Computer Vision I

Tim Luchterhand, Paul Nykiel (Group 17)

9. Juli 2018

1 Gaussian Pyramids

```
boats = im2double(imread('boats.tif'));

G = cell(1,6);
S = cell(1,6);
G{1} = boats;
S{1} = boats;

figure;

subplot(2,6,1);
imshow(G{1});
title({"Subsampled by 1","with Gaussian Blur"});
subplot(2,6,7);
imshow(S{1});
title({"Subsampled by 1","without Gaussian Blur"});


for c=2:6
    blurred = imgaussfilt(G{c-1});
    G{c} = blurred(1:2:end, 1:2:end);
    S{c} = S{c-1}(1:2:end, 1:2:end);
    subplot(2,6,c);
    imshow(G{c});
    title({"Subsampled by " + 2^(c-1),"with Gaussian Blur"});
    subplot(2,6,c+6);
    imshow(S{c});
    title({"Subsampled by " + 2^(c-1),"without Gaussian Blur"});
end

```

```
print("sh05ex01.eps", "-depsc");
```

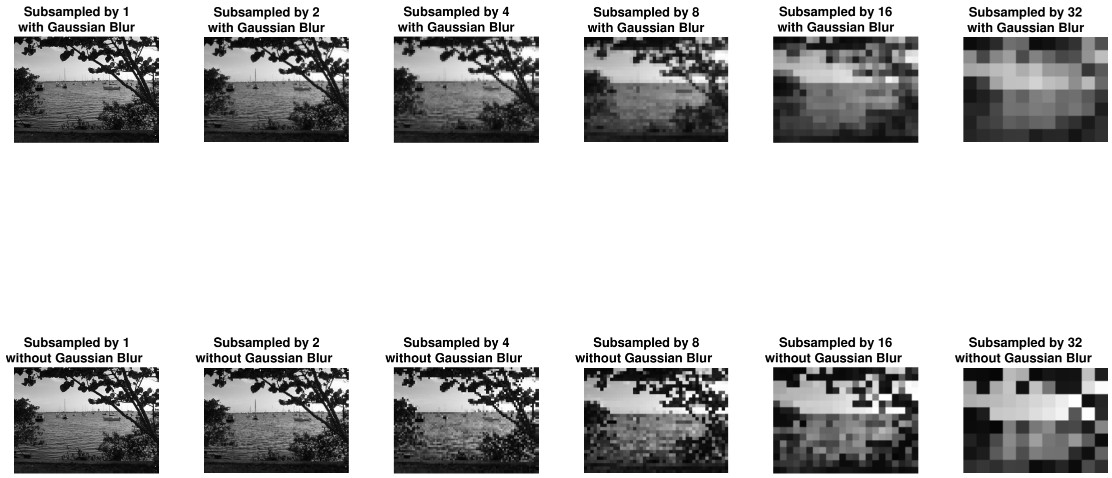

Abbildung 1: Output of the matlab script

2 Laplacian Pyramids

2.1

```
% Part 1
boats = im2double(imread('boats.tif'));
G = cell(1,6);
G{1} = boats;
for c=2:6
    blurred = imgaussfilt(G{c-1});
    G{c} = blurred(1:2:end, 1:2:end);
    S{c} = S{c-1}(1:2:end, 1:2:end);
end

Ge = cell(1,5);
for c=1:5
    Ge{c} = imresize(G{c+1}, 2);
end

L = cell(1,6);
L{6} = G{6};
for c=1:5
    L{c} = G{c} - Ge{c};
end
```

```
figure;
for c=1:6
    subplot(6, 3, (c-1)*3+1);
    imshow(G{7-c});
    title("G("+ (7-c) + ")");
    if c > 1
        subplot(6, 3, (c-1)*3+2);
        imshow(Ge{7-c});
        title("G_e("+ (7-c) + ")");
    end
    subplot(6, 3, (c-1)*3+3);
    imshow(L{7-c});
    title("L("+ (7-c) + ")");
end
print("sh05ex02_1.eps", "-depsc");
```

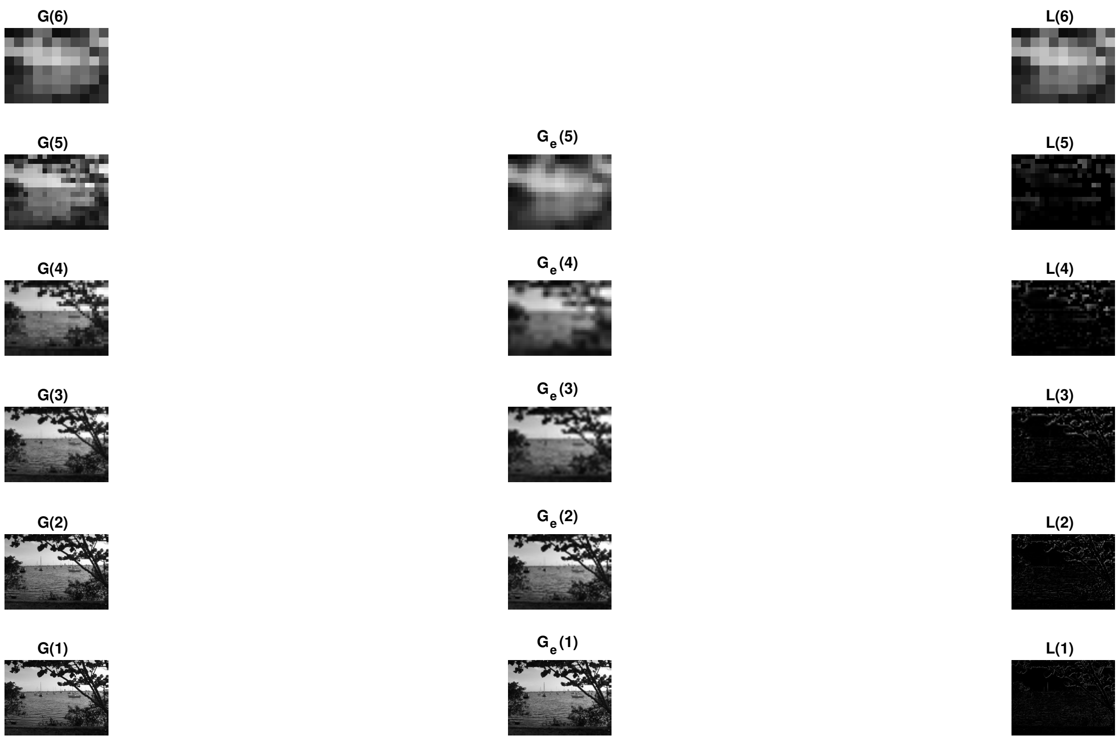

Abbildung 2: Output of the matlab script

2.2

```
% Part 2
```

```
E = cell(1, 5);
Gr = cell(1,5);
E{5} = imresize(L{6}, 2);
for c = 1:5
      Gr{6-c} = E{6-c} + L{6-c};
      if(c < 5)
            E{5-c} = imresize(Gr{6-c}, 2);
      end
end


figure;
for c=1:6
    subplot(6, 3, (c-1)*3+1);
    imshow(L{7-c});
    title("L("+ (7-c) + ")");
    if c > 1
          subplot(6, 3, (c-1)*3+2);
          imshow(E{7-c});
          title("E("+ (7-c) + ")");
          subplot(6, 3, (c-1)*3+3);
          imshow(Gr{7-c});
          title("G_r("+ (7-c) + ")");
    end
end
print("sh05ex02_2.eps", "-depsc");
```

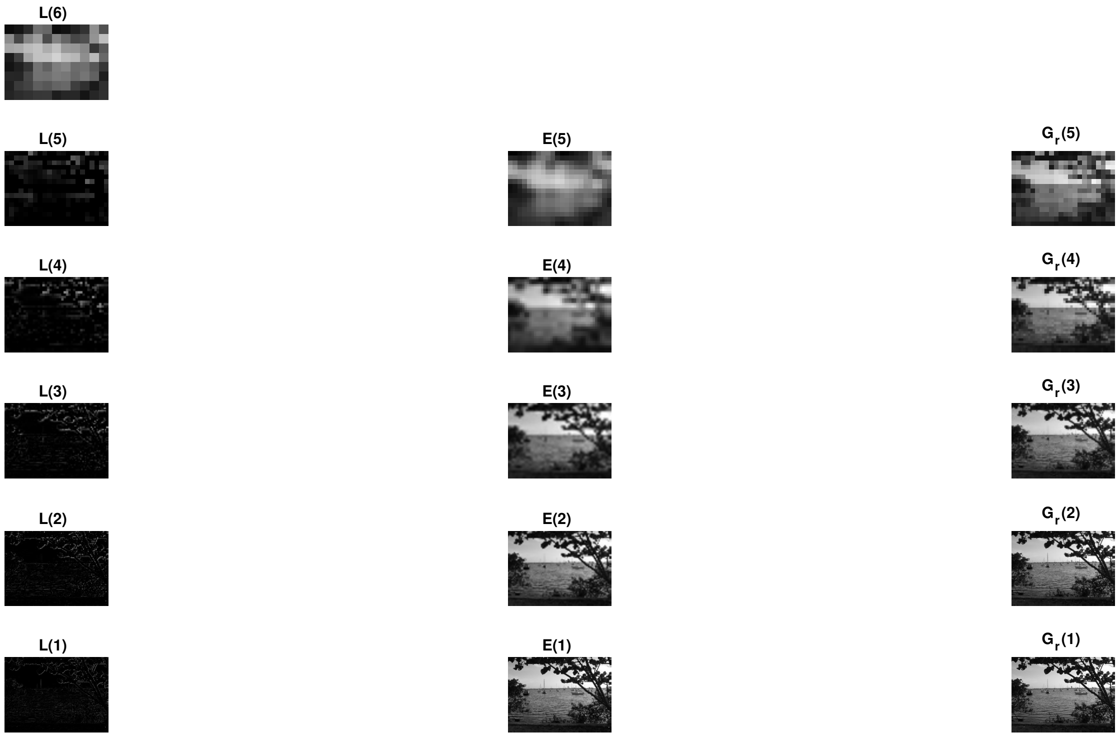

Abbildung 3: Output of the matlab script

2.3

```
% Part 3
L_thresh = cell(1,6);
L_thresh{6} = L{6};
for c = 1:5
    m = max(max(L{c}));
    t = m * 0.2;
    L_thresh{c} = L{c};
    L_thresh{c}(find(abs(L_thresh{c}) < t)) = 0;
end

E_reduced = cell(1, 5);
Gr_reduced = cell(1,5);
E_reduced{5} = imresize(L_thresh{6}, 2);
for c = 1:5
    Gr_reduced{6-c} = E_reduced{6-c} + L_thresh{6-c};
    if(c < 5)
        E_reduced{5-c} = imresize(Gr_reduced{6-c}, 2);
    end
```

```
end

figure;
for c=1:6
    subplot(6, 3, (c-1)*3+1);
    imshow(L{7-c});
    title("L("+ (7-c) + ")");
    if c > 1
        subplot(6, 3, (c-1)*3+2);
        imshow(E_reduced{7-c});
        title("E("+ (7-c) + ") with Threshold");
        subplot(6, 3, (c-1)*3+3);
        imshow(Gr_reduced{7-c});
        title("G_r("+ (7-c) + ") with Threshold");
    end
end
print("sh05ex02_3_1.eps", "-depsc");

figure();
subplot(1,2,1);
imshow(Gr{1});
subplot(1,2,2);
imshow(Gr_reduced{1});
print("sh05ex02_3_2.eps", "-depsc");
```

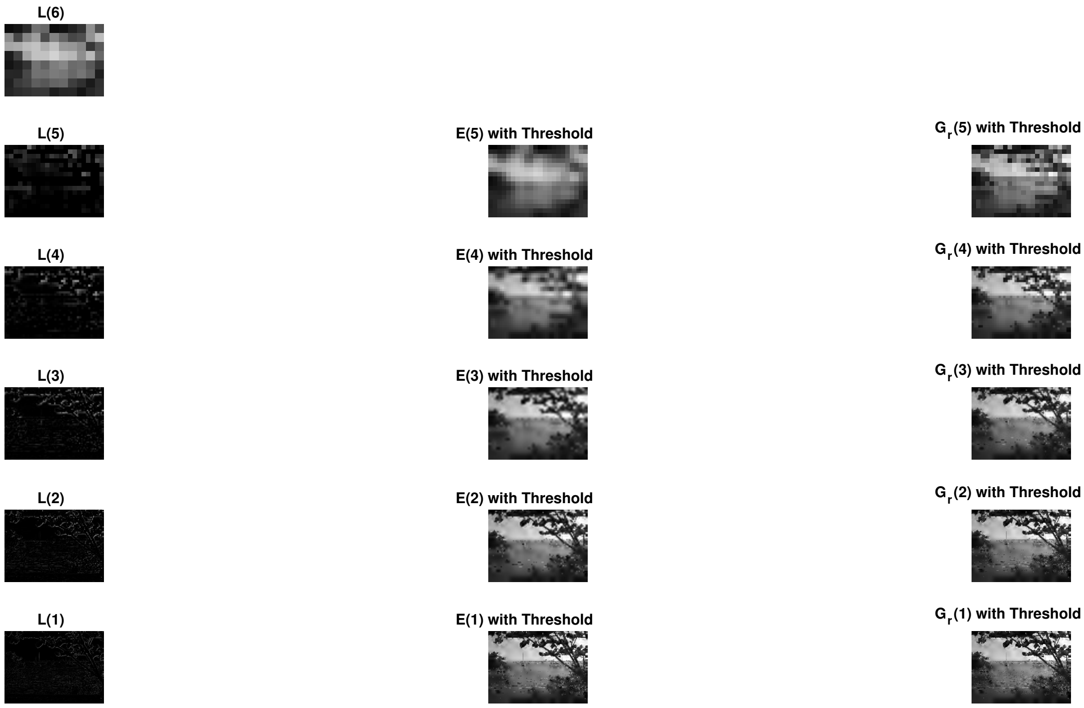

Abbildung 4: Output of the matlab script

Abbildung 5: Difference between the reconstructed and the original image

2.4

```
% Part 4
x = 0:0.05:0.3;
y = zeros(size(x));

for l=1:length(x)
    L_thresh = cell(1,6);
```

```
        L_thresh{6} = L{6};
        for c = 1:5
            m = max(max(L{c}));
            t = m * x(l);
            L_thresh{c} = L{c};
            L_thresh{c}(find(abs(L_thresh{c}) < t)) = 0;
        end

        E_reduced = cell(1, 5);
        Gr_reduced = cell(1,5);
        E_reduced{5} = imresize(L_thresh{6}, 2);
        for c = 1:5
             Gr_reduced{6-c} = E_reduced{6-c} + L_thresh{6-c};
             if(c < 5)
                  E_reduced{5-c} = imresize(Gr_reduced{6-c}, 2);
             end
        end
        mse = sum(sum((G{1} - Gr_reduced{1}).^2)) / (size(G{1},1) *
            size(G{1},2));
        y(l) = mse;
end

figure;
plot(x, y);
xlabel("\lambda");
ylabel("MSE");
title("Mean square error as a function of the threshold");
print("sh05ex02_4.eps", "-depsc");
```

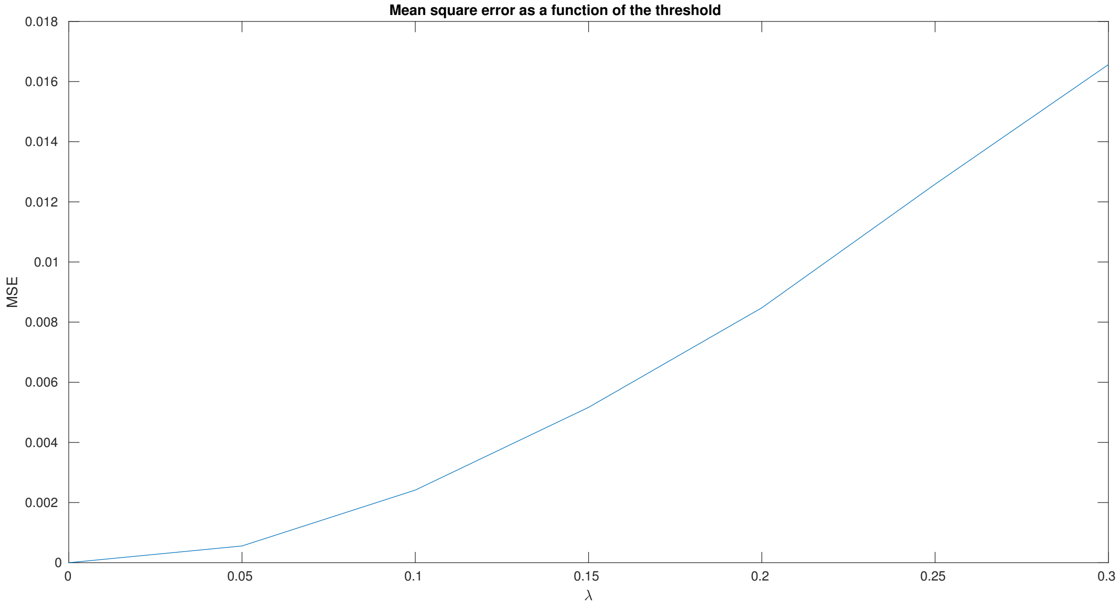

Abbildung 6: Output of the matlab script

3 Gabor Wavelets

```
% Part 1
basket = im2double(imread('basket.jpg'));
thetas = [0 pi/4 pi/2] * 180/pi;
fs = [0.64 0.32 0.08];

filters = cell(1,9);
b = 2.32;
x = zeros(1,9);
y = zeros(1,9);

for theta = 1:length(thetas)
    for f = 1:length(fs)
        sigma = b / fs(f);
        filters{(theta-1)*3 + f} = getGabor(fs(f), sigma, thetas(
            theta));
        x((theta-1)*3 + f) = thetas(theta);
        y((theta-1)*3 + f) = fs(f);
    end
end

% Part 2
```

```
energy = zeros(1,9);
filtered = cell(1,9);

for c = 1:9
    filtered{c} = imfilter(basket, filters{c}, 'conv');
    figure;
    imshow(abs(filtered{c}));
    energy(c) = sqrt(sum(sum(abs(filtered{c}).^2)));
end

% Part 3
figure;
stem3(x, y, energy);
xlabel("\theta");
ylabel("f");
print("sh05ex03.eps", "-depsc");
```

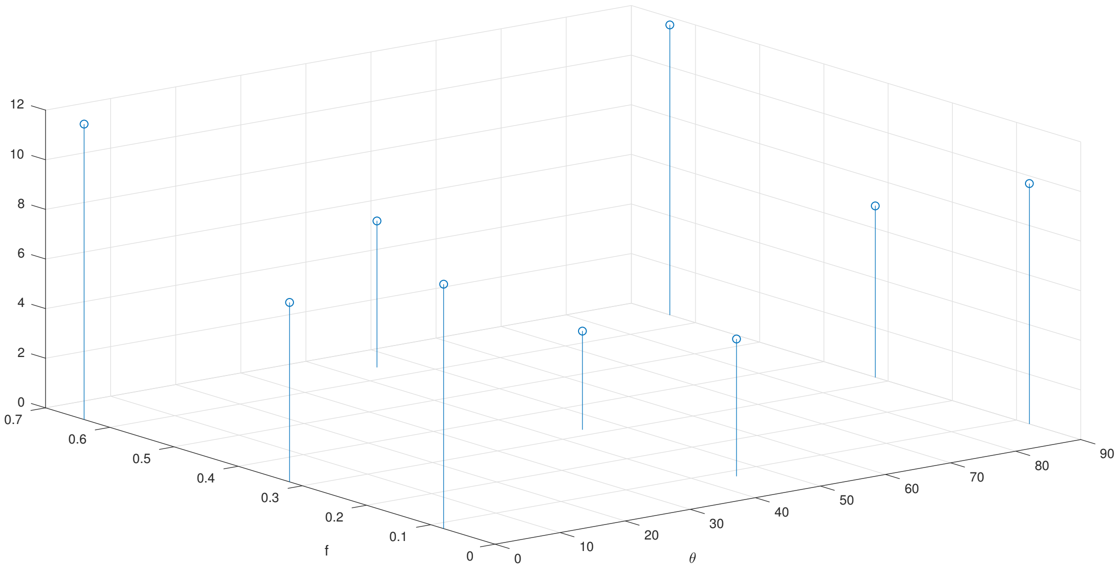

Abbildung 7: Output of the matlab script

For the orientation $\theta = 45°$ all the frequency energys are lower compared to the other orientations since there are no significant image structures like edges.

In the other directions ($\theta = 0°$ and $\theta = 90°$) there are peaks at high frequencys due to prominent edges, as well as peaks at low frequencys as a result of partitally homogeneous regions.