# 1 Matlab correction

## 1.1 Pyramidal decomposition and reconstruction

### 1.1.1 Decomposition

The following function makes the decomposition of the Laplacian and Gaussian pyramids at the same time. The Laplacian pyramid can be reconstructed without any additional information. This is illustrated in Fig. 1.

```
function [pyrG, pyrL] = LaplacianPyramidDecomposition(Image, levels, mode)
% Gaussian and Laplacian Pyramid decomposition
% Image must be either 2D (grayscale) or 3D (color)
% levels: number of levels of decomposition (not including the original level)
% mode: approximation mode 'bilinear', 'nearest', etc. for use in imresize
%       bilinear by default
% pyrG: Gaussian pyramid
% pyrL: Laplacian pyramid
%
% The Laplacian pyramid is of size levels+1. The last image pyrL{levels+1}
% is the approximation image of the last level. The original image is
% exactly reconstructed by the LaplacianPyramidReconstruction
% function.

% pyramids
pyrL = cell(levels+1, 1);
pyrG = cell(levels+1, 1);

% gaussian filter
H = fspecial('gaussian');

if ~exist('mode', 'var')
    mode = 'bilinear';
end

for i=1:levels
    ImagePrec = Image; % previous image
    g=imfilter(Image,H,'same');

    Image = imresize(g, .5, mode);
    ImagePrime = imresize(Image, [size(g,1), size(g,2)], mode);

    pyrL{i} =  ImagePrec - ImagePrime;
    pyrG{i} =  ImagePrec;

end
pyrL{levels+1} = Image;
```

```
pyrG{levels+1} = Image;
```

**Informations**

Notice that the images are of type double, which implies a special care when displaying them, for example by using the command imshow(I, []).

### 1.1.2 Reconstruction

The reconstruction is straighforward and exact because of the construction of the residue. The details can be filtered (removed for example), thus giving the following result Fig. 2.

```
function Image = LaplacianPyramidReconstruction(pyr, mode)
% Laplacian Pyramid Reconstruction
% pyr: Laplacian pyramid
% mode: approximation mode for imresize, bilinear by default

if ~exist('mode', 'var')
    mode = 'bilinear';
end

levels = size(pyr, 1)-1;

Image = pyr{levels+1};
% reconstruction from bottom to top
for i=levels:-1:1
    Image = pyr{i} + imresize(Image, ...
                    [size(pyr{i},1), size(pyr{i}, 2)], mode);
end
```

## 1.2 Scale-space decomposition and multiscale filtering

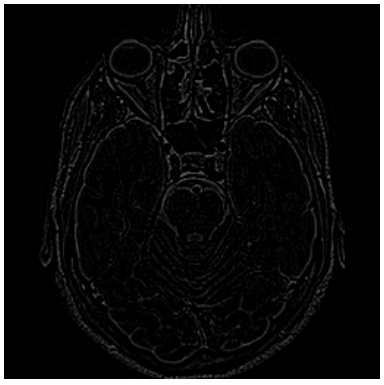
```
% Morphological scale-space decomposition
k=3; % number of decompositions
ss=cell(2,k);
for i=1:k
    se = strel('disk',2*i); % structuring element
    ss{1,i}=imdilate(A,se); % dilation
    ss{2,i}=imerode(A,se);  % erosion
end
```

(a) Original image.

(b) Gaussian pyramid level 1.

(c) Level 2.

(d) Level 3.



(e) Laplacian pyramid level 1.

(f) Level 2.

(g) Level 3.

Figure 1: Gaussian and Laplacian pyramids, for 3 levels of decomposition. The Laplacian pyramid in addition to the last level of the Gaussian pyramid is required to exactly reconstruct the original image.

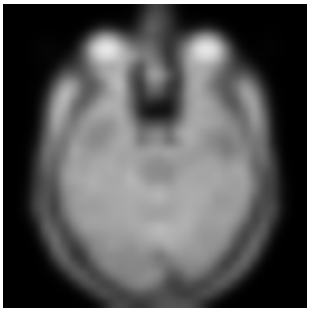
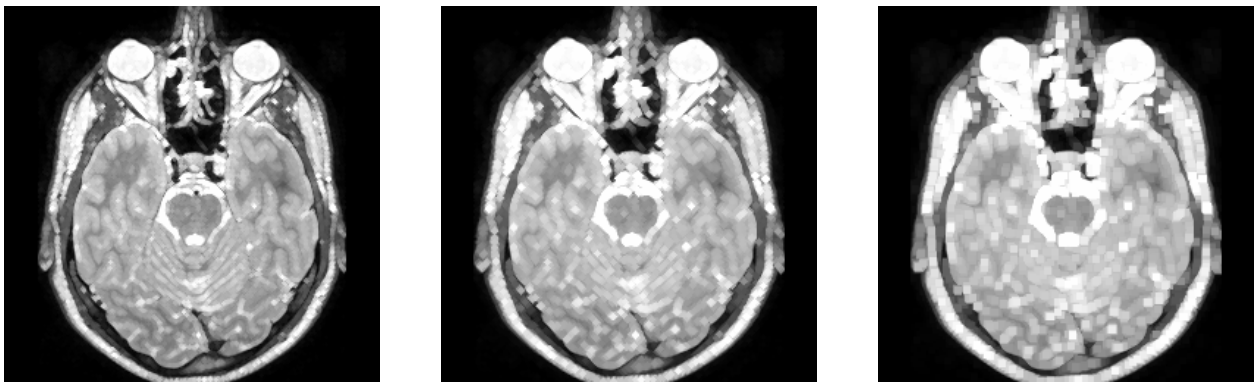
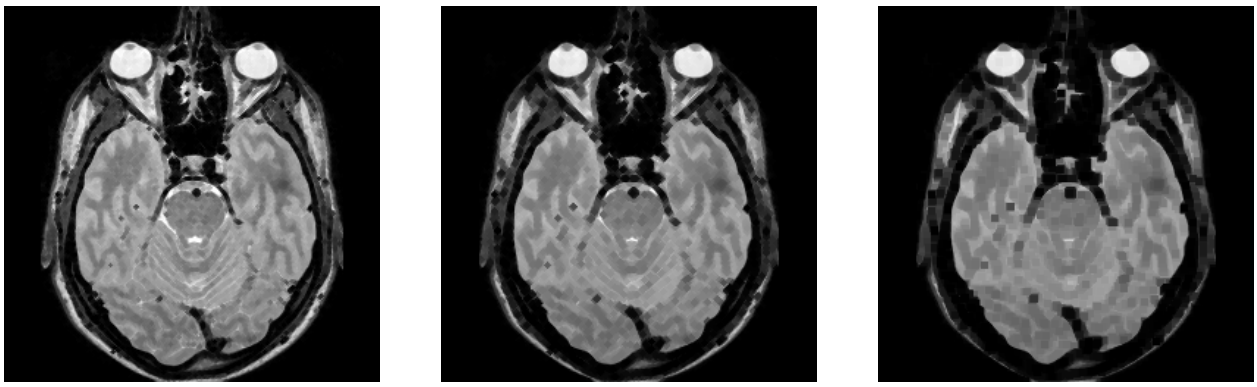


Figure 2: Reconstruction of the pyramid without any detail.



(a) Dilation scale 1. (b) Dilation scale 2. (c) Dilation scale 3.

Figure 3: Morphological multiscale decomposition by dilation.



(a) Erosion scale 1. (b) Erosion scale 2. (c) Erosion scale 3.

Figure 4: Morphological multiscale decomposition by erosion.

## 1.3 Kramer and Bruckner multiscale decomposition

The results are illustrated in Fig.5.

(a) $MK_B^1$. (b) $MK_B^2$. (c) $MK_B^3$.

Figure 5: Kramer and Bruckner multiscale decomposition, for $r = 5$.

```
sskb=cell(1,k+1);
sskb{1, 1} = A;
r  = 5;
for i=2:k+1
    sskb{1,i}=kb(sskb{1,i-1}, r);
end
```

```
function K = kb(I, r)
% Kramer and Bruckner iterative filter
% I: originale image
% r: size of neighborhood
%
% return filtered image
se = strel('disk', r);
D = imdilate(I, se);
E = imerode(I, se);
difbool=double((D-I)<(I-E));

K = D.*difbool+E.*(1-difbool);
```