Detectores de Bordas

Pablo G. Cavalcanti

Detecção de Bordas

- Derivada de primeira ordem (gradiente)

Gradiente simples

-1
+1

-1	+1

$$\nabla I(x,y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial I(x,y)}{\partial x} \\ \dfrac{\partial I(x,y)}{\partial y} \end{bmatrix}$$

Prewitt

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

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

$$M(x,y) = |I(x,y)| = \left[G_x^2 + G_y^2\right]^{1/2}$$

Sobel

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

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

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

G_x G_y

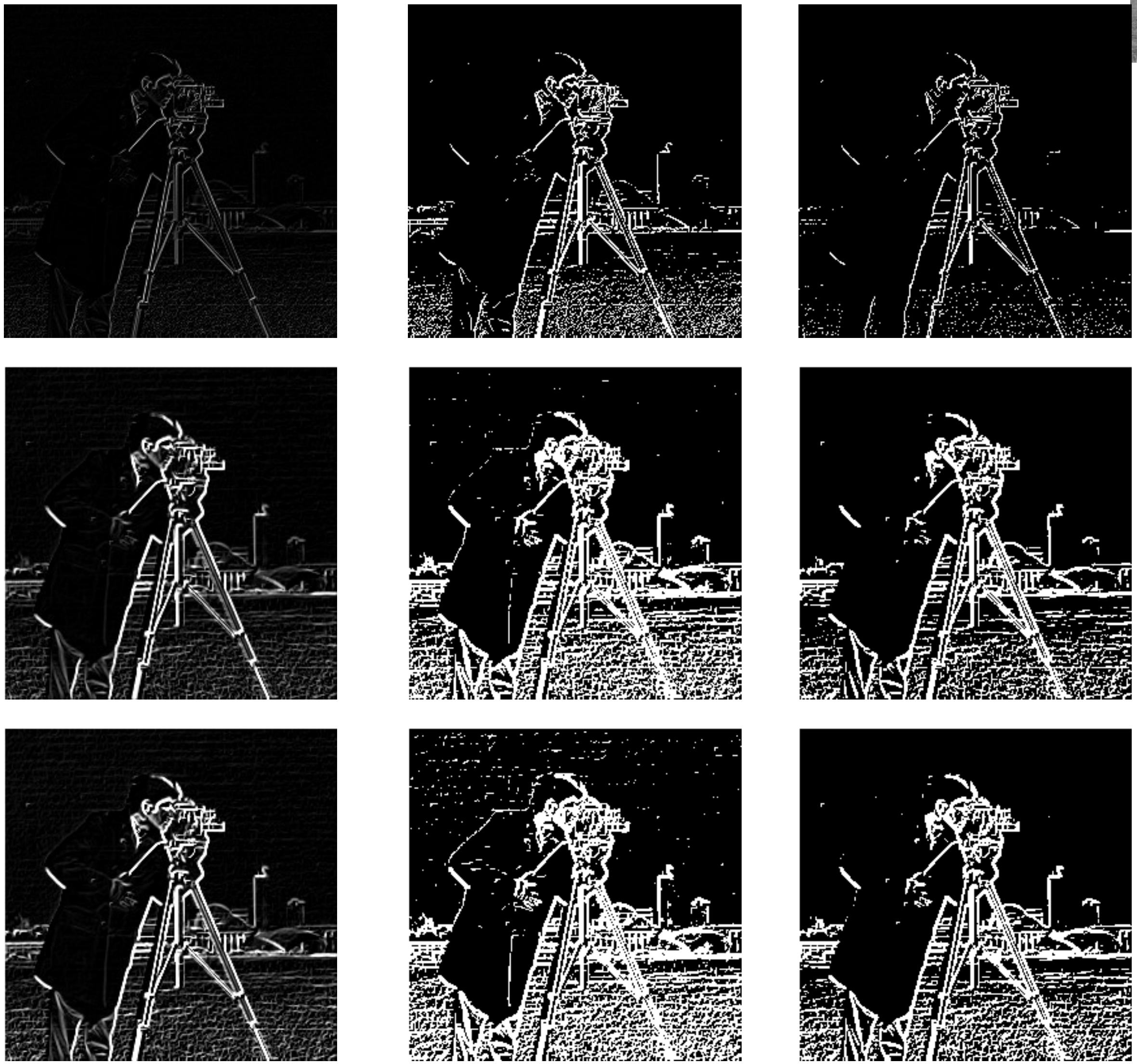
Cameraman

Detecção de Bordas

Gradiente

Prewitt

Sobel



Th= 20 *Th*= 40

Detecção de Bordas

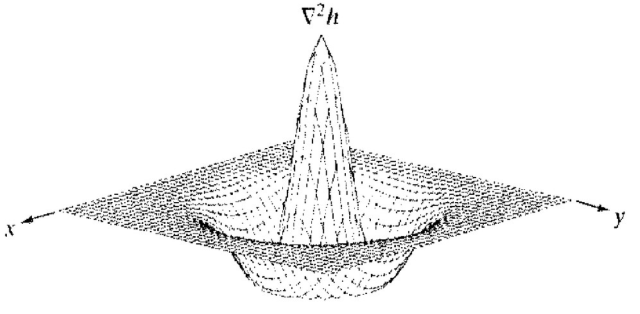
- Derivada de segunda ordem (Laplaciano)

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

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

-1	-1	-1
-1	+8	-1
-1	-1	-1

Normalmente é utilizada juntamente com um filtro de suavização *h*.
Exemplo de um Laplaciano de Gaussiana (LoG):



(Gonzalez & Woods, 2002)

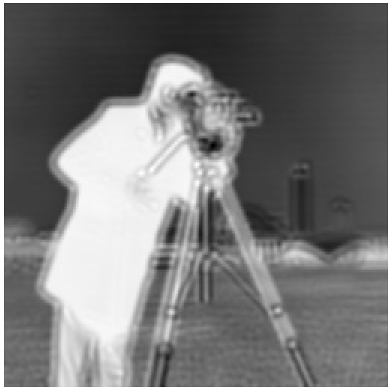
Cameraman

Detecção de Bordas

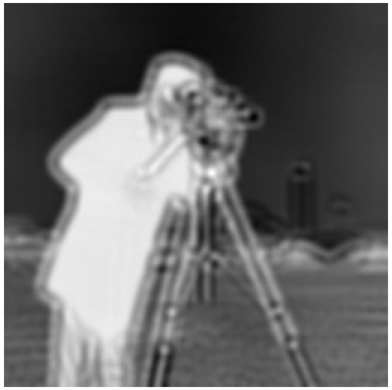
LoG

Cruzamentos por Zero.

11 x 11
$\sigma = 2$



15 x 15
$\sigma = 3.$



Detecção de Bordas

Método de Canny

1) Suaviza a imagem com um filtro Gaussiano;

2) Determina a magnitude e a direção do gradiente para cada pixel;

3) Se a magnitude de um pixel for maior que as de seus 2 vizinhos na direção (arredondada) do gradiente, marca esse pixel como borda - caso contrário, marca como fundo;

4) Remove as bordas "fracas" com histerese (2 limiares).

Histerese

Entradas: Imagem de bordas (ex: gradientes);
Limiares Th_0 e Th_1.
Saída: Imagem de bordas “fortes”/relevantes.

Algoritmo:

1) Marcar todas bordas com magnitude maior que Th_1 como borda;

2) Percorrem todos pixels com magnitude entre Th_0 e Th_1;

3) Se um desses pixels for vizinho à outro já marcado como borda, marcá-lo também;

4) Repetir a partir do passo 2 até atingir estabilidade.

Cameraman

Detecção de Bordas

Canny

$Th_0 = 10;$
$Th_1 = 30;$
$\sigma = 1.$

$Th_0 = 20;$
$Th_1 = 40;$
$\sigma = 2.$

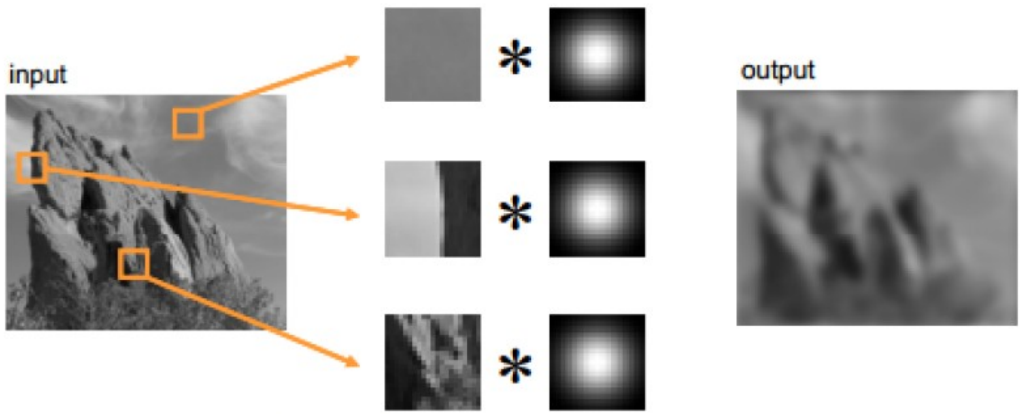
Filtros mais complexos

Pablo G. Cavalcanti

Fonte: http://www.decom.ufop.br/guillermo/BCC326/slides/Processamento%20de%20Imagens%20-%20Filtros%20Nao%20Lineares.pdf

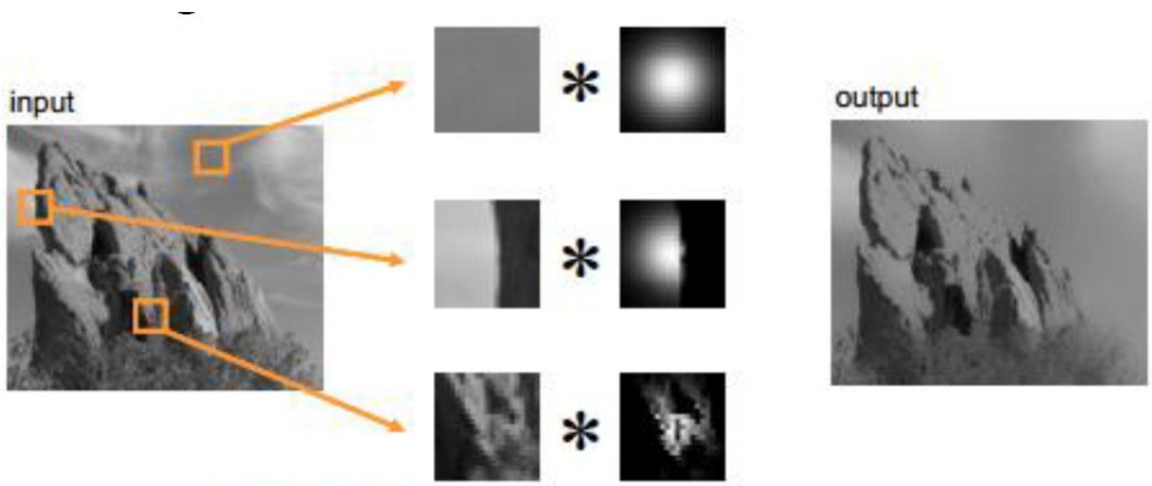
Filtragem Bilateral

Filtro Gaussiano



Same Gaussian kernel everywhere.

Filtro Bilateral



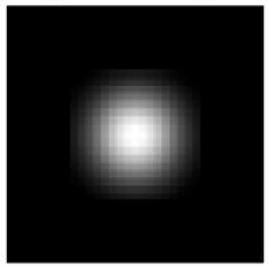
The kernel shape depends on the image content.

Fonte: http://www.decom.ufop.br/guillermo/BCC326/slides/Processamento%20de%20Imagens%20-%20Filtros%20Nao%20Lineares.pdf

A mesma idéia: média ponderada dos pixels

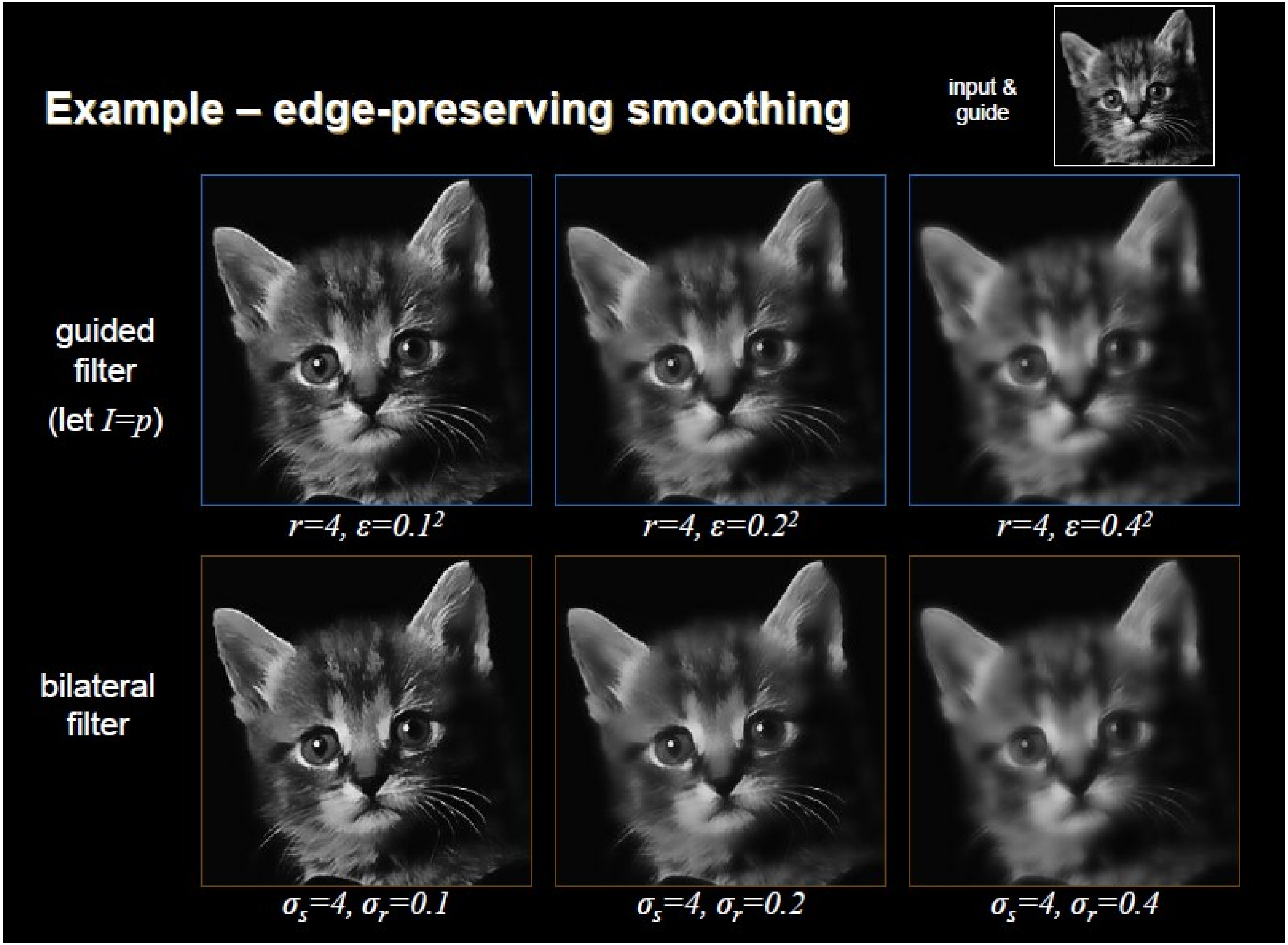
$$BF[I]_{\mathbf{p}} = \overset{\text{new}}{\underbrace{\frac{1}{W_{\mathbf{p}}}}_{\text{normalization factor}}} \sum_{\mathbf{q}\in S} \overset{\text{not new}}{\underbrace{G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)}_{\textit{space}\text{ weight}}} \overset{\text{new}}{\underbrace{G_{\sigma_r}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)}_{\textit{range}\text{ weight}}} I_{\mathbf{q}}$$

$$W_p = \sum_{q\in S} G_{\sigma_s}(\| p-q \|) G_{\sigma_r}(\| I_p - I_q \|)$$

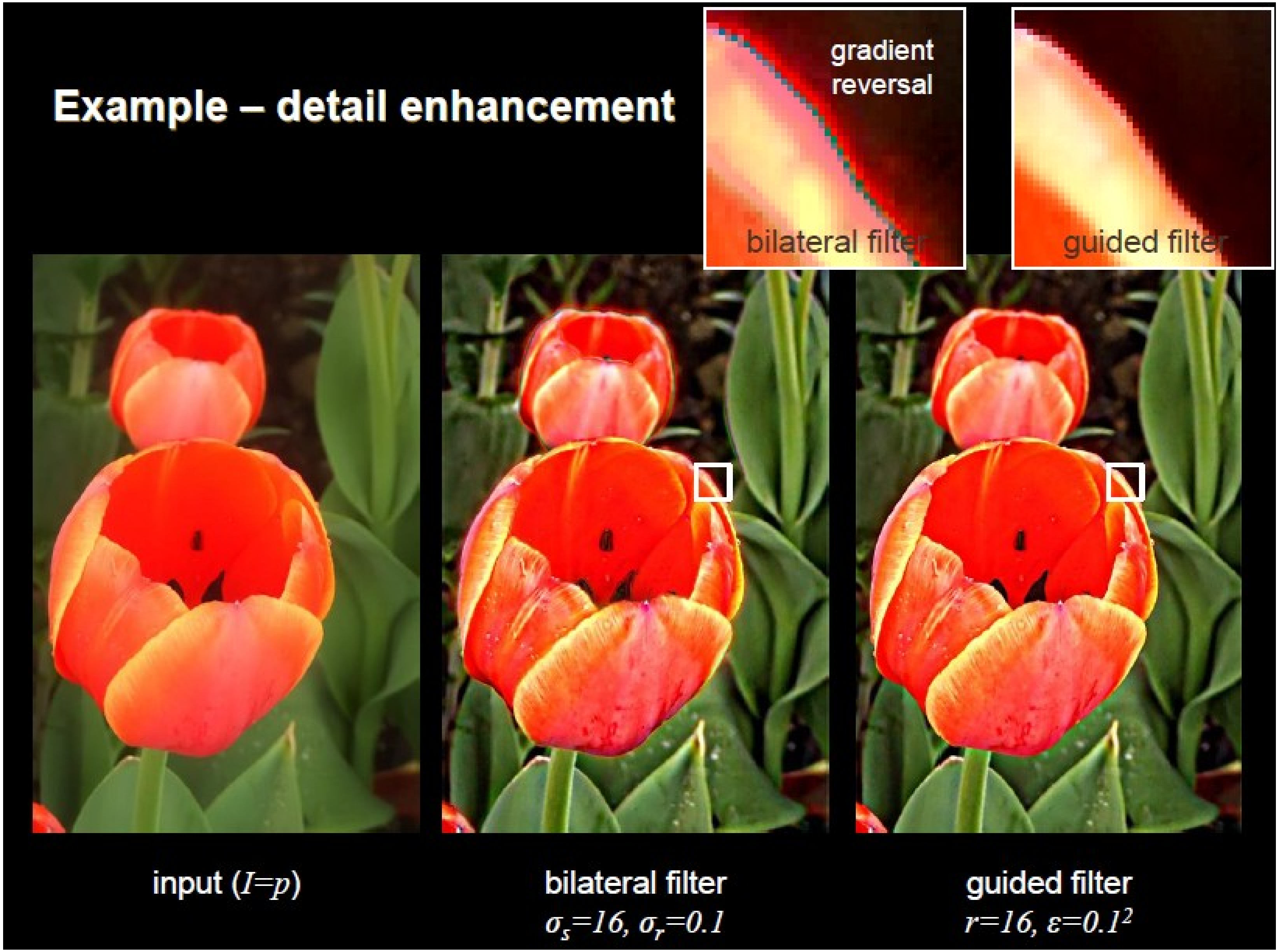


Sigma_p = 7
Sigma_q = 9
window = 3

Fonte: http://research.microsoft.com/en-us/um/people/kahe/eccv10/eccv10ppt.pdf



Fonte: http://research.microsoft.com/en-us/um/people/kahe/eccv10/eccv10ppt.pdf



A Total Variation Approach for Customizing Imagery to Improve Visual Acuity

CARLOS MONTALTO, IGNACIO GARCIA-DORADO, and DANIEL ALIAGA
Purdue University
MANUEL M. OLIVEIRA
Instituto de Informática - UFRGS
and
FENG MENG
Purdue University

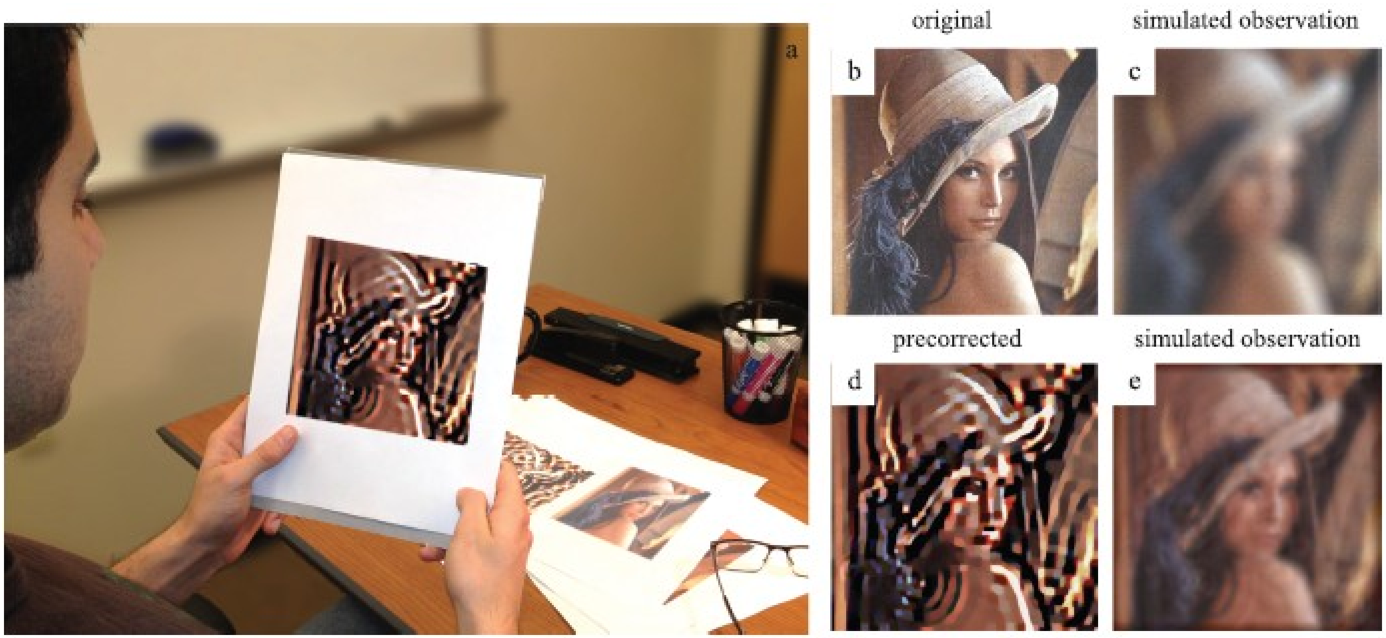
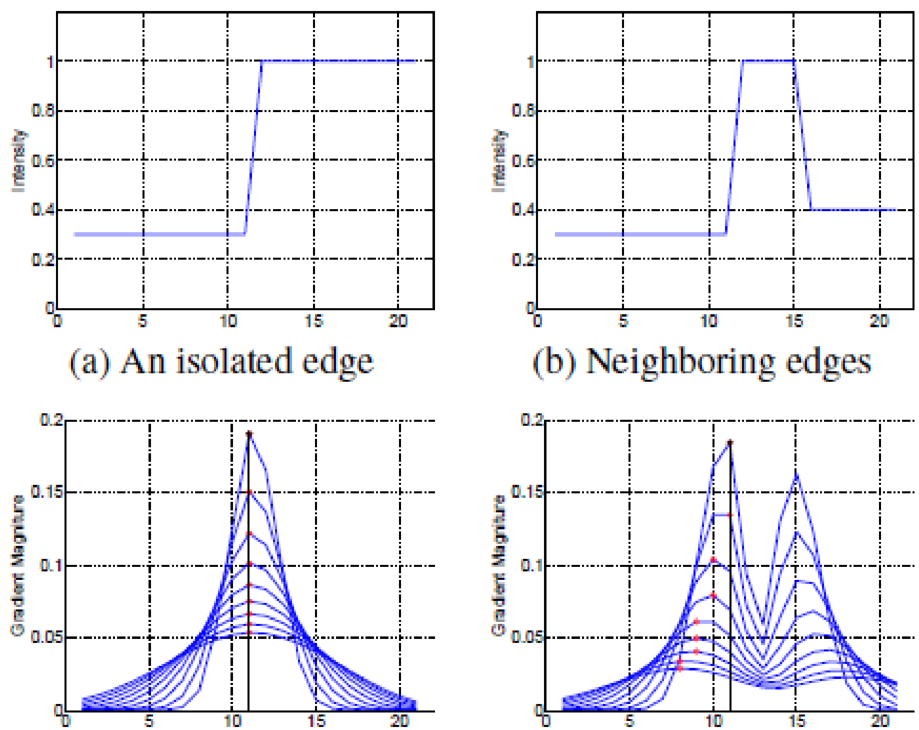


Fig. 1. Precorrection on Paper. (a) We depict the scenario where an observer sees our precorrected image, without his corrective eyewear, and perceives a sharper picture than viewing the original image; (b) original image (kindly provided by the USC SIPI Image Database); In c-e we show images from a scenario where the human observer is at a distance of 2.5m from the image and suffers -2.5D of refractive error; (c) simulated result of the observer looking at (b); (d) precorrected image; (e) simulated result of the observer looking at (d).

ADAPTIVE SCALE SELECTION FOR MULTIRESOLUTION DEFOCUS BLUR ESTIMATION

Ali Karaali and Claudio Rosito Jung

Federal University of Rio Grande do Sul (UFRGS)
Institute of Informatics
Porto Alegre - RS - Brazil
{akaraali, crjung}@inf.ufrgs.br



(a) An isolated edge (b) Neighboring edges

(c) Multiscale gradients of (a) (d) Multiscale gradients of (b)

Fig. 1. (a) A preserved edge profile and its gradient profile (c), (b) an edge profile with neighbouring edge interference and its gradient profile (d)

Fonte: https://homes.cs.washington.edu/~shanqi/res/deblur_siggraph08.pdf

High-quality Motion Deblurring from a Single Image *

Qi Shan, Jiaya Jia
Department of Computer Science and Engineering
The Chinese University of Hong Kong

Aseem Agarwala
Adobe Systems, Inc.

Figure 1 High quality single image motion-deblurring. The left sub-figure shows one captured image using a hand-held camera under dim light. It is severely blurred by an unknown kernel. The right sub-figure shows our deblurred image result computed by estimating both the blur kernel and the unblurred latent image. We show several close-ups of blurred/unblurred image regions for comparison.

Fonte: https://homes.cs.washington.edu/~shanqi/res/deblur_siggraph08.pdf

High-quality Motion Deblurring from a Single Image *

Qi Shan Jiaya Jia
Department of Computer Science and Engineering
The Chinese University of Hong Kong

Aseem Agarwala
Adobe Systems, Inc.

(a) Blurred image (b) Iteration 1 (c) Iteration 6 (d) Iteration 10

Figure 10 Illustration of our optimization in iterations. (a) The blurred images. The ground truth blur kernels are shown in the green rectangles. Our simple initialized kernels are shown in the red rectangles. (b)-(d) The restored images and kernels in iteration 1, 6, and 10.