight is proportional to (reciprocal distance)* 2 / 3
- It the direction between horizontal and diagonal we use assumption that the weight is proportional to (reciprocal distance)* 5 / 6

After calculation we found that b, c coefficients equal to (weights 1/60. 1/60):

-1	-1	0	1	1
-2	-2	0	2	2
-3	-6	0	6	3
-2	-2	0	2	2
-1	-1	0	1	1

-1	-2	-3	-2	-1
-1	-10	0	-10	-8
0	0	0	0	0
-8	-10	0	10	8
-5	-4	0	4	5

It will be interesting to develop Scharr weight matrix 7x7, which is consistent with 3x3 and 5x5 weight matrices

6. Prewitt, Sobel, and Scharr 3x3 and 5x5 gradient operators Comparison.

For comparison of gradient operators we use classic test image Lenna of the size 512x512 from http://hlevkin.com/06testimages.htm.
For every gradient operator we calculate 6 corresponding 2-bytes image (3 for 3x3 gradient operators and another 3 for 5x5 gradient operators).
We are working with grey scale Lenna image, but gradient matrices are represented in the color form: red color is used for pixels with positive gradient, blue color for pixels with negative gradient.

After that we compare result images for 3x3 Prewitt, Sobel and Scharr and result images for 5x5 Prewitt, Sobel and Scharr. Comparison 3x3 processed images demonstrate small (practically invisible) difference between them. It's because of the fact, that Lenna image is essentially linear (can be presented in 3x3 windows by planes) and 3x3 Prewitt, Sobel, Scharr gradients gives right results in this case.
In some articles Prewitt and Sobel gradient matrices are used without coefficients (1/6 and 1/8 correspondently). As a result they had Sobel gradient image 33% brighter than Prewitt's one. Similar results will happen when using Scharr gradient operator without coefficient (1/32).

5x5 Prewitt operator will give us not good but smoothed gradient image (look at Picture 1). Sobel and Scharr gives better results because of the fact that far pixels in 5x5 window have less weights then pixels close to the center (look at Picture 2). 5x5 Sobel and Scharr gradient images are practically the same for Lenna image.
Some images have many horizontal and vertical edges. In addition to the 1-dimensional gradient operators it will be useful to use the Scharr gradient operators because of its anisotropic nature.

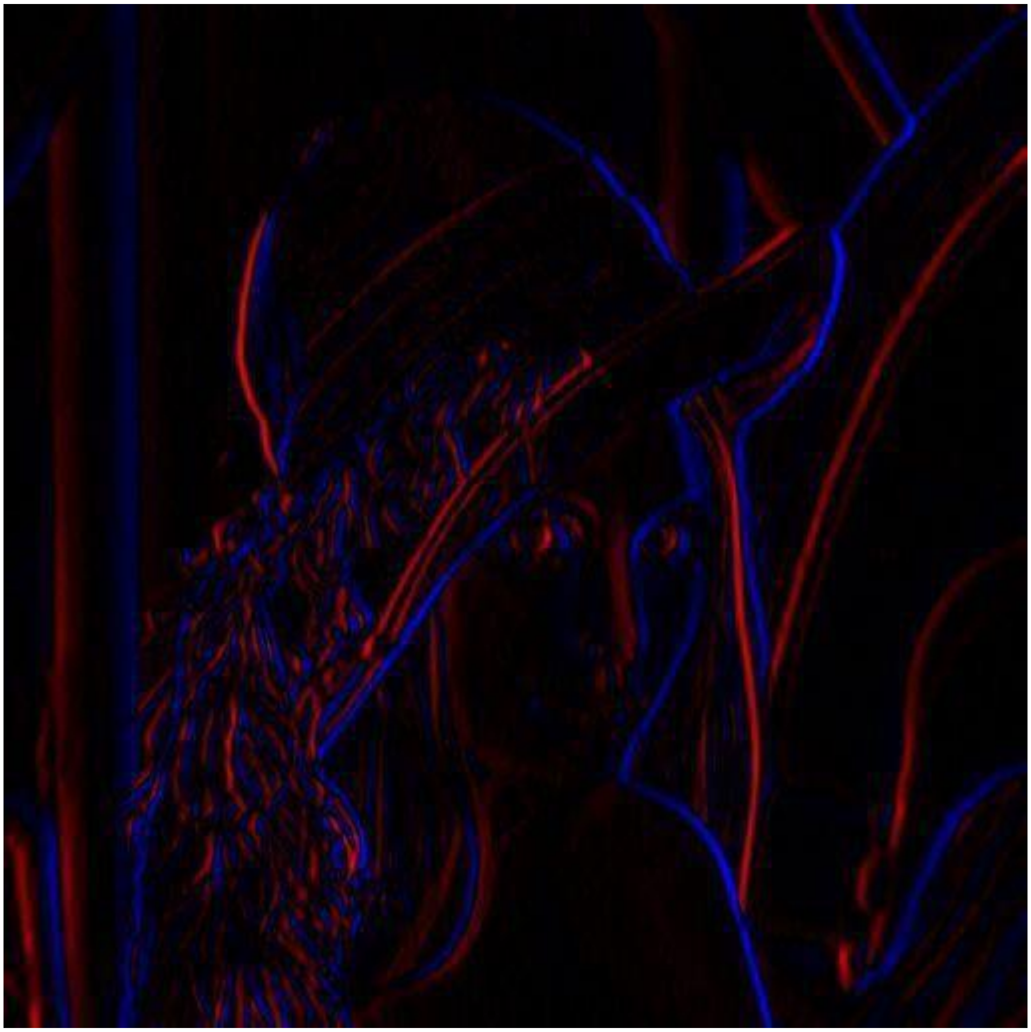
The best way of gradient image calculation is:
- Find linear and non linear segments of image (look at [1] how to do it)
- Process linear parts by Prewitt 3x3 gradient kernel, the rest by Sobel 5x5 kernel.

7. Conclusions.

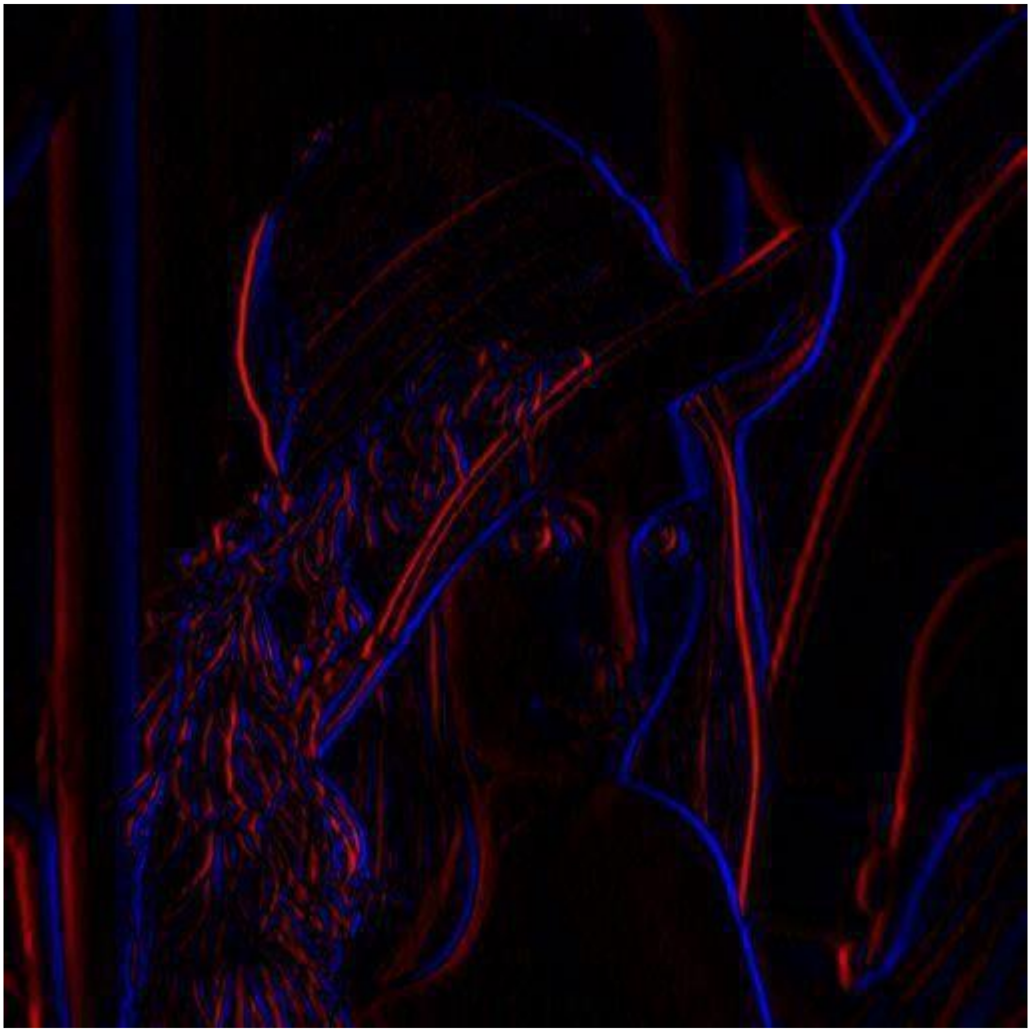
LPA together with weight matrix selection is useful for building of smoothing and gradient operators for image processing. For every gradient operator we have set of smoothing operators, which using makes consistency in simultaneous use of smoothing and gradient operators.
For big images, it is often convenient to use 5x5 and 7x7 convolution kernels.
It will be interesting to use other weight matrices, which build by using other distance metrics.

8. Acknowledgements.
I gratefully acknowledge V. Levkine for the support in writing this article.

References:
[1] G. Levkine, “Image Local Polynomial Approximation (LPA) and its Applications”
[2] J. M. S. Prewitt. “Object Enhancement and Extraction” in “Picture Processing and Psychopictorics”, Academic Press, 1970
[4] Reference to Sobel operator in Duda, Hart “Pattern Classification and scene analysis”, Wiley & Sons, 1973
[5] Reference to Scharr operator in Wikipedia article “Sobel operator”
[6] http://hlevkin.com/06testimages.htm



Picture 1.



Picture 2.