Image preprocessing in spatial domain

Sharpening, image derivatives, Laplacian, edges

Revision: 1.3, dated: May 6, 2010

Tomáš Svoboda

Czech Technical University, Faculty of Electrical Engineering
Center for Machine Perception, Prague, Czech Republic

svoboda@cmp.felk.cvut.cz

http://cmp.felk.cvut.cz/~svoboda

Spatial Filtering — overview

We have learned

- smoothing
- remove noise
- pattern matching (normalised cross–correlation)

Spatial Filtering — overview

We have learned

- smoothing
- remove noise
- pattern matching (normalised cross–correlation)

We will learn today

- sharpening
- image derivatives
- edges

Sharpening

Enhancing differences. So, the kernels involve differences — combine positive and negative weights.

- unsharp masking
- 1st and 2nd derivatives

Unsharp masking

- Often appears in Image manipulation packages (Gimp, ImageMagick)
- Quite powerful it cannot do miracles, though.

Idea: Subtract out the blur.

Procedure:

1. Blur the image
2. Subtract from original
3. Multiply by a weight
4. Combine (add to) with the original

Unsharp masking — Mathematically

$$g = f + \alpha(f - f_b)$$

- f original image
- f_b blurred image
- g sharpened result
- α controls the sharpening

What is the unsharp mask?

Unsharp masking — Mathematically

$$g = f + \alpha(f - f_b)$$

- f original image
- f_b blurred image
- g sharpened result
- α controls the sharpening

What is the unsharp mask?

$$g \;=\; \mathbf{1} * f + \alpha(\mathbf{1} * f - B * f)$$

.

Unsharp masking — Mathematically

$$g = f + \alpha(f - f_b)$$

- f original image
- f_b blurred image
- g sharpened result
- α controls the sharpening

What is the unsharp mask?

$$\begin{aligned} g &= \mathbf{1} * f + \alpha(\mathbf{1} * f - B * f) \\ &= (\mathbf{1} + \alpha(\mathbf{1} - B)) * f \end{aligned}$$

.

Unsharp masking — Mathematically

$$g = f + \alpha(f - f_b)$$

- f original image
- f_b blurred image
- g sharpened result
- α controls the sharpening

What is the unsharp mask?

$$\begin{aligned} g &= \mathbf{1} * f + \alpha(\mathbf{1} * f - B * f) \\ &= (\mathbf{1} + \alpha(\mathbf{1} - B)) * f \\ &= U * f \end{aligned}$$

where U is the desired unsharp mask.

Unsharp masking — Blur image

Unsharp masking — Subtract from original

−

=

Unsharp masking — Adding to the original

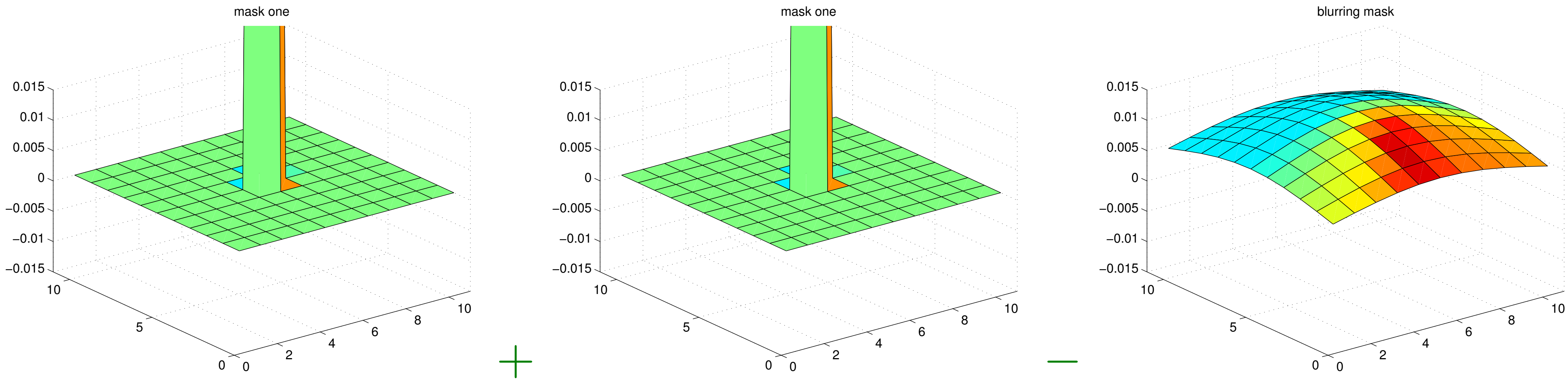
+

=

Unsharp masking — Result

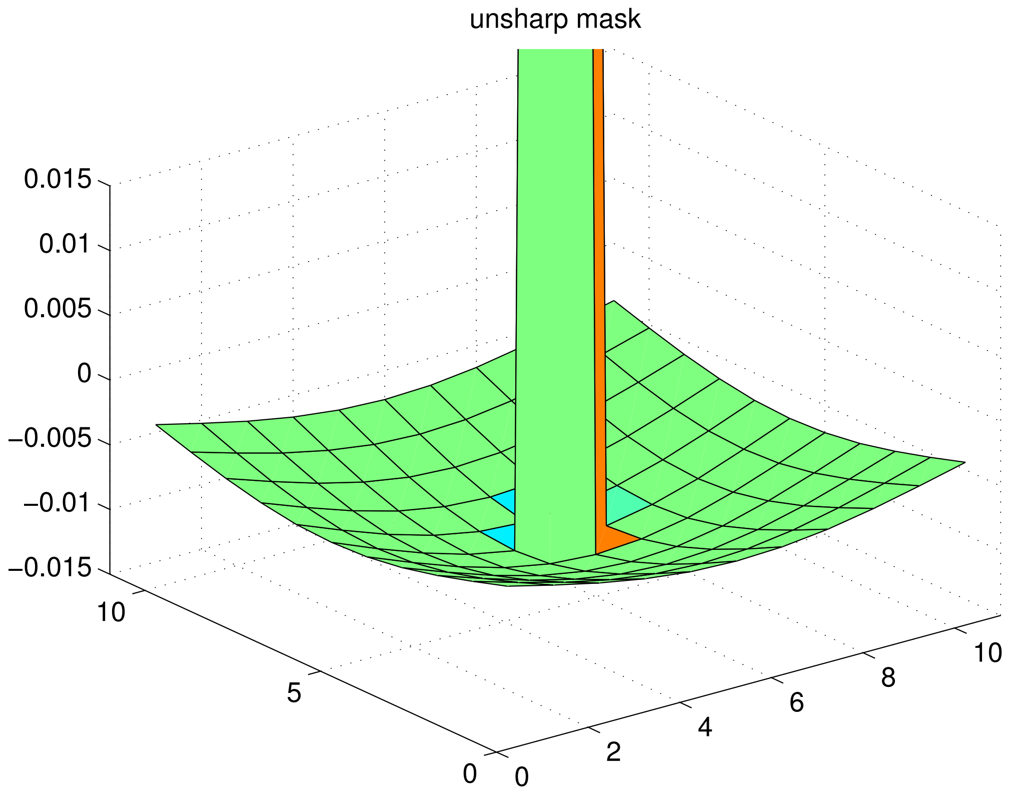
Unsharp masking — unsharp mask U

$$U = \mathbf{1} + \alpha(\mathbf{1} - B)$$



Unsharp masking — unsharp mask U

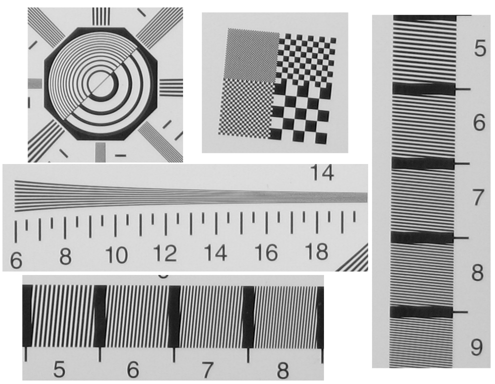
$$U = \mathbf{1} + \alpha(\mathbf{1} - B)$$



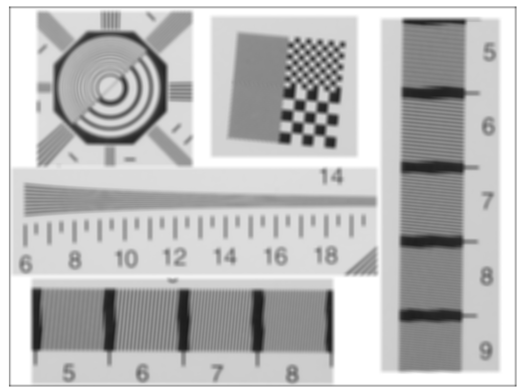
−0.0044	−0.0053	−0.0061	−0.0067	−0.0071	−0.0073	−0.0071	−0.0067	−0.0061	−0.0053	−0.0044
−0.0053	−0.0063	−0.0073	−0.0080	−0.0085	−0.0087	−0.0085	−0.0080	−0.0073	−0.0063	−0.0053
−0.0061	−0.0073	−0.0083	−0.0092	−0.0098	−0.0100	−0.0098	−0.0092	−0.0083	−0.0073	−0.0061
−0.0067	−0.0080	−0.0092	−0.0102	−0.0108	−0.0110	−0.0108	−0.0102	−0.0092	−0.0080	−0.0067
−0.0071	−0.0085	−0.0098	−0.0108	−0.0115	−0.0117	−0.0115	−0.0108	−0.0098	−0.0085	−0.0071
−0.0073	−0.0087	−0.0100	−0.0110	−0.0117	1.9880	−0.0117	−0.0110	−0.0100	−0.0087	−0.0073
−0.0071	−0.0085	−0.0098	−0.0108	−0.0115	−0.0117	−0.0115	−0.0108	−0.0098	−0.0085	−0.0071
−0.0067	−0.0080	−0.0092	−0.0102	−0.0108	−0.0110	−0.0108	−0.0102	−0.0092	−0.0080	−0.0067
−0.0061	−0.0073	−0.0083	−0.0092	−0.0098	−0.0100	−0.0098	−0.0092	−0.0083	−0.0073	−0.0061
−0.0053	−0.0063	−0.0073	−0.0080	−0.0085	−0.0087	−0.0085	−0.0080	−0.0073	−0.0063	−0.0053
−0.0044	−0.0053	−0.0061	−0.0067	−0.0071	−0.0073	−0.0071	−0.0067	−0.0061	−0.0053	−0.0044

We may combine only masks not the whole images!

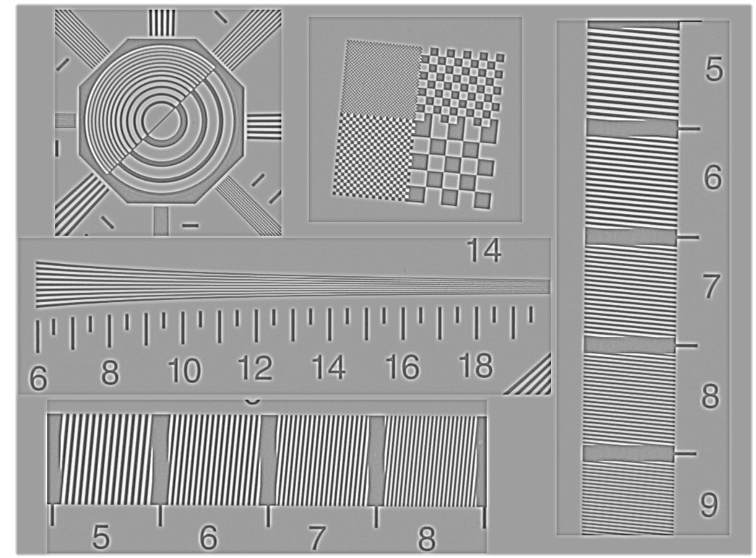
Unsharp masking — Subtract from original



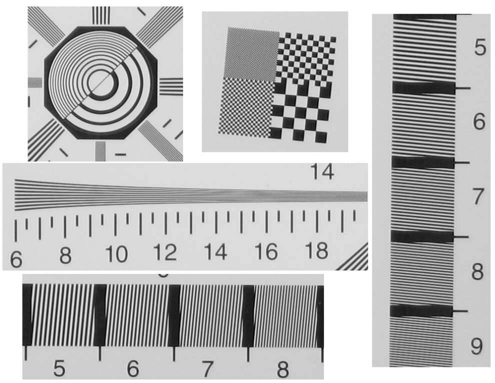
−



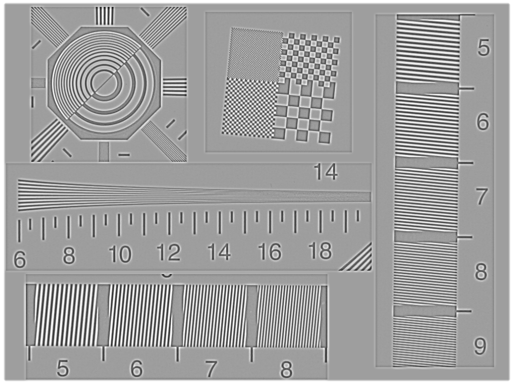
=



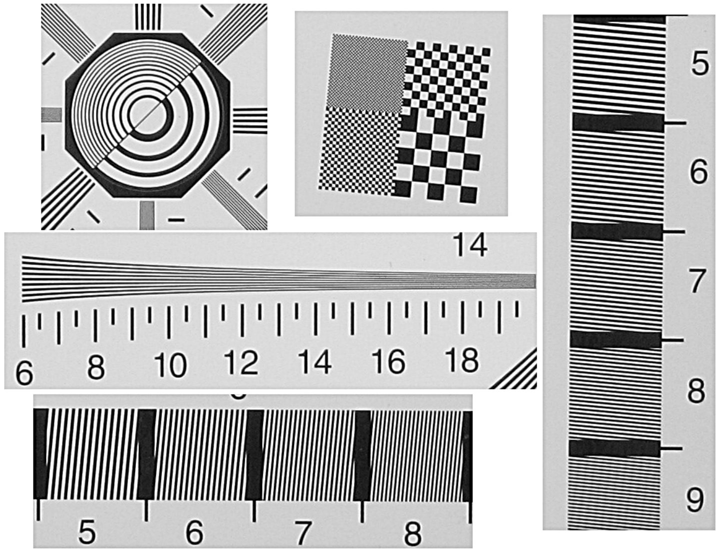
Unsharp masking — Adding to the original



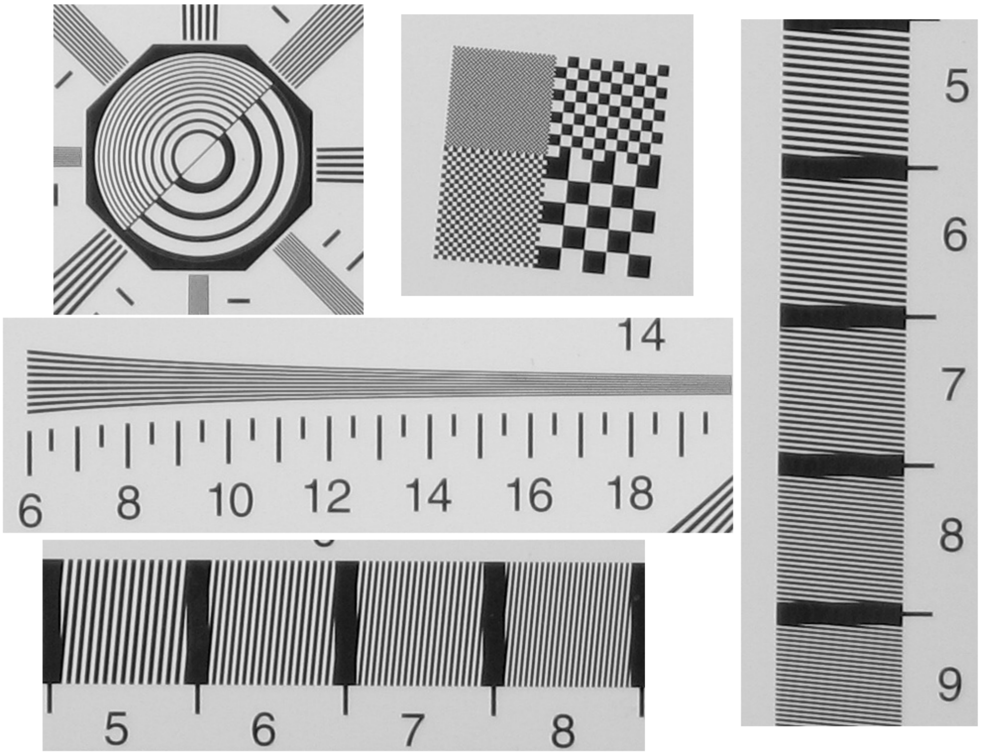
+

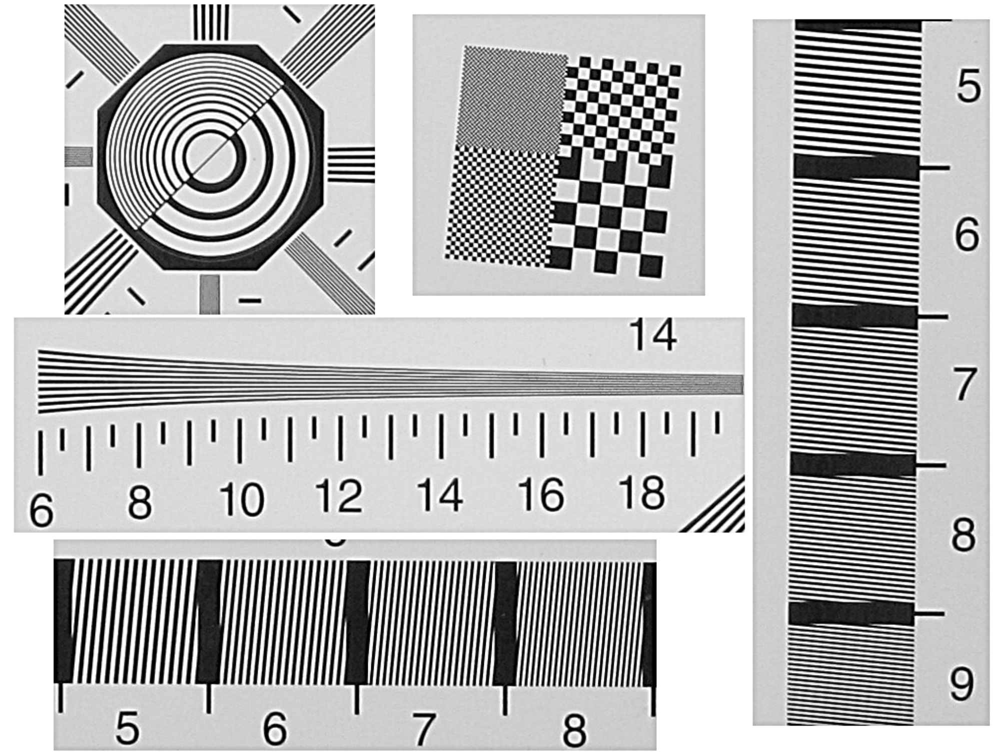


=

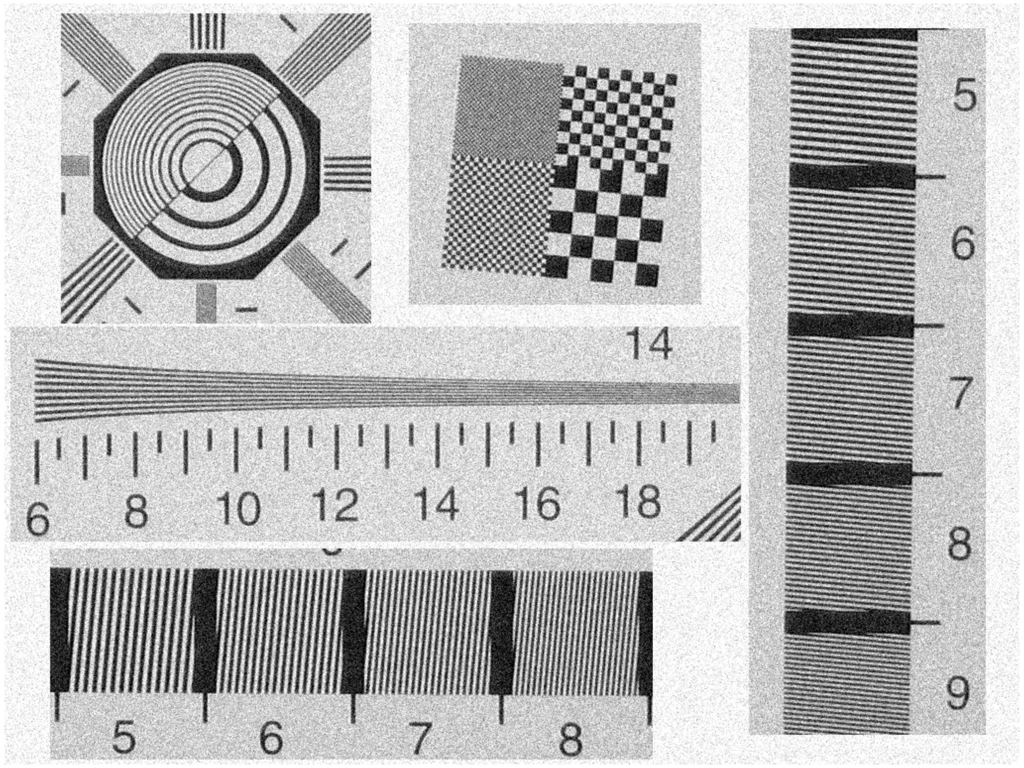


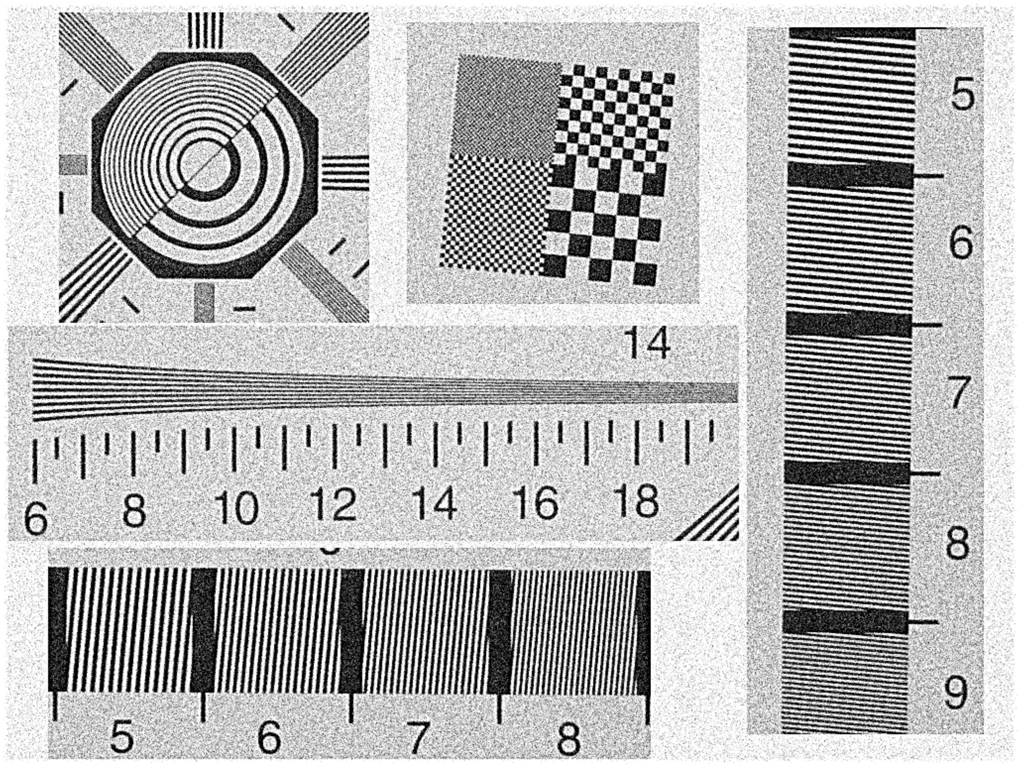
Unsharp masking — Result



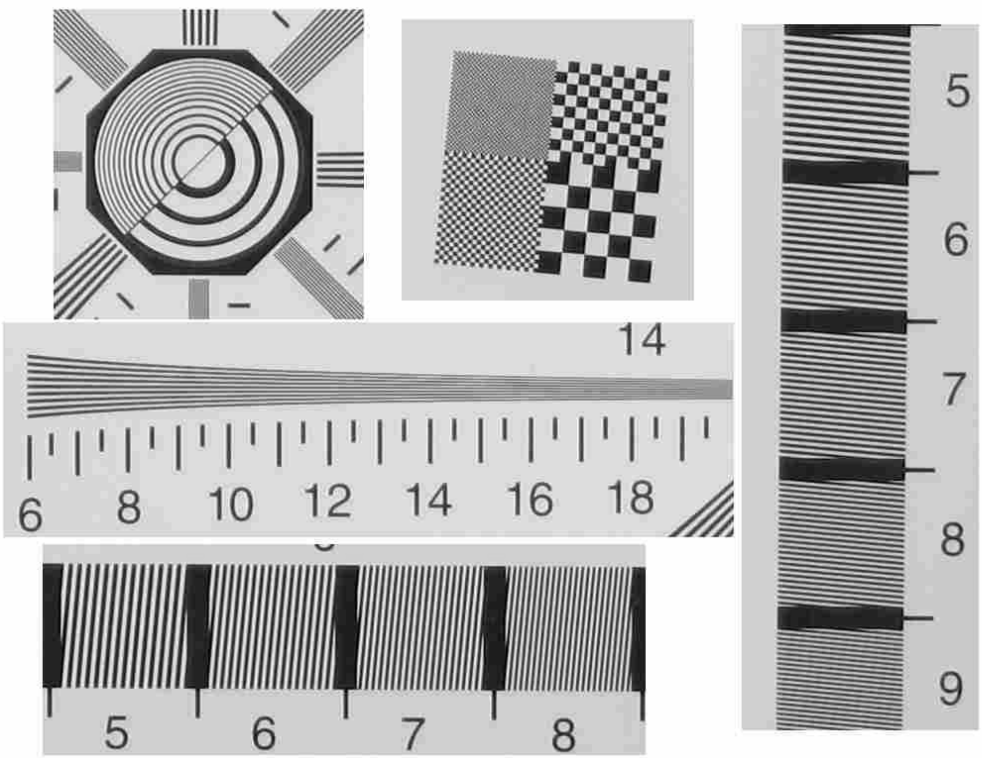


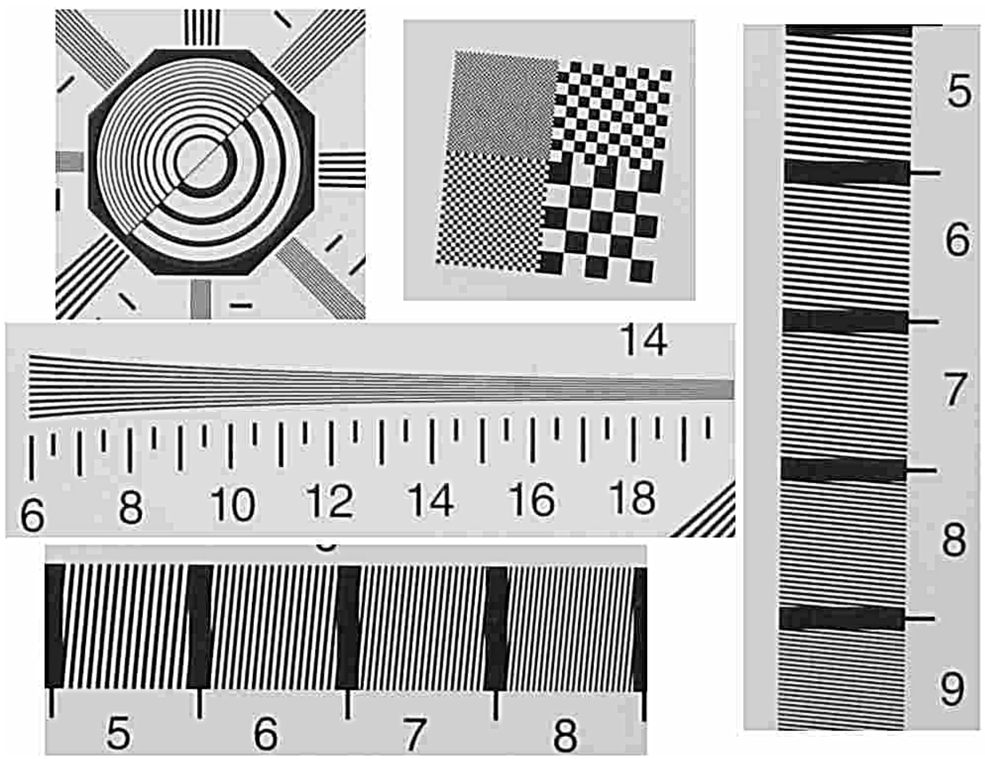
Unsharp masking — Problems with noise





Unsharp masking — Problems lossy JPG compression





Unsharp masking — revisited

- Often appears in Image manipulation packages (Gimp, ImageMagick).
- It may help in practice. Low-cost lenses blur the image.
- Quite powerful it cannot do miracles, though.
- It also emphasises noise and JPG artifacts.

Image derivatives

- Measure local image geometry
- Differential geometry a branch of mathematics built around
- We can use convolution to compute them

Image derivatives

- Measure local image geometry
- Differential geometry a branch of mathematics built around
- We can use convolution to compute them

- First derivative — local changes to the signal. (from physics: speed is derivative of a position with respect to time)
- Second derivative — changes to change (from physics: acceleration is . . .)

Derivative — reminder from calculus

Consider a 1D signal $f(x)$

$$\frac{d}{dx} f(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Derivative — reminder from calculus

Consider a 1D signal $f(x)$

$$\frac{d}{dx}f(x) = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

However, for sampled (discrete) signals, the smallest difference h is one. So,

$$\frac{d}{dx}f(x) \approx \frac{f(x+1)-f(x)}{1}$$

This called forward difference

Backward difference

Remind that the limit $\lim_{h\to 0}$

$$\frac{d}{dx}f(x) = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

must exist for both $\lim_{h\to 0+}$ and $\lim_{h\to 0-}$

Backward difference

Remind that the limit $\lim_{h\to 0}$

$$\frac{d}{dx}f(x) = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

must exist for both $\lim_{h\to 0+}$ and $\lim_{h\to 0-}$

So going from negative side of h

$$\frac{d}{dx}f(x) = \lim_{h\to 0}\frac{f(x)-f(x-h)}{h}$$

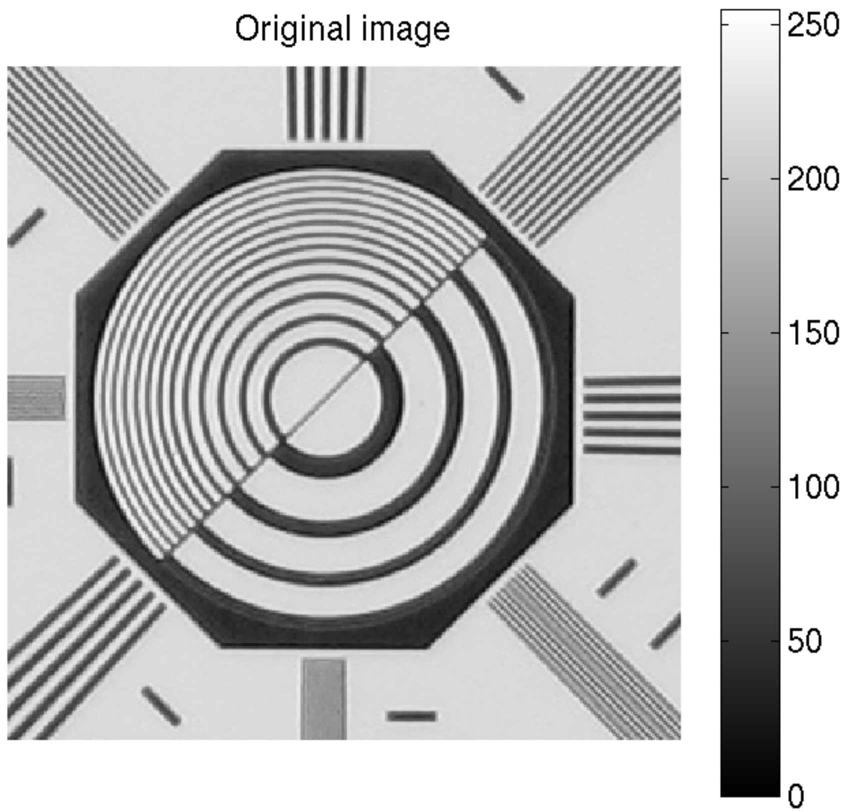
Sampled variant

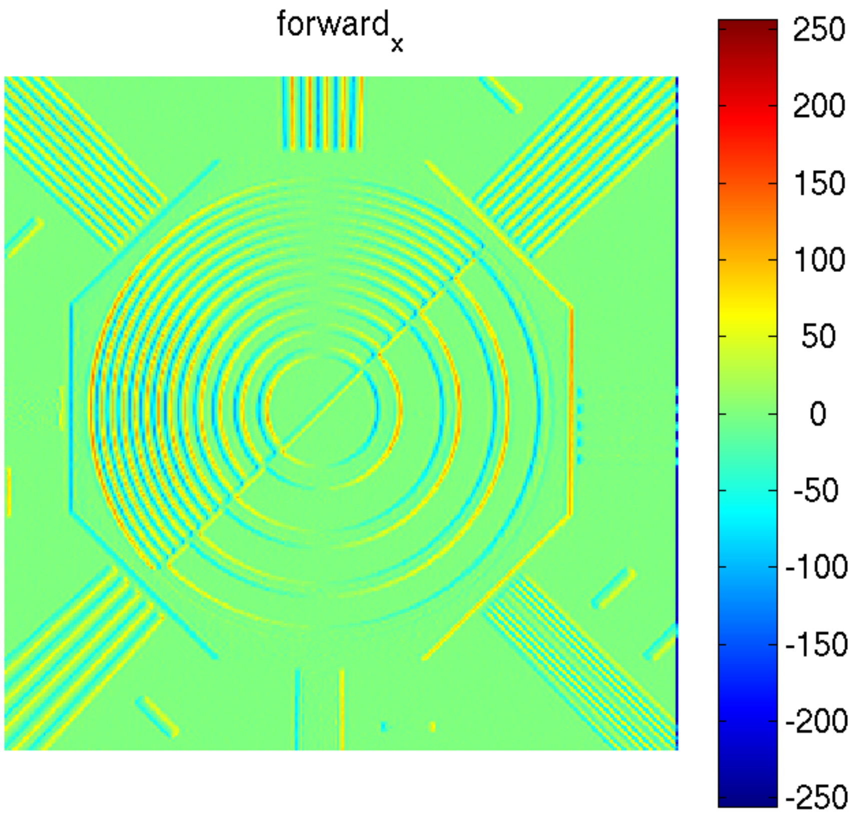
$$\frac{d}{dx}f(x) \approx \frac{f(x)-f(x-1)}{1}$$

Kernels for derivatives

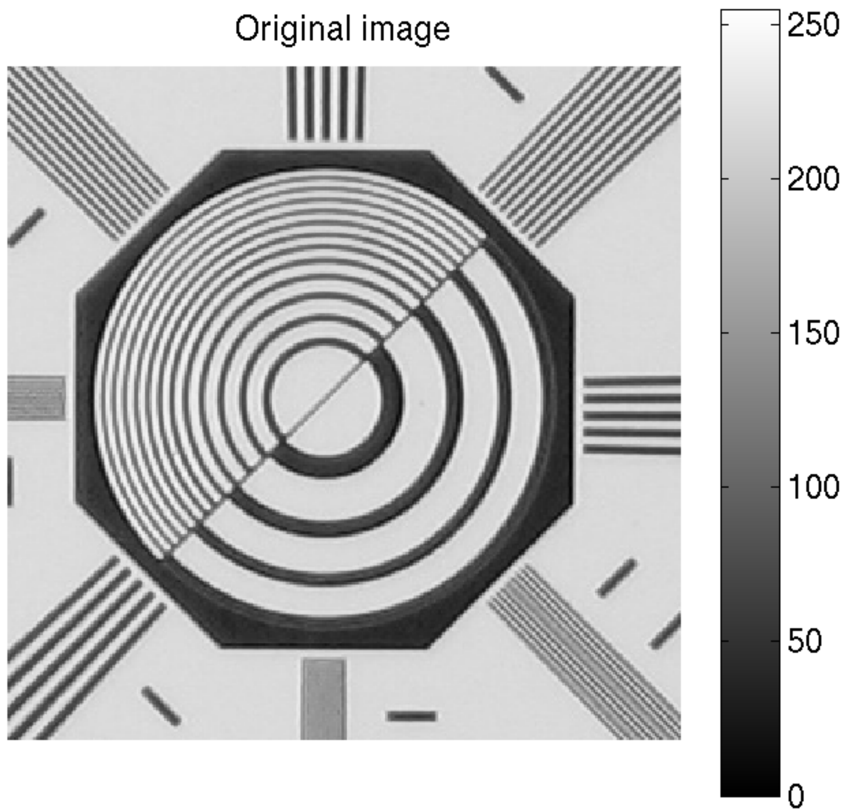
Image is 2D function $f(x, y)$. Derivatives may also be along $y-$ direction

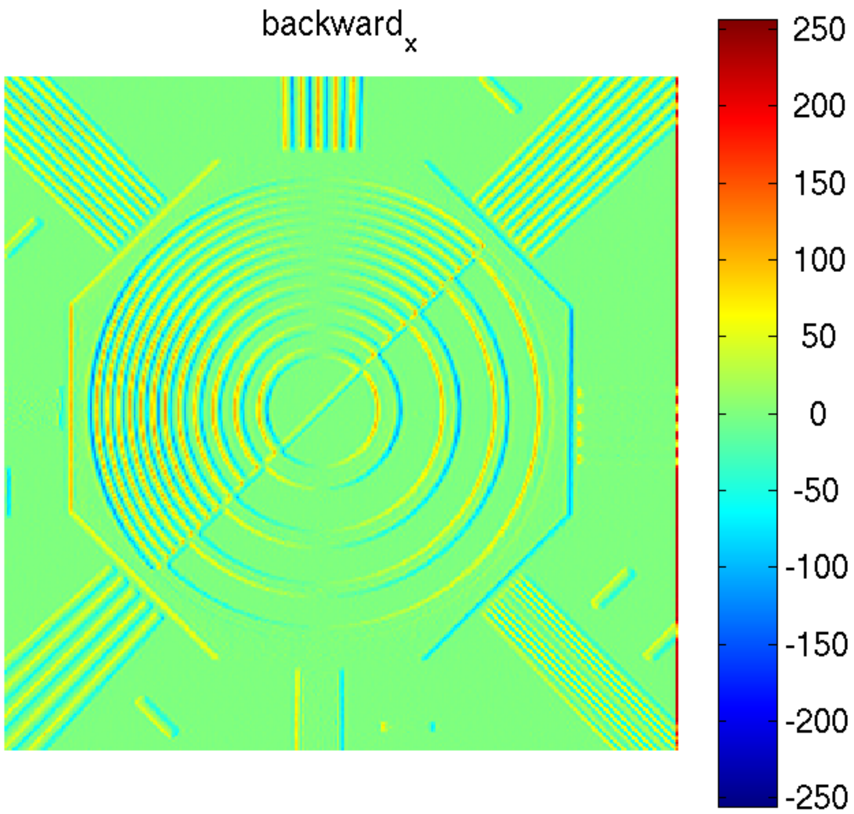
Forward difference — x direction



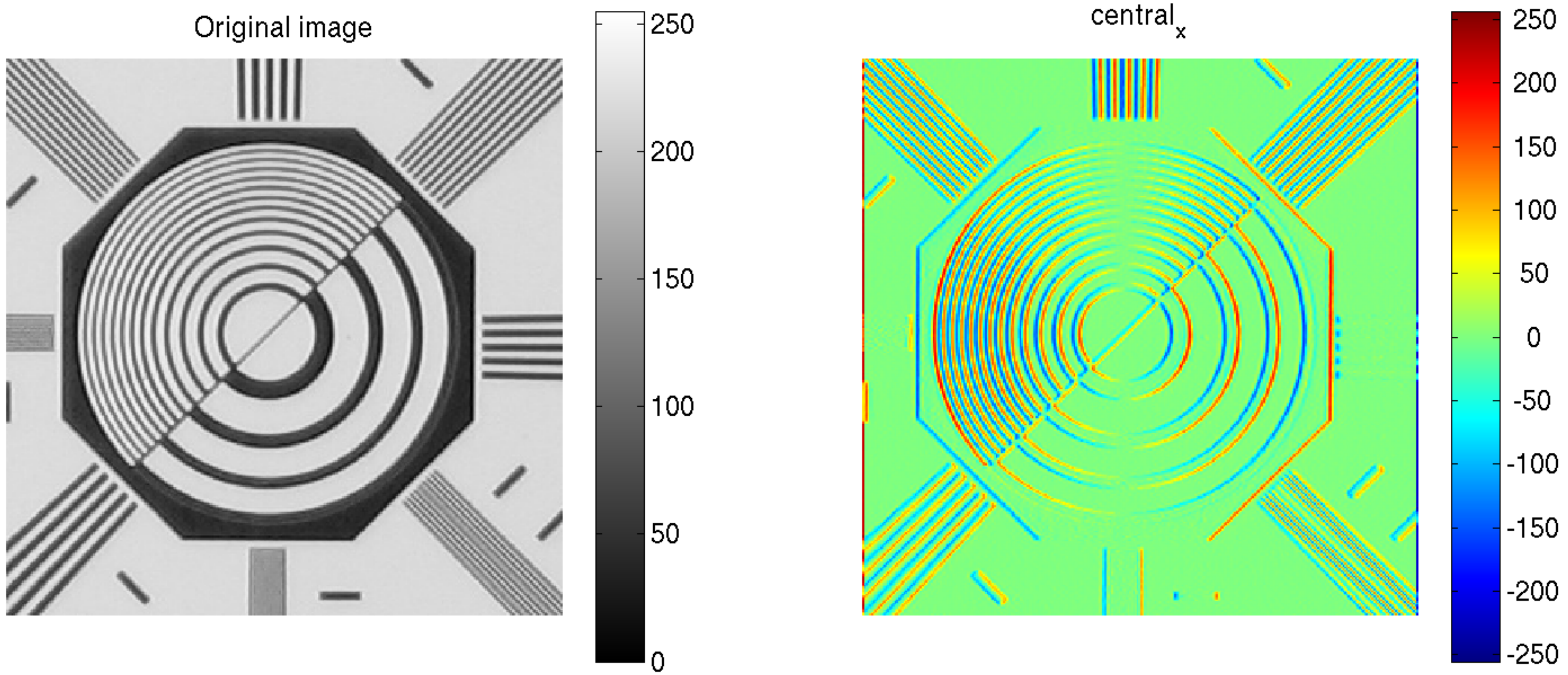


Backward difference — x direction

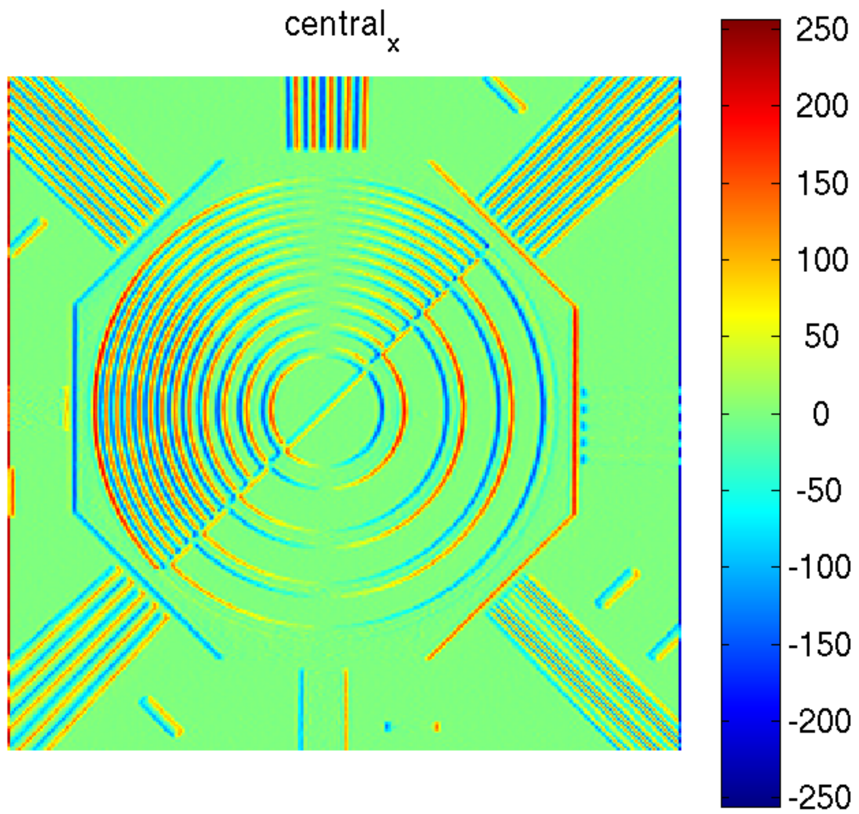


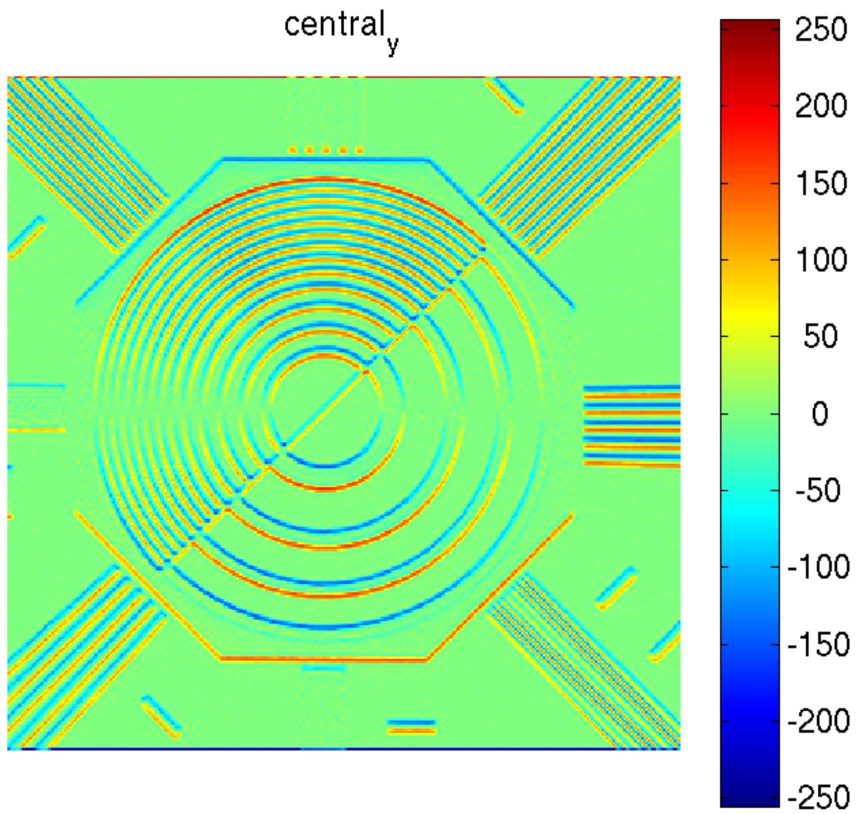


Central difference — x direction



Central difference — x and y direction





Second derivatives

Forward

$$\frac{d}{dx}f(x) \approx f(x+1) - f(x)$$

Backward

$$\frac{d}{dx}f(x) \approx f(x) - f(x-1)$$

Second derivatives

Forward

$$\frac{d}{dx}f(x) \approx f(x+1) - f(x)$$

Backward

$$\frac{d}{dx}f(x) \approx f(x) - f(x-1)$$

Difference of differences

$$\begin{aligned}\frac{d^2}{dx^2}f(x) &\approx (f(x+1) - f(x)) - (f(x) - f(x-1)) \\ &= f(x+1) - 2f(x) + f(x-1)\end{aligned}$$

Second derivatives

Forward

$$\frac{d}{dx}f(x) \approx f(x+1) - f(x)$$

Backward

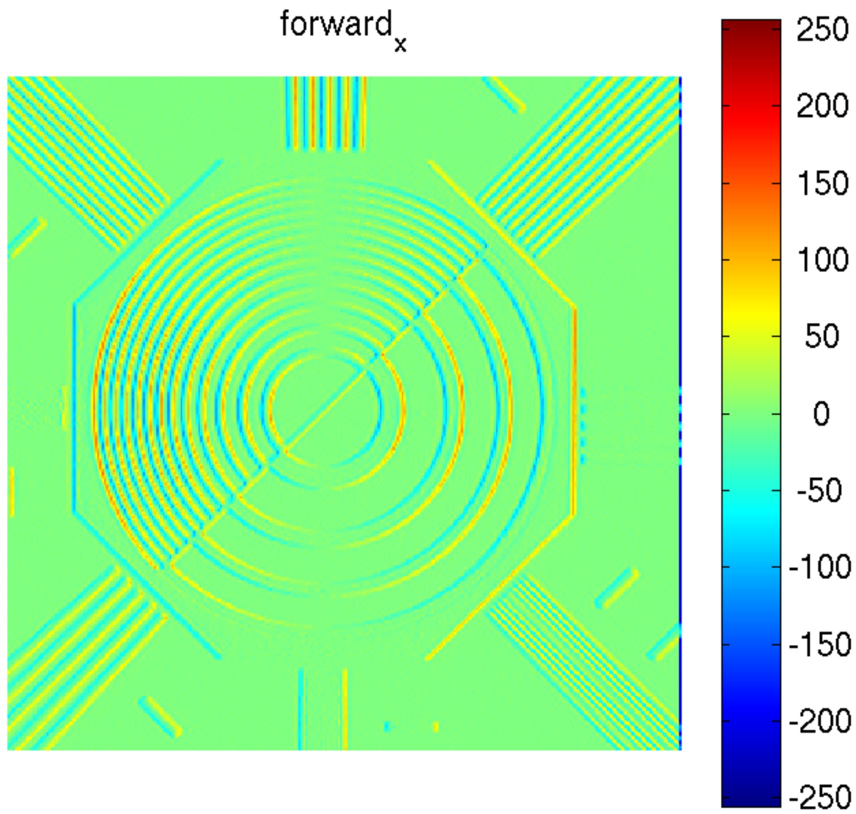
$$\frac{d}{dx}f(x) \approx f(x) - f(x-1)$$

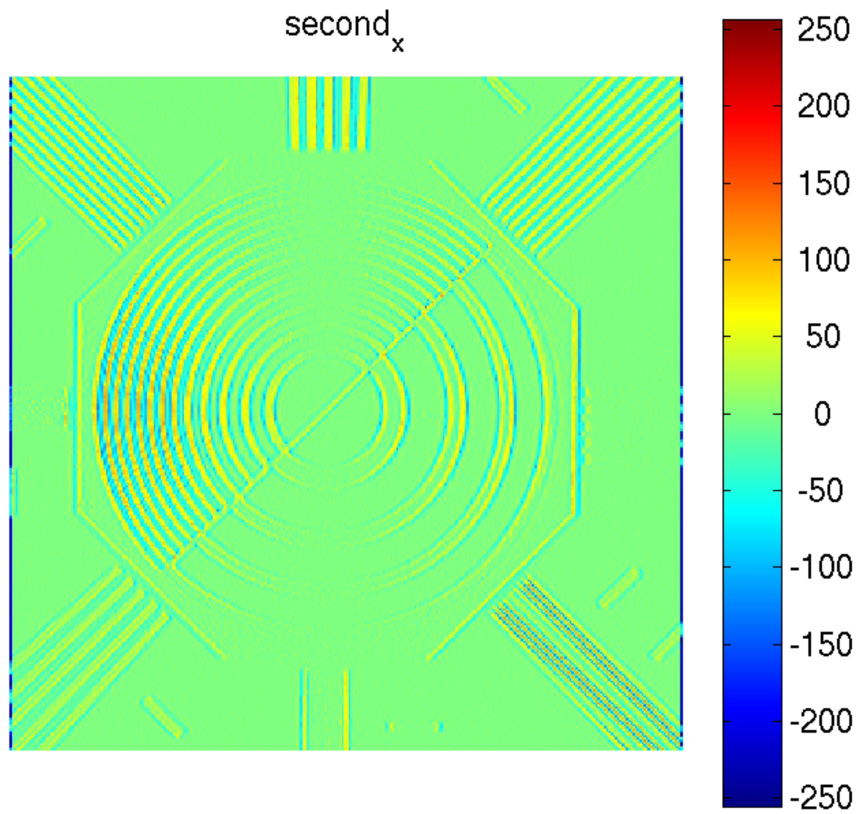
Difference of differences

$$\begin{aligned}\frac{d^2}{dx^2}f(x) &\approx (f(x+1) - f(x)) - (f(x) - f(x-1)) \\ &= f(x+1) - 2f(x) + f(x-1)\end{aligned}$$

$+1$	**-1**	$*$	$+1$	**-1**	$=$	$+1$	**-2**	$+1$

Second derivatives — derivative of derivative





2D derivatives

Differentiate in one dimension, ignore the other

$\frac{\partial}{\partial x}$

0	0	0
-1	0	+1
0	0	0

$\frac{\partial}{\partial y}$

0	-1	0
0	0	0
0	+1	0

$\frac{\partial^2}{\partial x^2}$

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

$\frac{\partial^2}{\partial y^2}$

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

2D derivatives with smoothing

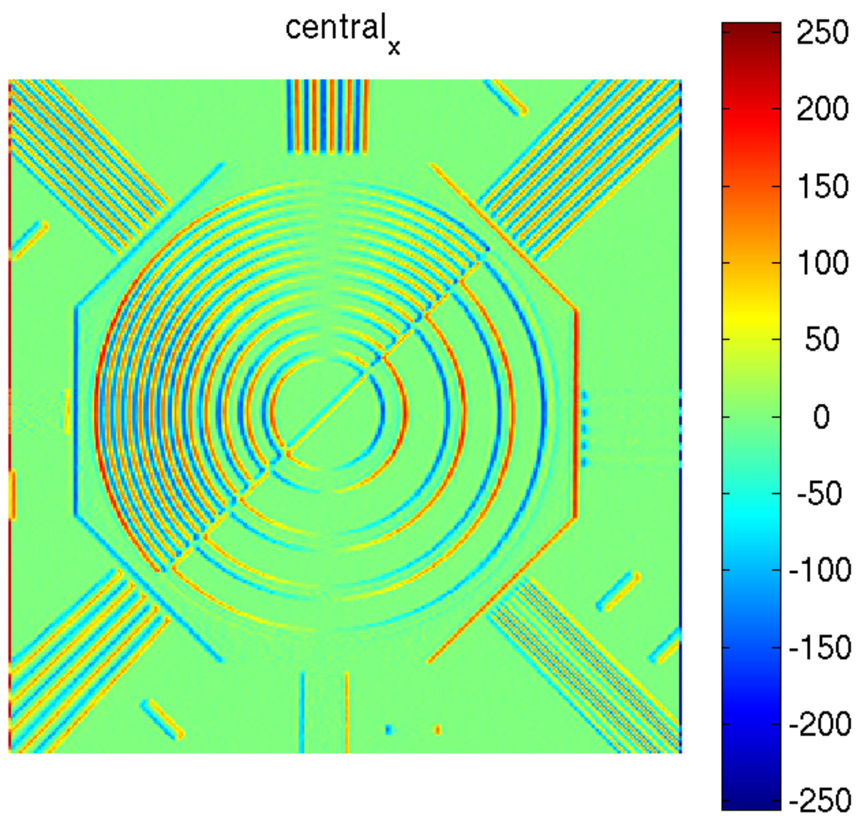
Differentiate in one dimension and smooth in the other

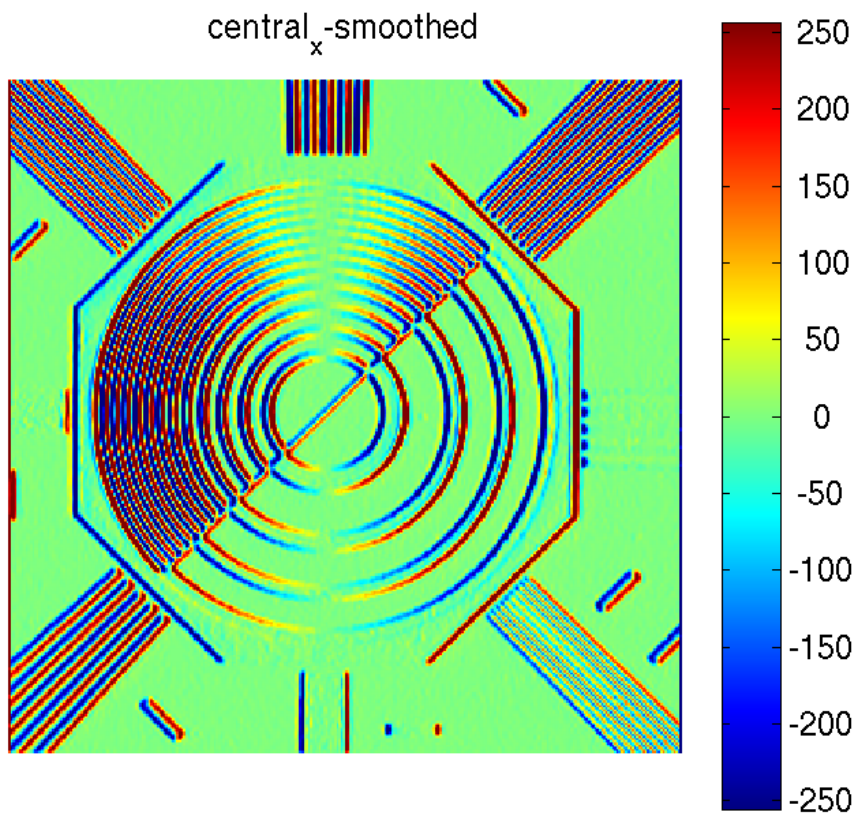
$$\begin{array}{|c|c|c|}\hline -1 & 0 & +1 \\ \hline \end{array} * \begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline 1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline -1 & 0 & +1 \\ \hline -1 & 0 & +1 \\ \hline -1 & 0 & +1 \\ \hline \end{array}$$

2D derivatives with smoothing

Differentiate in one dimension and smooth in the other

$$\begin{array}{|c|c|c|}\hline -1 & 0 & +1 \\ \hline \end{array} * \begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline 1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline -1 & 0 & +1 \\ \hline -1 & 0 & +1 \\ \hline -1 & 0 & +1 \\ \hline \end{array}$$





The Gradient

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- Magnitude

$$\|\nabla f(x,y)\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}\,,$$

  is steepness in

- direction

$$\psi = \mathsf{atan}\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)\,,$$

A way to do the edge detection. Edge direction is perpendicular to ψ.

The Laplacian

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Sum of second derivatives in x and y directions.
- Sort of an overall curvature.

The Laplacian

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Sum of second derivatives in x and y directions.
- Sort of an overall curvature.

With kernels:

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

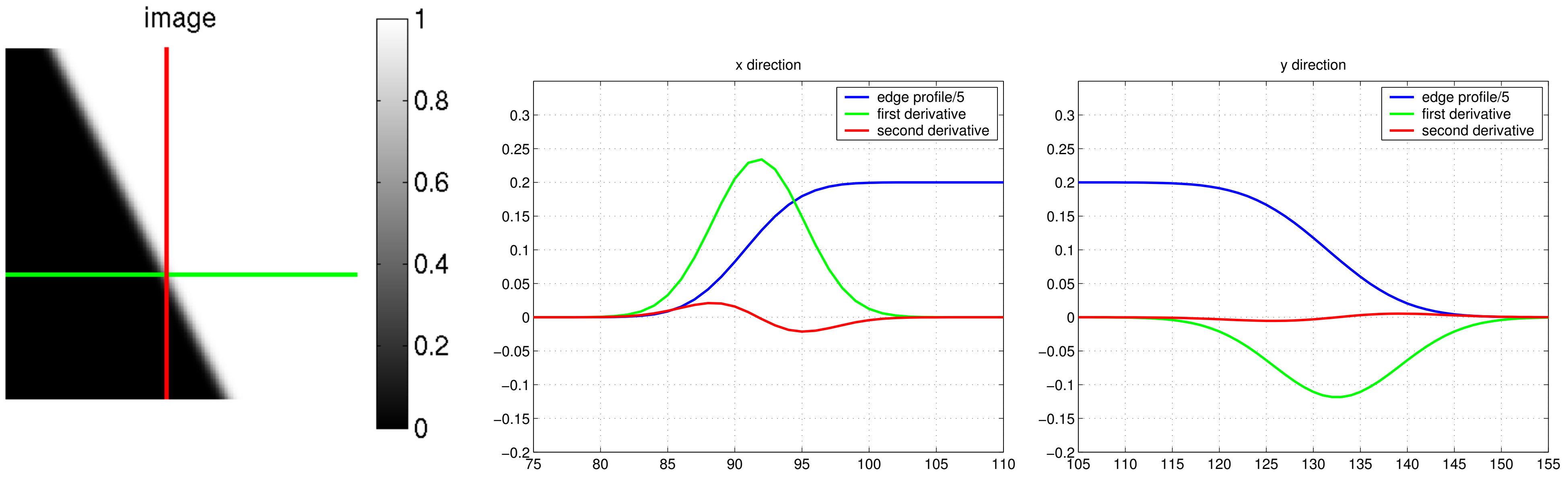
$+$

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

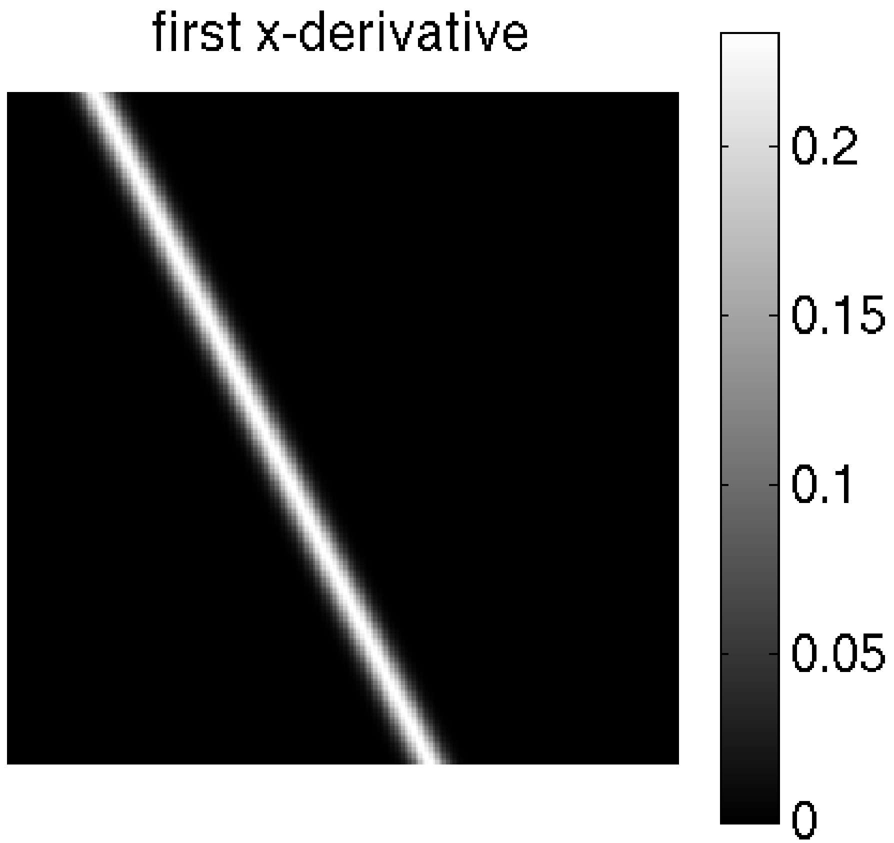
$=$

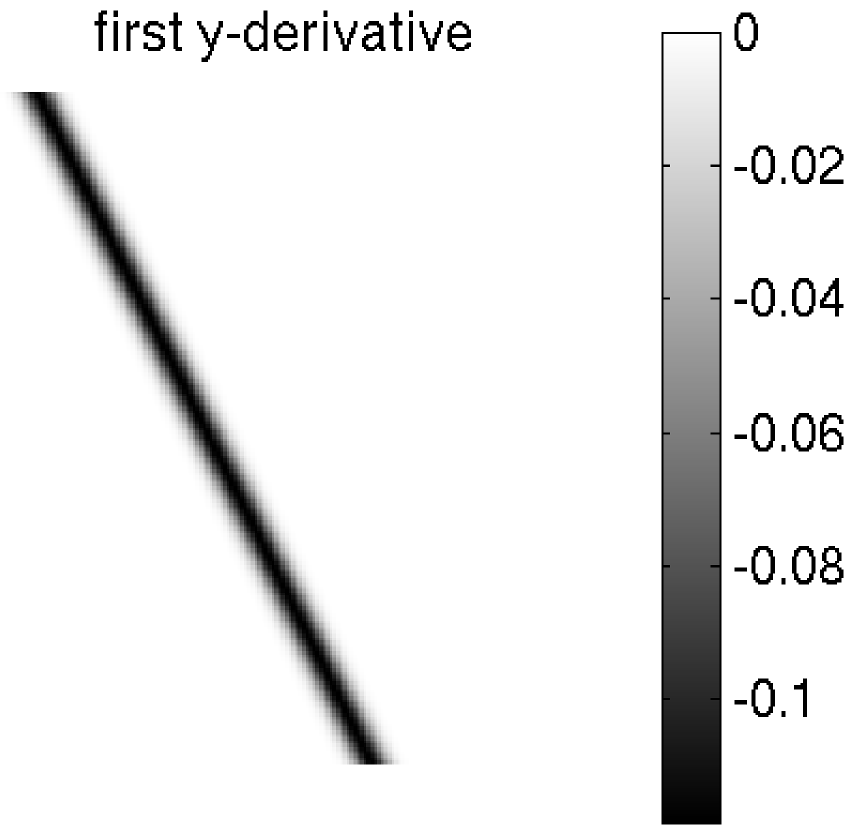
0	+1	0
1	-4	1
0	+1	0

What is an edge?



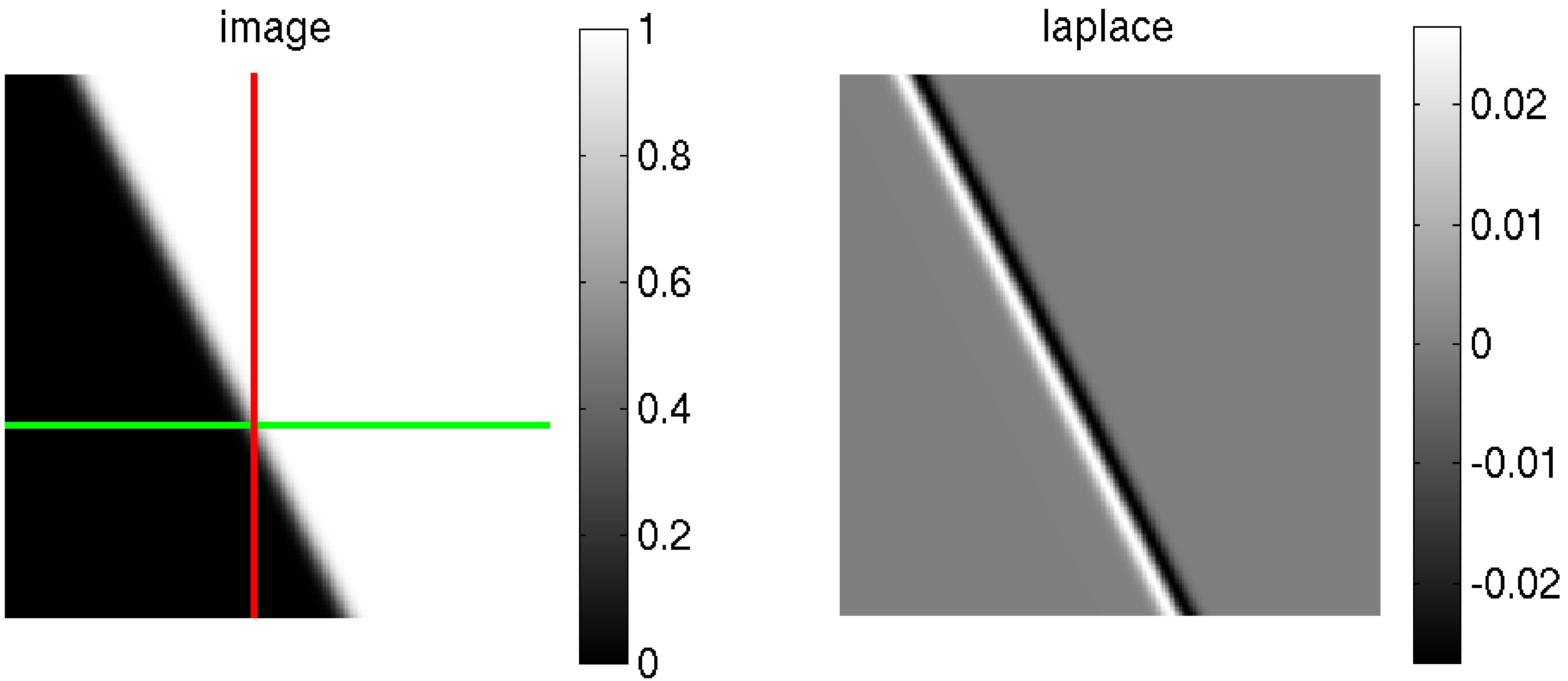
Partial derivatives





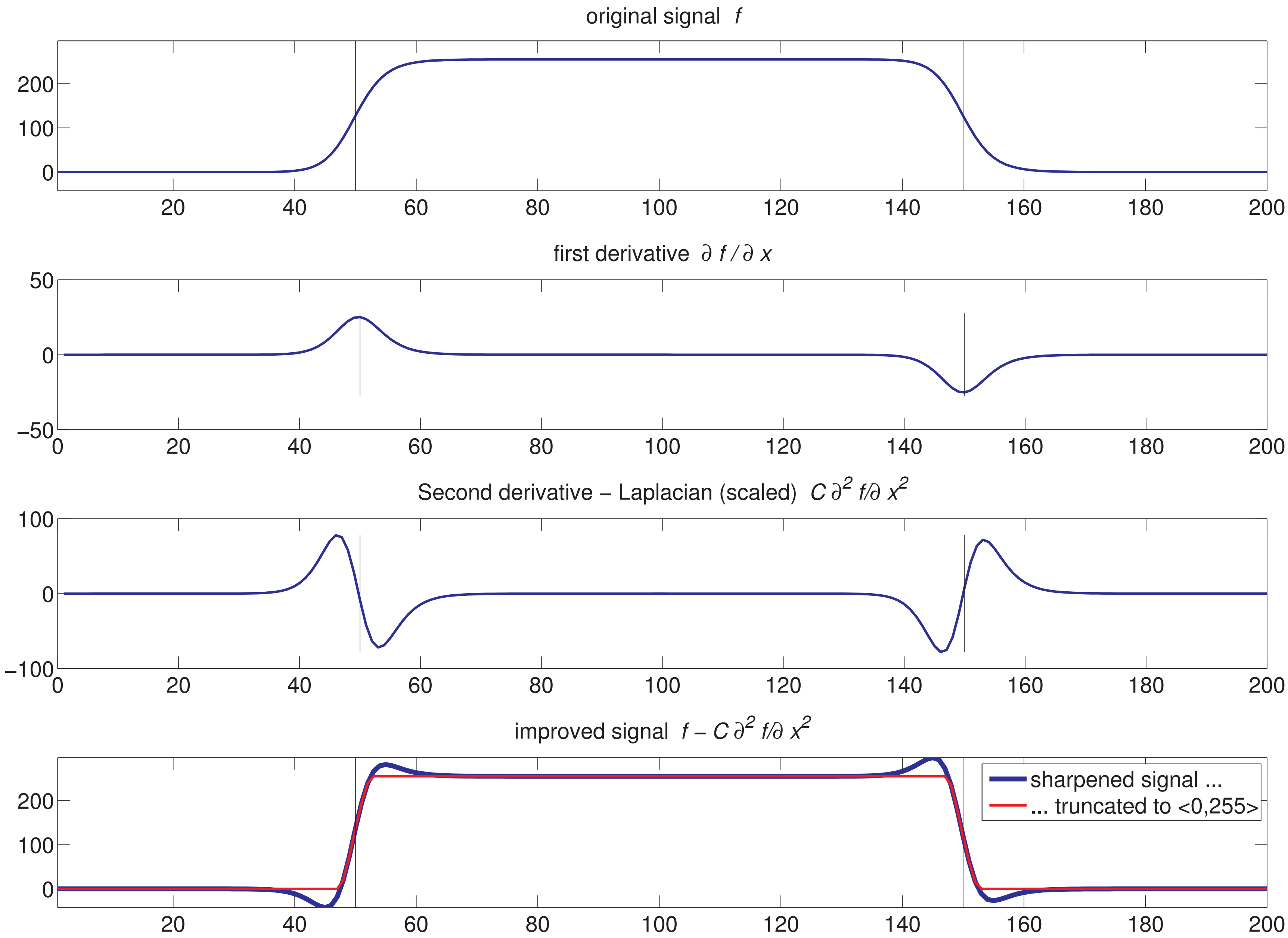
Extrema of partial derivatives are good candidates for edges.

Laplacian



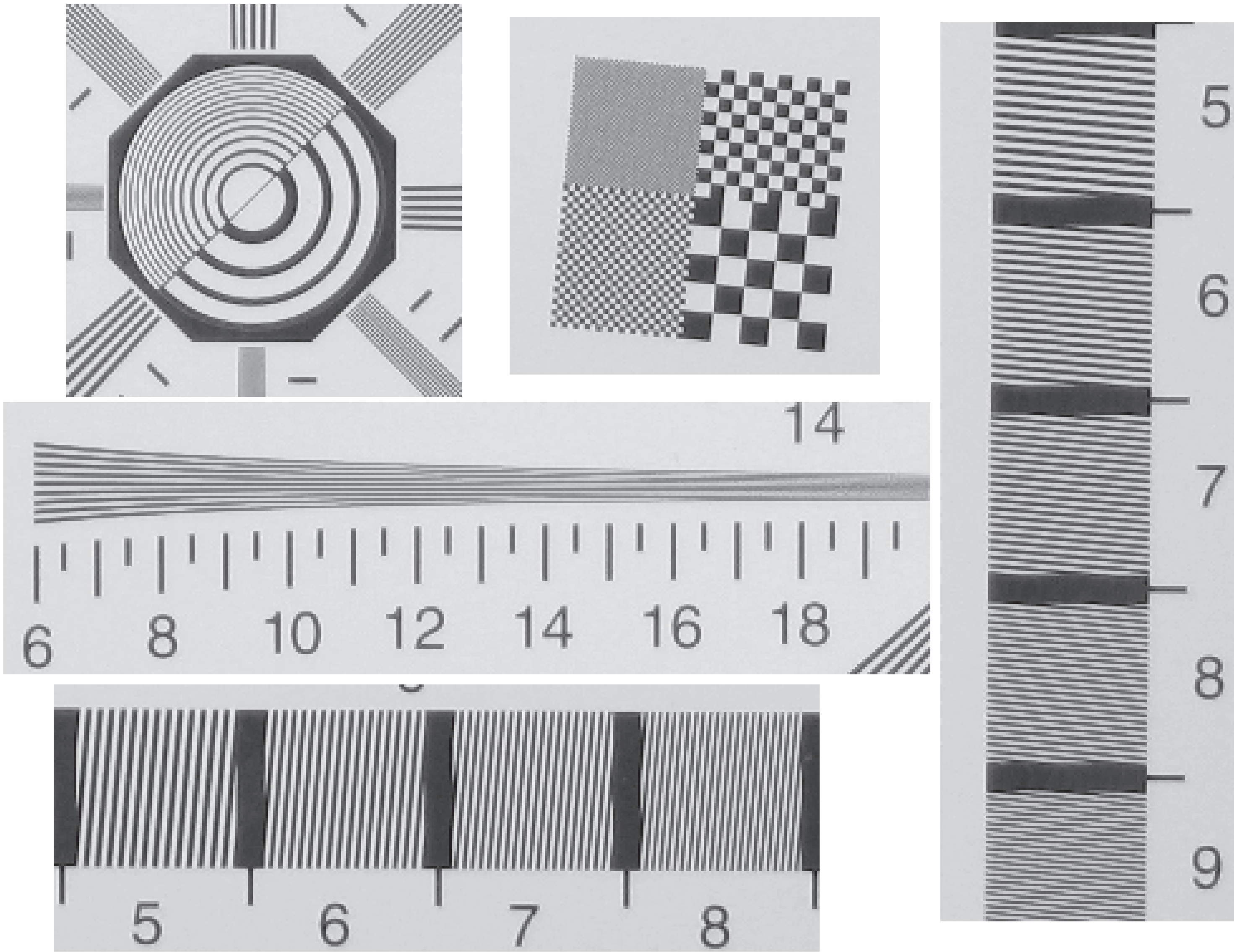
Places where the Laplacian changes from positive to negative are also good potential edges.

Laplacian for sharpenning

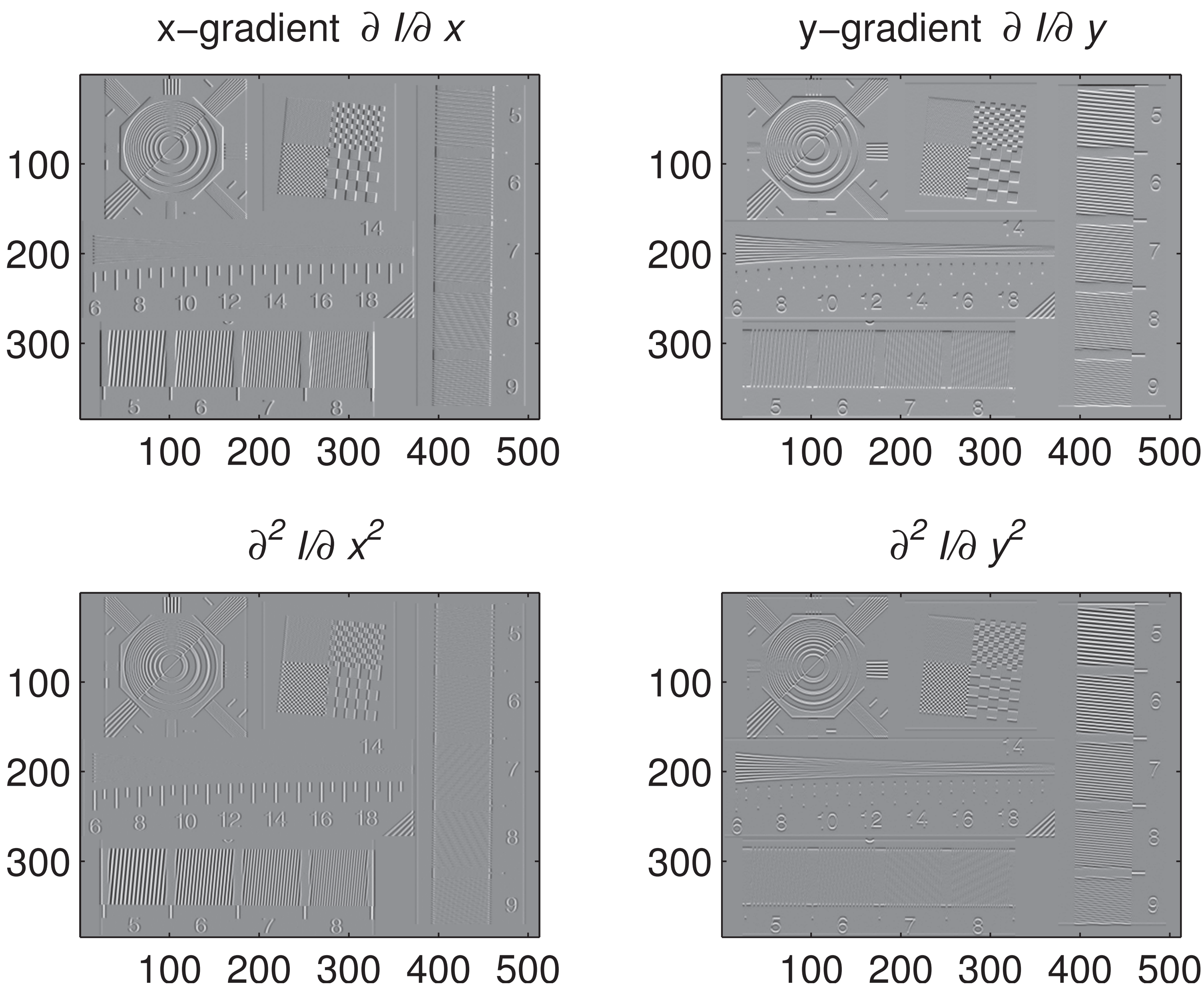


Laplacian for sharpenning – input

Original image

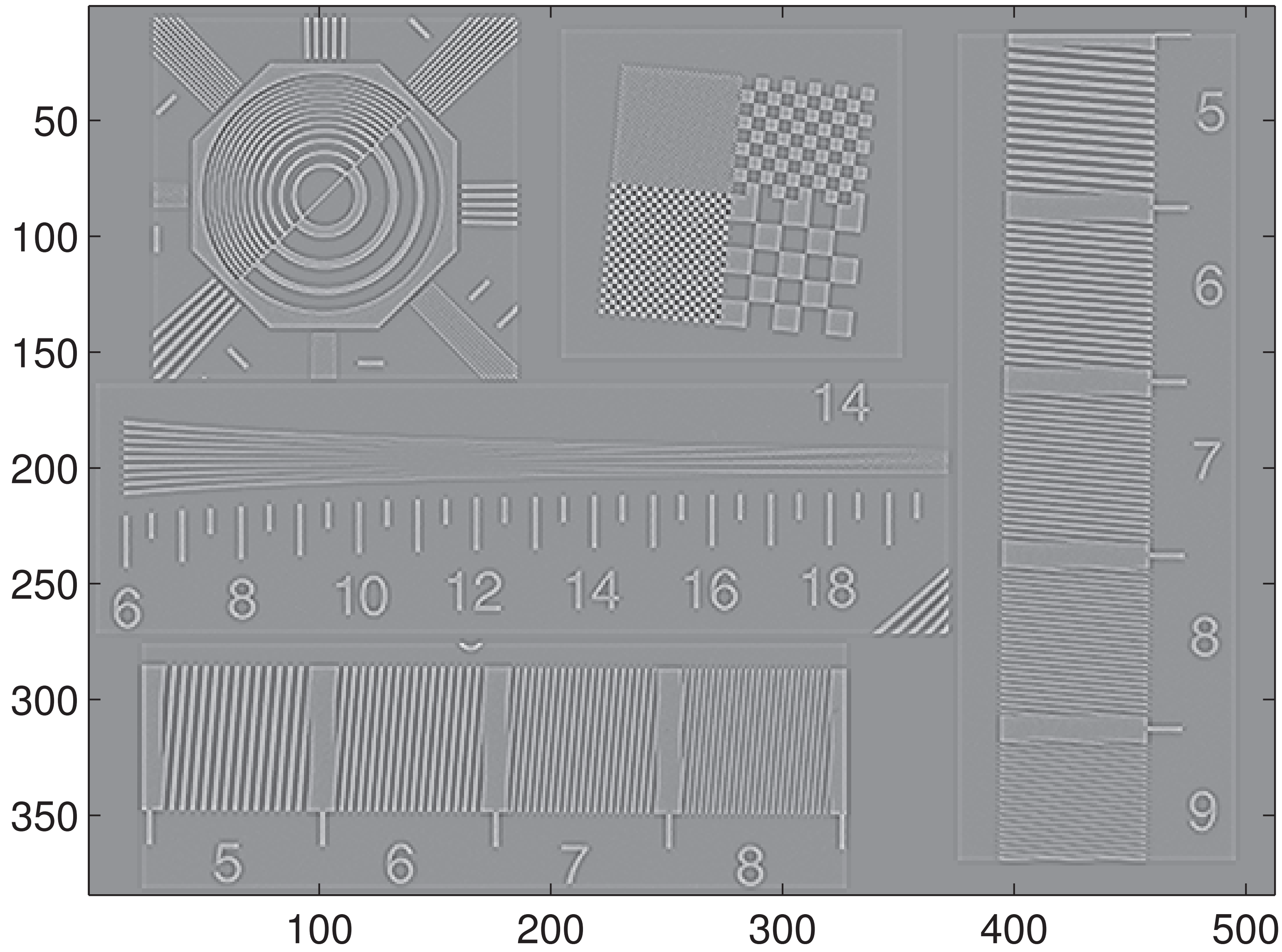


Laplacian for sharpenning – gradients



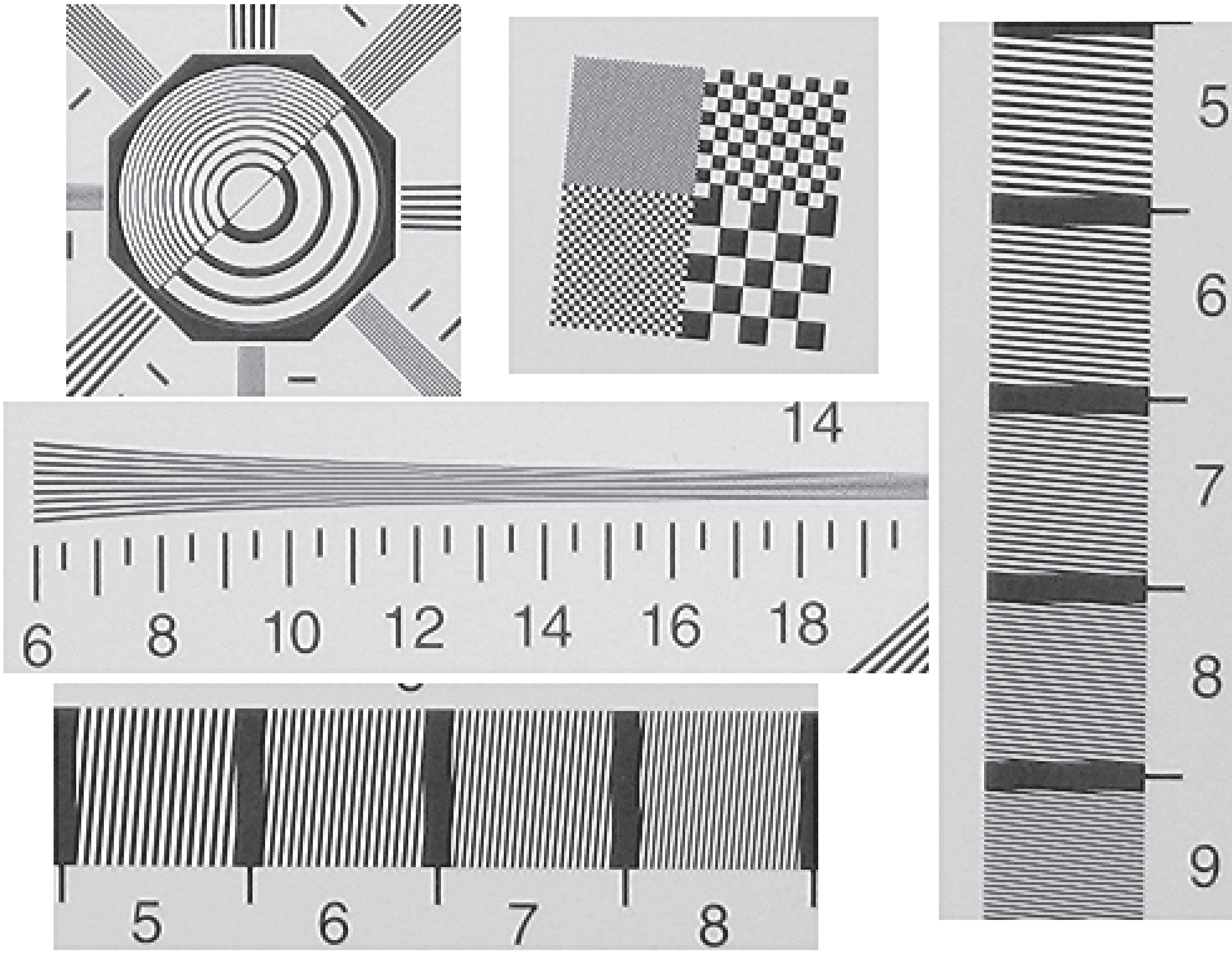
Laplacian for sharpenning – Laplacian

Laplacian: $\nabla = \partial^2 I / \partial x^2 + \partial^2 I / \partial y^2$



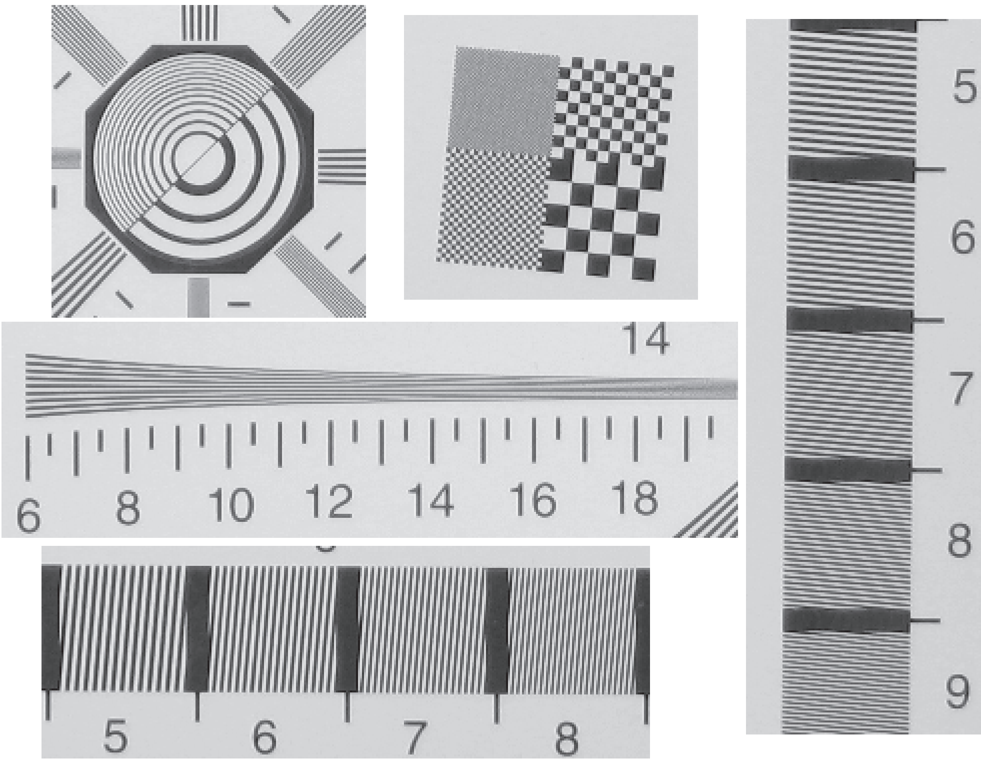
Laplacian for sharpenning – result

Sharpened image, C=0.5

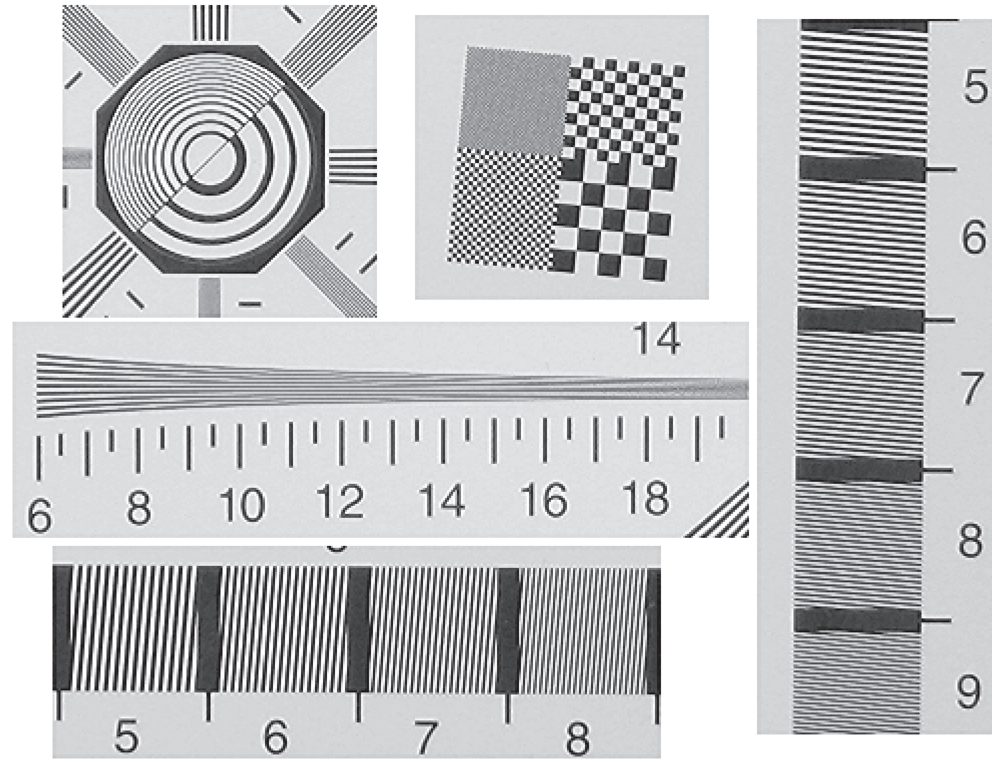


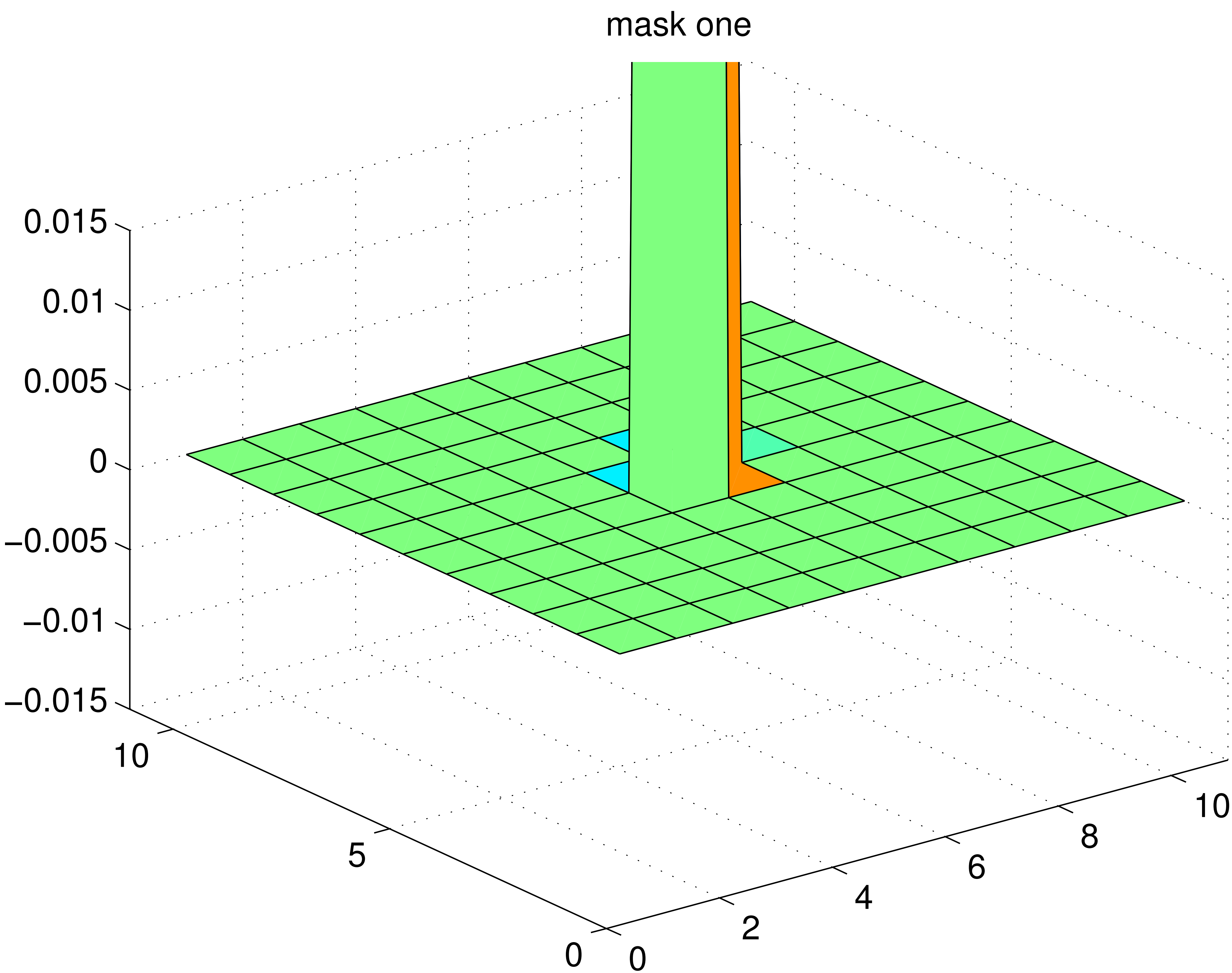
Laplacian for sharpenning – side by side

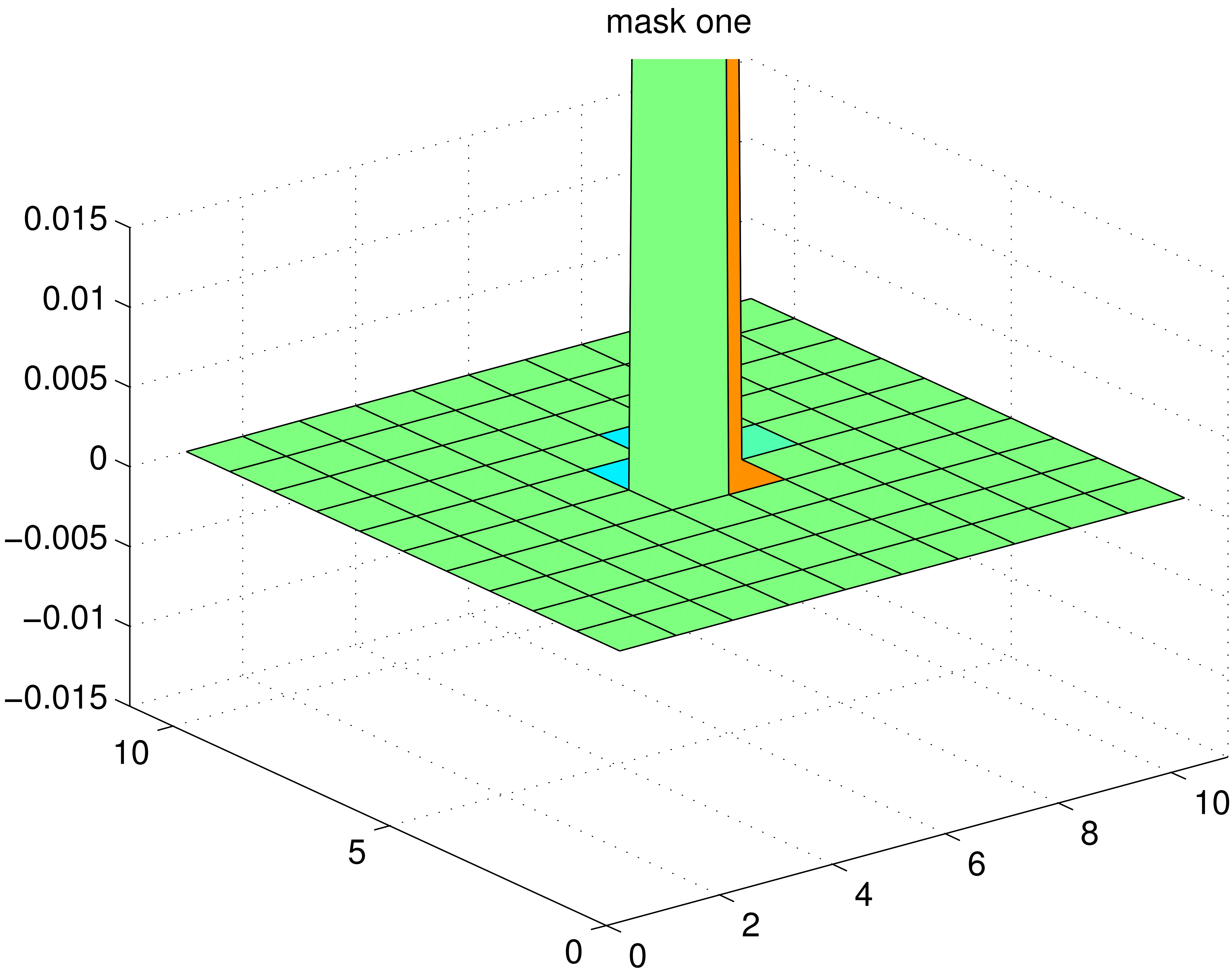
Original image

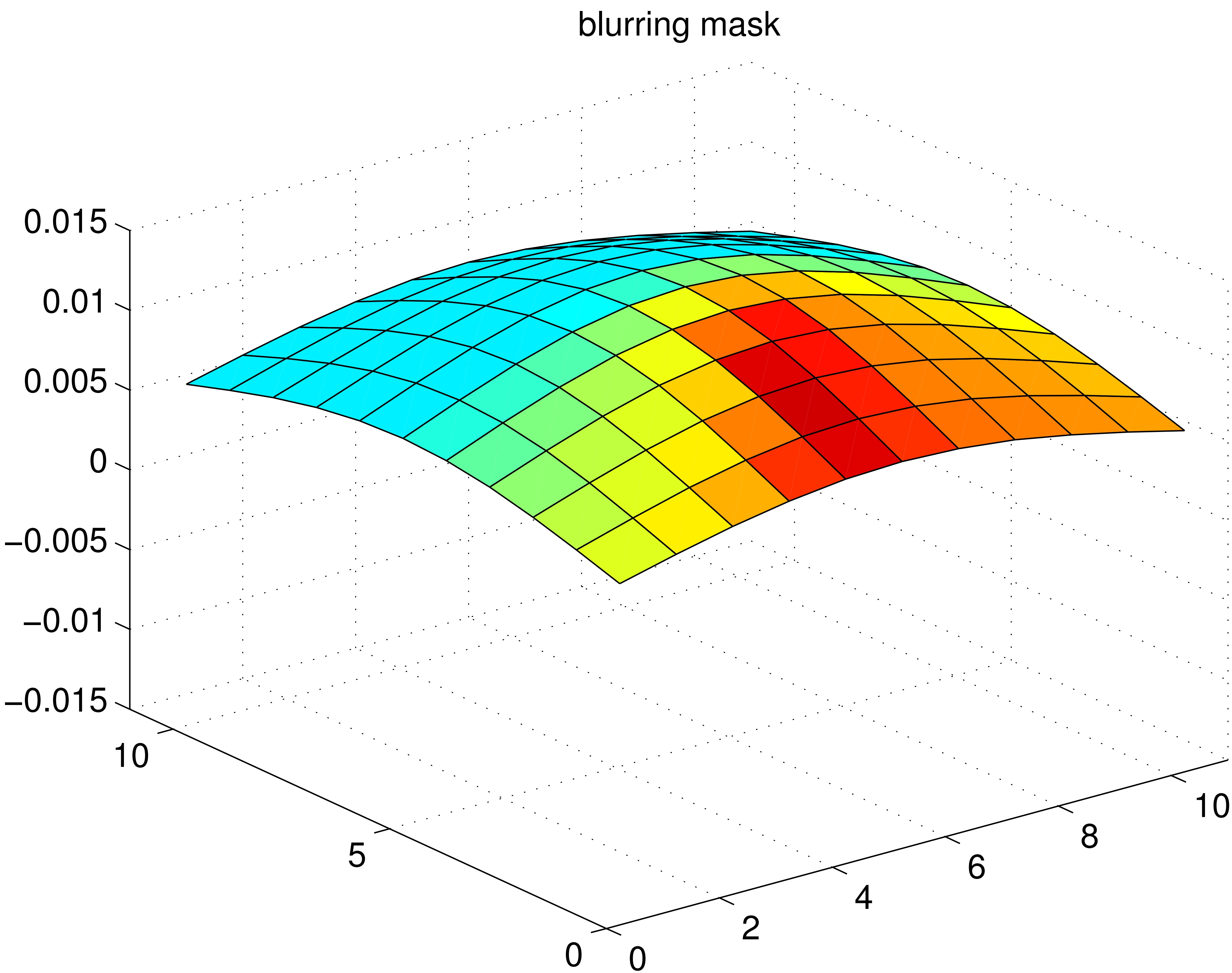


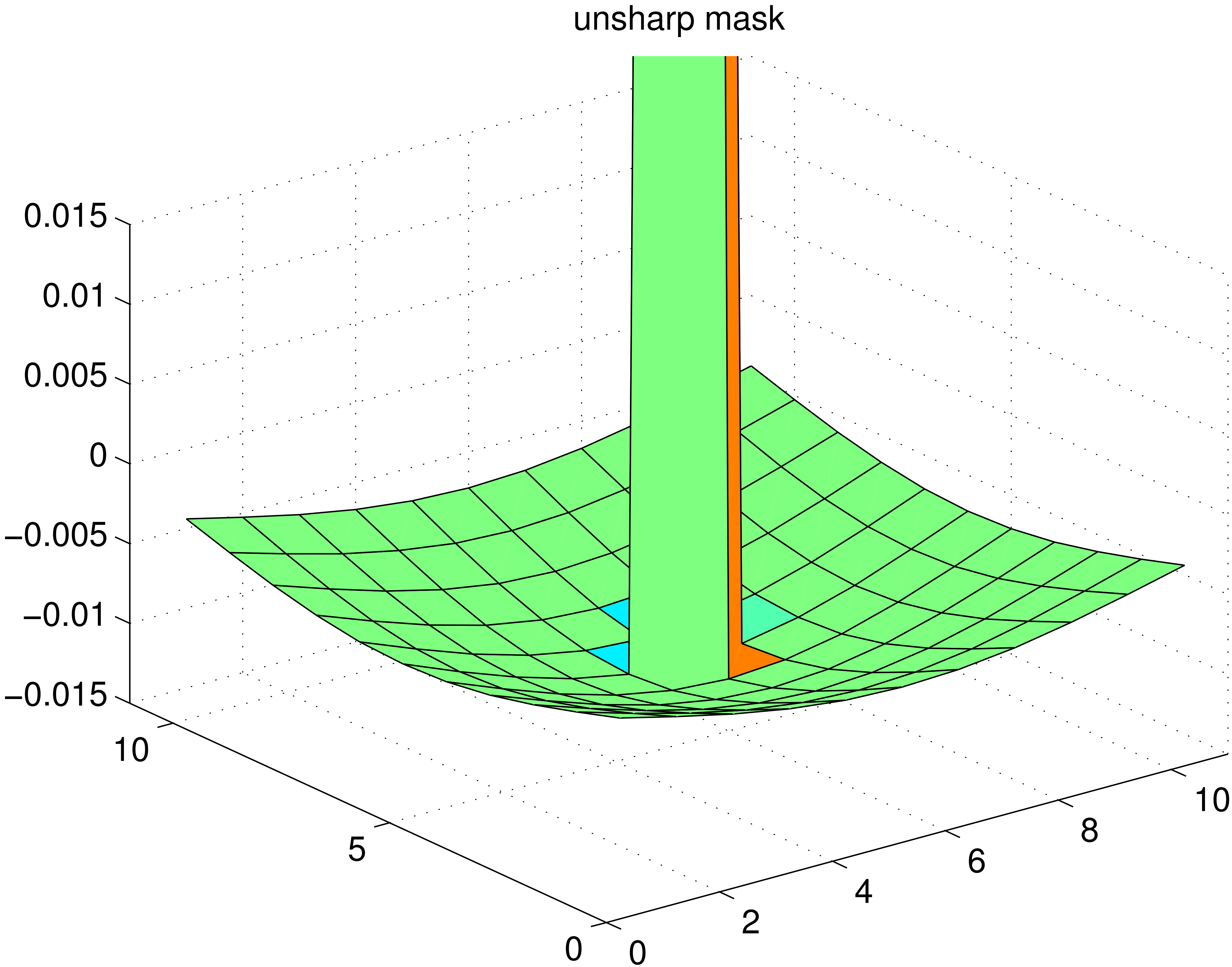
Sharpened image, C=0.5



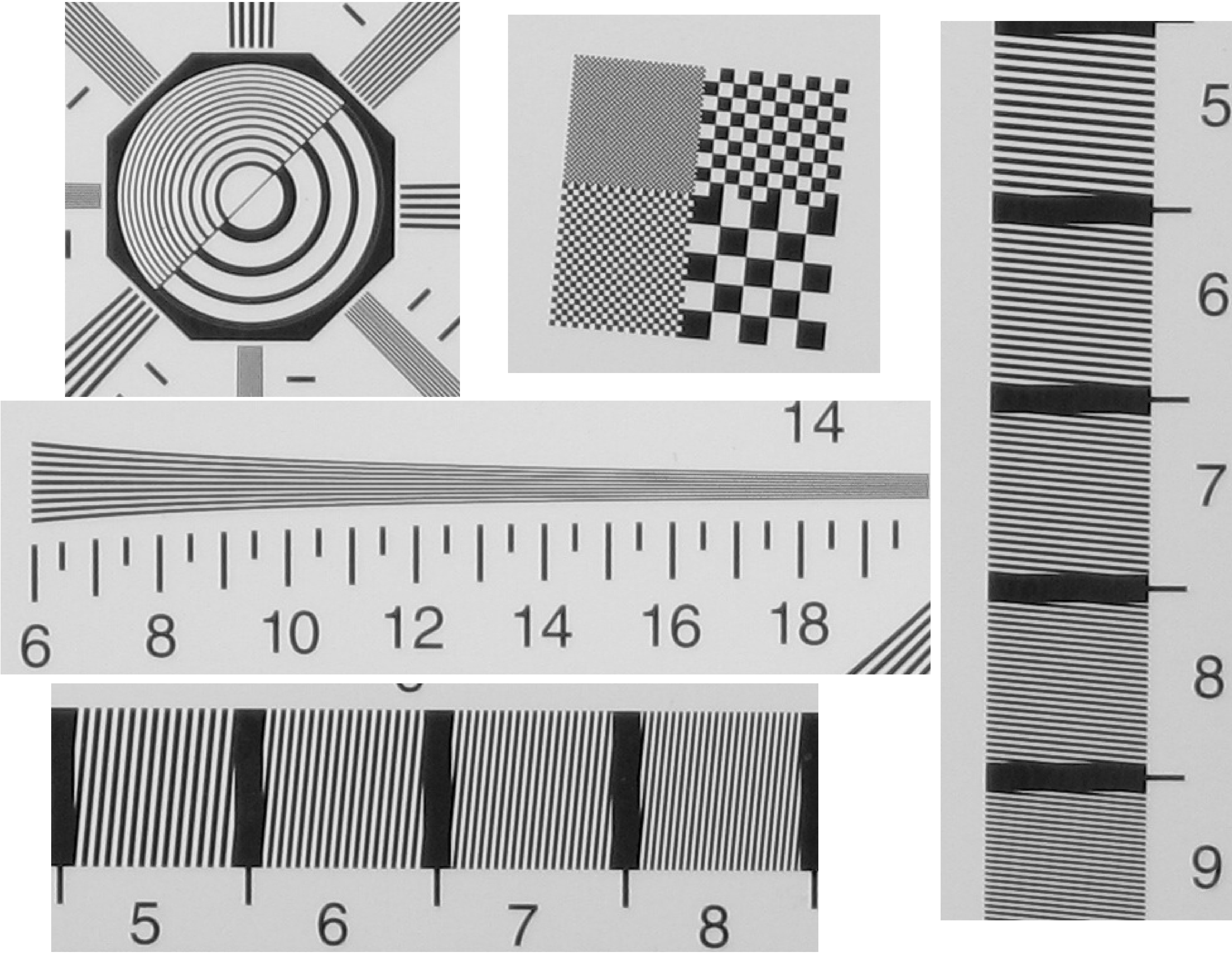

mask one
0.015
0.01
0.005
0
−0.005
−0.01
−0.015
10
5
0
0
2
4
6
8
10

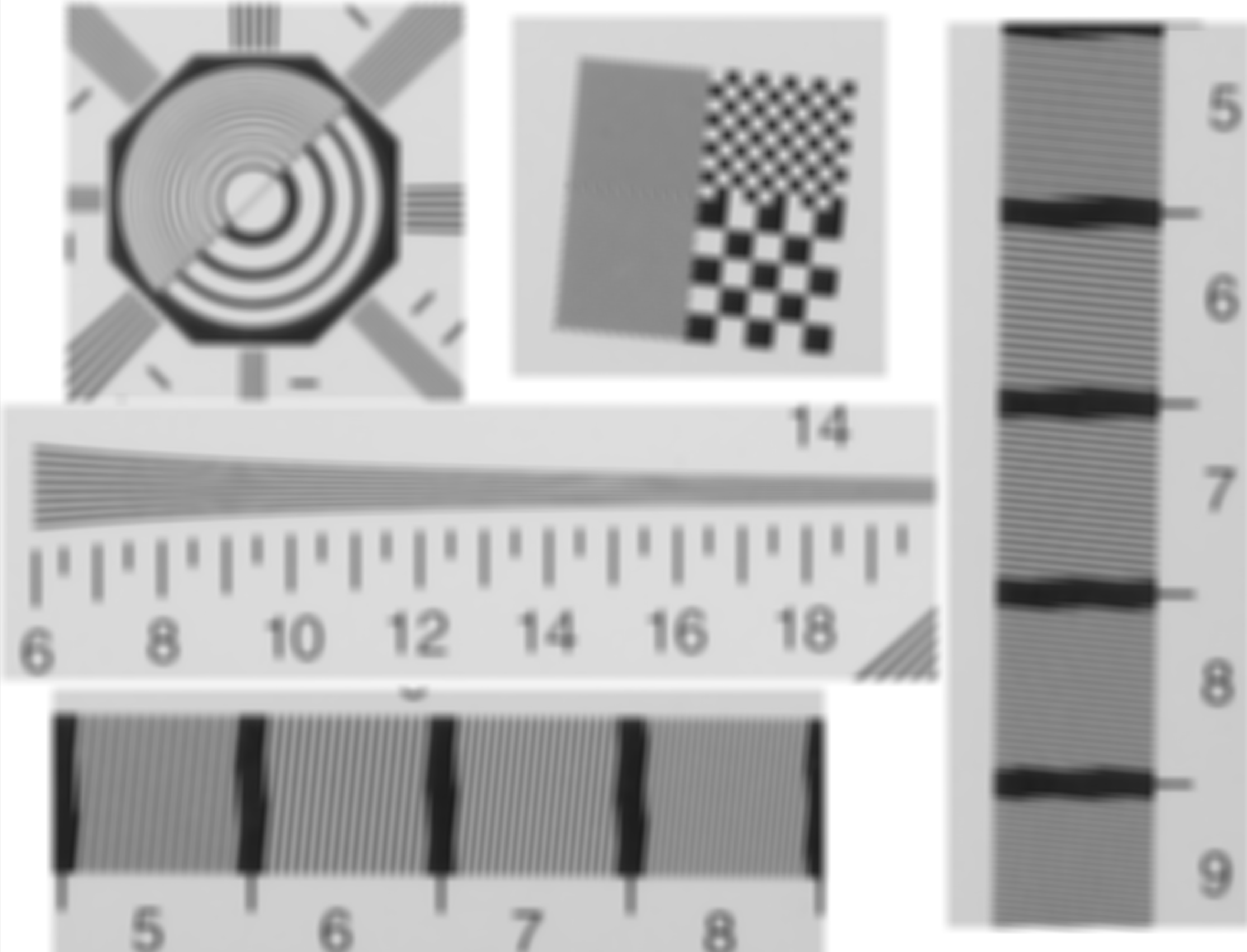

mask one
0.015
0.01
0.005
0
−0.005
−0.01
−0.015
10
5
0
0
2
4
6
8
10

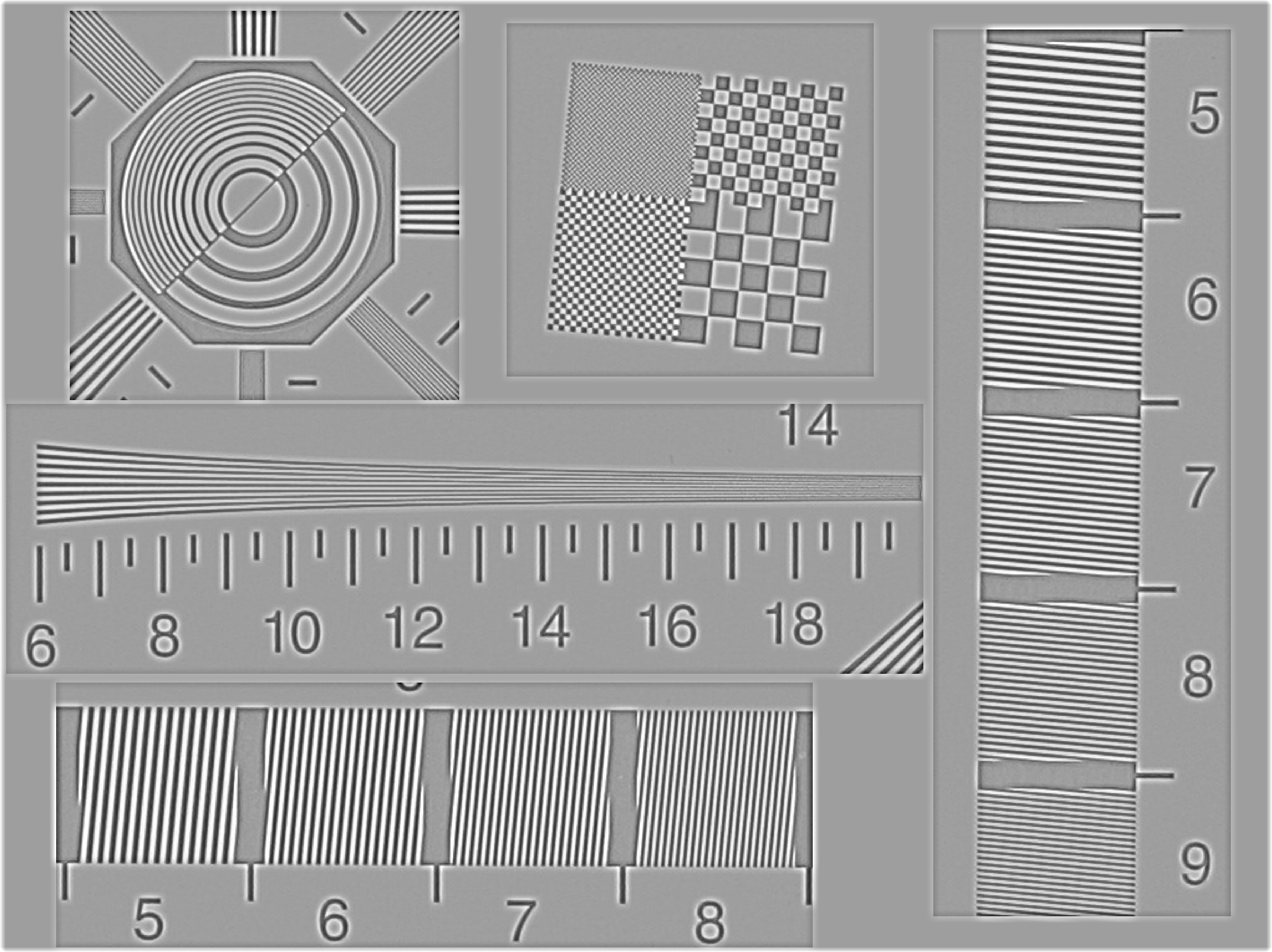

blurring mask
0.015
0.01
0.005
0
−0.005
−0.01
−0.015
10
5
0
0
2
4
6
8
10

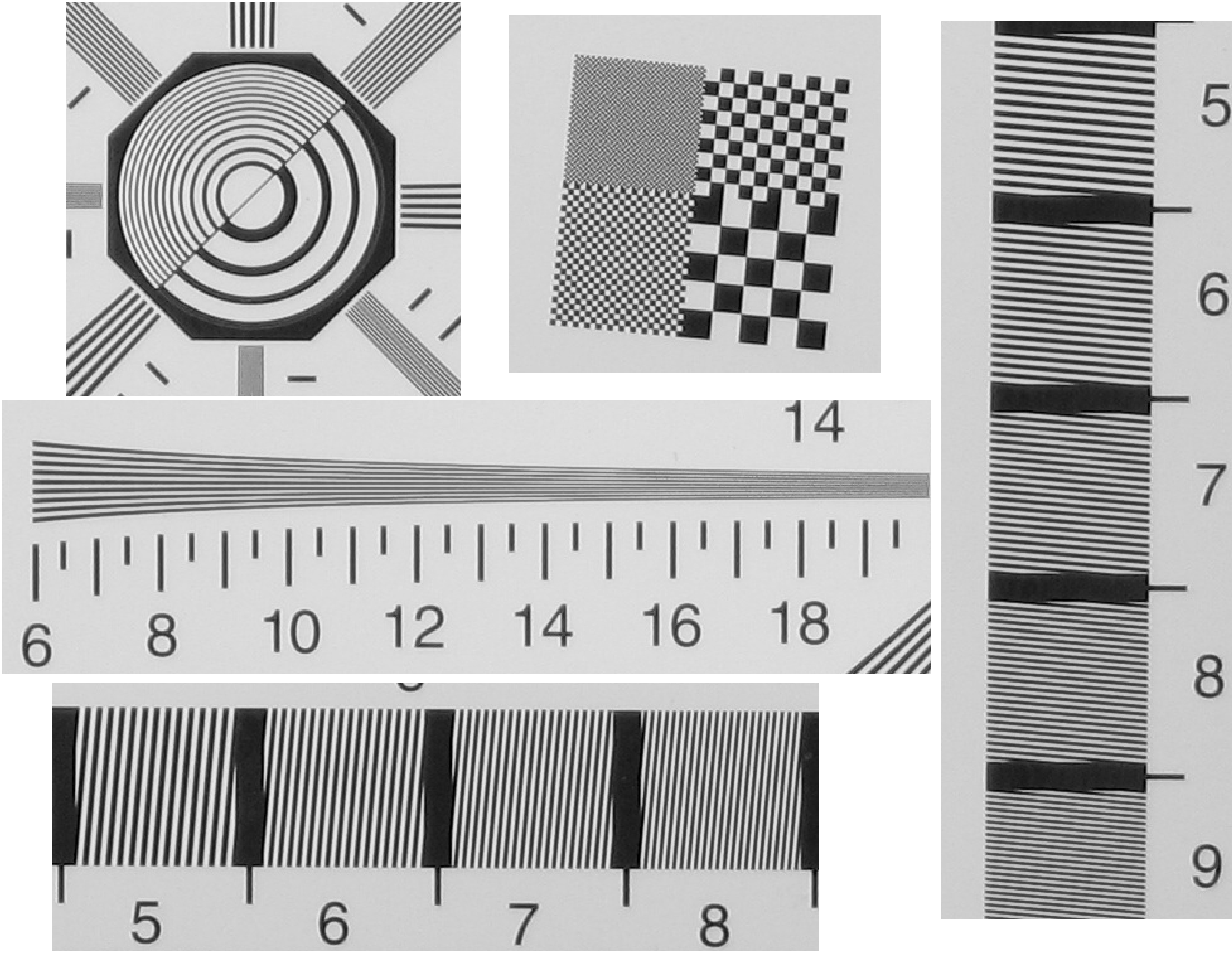

unsharp mask
0.015
0.01
0.005
0
−0.005
−0.01
−0.015
10
5
0
0
2
4
6
8
10

−0.0044	−0.0053	−0.0061	−0.0067	−0.0071	−0.0073	−0.0071	−0.0067	−0.0061	−0.0053	−0.0044
−0.0053	−0.0063	−0.0073	−0.0080	−0.0085	−0.0087	−0.0085	−0.0080	−0.0073	−0.0063	−0.0053
−0.0061	−0.0073	−0.0083	−0.0092	−0.0098	−0.0100	−0.0098	−0.0092	−0.0083	−0.0073	−0.0061
−0.0067	−0.0080	−0.0092	−0.0102	−0.0108	−0.0110	−0.0108	−0.0102	−0.0092	−0.0080	−0.0067
−0.0071	−0.0085	−0.0098	−0.0108	−0.0115	−0.0117	−0.0115	−0.0108	−0.0098	−0.0085	−0.0071
−0.0073	−0.0087	−0.0100	−0.0110	−0.0117	1.9880	−0.0117	−0.0110	−0.0100	−0.0087	−0.0073
−0.0071	−0.0085	−0.0098	−0.0108	−0.0115	−0.0117	−0.0115	−0.0108	−0.0098	−0.0085	−0.0071
−0.0067	−0.0080	−0.0092	−0.0102	−0.0108	−0.0110	−0.0108	−0.0102	−0.0092	−0.0080	−0.0067
−0.0061	−0.0073	−0.0083	−0.0092	−0.0098	−0.0100	−0.0098	−0.0092	−0.0083	−0.0073	−0.0061
−0.0053	−0.0063	−0.0073	−0.0080	−0.0085	−0.0087	−0.0085	−0.0080	−0.0073	−0.0063	−0.0053
−0.0044	−0.0053	−0.0061	−0.0067	−0.0071	−0.0073	−0.0071	−0.0067	−0.0061	−0.0053	−0.0044

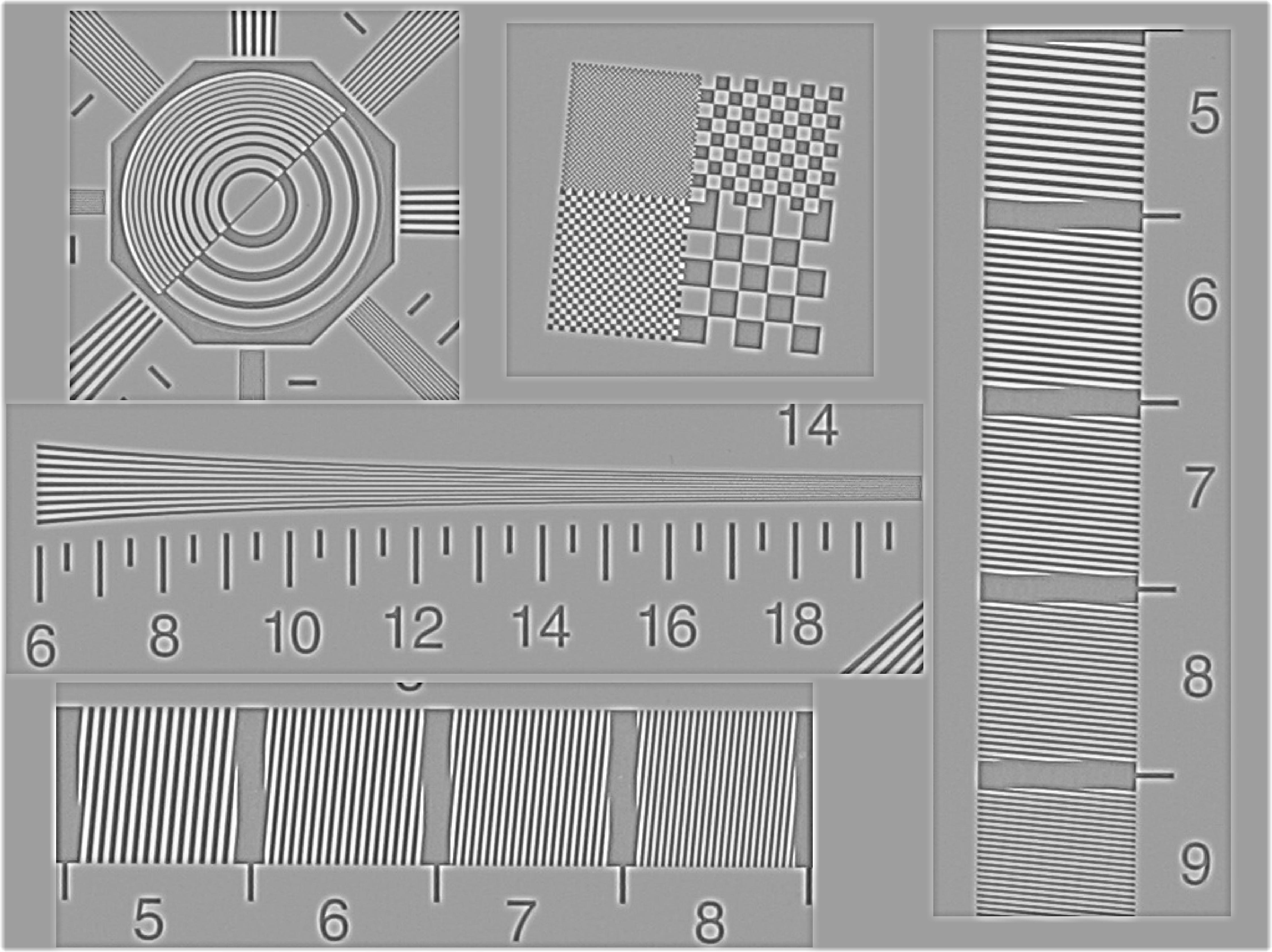

14
6
8
10
12
14
16
18
5
6
7
8
5
6
7
8
9

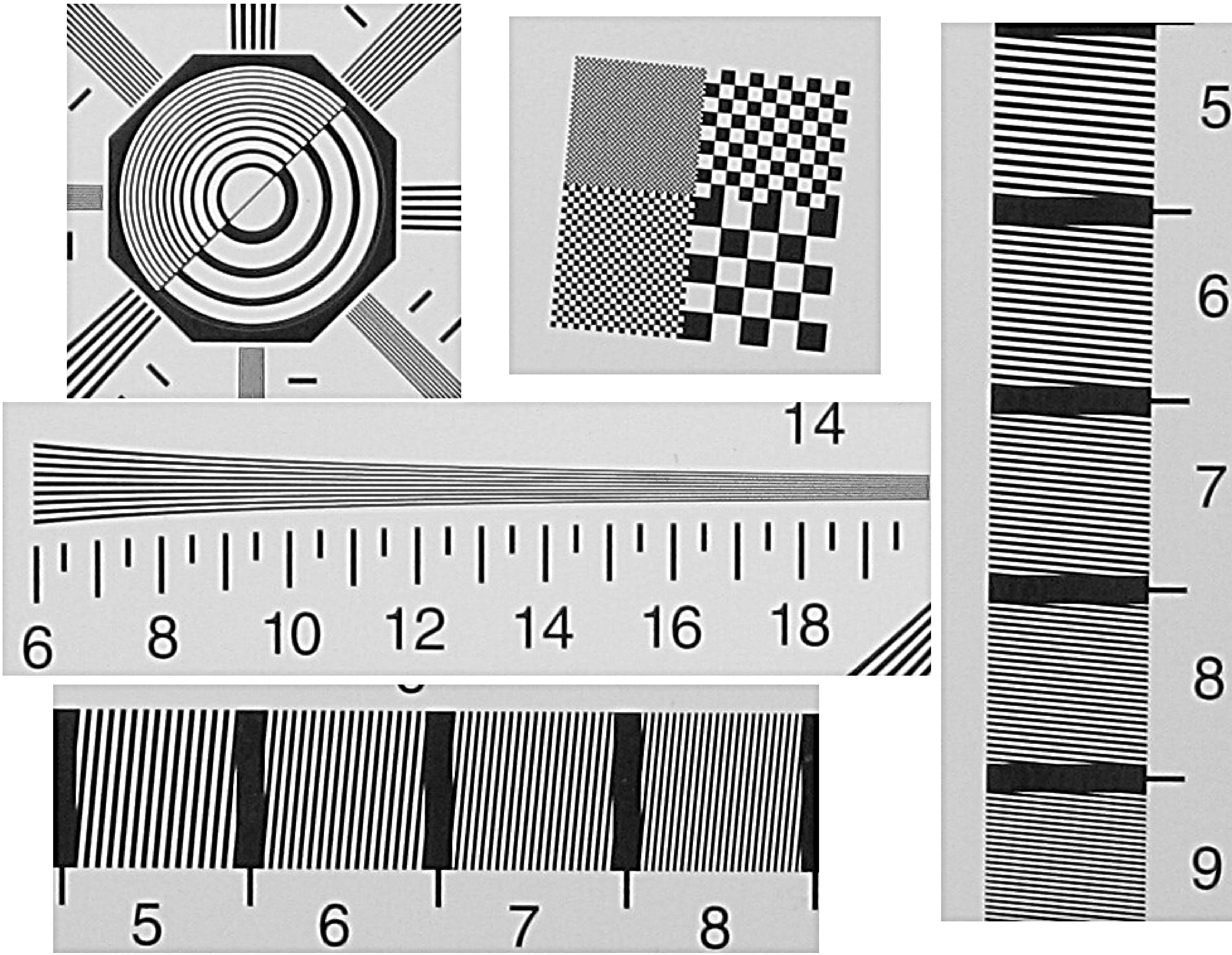

14
6 8 10 12 14 16 18
5 6 7 8
5 6 7 8 9


14
6
8
10
12
14
16
18
5
6
7
8
5
6
7
8
9

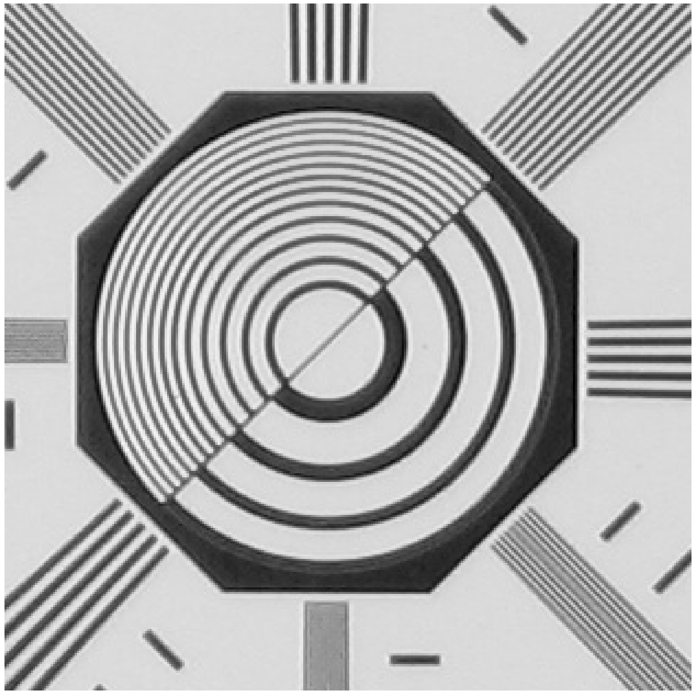


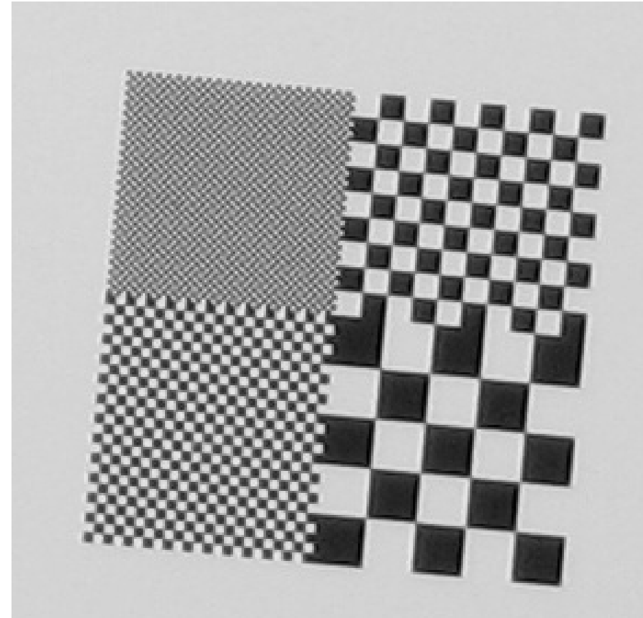
14
6 8 10 12 14 16 18
5 6 7 8
5 6 7 8 9

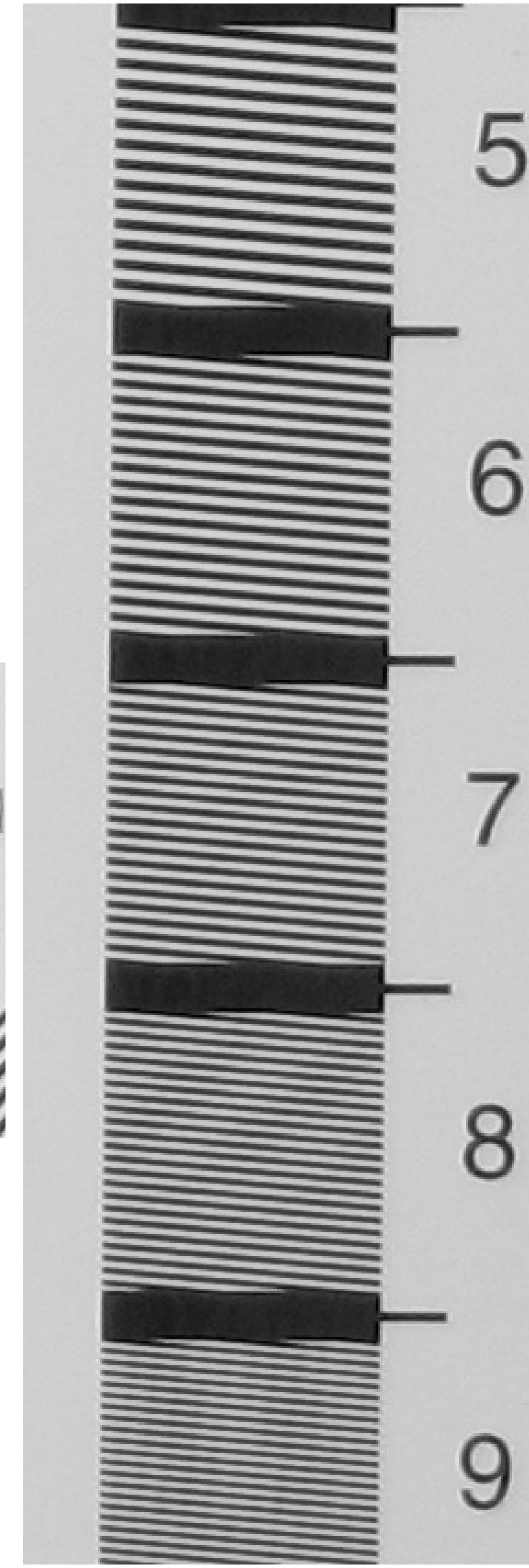

14
6 8 10 12 14 16 18
5 6 7 8
5 6 7 8 9

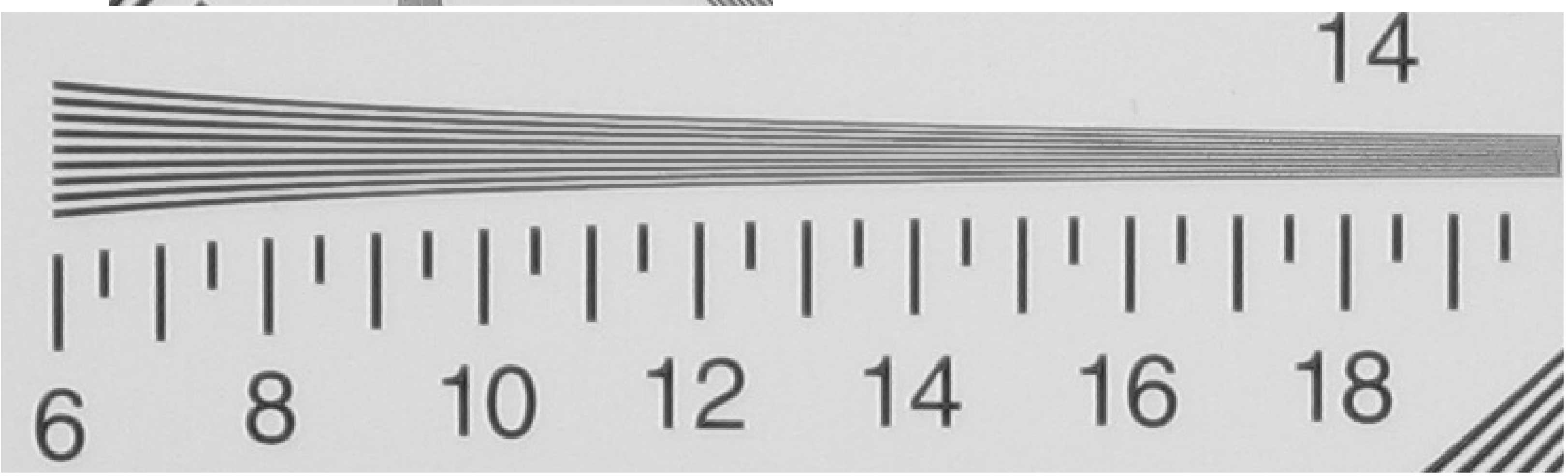


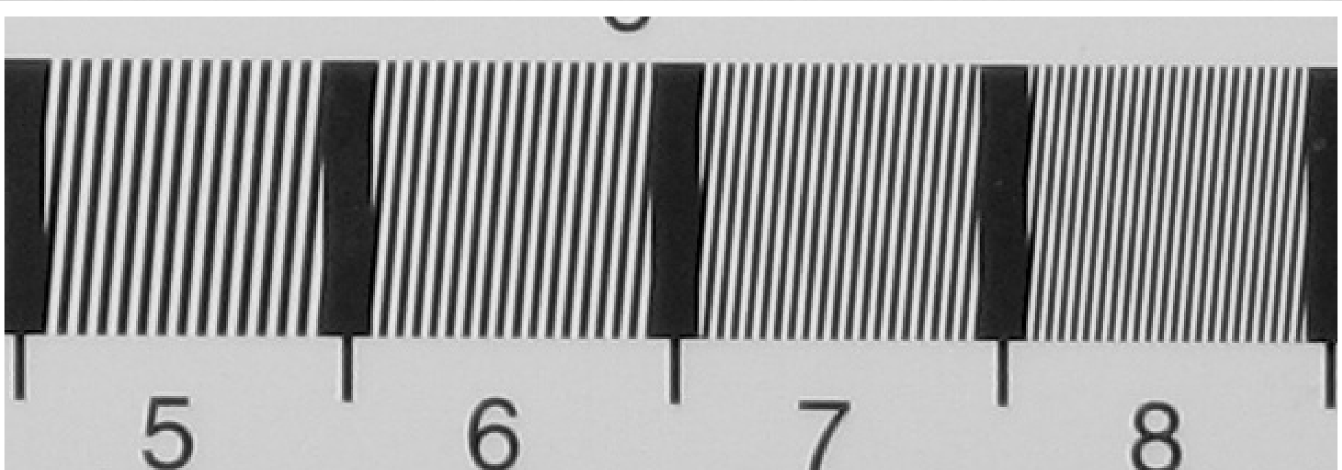
14
6 8 10 12 14 16 18
5 6 7 8
5 6 7 8 9

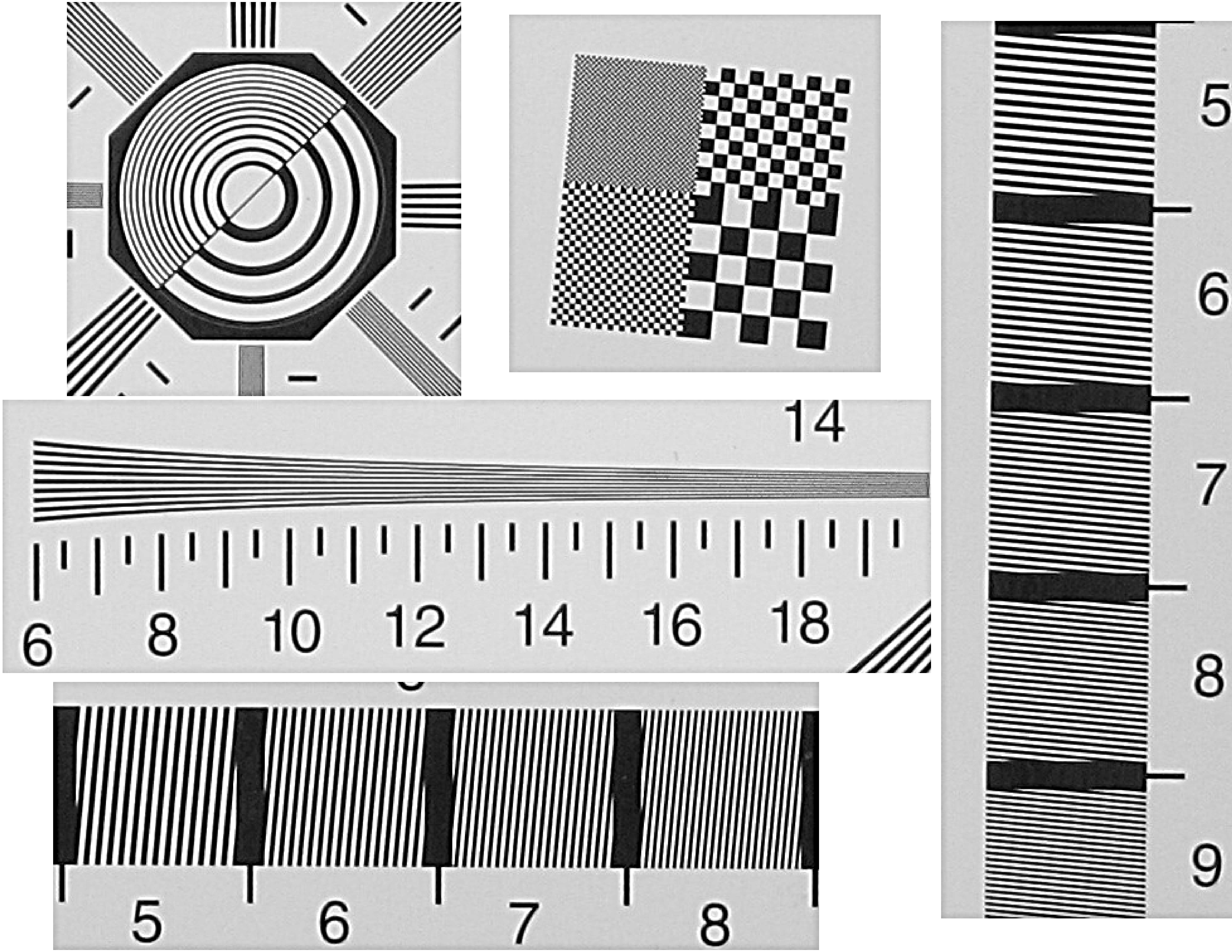


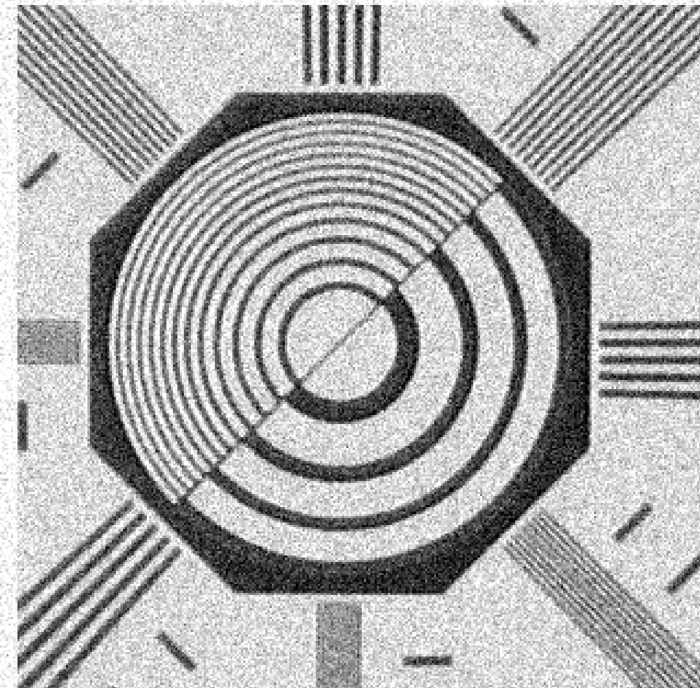


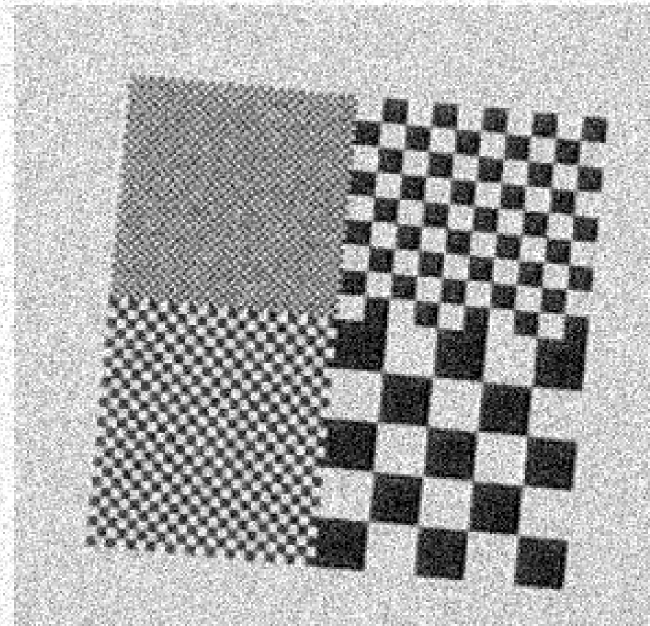

5
6
7
8
9

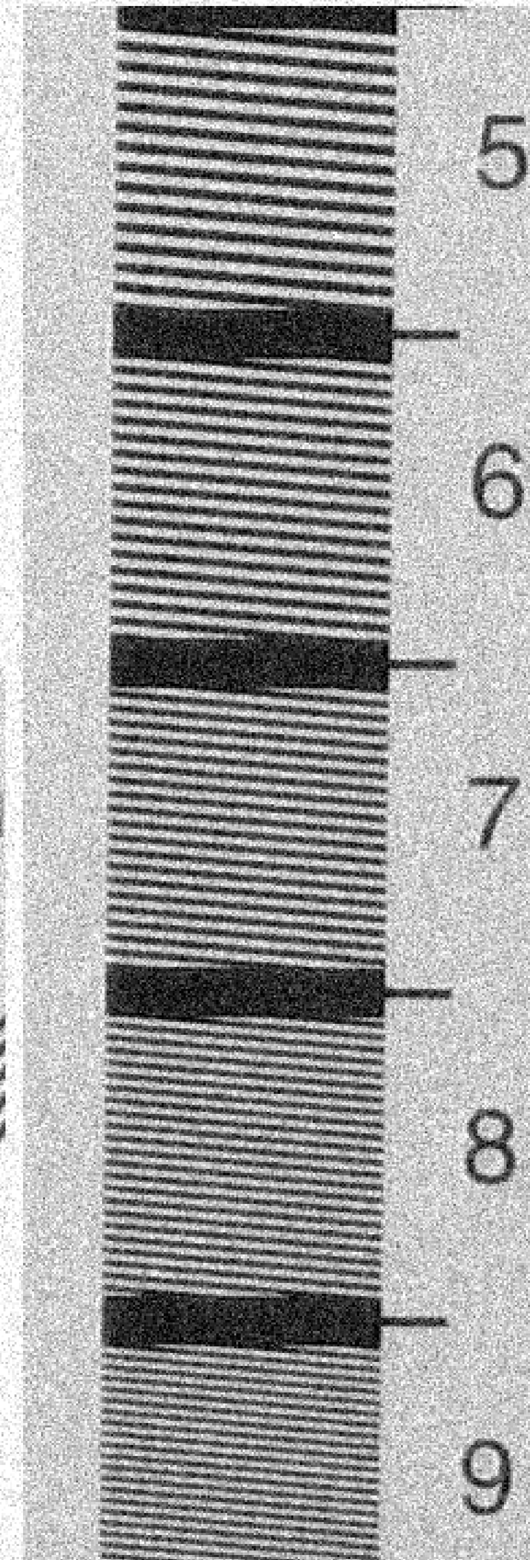

14
6
8
10
12
14
16
18

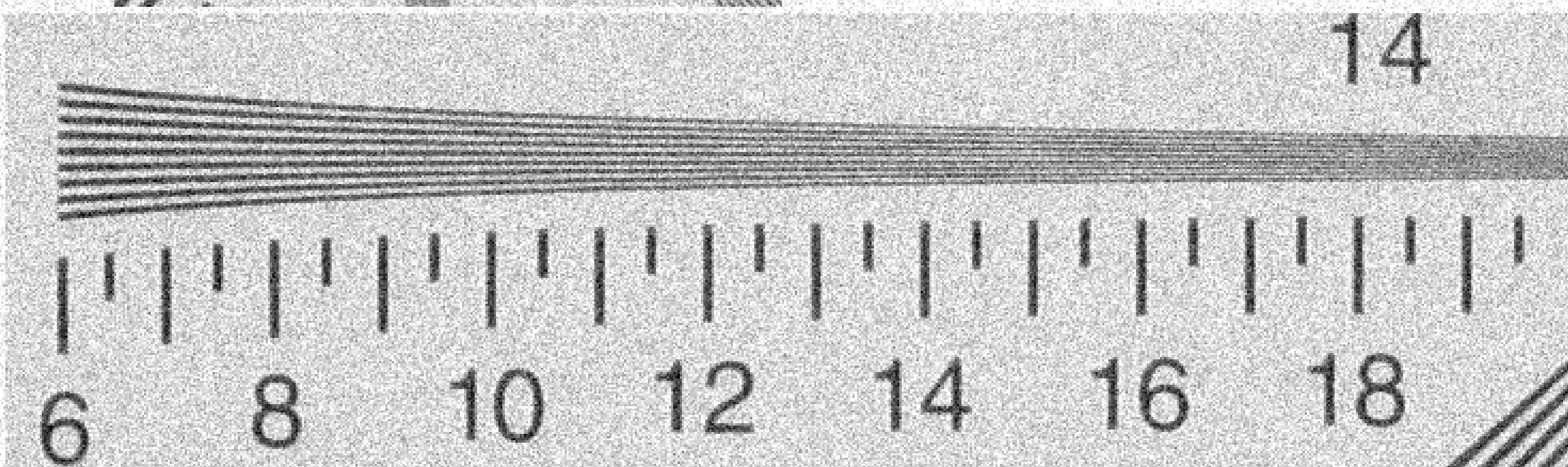

5
6
7
8

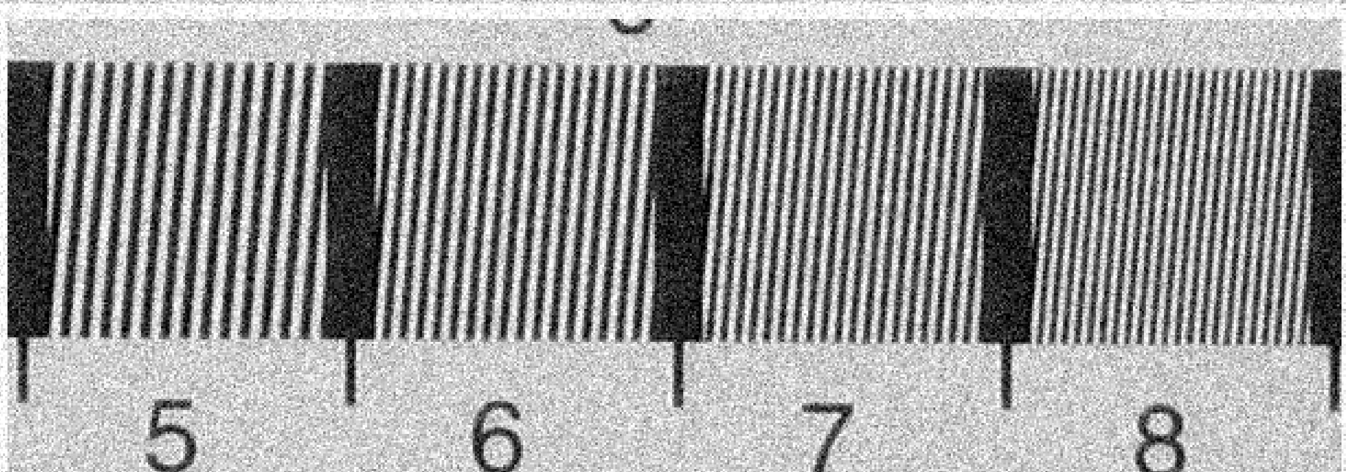

14
6 8 10 12 14 16 18
5 6 7 8
5 6 7 8 9

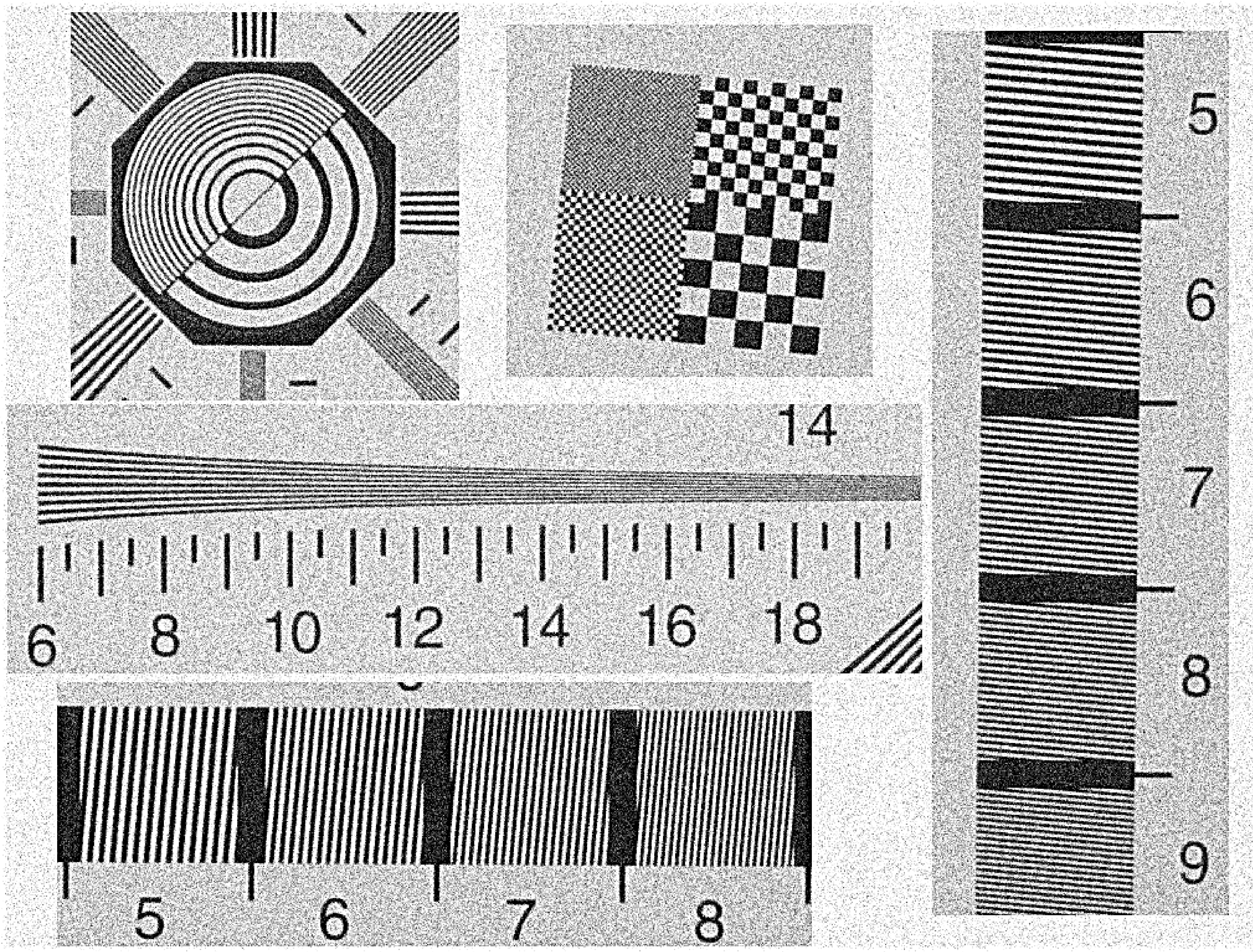


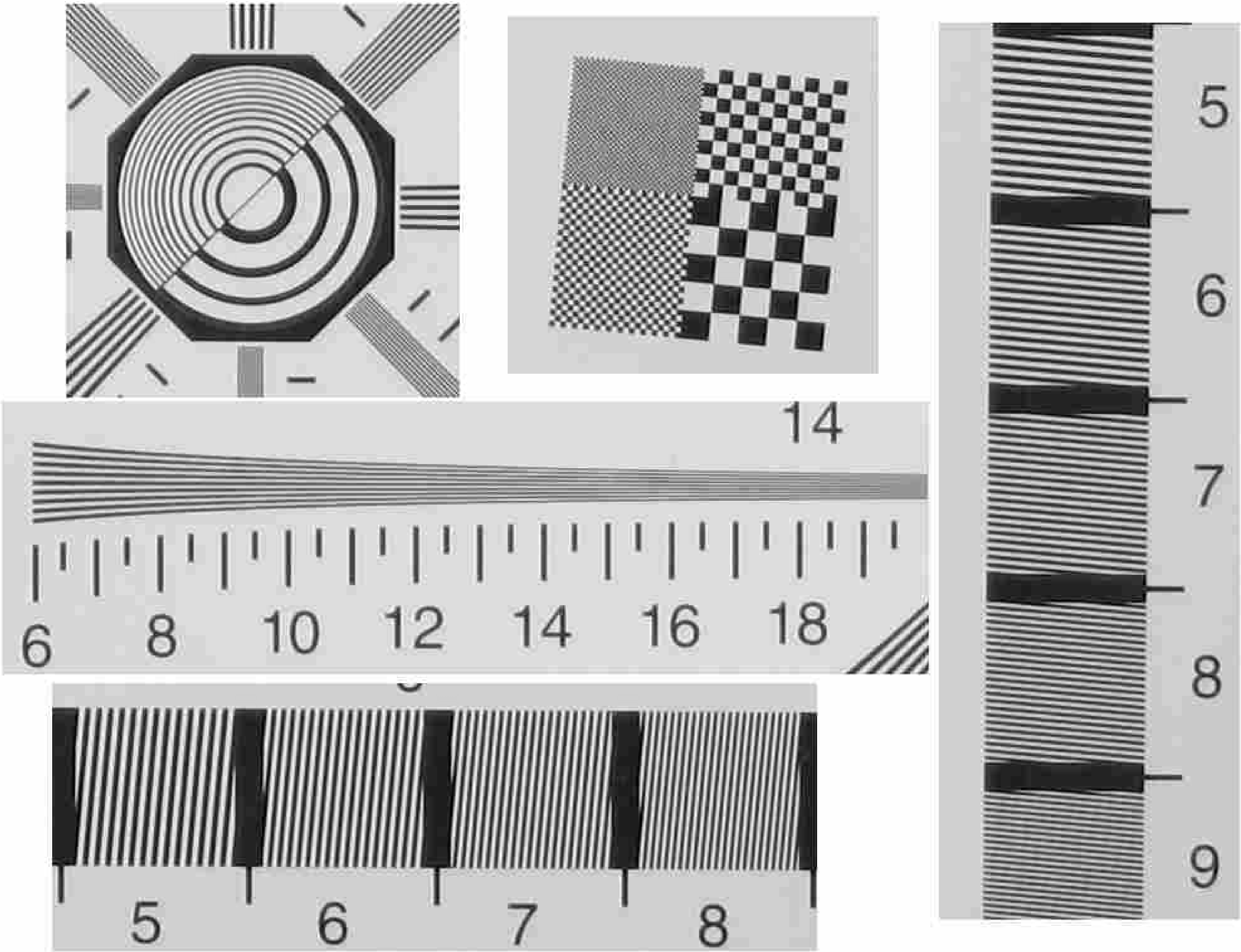


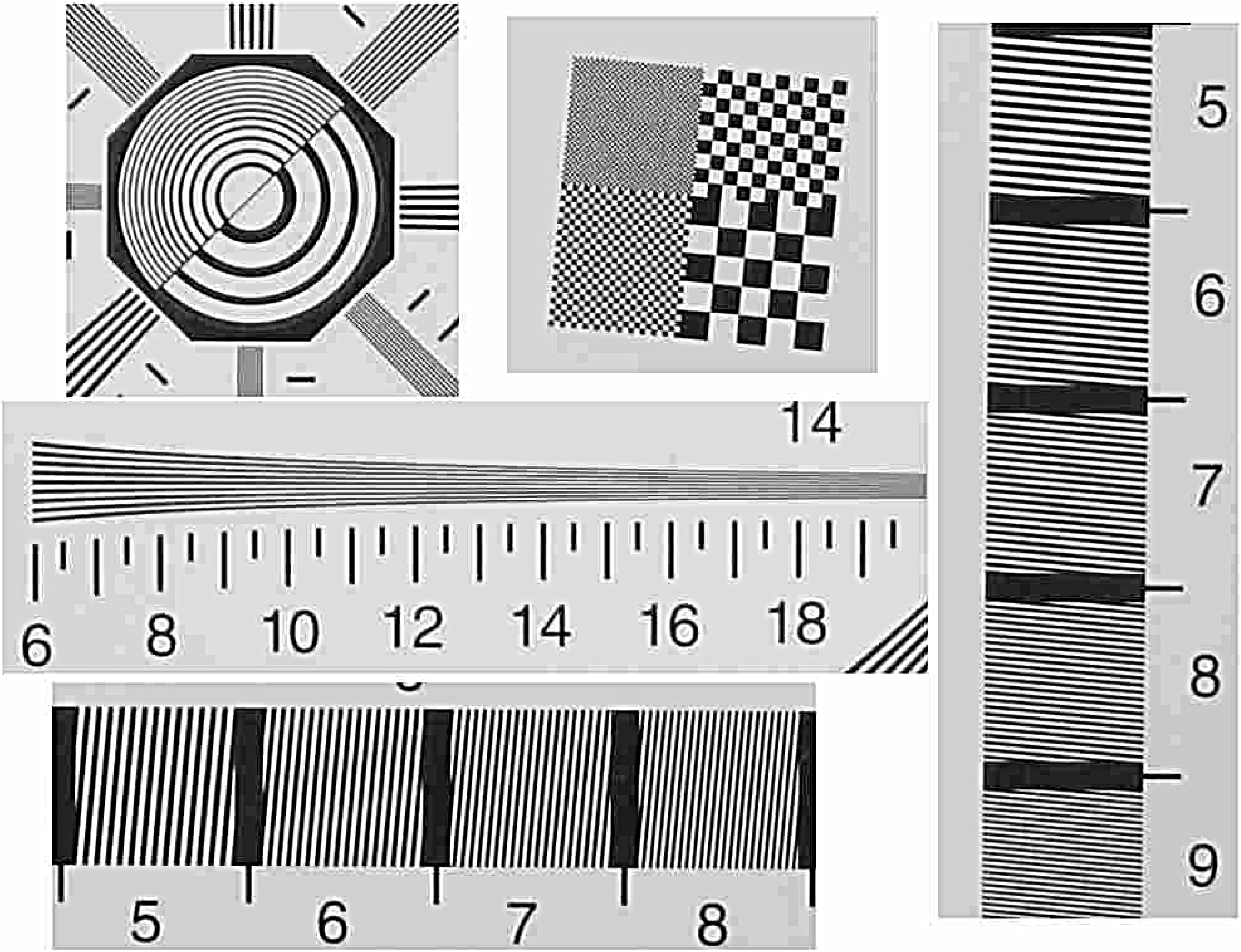

5
6
7
8
9

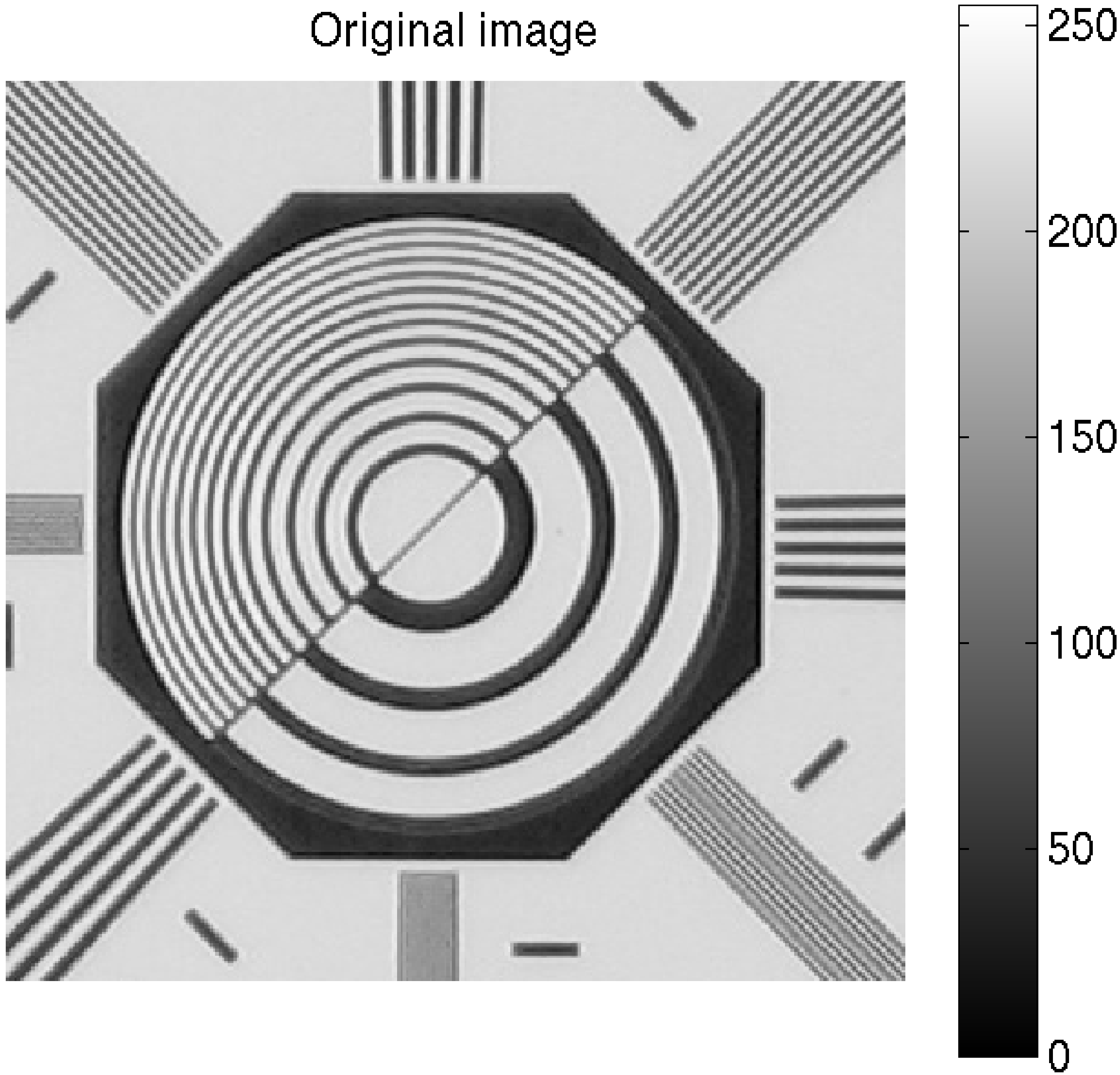

14
6
8
10
12
14
16
18

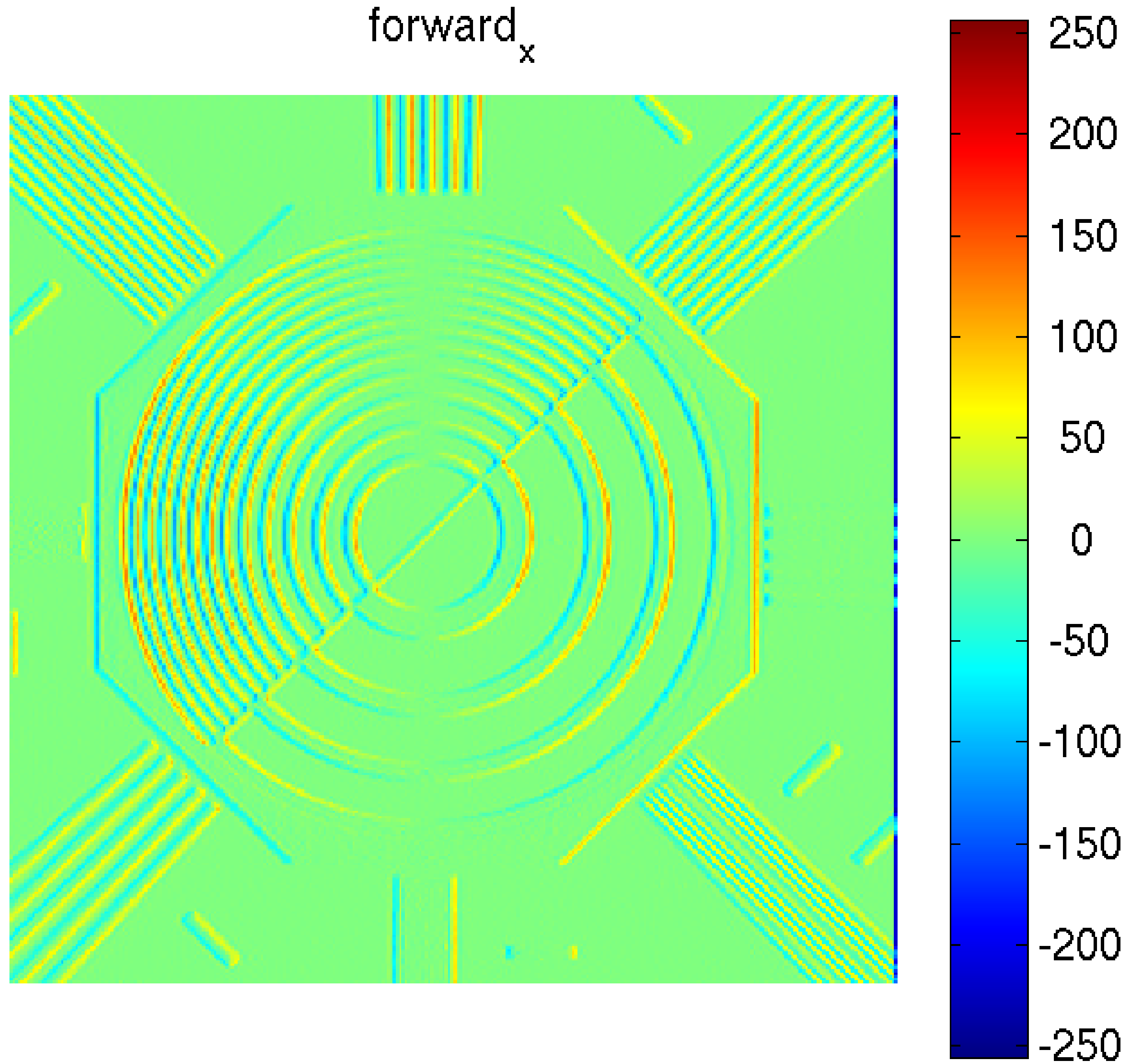

5
6
7
8

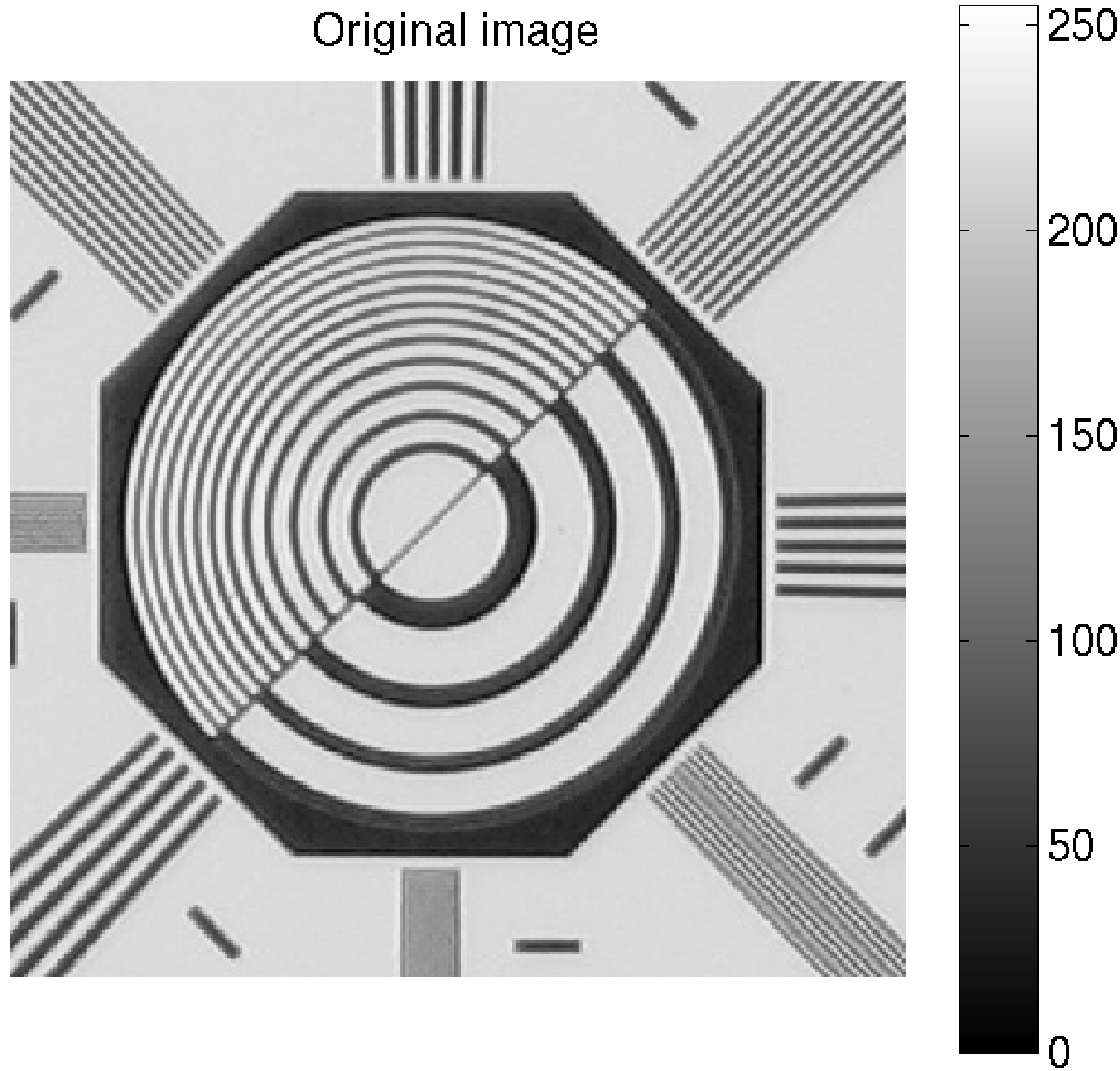

5
6
7
8
9
14
6
8
10
12
14
16
18
5
6
7
8

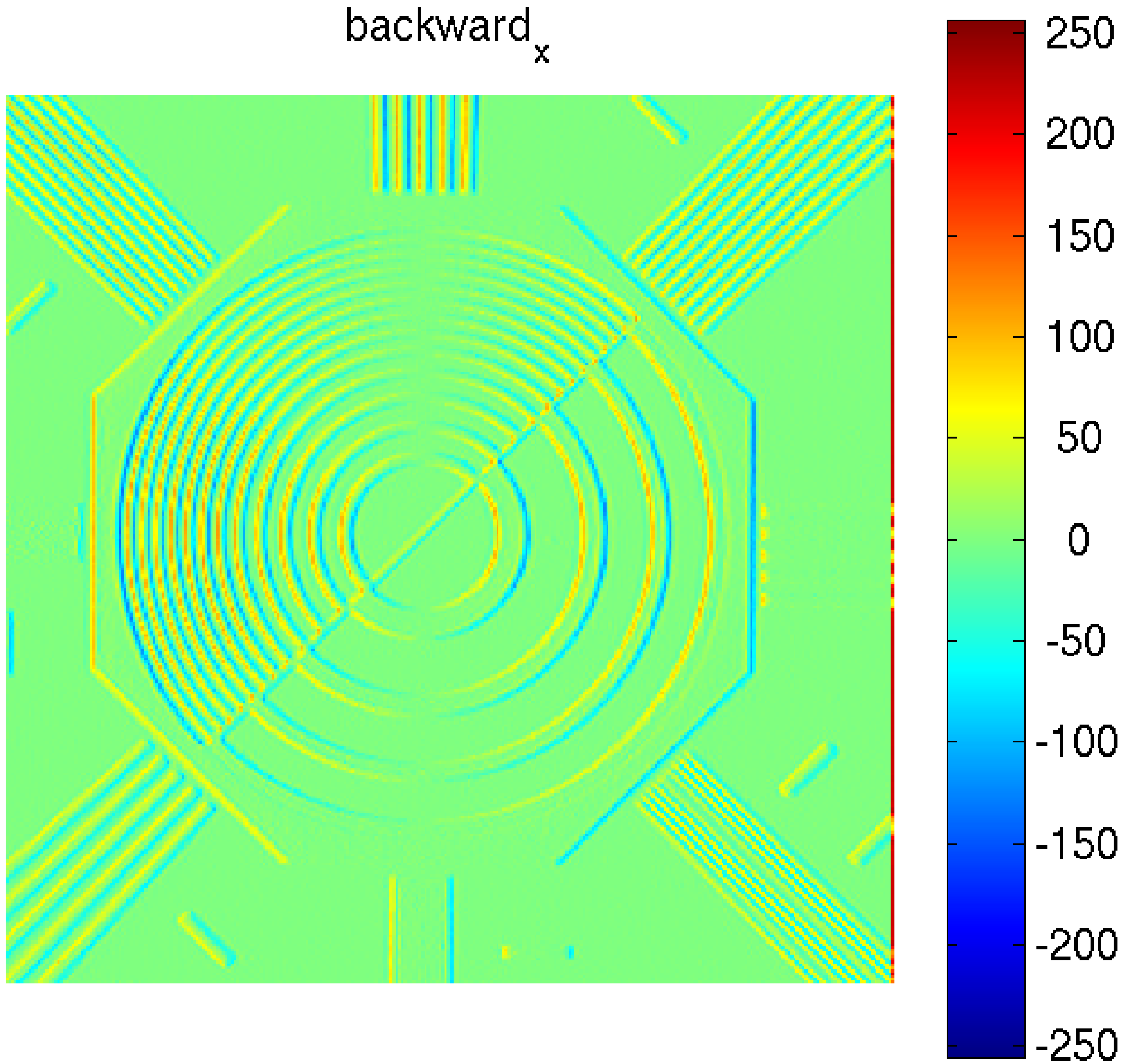

14
6
8
10
12
14
16
18
5
6
7
8
5
6
7
8
9


5
6
7
8
9
14
6
8
10
12
14
16
18
5
6
7
8

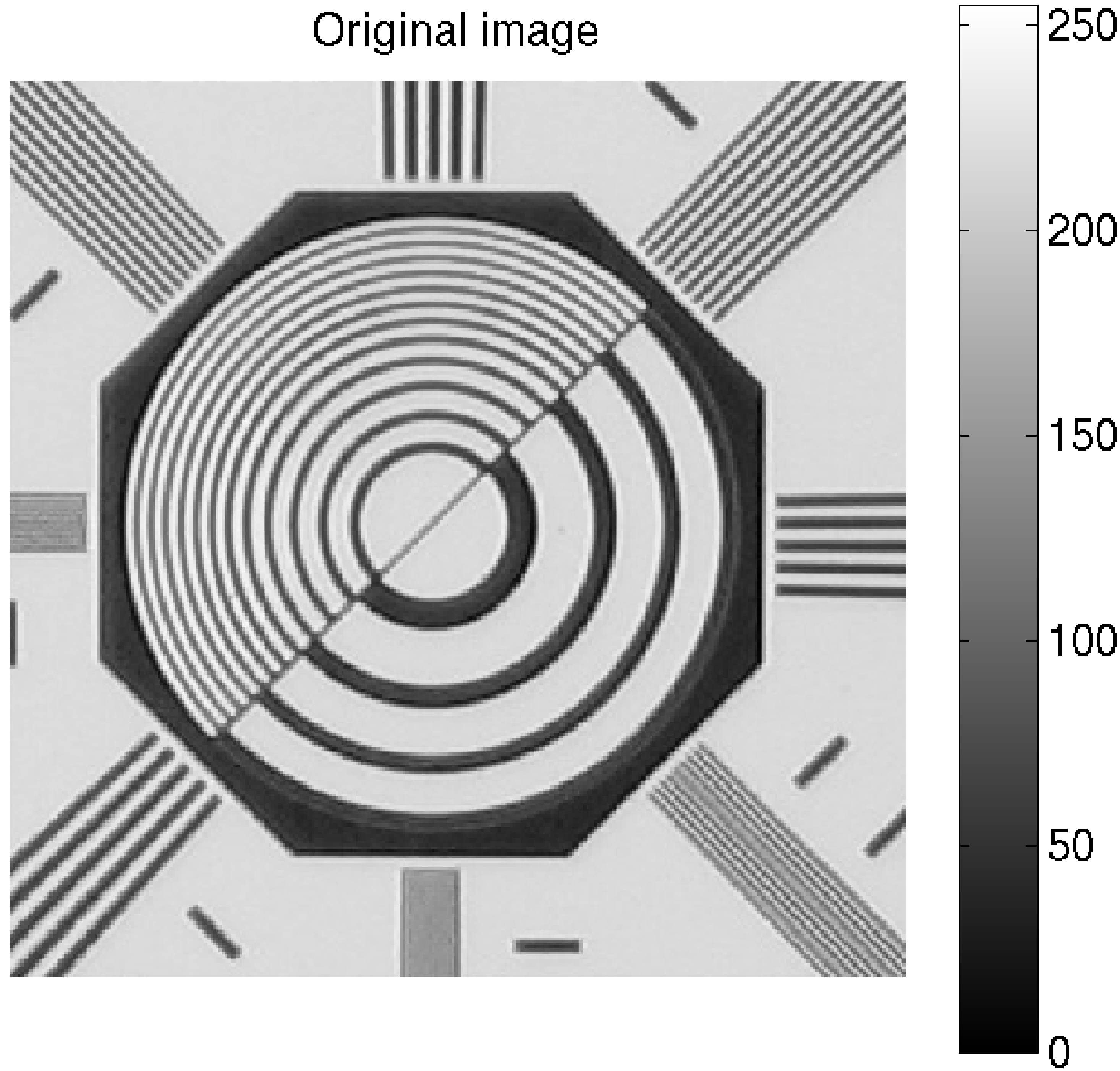

Original image
250
200
150
100
50
0

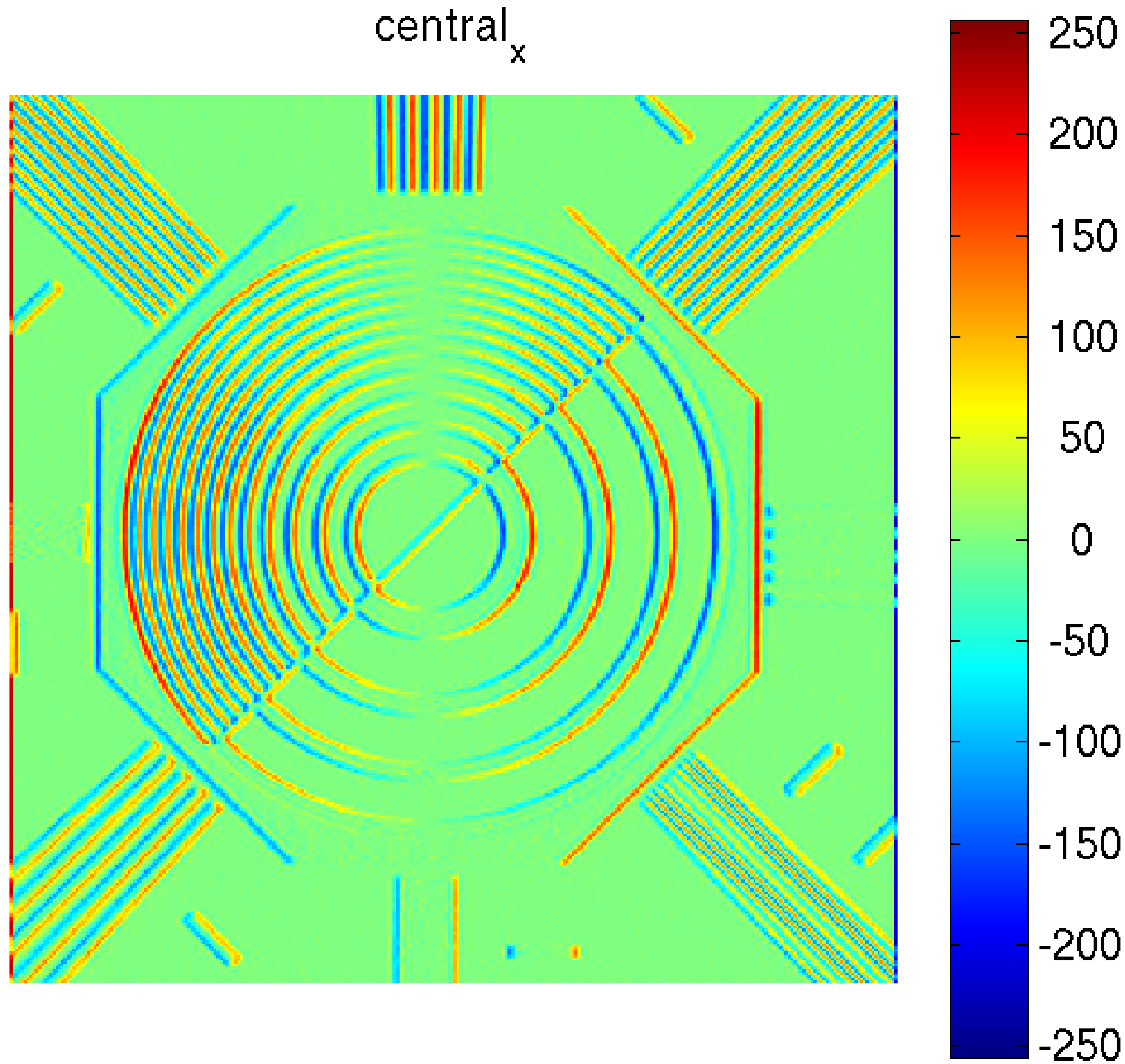

forward$_x$
250
200
150
100
50
0
-50
-100
-150
-200
-250


Original image
250
200
150
100
50
0

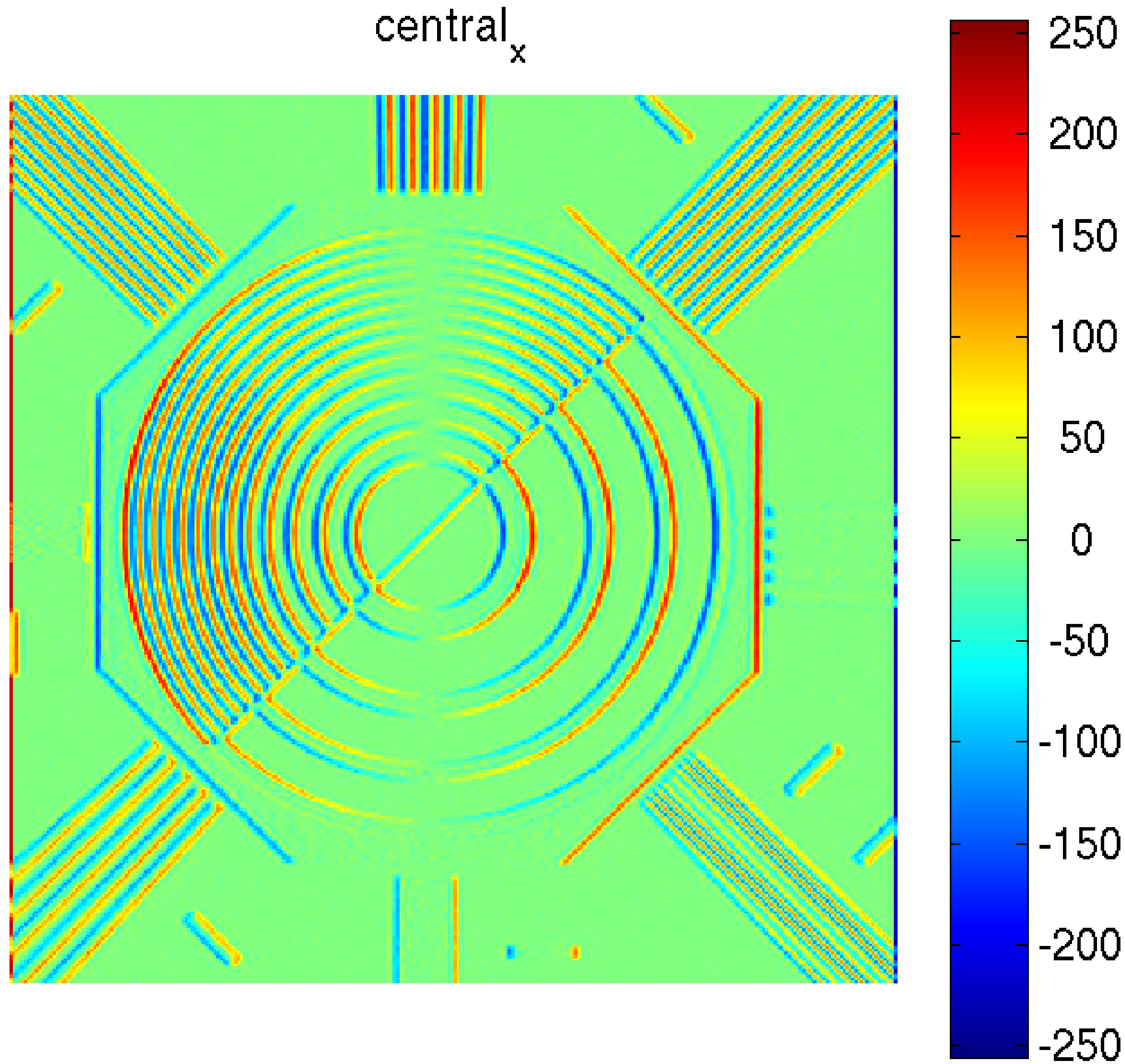


backward$_x$
250
200
150
100
50
0
-50
-100
-150
-200
-250

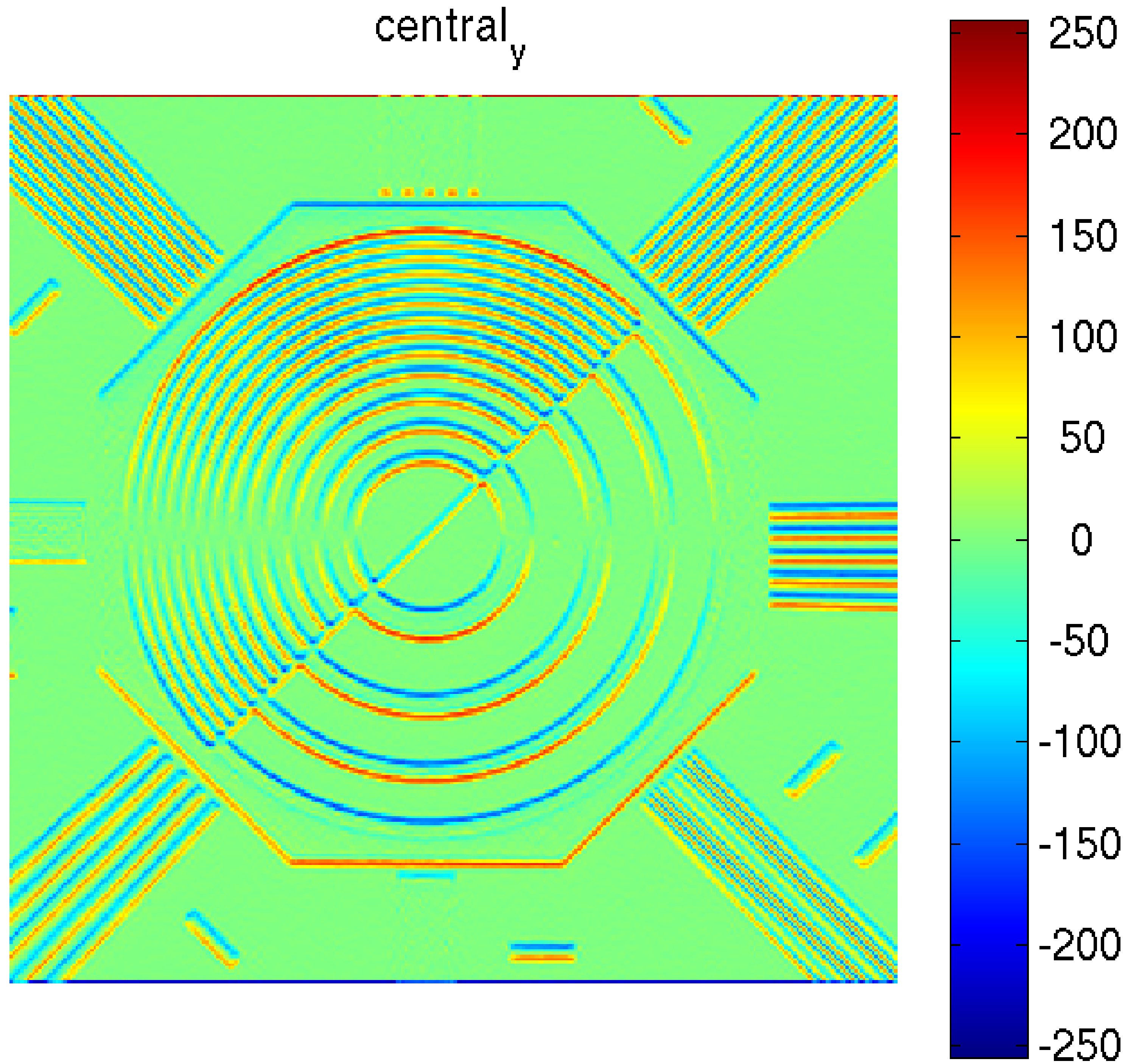

Original image
250
200
150
100
50
0

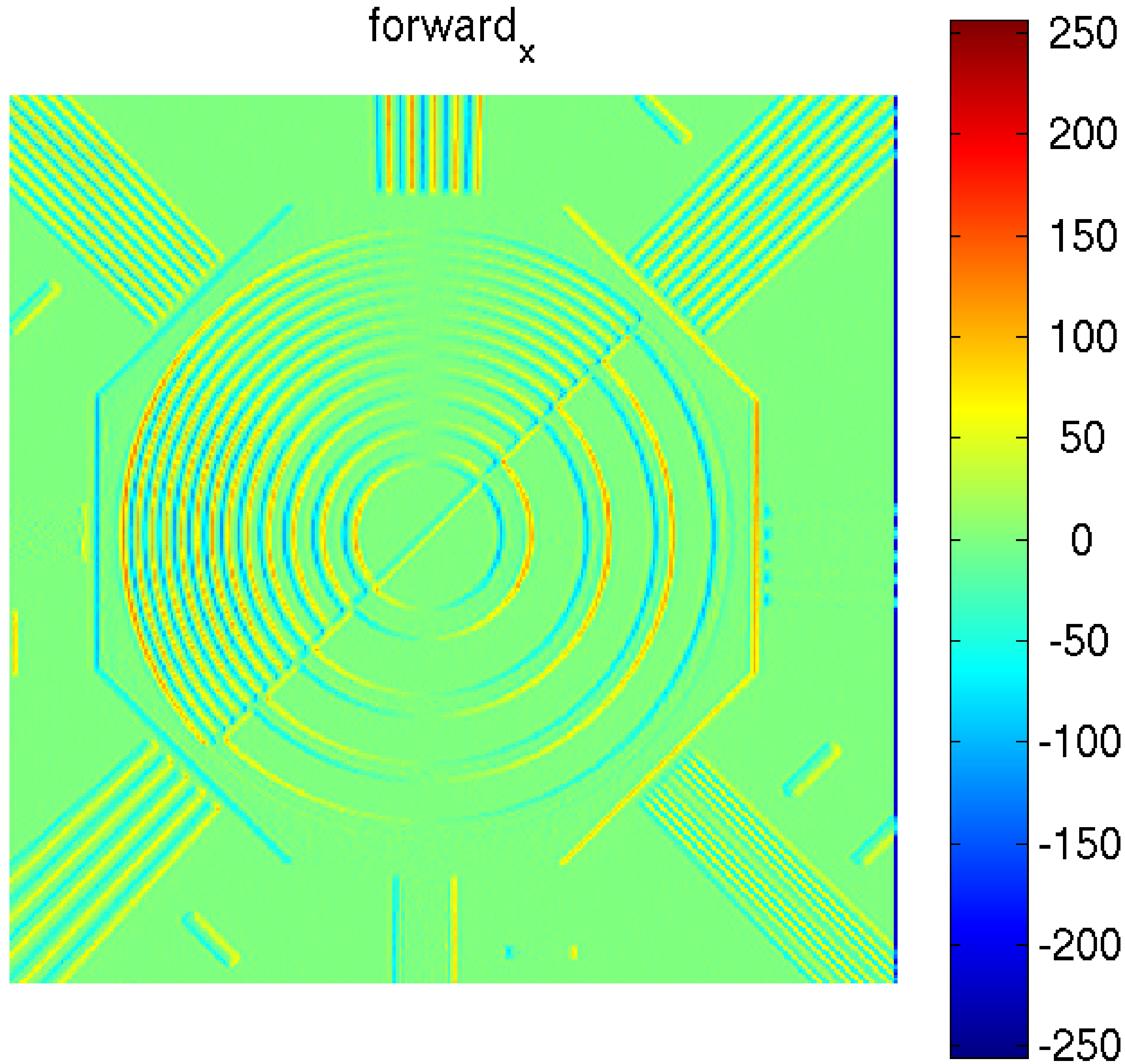


central$_x$
250
200
150
100
50
0
-50
-100
-150
-200
-250

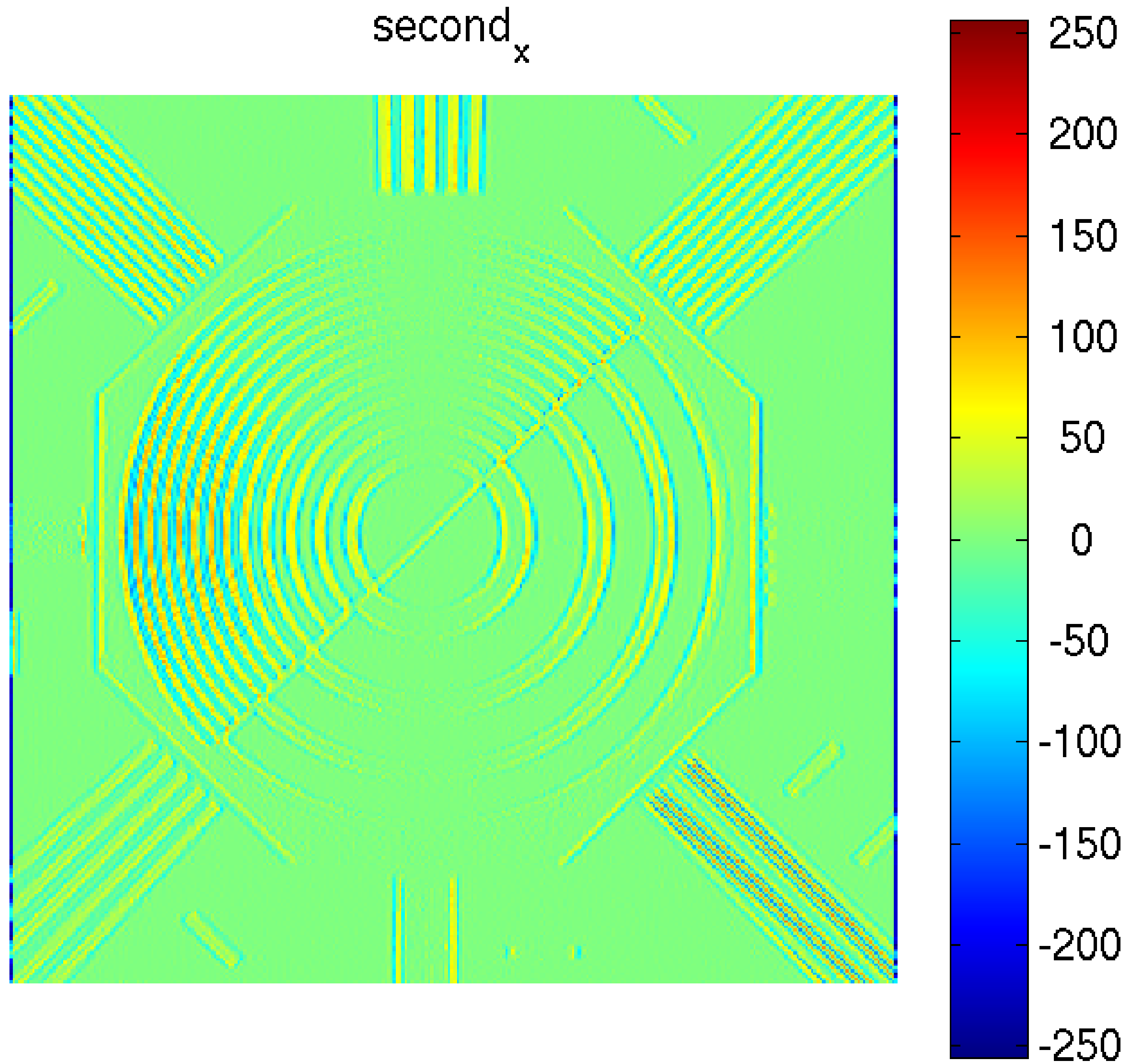


central$_x$
250
200
150
100
50
0
-50
-100
-150
-200
-250

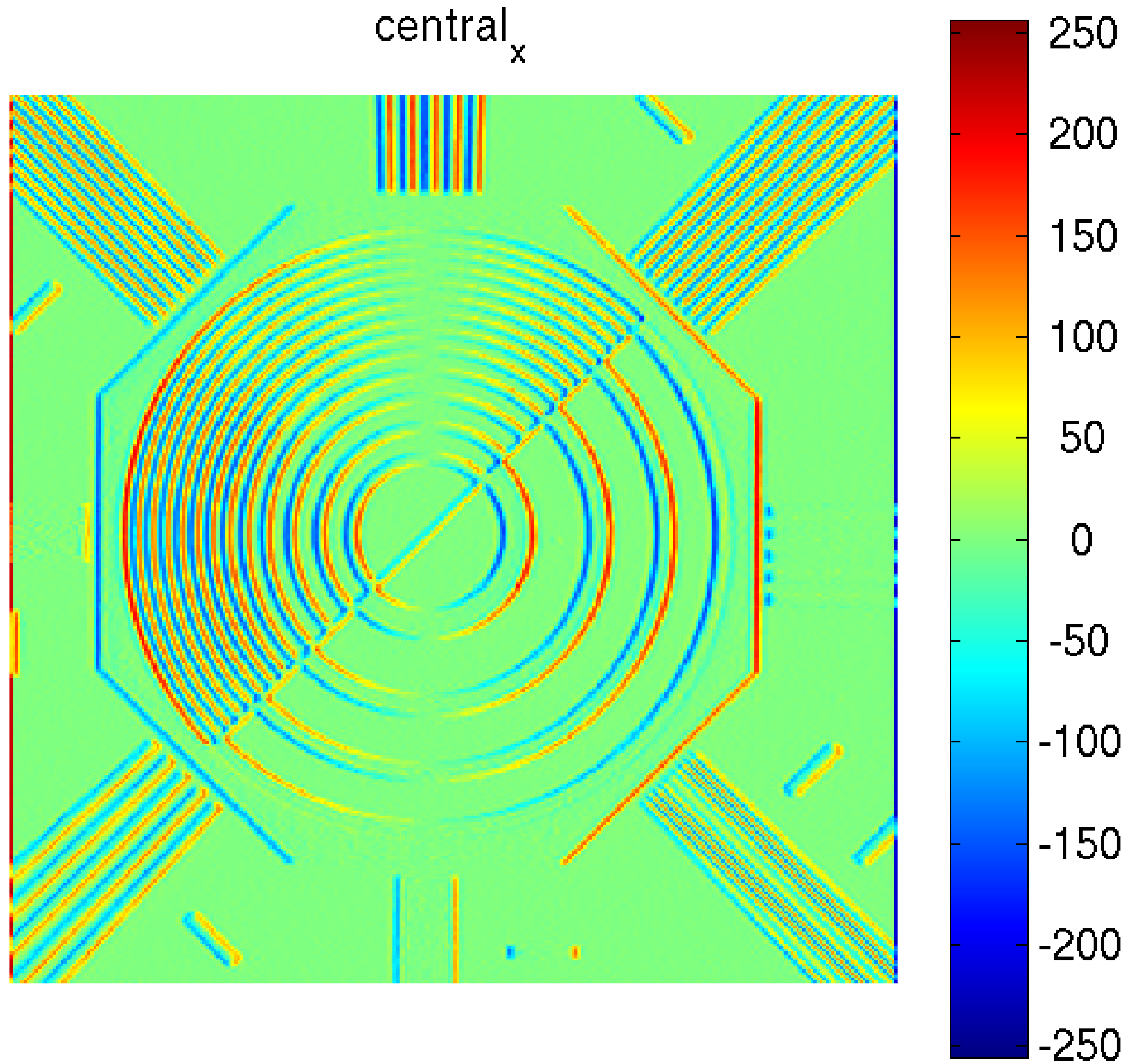

central$_y$
250
200
150
100
50
0
-50
-100
-150
-200
-250

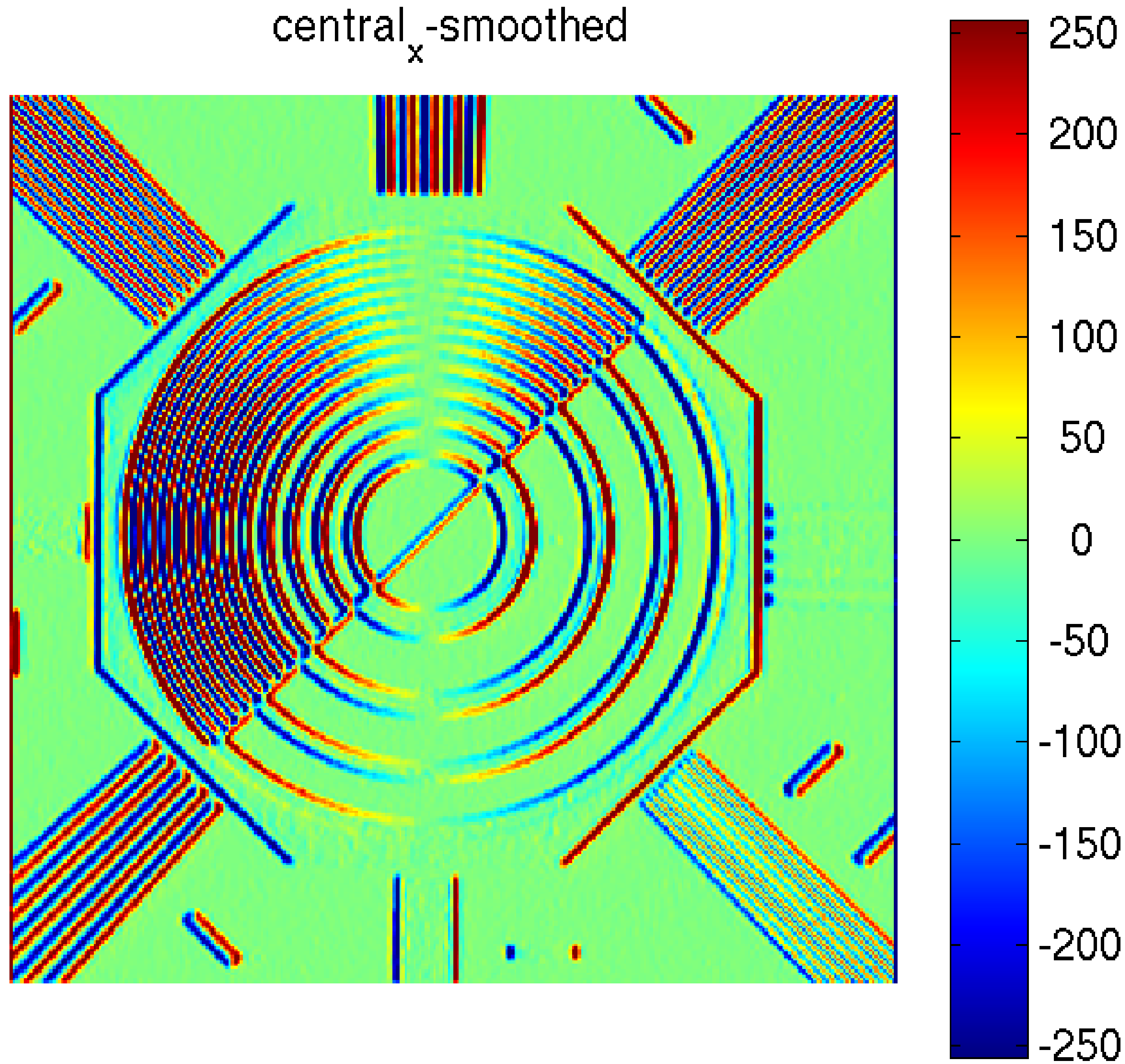


forward$_x$
250
200
150
100
50
0
-50
-100
-150
-200
-250



second$_x$
250
200
150
100
50
0
-50
-100
-150
-200
-250


$central_x$
250
200
150
100
50
0
-50
-100
-150
-200
-250


central$_x$-smoothed
250
200
150
100
50
0
-50
-100
-150
-200
-250

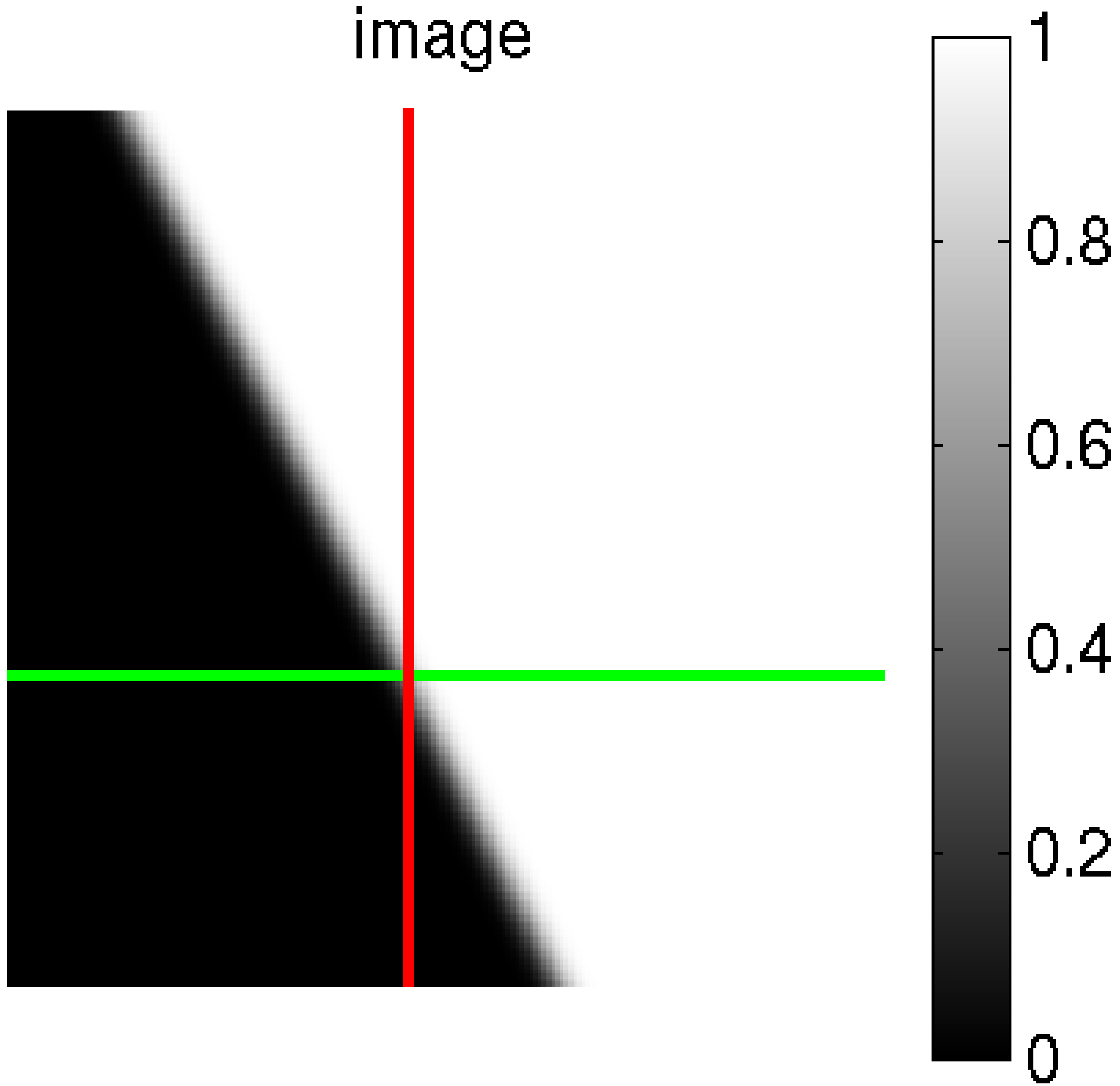
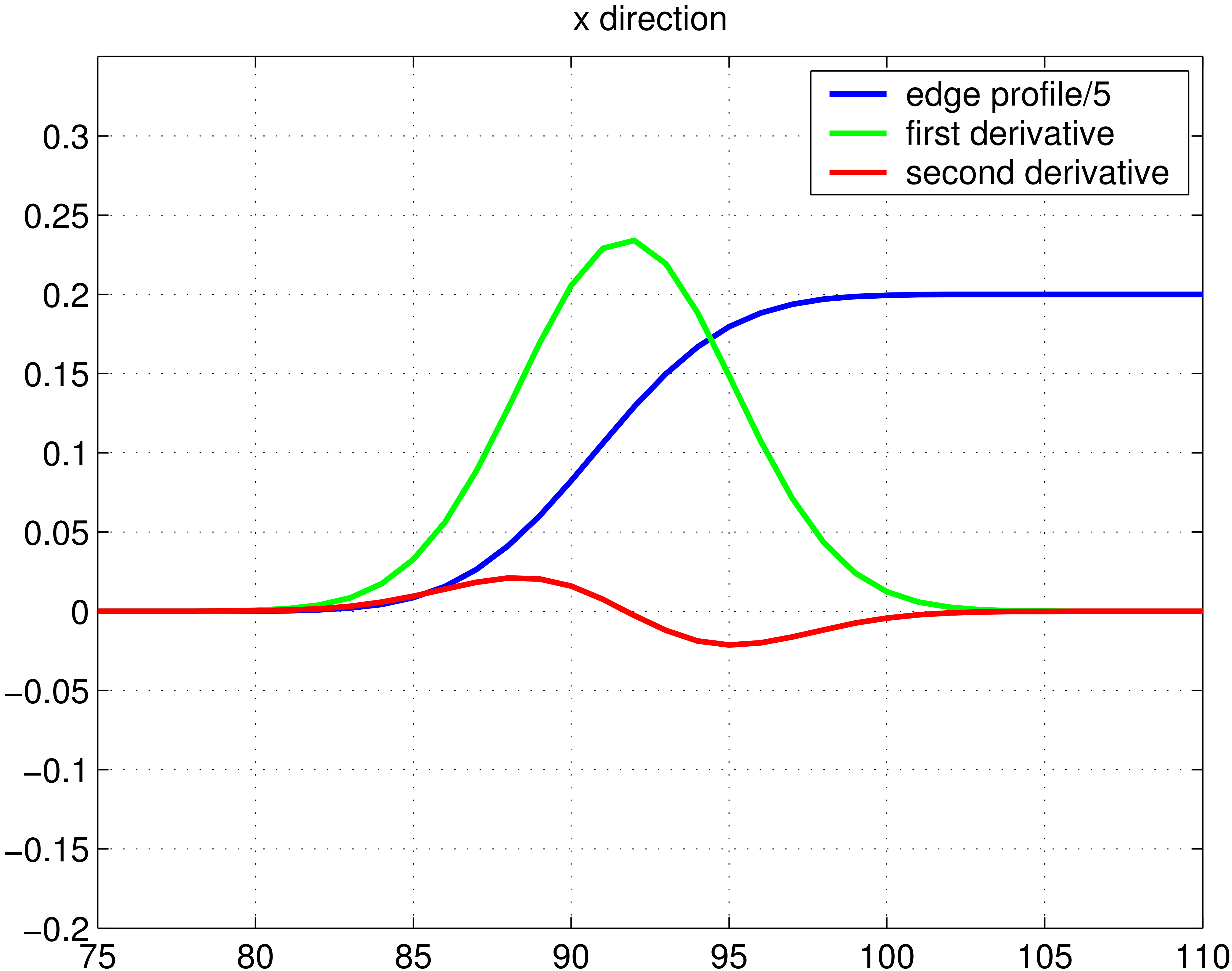
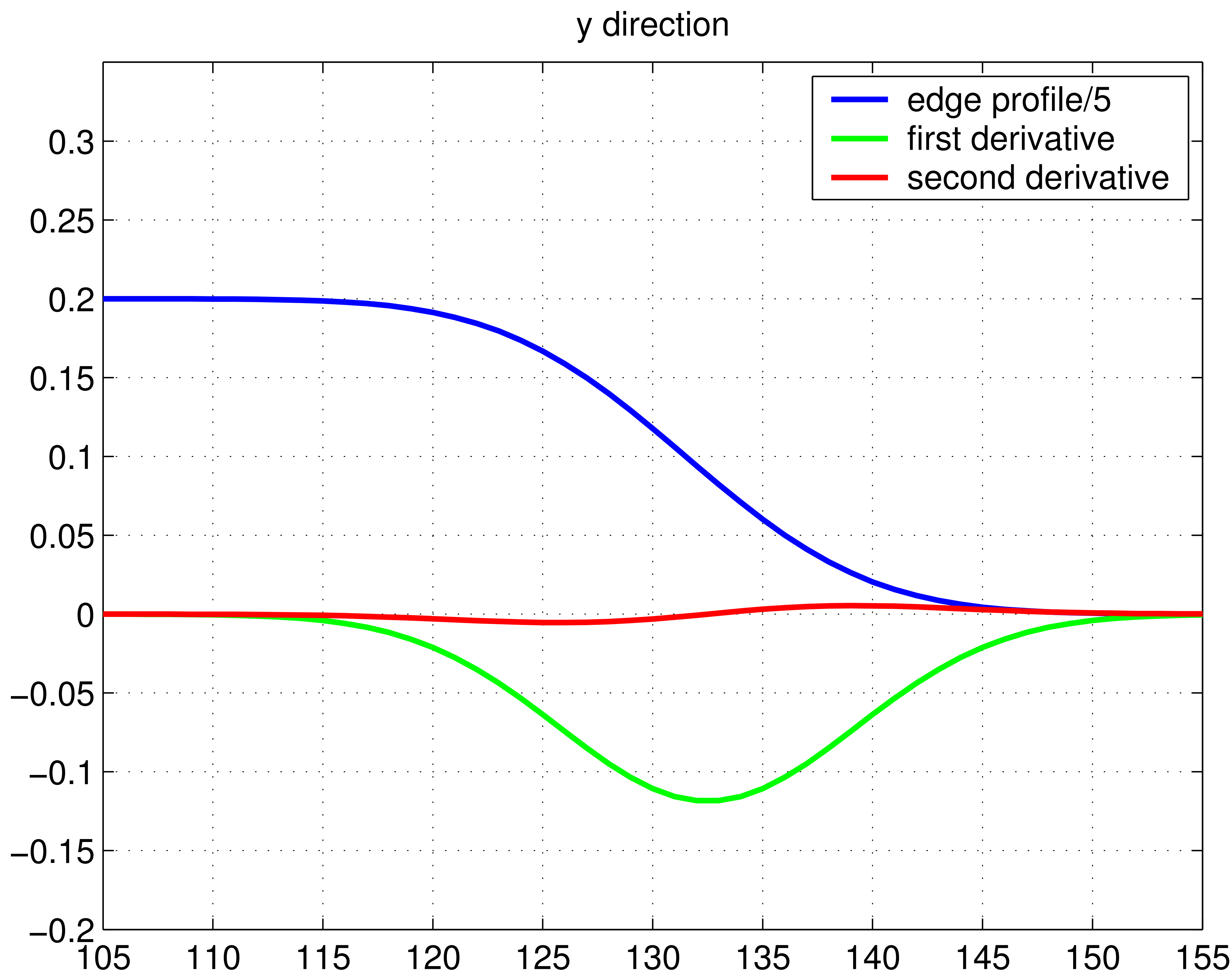
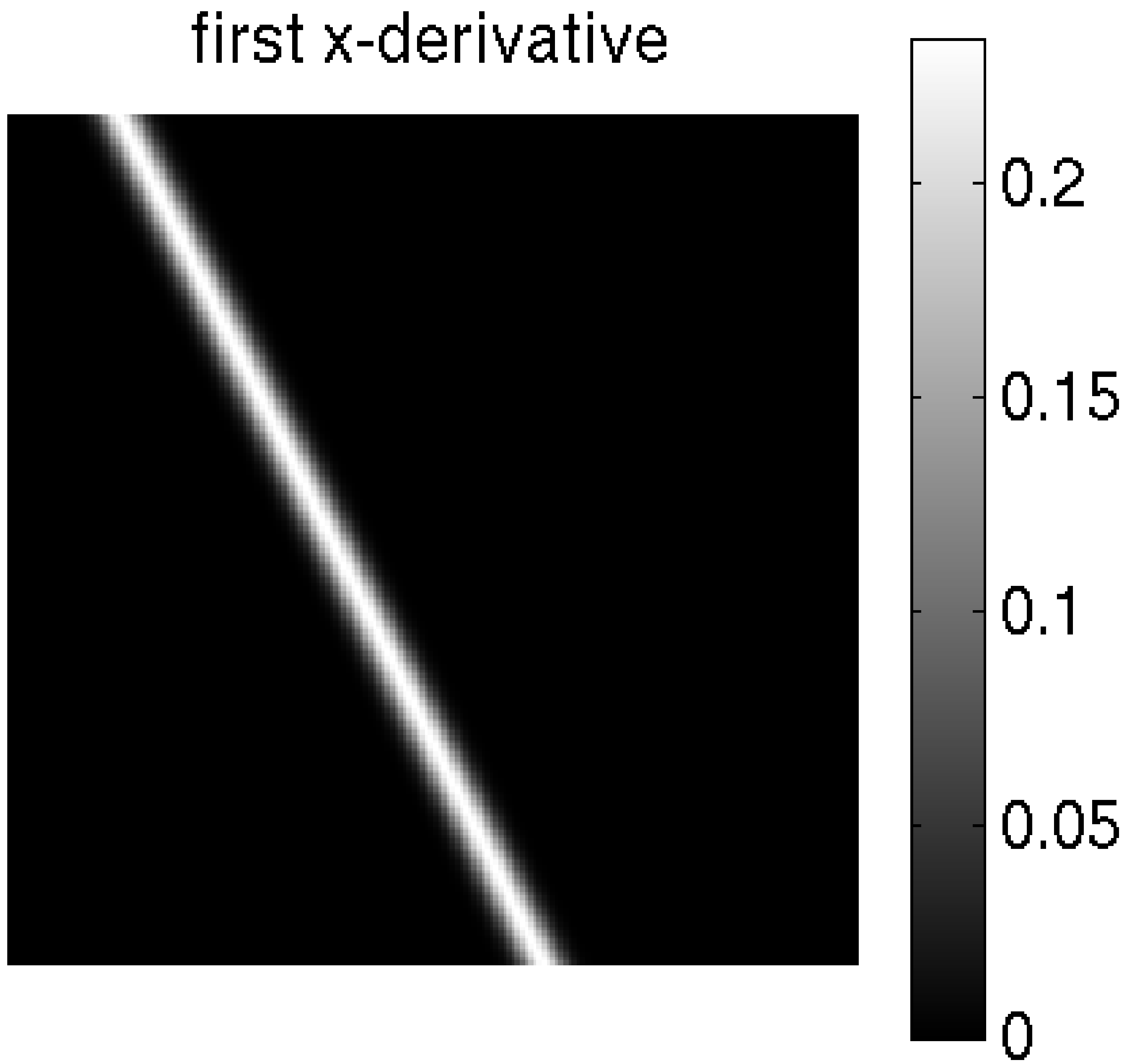


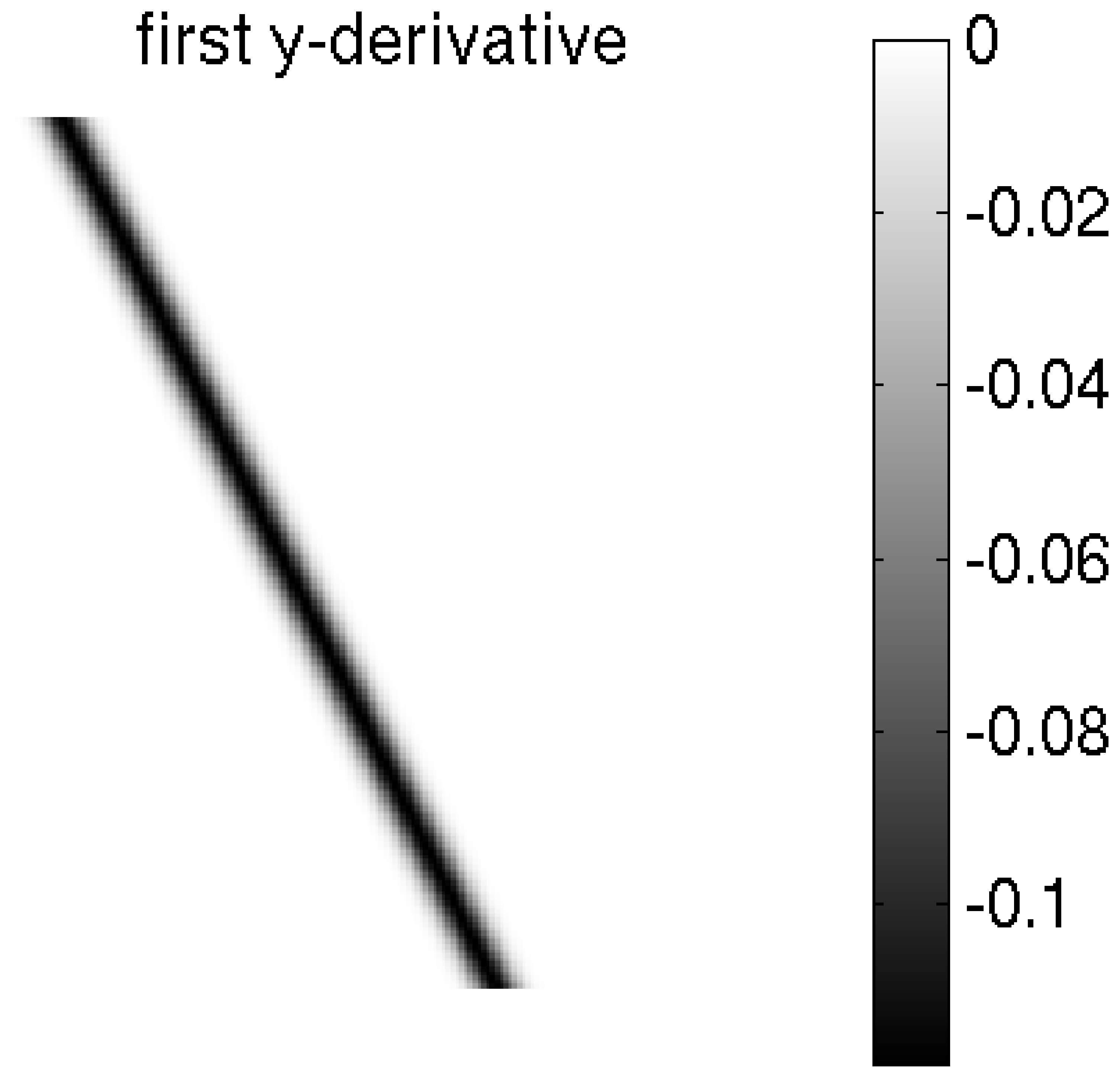
image
1
0.8
0.6
0.4
0.2
0


x direction
edge profile/5
first derivative
second derivative
0.3
0.25
0.2
0.15
0.1
0.05
0
−0.05
−0.1
−0.15
−0.2
75
80
85
90
95
100
105
110



y direction
edge profile/5
first derivative
second derivative
0.3
0.25
0.2
0.15
0.1
0.05
0
−0.05
−0.1
−0.15
−0.2
105
110
115
120
125
130
135
140
145
150
155


first x-derivative
0.2
0.15
0.1
0.05
0


first y-derivative
0
-0.02
-0.04
-0.06
-0.08
-0.1

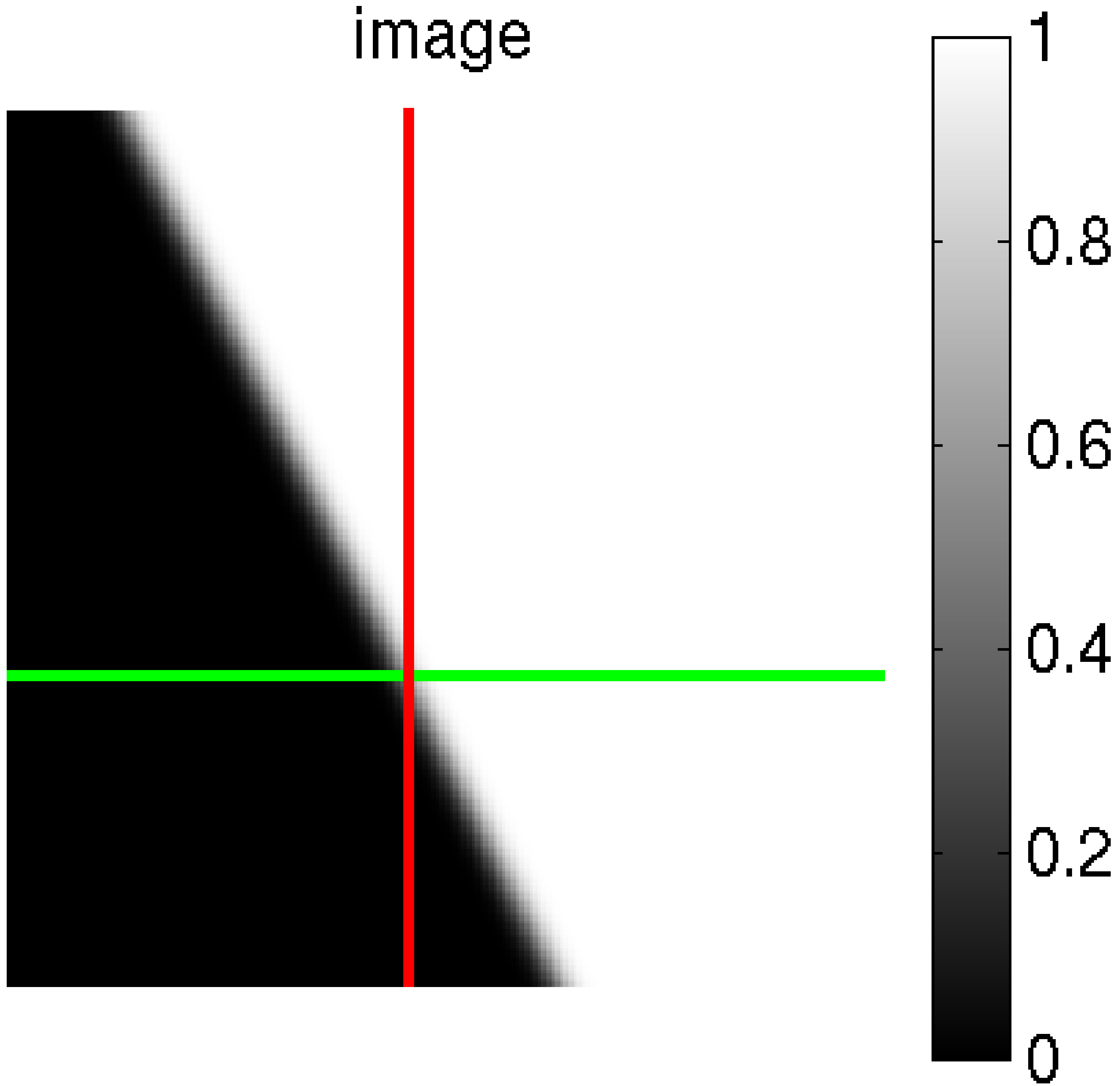
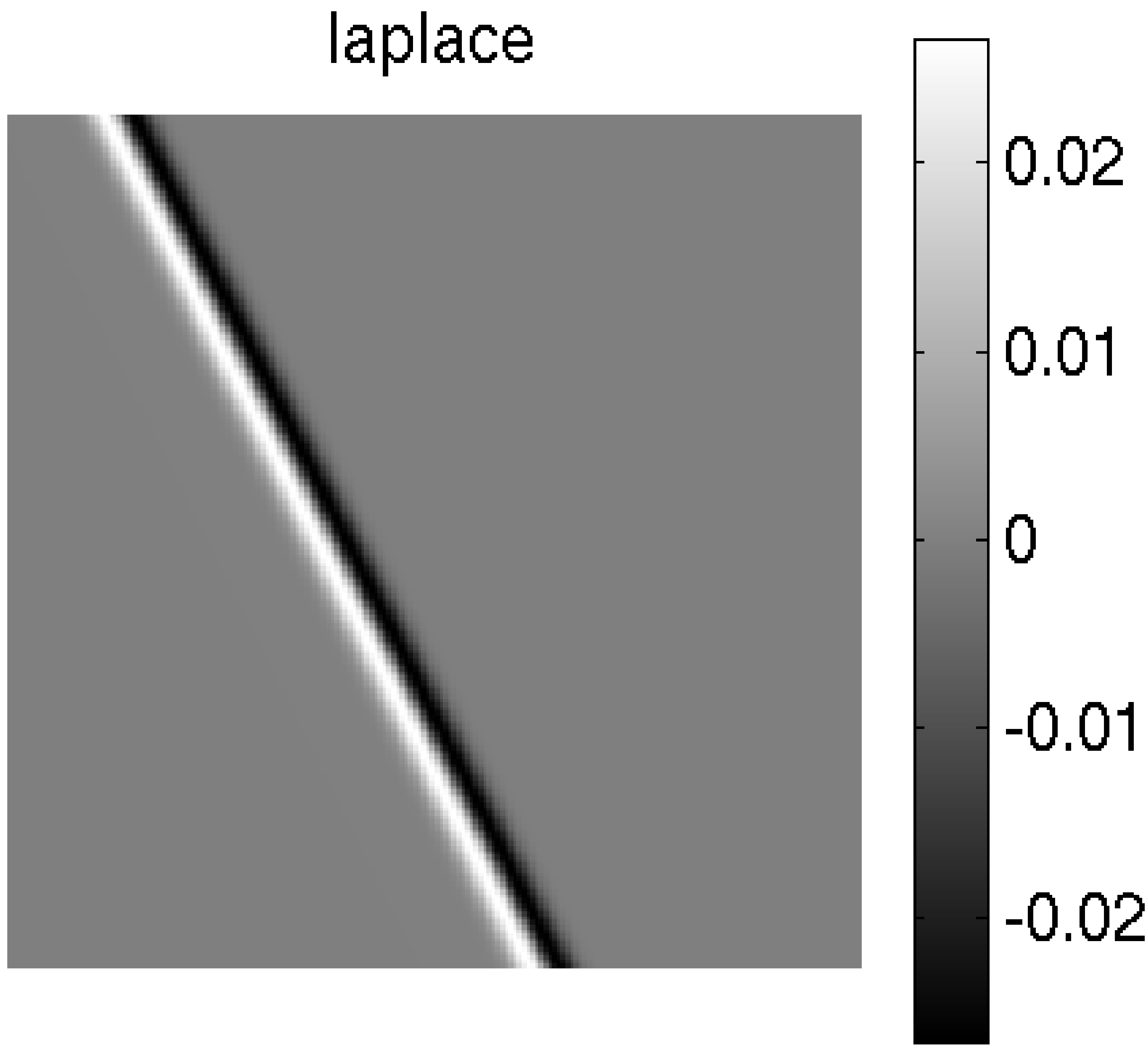
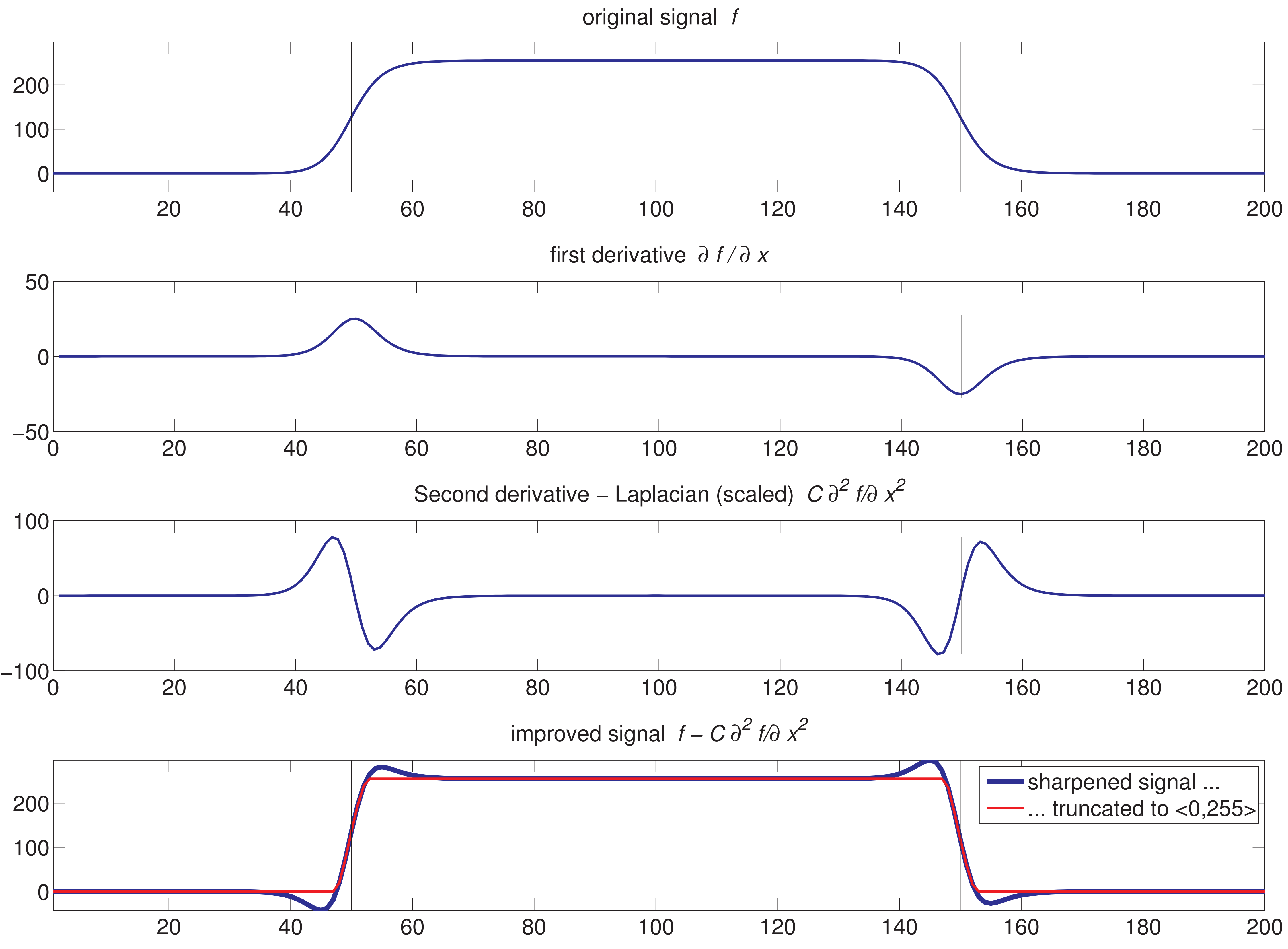


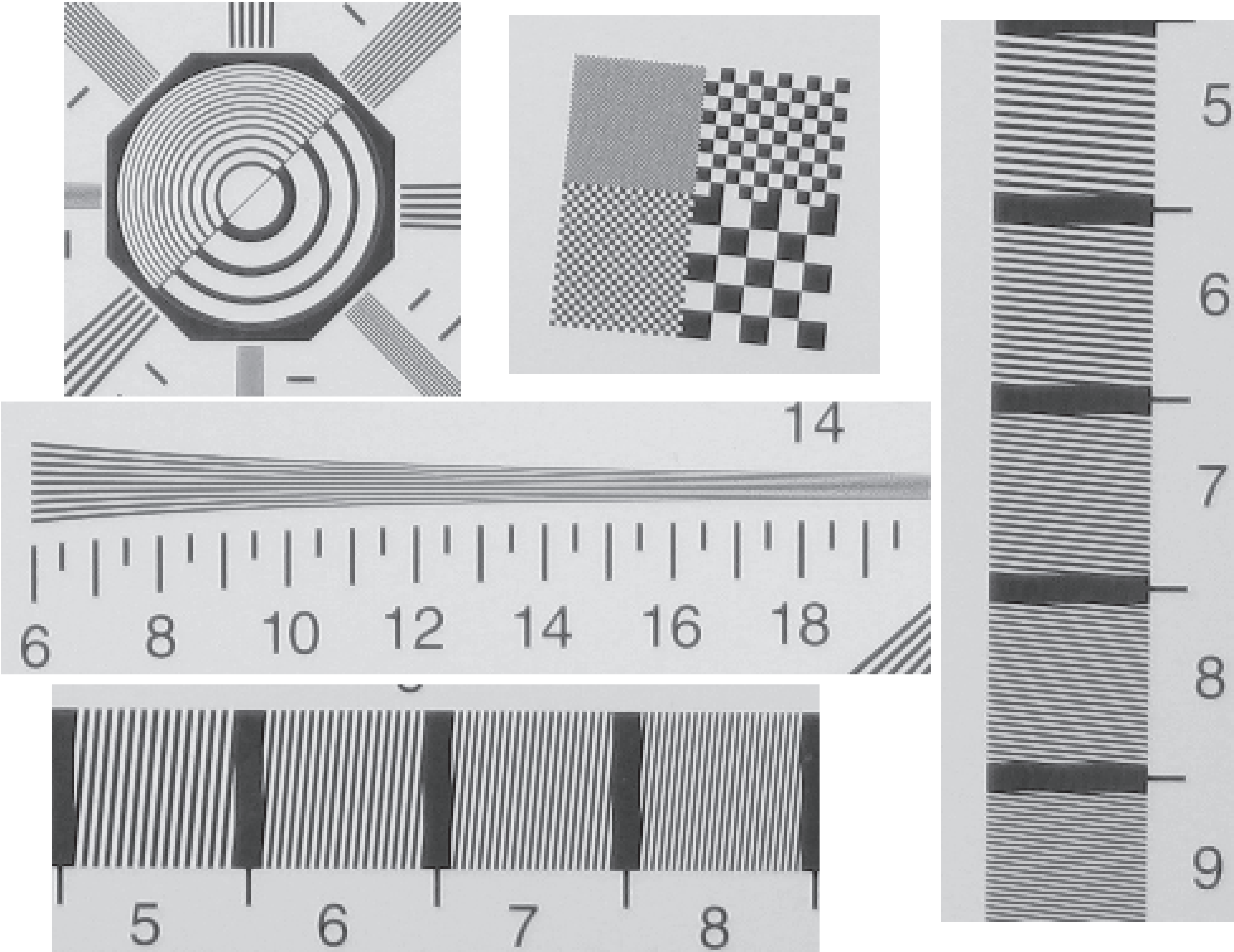
image
1
0.8
0.6
0.4
0.2
0

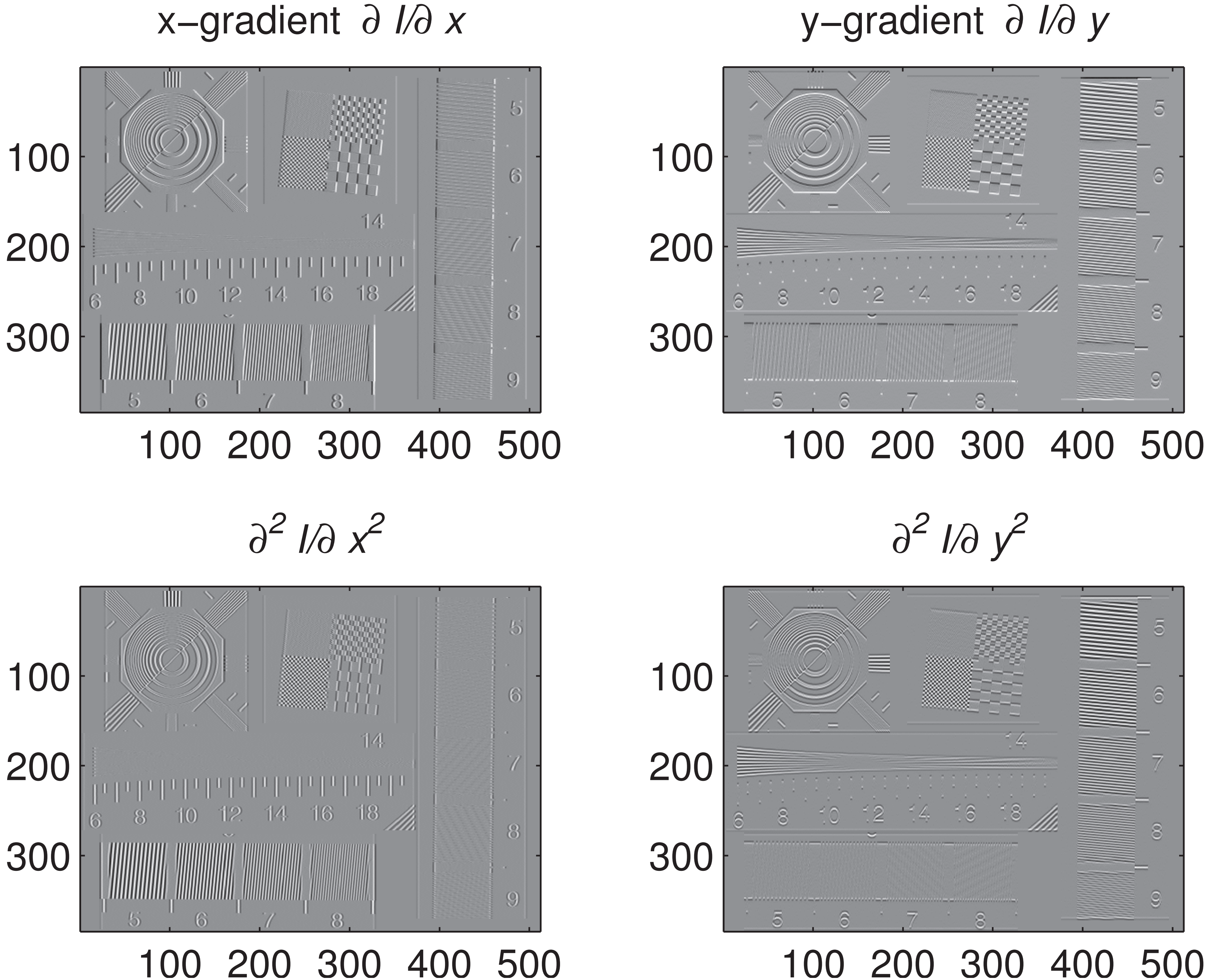

laplace
0.02
0.01
0
-0.01
-0.02



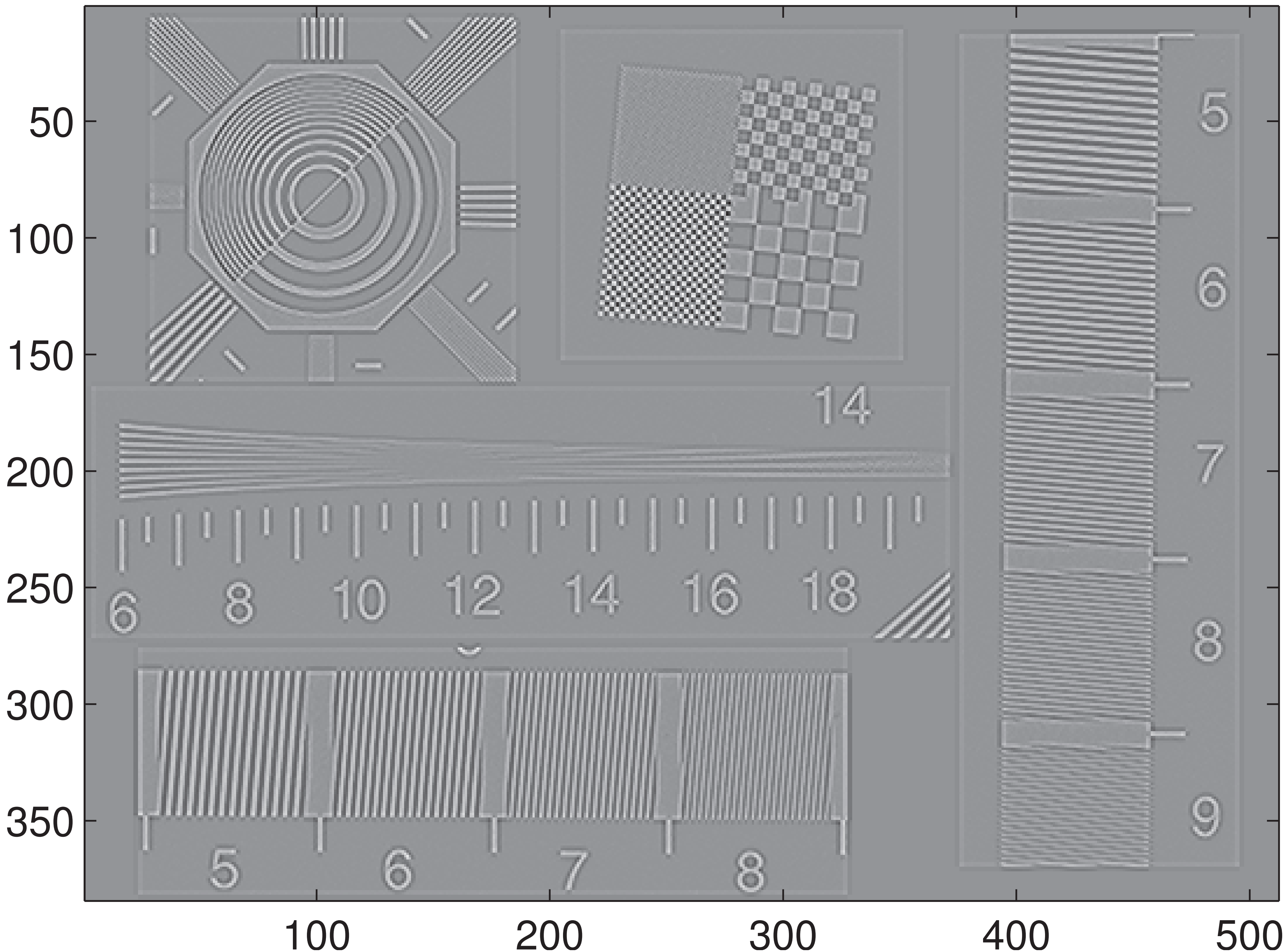
original signal f
first derivative ∂ f / ∂ x
Second derivative – Laplacian (scaled) C ∂2 f/∂ x2
improved signal f – C ∂2 f/∂ x2
sharpened signal ...
... truncated to <0,255>

Original image

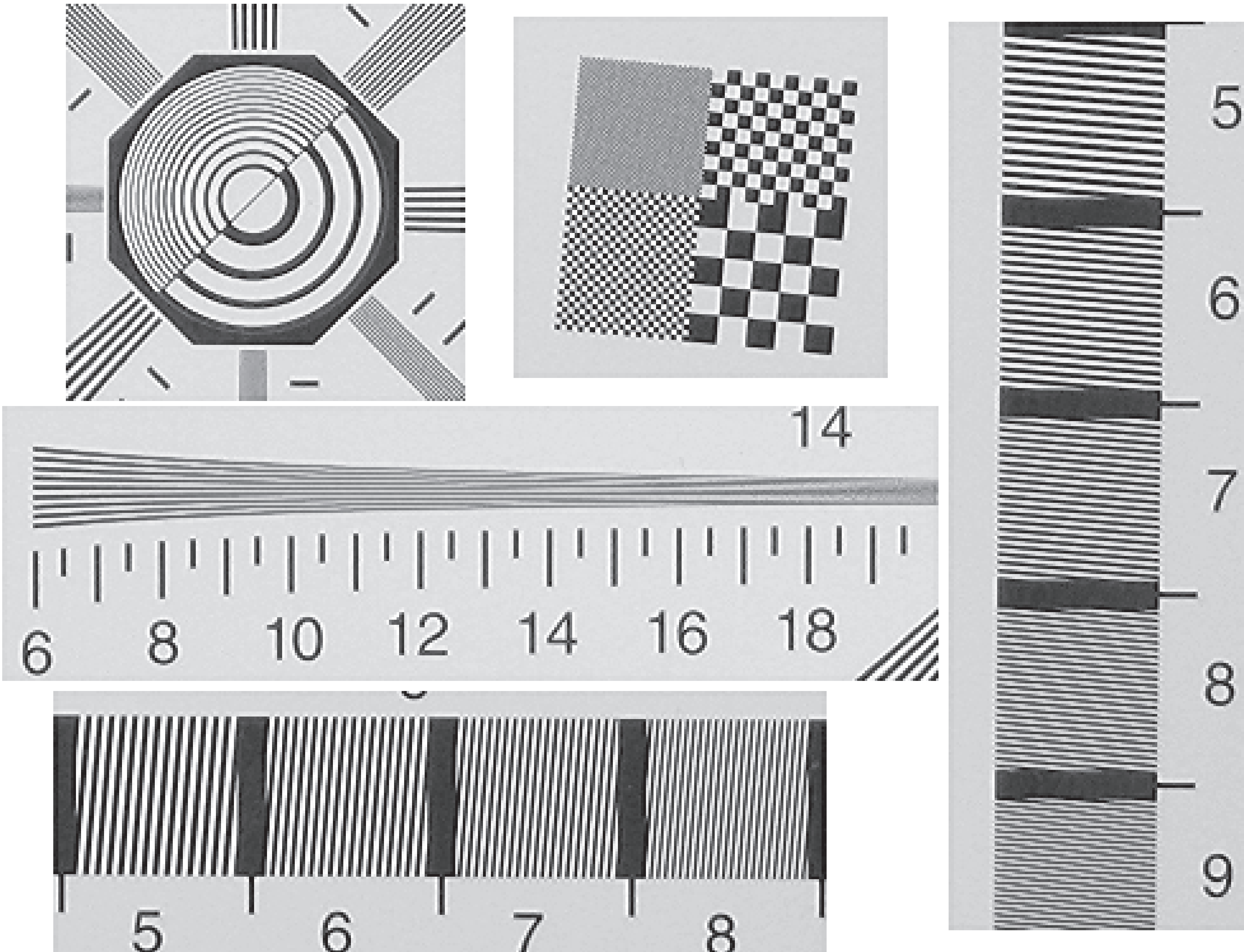



x−gradient ∂ I/∂ x
y−gradient ∂ I/∂ y
∂2 I/∂ x2
∂2 I/∂ y2
100
200
300
100 200 300 400 500

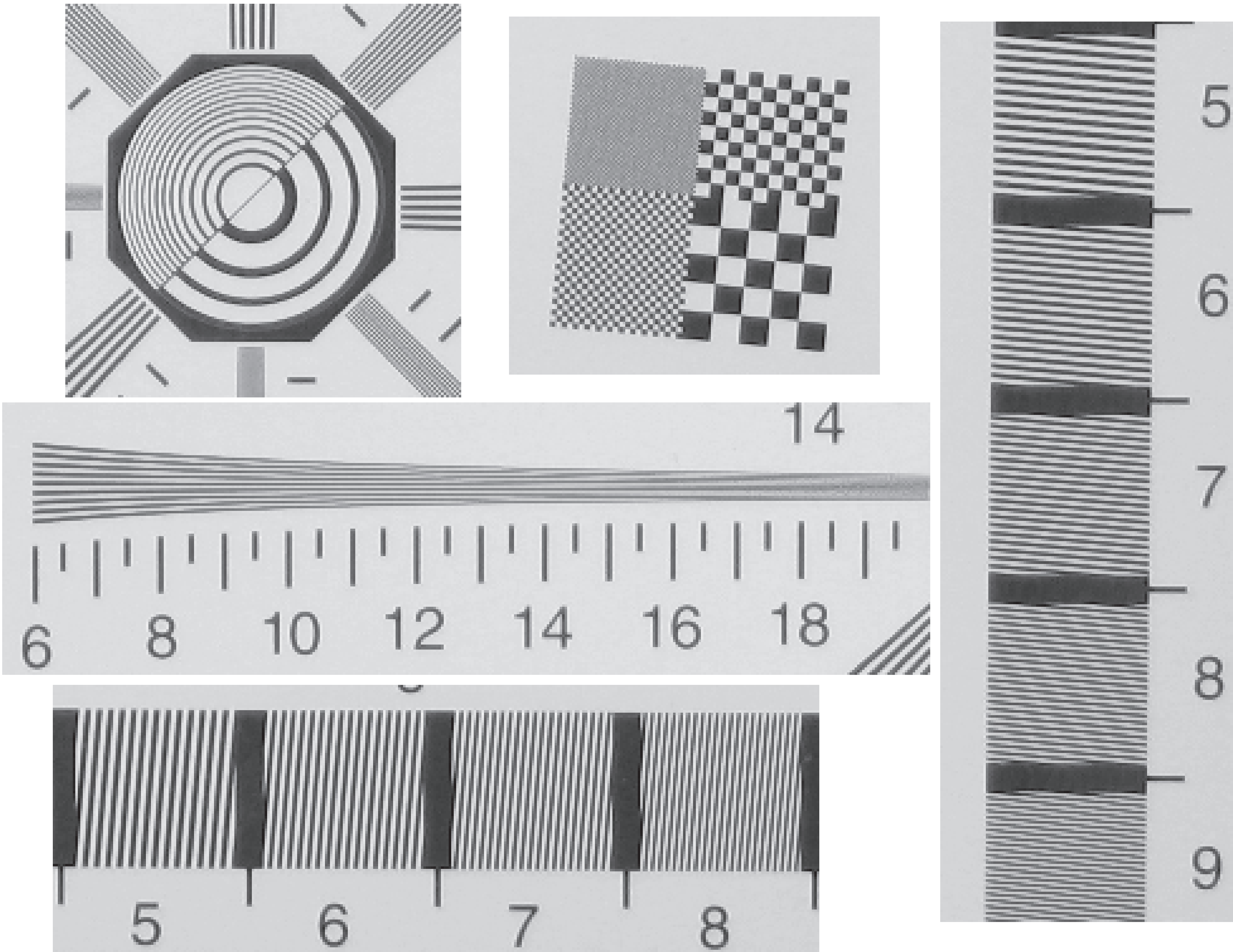
Laplacian: $\nabla = \partial^2 I/\partial x^2 + \partial^2 I/\partial y^2$



Sharpened image, C=0.5



Original image



Sharpened image, C=0.5