CSE185
Introduction to Computer Vision
Lab 03: Spatial Filters

Instructor: Prof. Ming-Hsuan Yang
TA: Chun-Han Yao & Yi-Wen Chen

Overviews

- Median filter:

Noisy input image

Median filter output

Overviews

- Sobel filter:

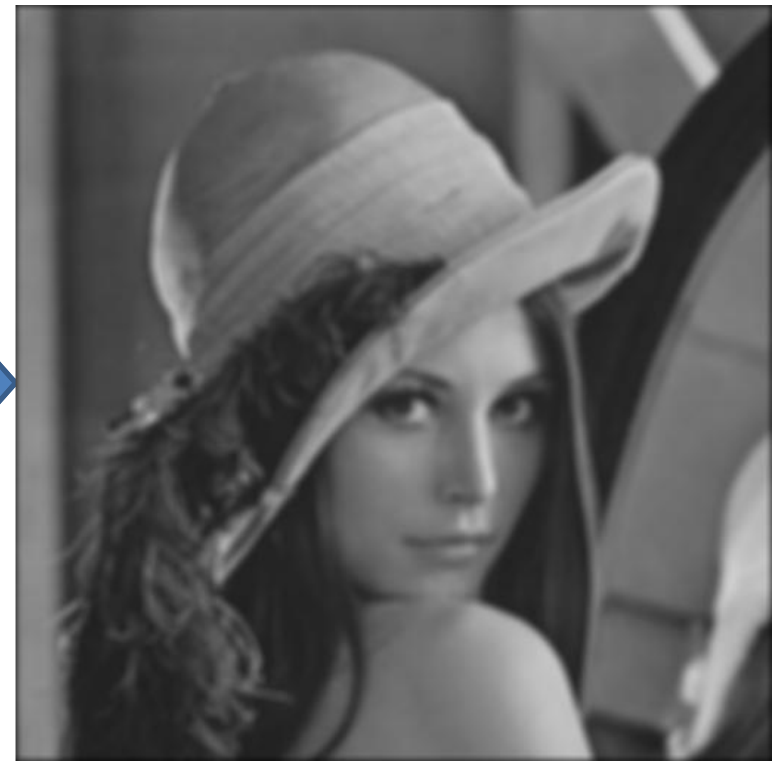
Input image

Sobel filter output

Overviews

- Gaussian filter:

Input image

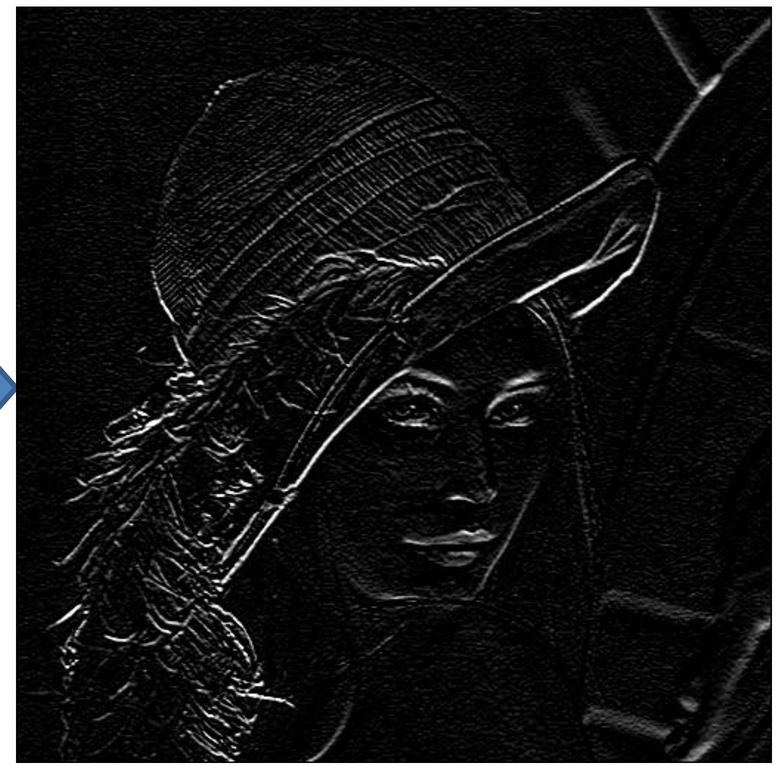


Gaussian filter output

Sobel Filter

- Sobel filter is a simple edge detector
- $H = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$ or $H = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$

Input image



Sobel filter output

Sobel Filter

- `sobel_filter.m`

```
function output = sobel_filter(img, kernel)
    % YOUR CODE HERE
end
```

- In `lab03.m`:

```
img = im2double(imread('lena.jpg'));

%% Sobel filter
H = [1, 2, 1; 0, 0, 0; -1, -2, -1]; % horizontal edge
%H = [1, 0, -1; 2, 0, -2; 1, 0, -1]; % vertical edge

img_sobel = sobel_filter(img, H);
figure, imshow(img_sobel);
imwrite(img_sobel, 'sobel_h.jpg');
```

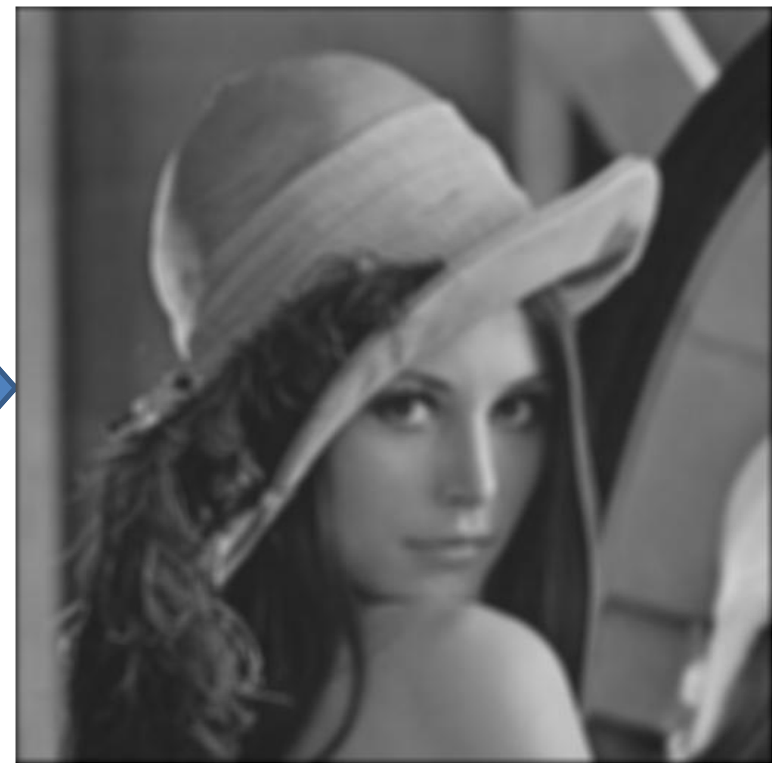
- Compare your result with `I = imfilter(img, H);`

Gaussian Filter

- Gaussian filter is a low-pass/smoothing filter

$$g(\Delta x, \Delta y) = \frac{1}{Z} \exp\left(-\frac{\Delta x^2 + \Delta y^2}{2\sigma^2}\right)$$

Input image



Gaussian filter output

Gaussian Filter

- Gaussian filter is a low-pass/smoothing filter

$$g(\Delta x, \Delta y) = \frac{1}{Z} \exp\left(-\frac{\Delta x^2 + \Delta y^2}{2\sigma^2}\right)$$

- $\Delta x, \Delta y$ are offsets from the kernel center, and Z is the normalized term to make the sum of all weights equal to 1
- In MATLAB:

0.0113	0.0149	0.0176	0.0186	0.0176	0.0149	0.0113
0.0149	0.0197	0.0233	0.0246	0.0233	0.0197	0.0149
0.0176	0.0233	0.0275	0.029	0.0275	0.0233	0.0176
0.0186	0.0246	0.029	0.0307	0.029	0.0246	0.0186
0.0176	0.0233	0.0275	0.029	0.0275	0.0233	0.0176
0.0149	0.0197	0.0233	0.0246	0.0233	0.0197	0.0149
0.0113	0.0149	0.0176	0.0186	0.0176	0.0149	0.0113

Gaussian Filter

- Gaussian filter is a low-pass/smoothing filter

$$g(\Delta x, \Delta y) = \frac{1}{Z}\exp\left(-\frac{\Delta \mathrm{x}^2 + \Delta \mathrm{y}^2}{2\sigma^2}\right)$$

- $\Delta x, \Delta y$ are offsets from the kernel center, and Z is the normalized term to make the sum of all weights equal to 1
- In MATLAB:

$\boxed{0.0113}$	0.0149	0.0176	0.0186	0.0176	0.0149	0.0113
0.0149	0.0197	0.0233	0.0246	0.0233	0.0197	0.0149
0.0176	0.0233	0.0275	0.029	0.0275	0.0233	0.0176
0.0186	0.0246	0.029	$\boxed{0.0307}$	0.029	0.0246	0.0186
0.0176	0.0233	0.0275	0.029	0.0275	0.0233	0.0176
0.0149	0.0197	0.0233	0.0246	0.0233	0.0197	0.0149
0.0113	0.0149	0.0176	0.0186	0.0176	0.0149	0.0113

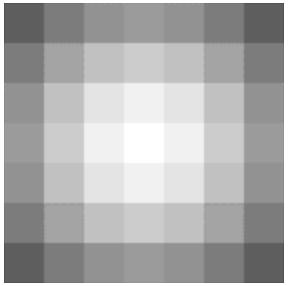
Gaussian Filter

- You don't need to implement Gaussian kernel by yourself (this is bonus).
- Use `fspecial`:

```
H = fspecial('gaussian', hsize, sigma);
```

`hsize` is the size of the kernel, `sigma` = σ

- Visualization of a Gaussian kernel:



Gaussian Filter

- `gaussian_filter.m`

```
function output = gaussian_filter(img, hsize, sigma)
    H = fspecial('gaussian', hsize, sigma);
    % YOUR CODE HERE
end
```

- In `lab03.m`:

```
%% Gaussian filter
hsize = 5; sigma = 2;
% hsize = 9; sigma = 4;

img_gaussian = gaussian_filter(img, hsize, sigma);
figure, imshow(img_gaussian);
imwrite(img_gaussian, 'gaussian_5.jpg');
```

- Compare your result with `I = imfilter(img, H);`

Lab assignment 03

1. Implement sobel_filter.m, use horizontal filter and save the image as sobel_h.jpg
2. Use vertical filter and save the image as sobel_v.jpg
3. Implement gaussian_filter.m, use hsize = 5, sigma = 2, and save the image as gaussian_5.jpg
4. Use hsize = 9, sigma = 4, and save the image as gaussian_9.jpg
5. Upload your output images and lab03.m, sobel_filter.m, and gaussian_filter.m

* Use 'lena.jpg' as the input image for all questions.

Bonus

- Implement boundary padding
  - zero padding (pixels outside the image are zero)
  - replicated/repeated padding (pixels outside the image are the same as pixels on the boundaries)
- You can try built-in function `padarray` or `wextend`, but you need to implement by yourself to get bonus points

- Implement Gaussian kernel
  - compute a matrix based on $g(\Delta x, \Delta y) = \exp\left(-\frac{\Delta x^2 + \Delta y^2}{2\sigma^2}\right)$
  - normalize the matrix to have sum equal to 1

Reference

- Spatial Filtering
- Linear Algebra and MATLAB Tutorial
- Awesome Computer Vision