Lab 3: Pyramid and Scale-space

Kush Gupta

1 Introduction

In the lab session we explored different methods for the the scale analysis of the scene in the scene.

2 Quad-tree decomposition

2.1 Exercise 1

Regions at the last level of a quad-tree decomposition, or leaf-regions of the quad-tree, are regions of one-pixel size which do not fulfill the condition function at any level. If the condition is designed to divide the image into homogeneous square-regions, these leaf-regions usually coincide with strong image edges.

In the quad tree example, the parameter K, defines the maximum number of levels in which the original Image will be divided. A larger value of K, breaks the image into more parts than a smaller value. The parameter alpha α decides the number of leaves in the end for a QT decomposition of the image.A higher value (0.1) of alpha results in fewer leaves and a smaller value(0.001) results into more leaves.

2.2 Exercise 2

In the second task, $quad_{t}ree_{e}x_{2}.m$ when producing a qt decomposition using the function $maxf(x) - minf(x) > alpha$. I observed that using a higher value of alpha (0.7) results in fewer leaves as the window size is big and the fine details of the image are smoothed out 2a. However, using a small value of alpha say (0.1), resulted image into more fine division of the image and some textures were also preserved in the image 2b.

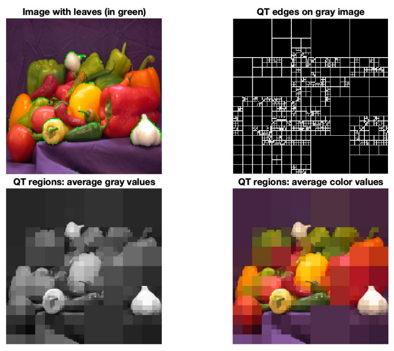


Image with leaves (in green)
QT edges on gray image
QT regions: average gray values
QT regions: average color values

(a) Qt decomposition using a higher alpha value (0.01).

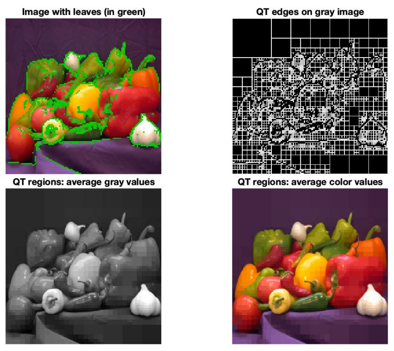


Image with leaves (in green)
QT edges on gray image
QT regions: average gray values
QT regions: average color values

(b) Qt decomposition using a higher alpha value (0.001).

Figure 1: Quadtree decomposition

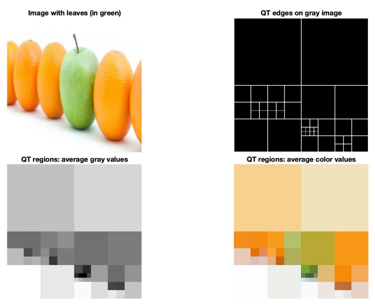


Image with leaves (in green)
QT edges on gray image
QT regions: average gray values
QT regions: average color values

(a) Qt decomposition using a higher alpha value (0.7).

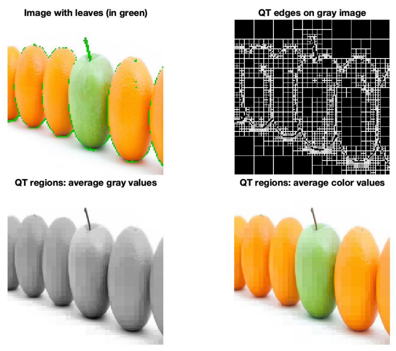
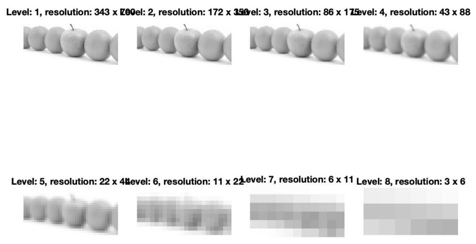


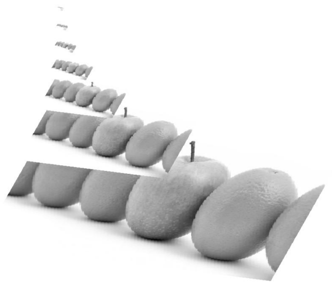
Image with leaves (in green)
QT edges on gray image
QT regions: average gray values
QT regions: average color values

(b) Qt decomposition using a lower alpha value (0.1).

Figure 2: $Quadtree_{e}xample_{2}$


Level: 1, resolution: 343 x 700
Level: 2, resolution: 172 x 350
Level: 3, resolution: 86 x 175
Level: 4, resolution: 43 x 88
Level: 5, resolution: 22 x 44
Level: 6, resolution: 11 x 22
Level: 7, resolution: 6 x 11
Level: 8, resolution: 3 x 6

(a) Image showing the down sampling and gaussian smoothing.



(b) Image pyramid containing images at various scales, down sampled by a factor of 2.

Figure 3: nlevel = 10 and scale factor =2

3 Gaussian Pyramids

3.1 Exercise 4

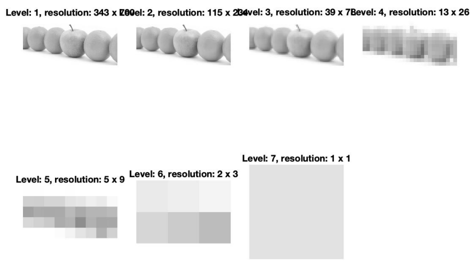
In the $GaussianPyramid_{e}xample.m$, the scale arrangement properties of the pyramid were tested for different values of nlevel and scale factor.

3.2 For values, nlevel = 10 and scale factor =2

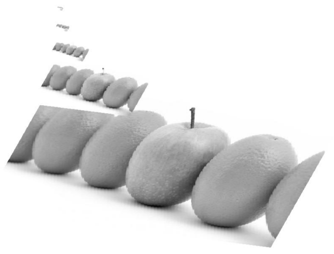
Upon selecting above values for the guassian pyramid, i observed that there were 10 images in the pyramid and each image was downsampled by a factor of 2. Due to gaussian smoothing, the edges of the objects (oranges and apples) were smoothed eventually as the picture was down sampled. 3a 3b

3.3 For values, nlevel=8 and scale factor = 3

These values for the guassian pyramid, showed 8 images in the feature pyramid and the orginal image was down sampled by a factor of 3 and due to gaussian smoothing, the edges of the smaller objects (oranges and apples) were smoothed quickly, due to the size of the kernel size and image down sampling. 4a 4b


Level: 1, resolution: 343 x 700
Level: 2, resolution: 115 x 234
Level: 3, resolution: 39 x 78
Level: 4, resolution: 13 x 26
Level: 5, resolution: 5 x 9
Level: 6, resolution: 2 x 3
Level: 7, resolution: 1 x 1

(a) Image showing the down sampling and gaussian smoothing.



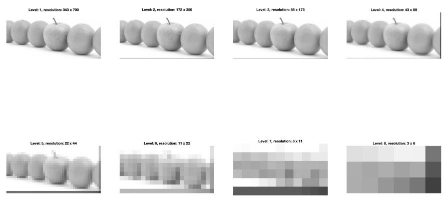
(b) Image pyramid containing images at various scales, down sampled by a factor of 3.

Figure 4: nlevel=8 and scale factor = 3

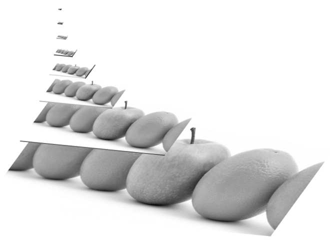
3.4 Exercise 5

3.5 doAvgPyramid.m

The AvgPyramid function uses a sliding window of size 2X2 and replaces the pixel value in the window with the window average. I used 8 levels and a scale factor of 2. I observed that in the images, there is a problem with pixels on left boundary which can be eliminated with padding as a pre-processing step. 5a 5b


Level: 1, resolution: 343 x 700
Level: 2, resolution: 172 x 350
Level: 3, resolution: 86 x 175
Level: 4, resolution: 43 x 88
Level: 5, resolution: 22 x 44
Level: 6, resolution: 11 x 22
Level: 7, resolution: 6 x 11
Level: 8, resolution: 3 x 6

(a) Image showing the down sampling and average filtering.



(b) Image pyramid containing images at various scales, down sampled by a factor of 2.

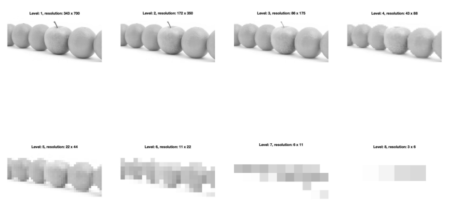
Figure 5: Avg Pyramid

3.6 doMaxPyramid.m

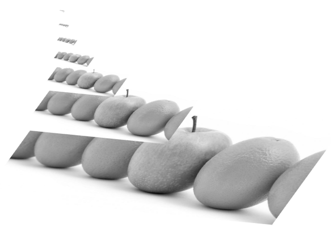
The MaxPyramid function uses a sliding window of size 2X2 and replaces the pixel value in the window with maximum value in the window. I used 8 levels and a scale factor of 2. I observed that in the images, the object boundaries disappeared more quickly when compared to avg function. The object boundaries are merged more quickly. 6a 6b

4 Exercise 6: Bandpass Pyramids

The bandpas pyramid function takes only images with standard size like 512X512 or 256X256. If the input image is of different size then it must be resized to a standard size. The Bandpass pyramid gives output as DoG, Difference of two Gaussian's at different scales which is computationally fast and effective.7a 7b For nlevels=10,7,6 and scale factor =2, i observed that the image pyramid contains only 8, 6 and 5 images respectively, because it's taking difference between two consecutive gaussians. Also, for the given image choosing scale factor of more or less than 2 was giving error.

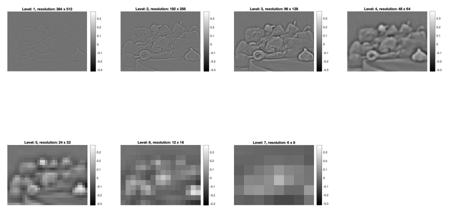

Level: 1, resolution: 343 x 700
Level: 2, resolution: 172 x 350
Level: 3, resolution: 86 x 175
Level: 4, resolution: 43 x 88
Level: 5, resolution: 22 x 44
Level: 6, resolution: 11 x 22
Level: 7, resolution: 6 x 11
Level: 8, resolution: 3 x 6

(a) Image showing the down sampling and bandpass filtering.

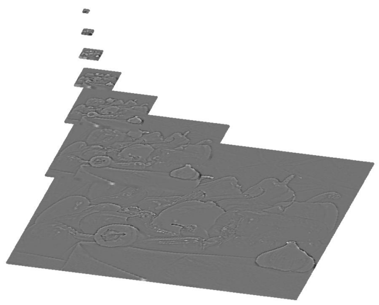


(b) Image pyramid containing images at various scales, down sampled by a factor of 2.

Figure 6: Maximum pyramid


Level: 1, resolution: 384 x 512
Level: 2, resolution: 192 x 256
Level: 3, resolution: 96 x 128
Level: 4, resolution: 48 x 64
Level: 5, resolution: 24 x 32
Level: 6, resolution: 12 x 16
Level: 7, resolution: 6 x 8

(a) Image pyramid consisting image at various scale, down sampled by a factor of 2.



(b) Image pyramid consisting image at various scale, down sampled by a factor of 2.

Figure 7: Bandpass pyramid

5 Gaussian Scale-space

The Gaussian scale-space method is a method to determine local maxima and minima in the image.

5.1 Exercise 8

I used three different scales and t0 values in the gaussian scale space function and could see that for the 1st run when the nscale =300 and t0=18, the size of the gaussian kernel was 17.32 and we can see many local maxima and minima.

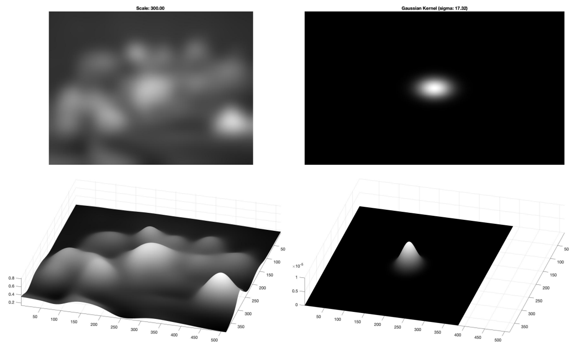


Figure 8: scale space representation with nscale=300 and t0=1.

For nscale=600 and t0=3, the kernel size is larger(42.43) and we can see fewer local maxima and minima's as most of the smaller edges are blurred out.9

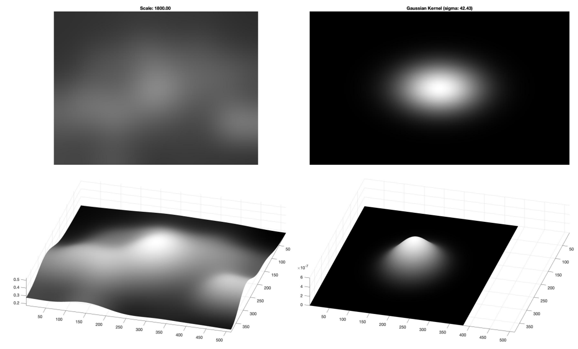


Figure 9: scale space representation with nscale=600 and t0=3.

Using, nscale=400 and t0=2, the scale size was 800 and the kernel size was 28.28. We can see more number of local maxima and minima's when compared to the previous example where the scale was 1800 but fewer maxima and minima's in comparison to the scale size of 300.10

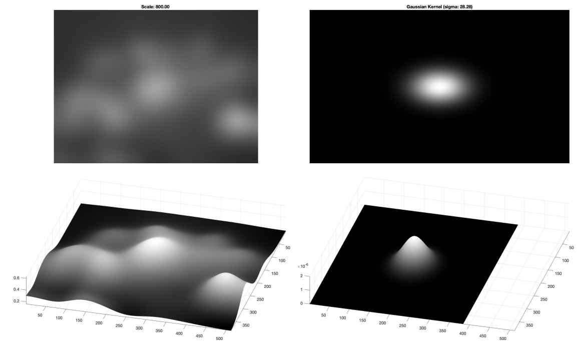
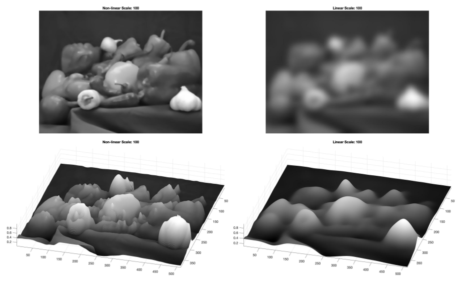


Figure 10: scale space representation with nscale=400 and t0=2.

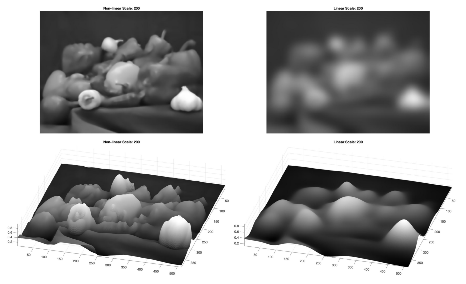
6 Non-linear Scale-space

In this method the variation of the parameters t0 and nscale results also the same change in the output. But the homogenity of the scaled regions are controlled by other 2 parameters $(perc, psi_c)$, which determine the conductivity. As it can be seen in the figures, the scaling does not go further, if here the further regulations are not met. On the other hand, this method results more accurate output.

6.1 Exercise 10



(a) t0=1, perc=0.7,nscales=100, psi$_c$ = 3



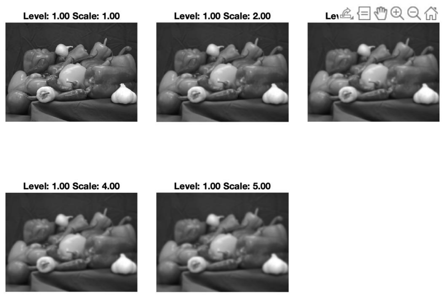
(b) t0=1, perc=0.8,nscales=200, psi$_c$ = 4

Figure 11: Non-linear scale space

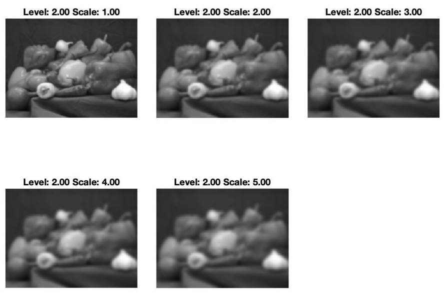
7 Global Exercise

Created a scale-space gaussian pyramid, in which each level contains multiple scales(for e.g 5) of scale-space decomposition. I've initialised image for each new level by sub-sampling the image at middle scale of the previous level. The code for $task12_doscalespacepyramid.m$ can be found in scale space folder.As a result, i observed that the images at level subsequent levels and for different scales are smoothed using gaussian kernel due to which we loose information.

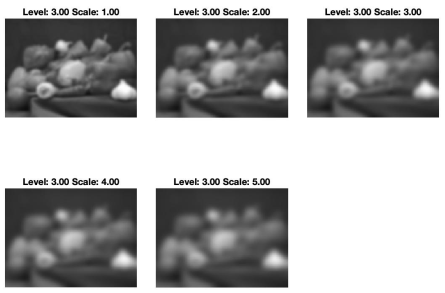
7.1 Exercise 12



(a) Level 1 of pyramid at 5 scales.



(b) Level 2 of pyramid at 5 scales.

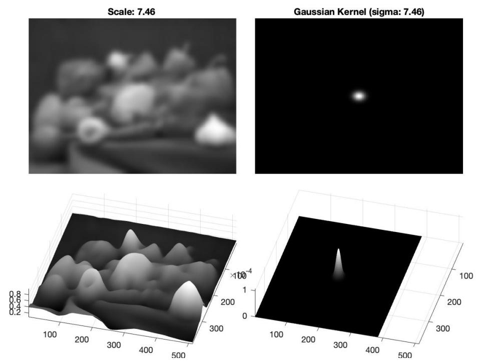


(c) Level 3 of pyramid at 5 scales.

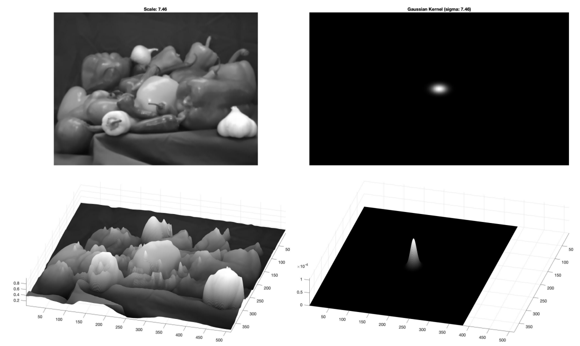
Figure 12: Three levels of gaussian pyramid, each having 5 scales.

7.2 Exercise 13

For exercise 13, i used the scale-space and non-linear scale-space codes to create decomposition for a fixed number of scales (10), including jumps in levels.Also, I used the formula given in 13c for calculating Gaussian standard deviations. As a result, i observed that the linear scale-space blurs out the edges and important information in the image where as the non-linear scale space preserves the edges and image is not blurred out, for a given sigma.



(a) scale-space decomposition for a fixed scales.



(b) Non-Linear scale-space decomposition for a fixed scales.

$$\sigma_j(o,s) = \sigma_0 \cdot 2^{o+s/S},$$

$$o \in [0,\ldots,O-1], s \in [0,\ldots,S-1],\ \ j \in [0,\ldots,N-1]$$

(c) Formula used for calculating the gaussian standard deviations.

Figure 13: Results for exercise 13.