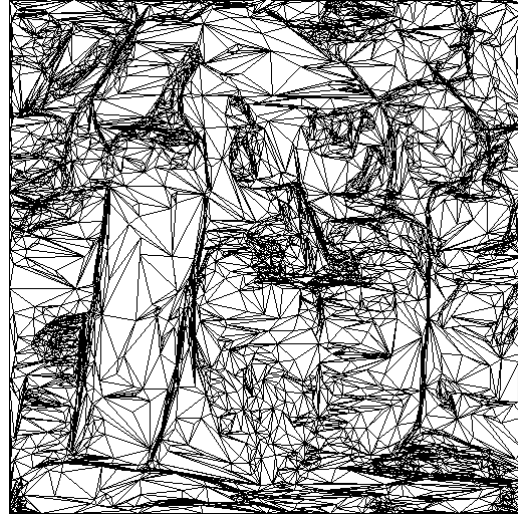


Figure 6.14 - Reconstructed image of Peppers. Reconstructed image and corresponding triangular meshes from (a-b) level=6, (c-d) level=4 and (e-f) level=2.



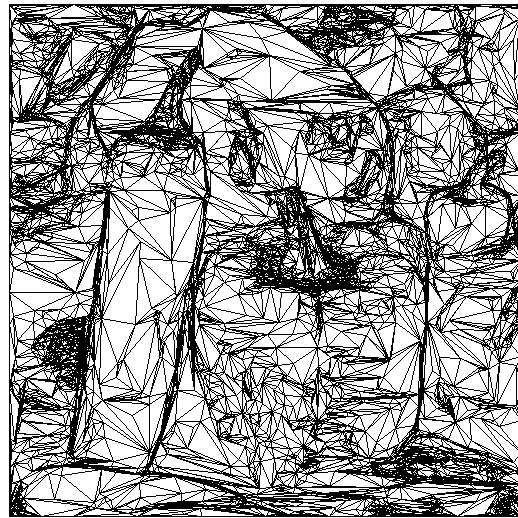
(g)



(h)



(i)



(j)

Figure 6.14 (continued) - Reconstructed image of Peppers. Reconstructed image and corresponding triangular meshes from (g-h) level=0, and (i-j) level=-2.

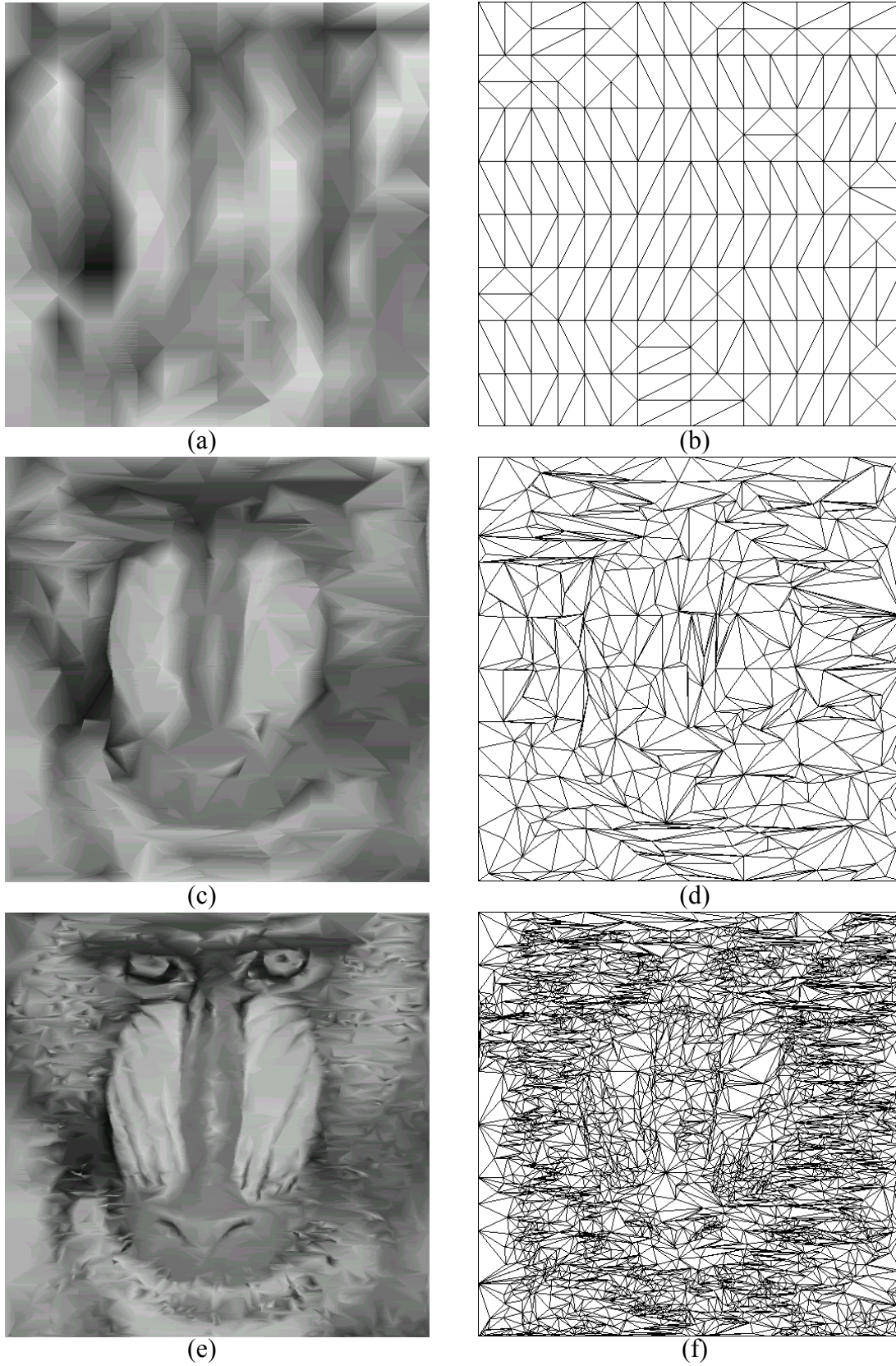
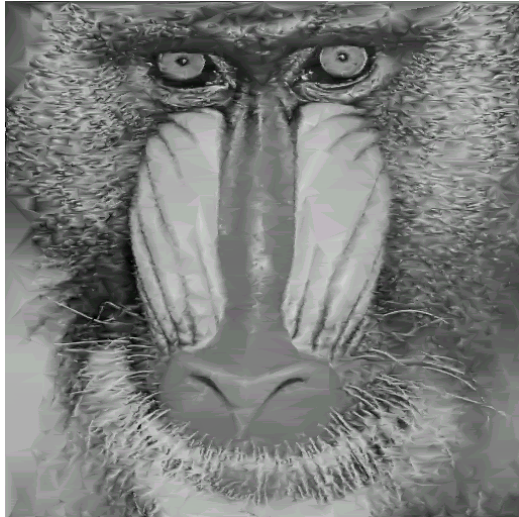
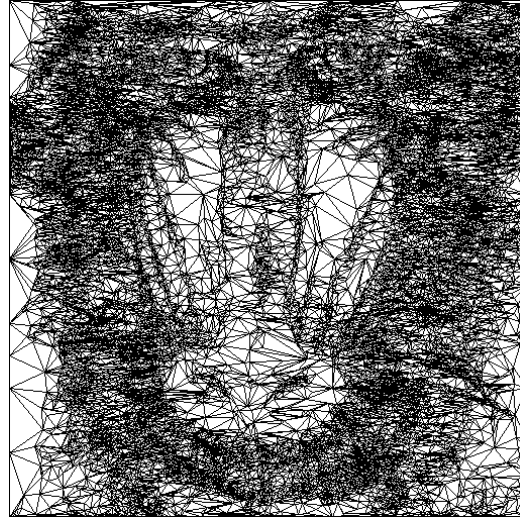


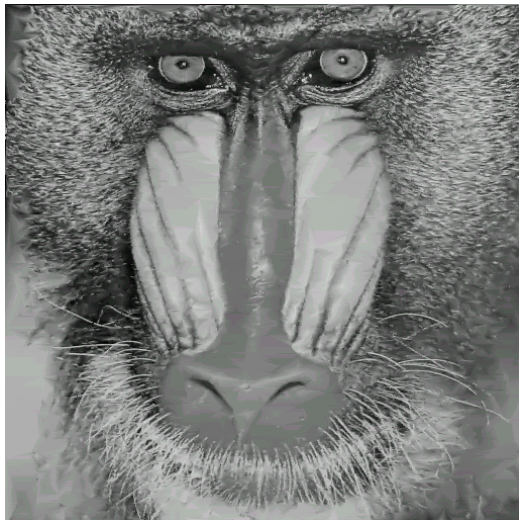
Figure 6.15 - Reconstructed image of Mandrill. Reconstructed image and corresponding triangular meshes from (a-b) level=6, (c-d) level=4, and (e-f) level=2.



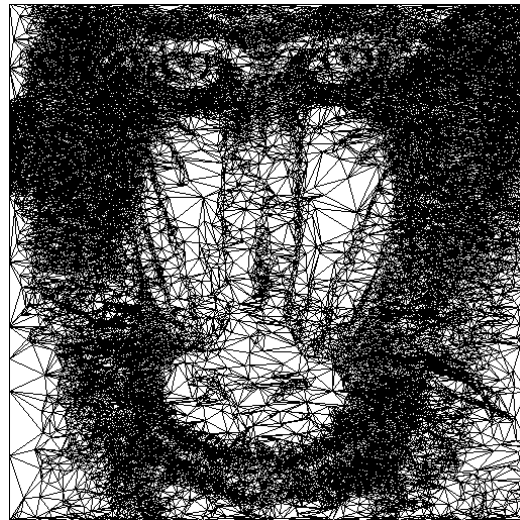
(g)



(h)



(i)



(j)

Figure 6.15 (continued) - Reconstructed image of Mandrill. Reconstructed image and corresponding triangular meshes from (g-h) level=0, and (i-j) level=-2.

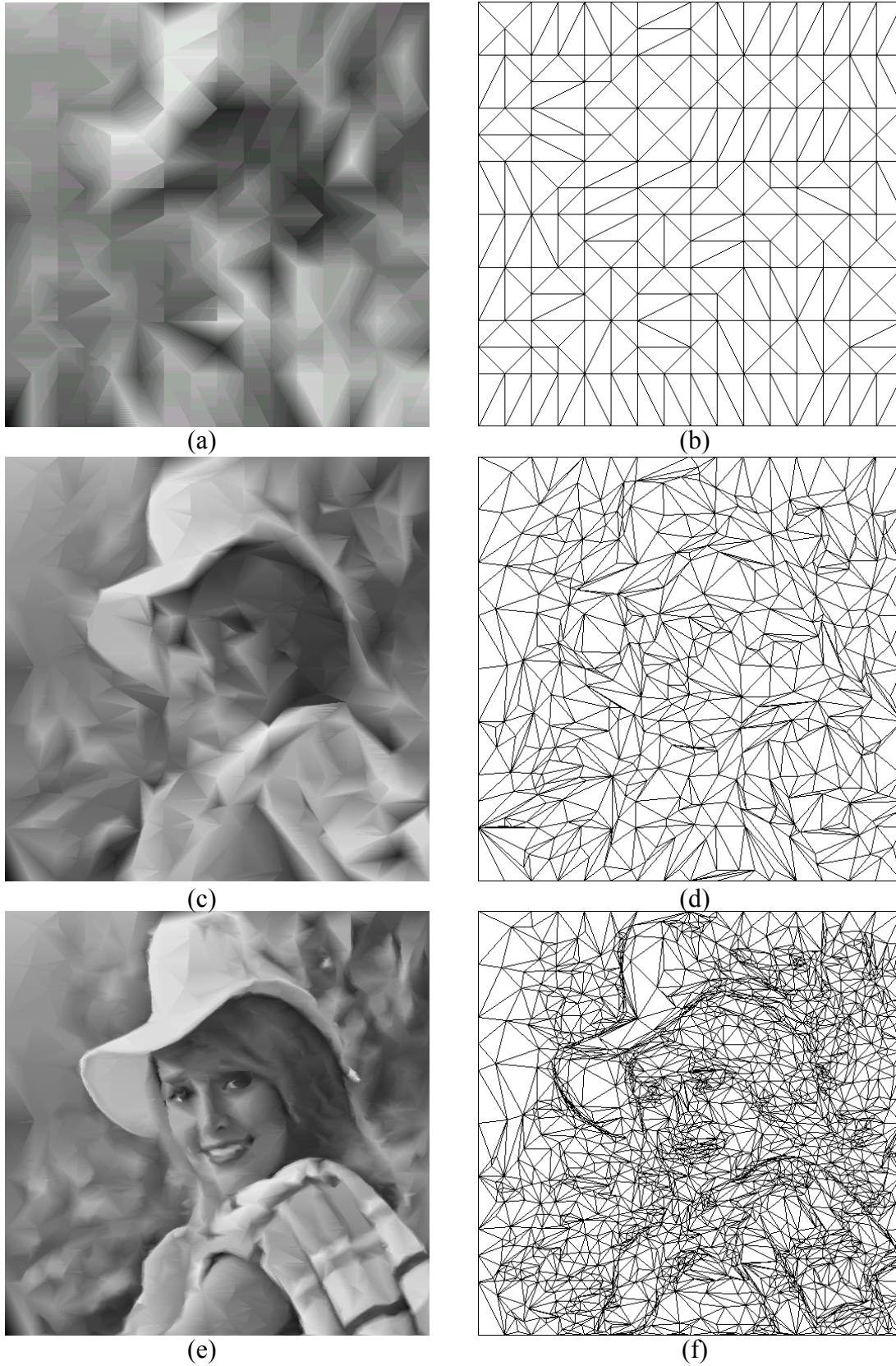
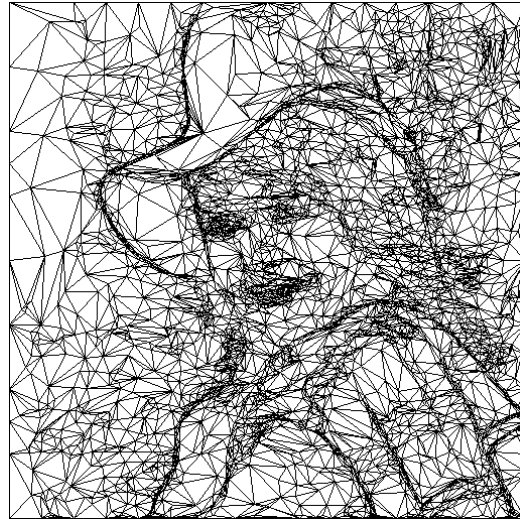


Figure 6.16 - Reconstructed image of Elaine. Reconstructed image and corresponding triangular meshes from (a-b) level=6, (c-d) level=4, and (e-f) level=2.



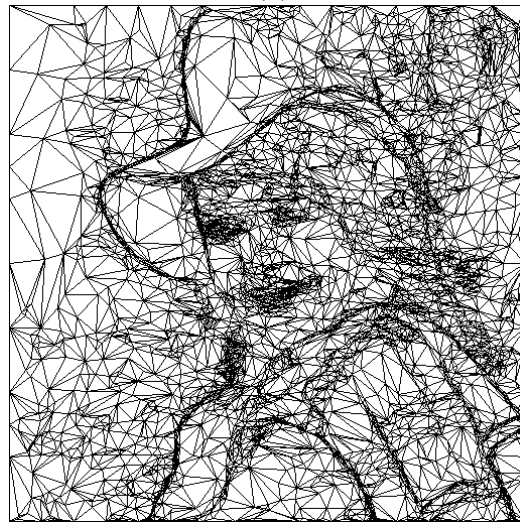
(g)



(h)



(i)



(j)

Figure 6.16 (continued) - Reconstructed image of Elaine. Reconstructed image and corresponding triangular meshes from (g-h), level=0 and (i-j) level=-2.

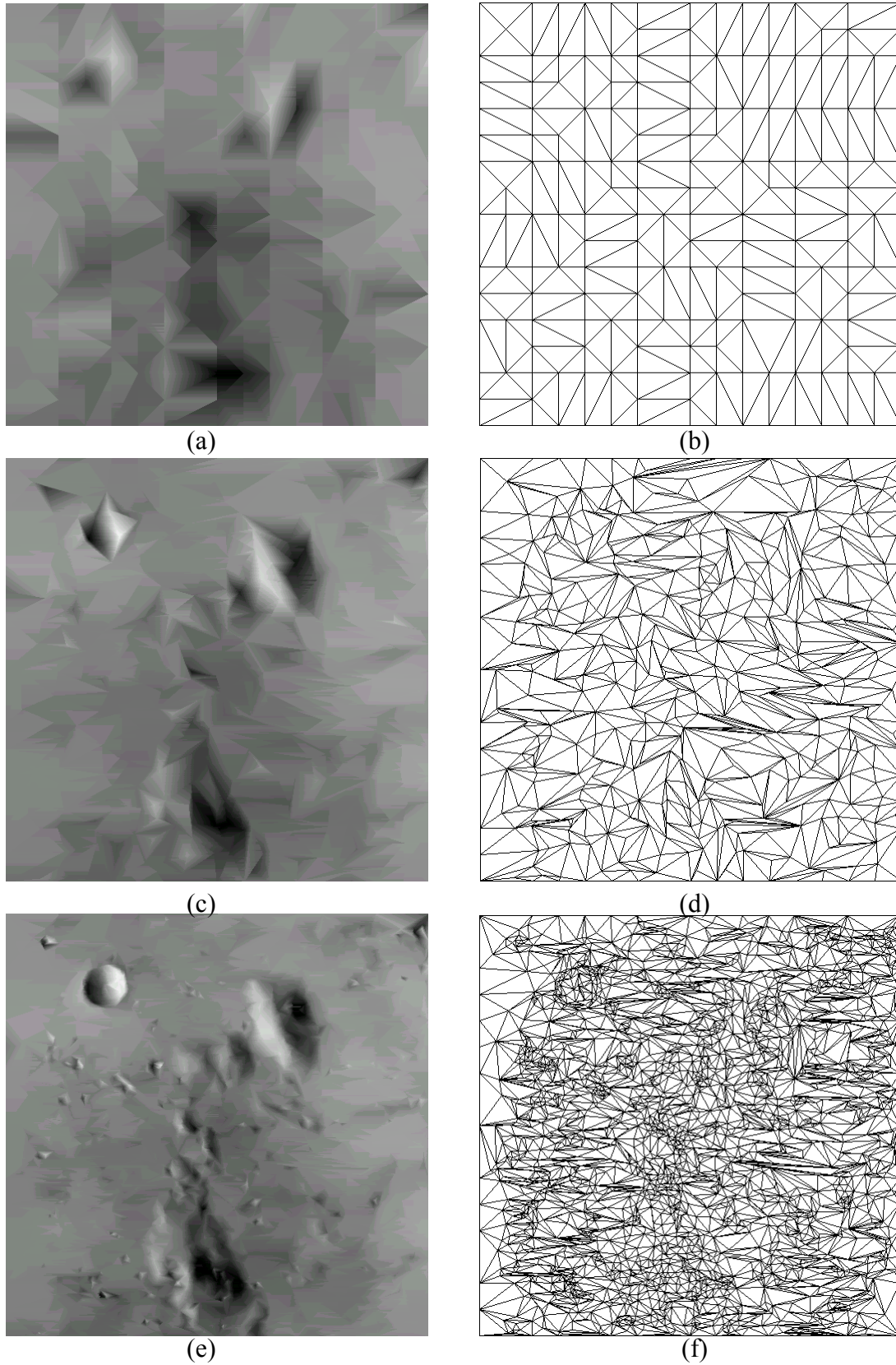
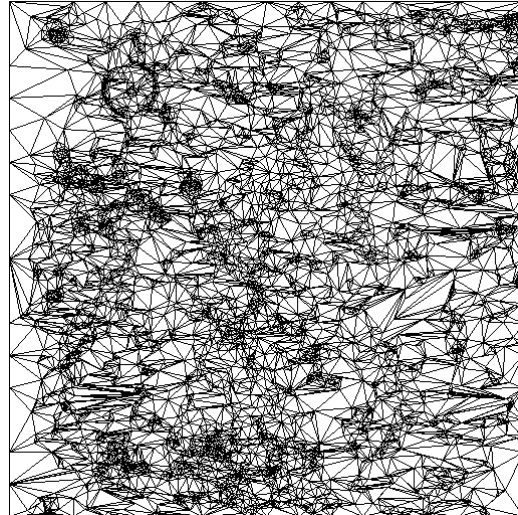


Figure 6.17 - Reconstructed image of Moon. Reconstructed image and corresponding triangular meshes from (a-b) level=6, (c-d) level=4, and (e-f) level=2.



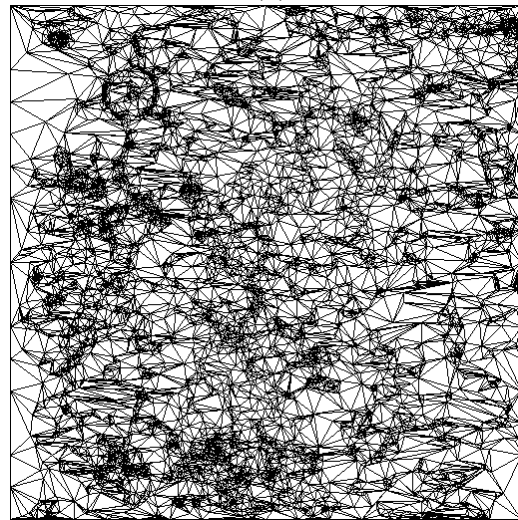
(g)



(h)



(i)



(j)

Figure 6.17 (continued) - Reconstructed image of Moon. Reconstructed image and corresponding triangular meshes from (g-h) level=0, and (i-j) level=-2.



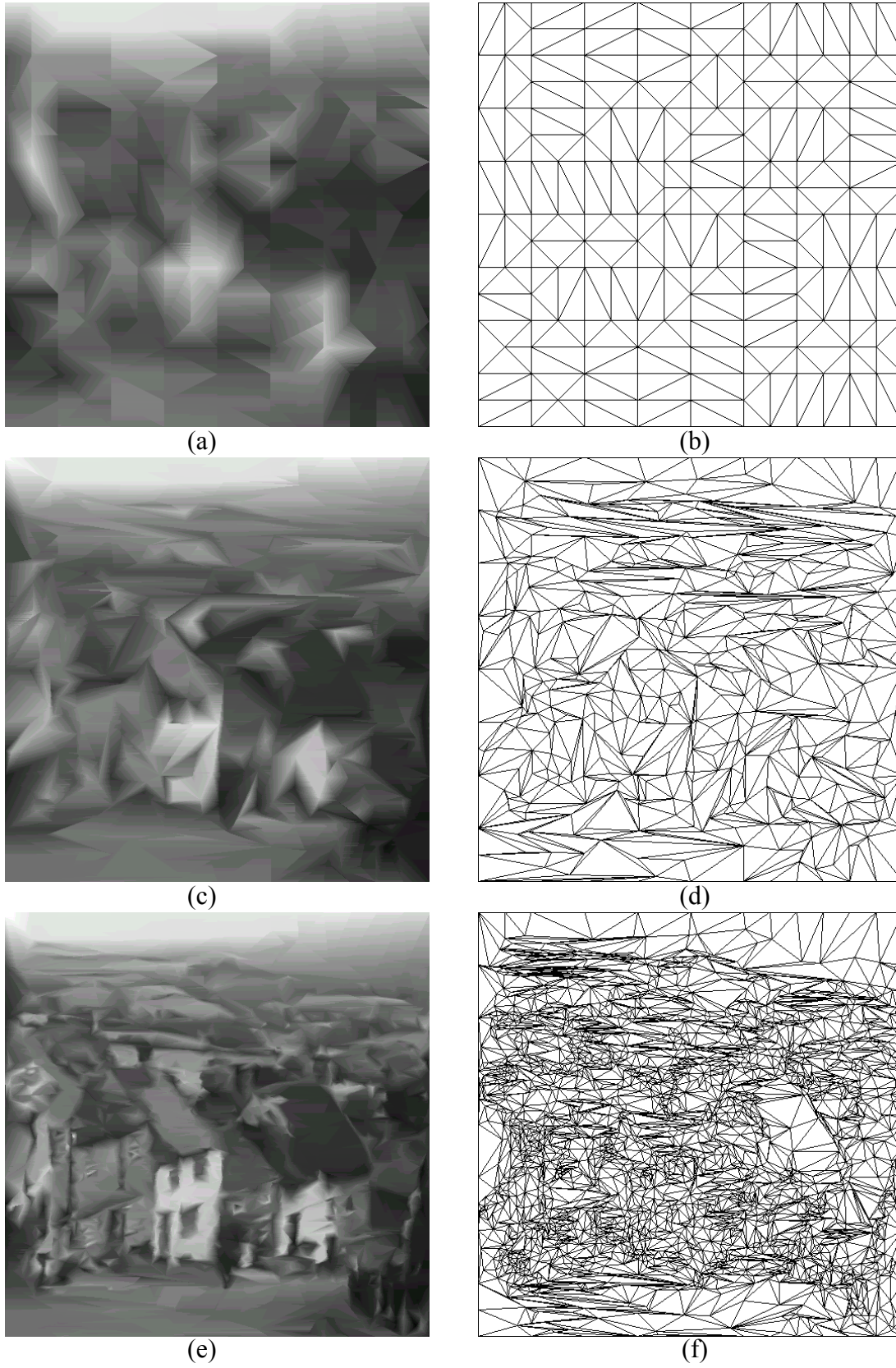
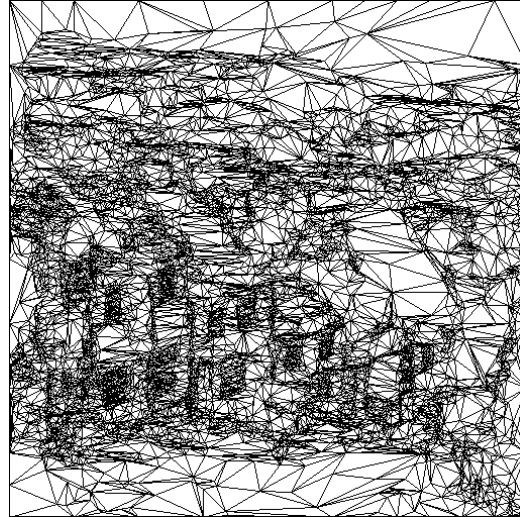


Figure 6.18 - Reconstructed image of Goldhill. Reconstructed image and corresponding triangular meshes from (a-b) level=6, (c-d) level=4, and (e-f) level=2.



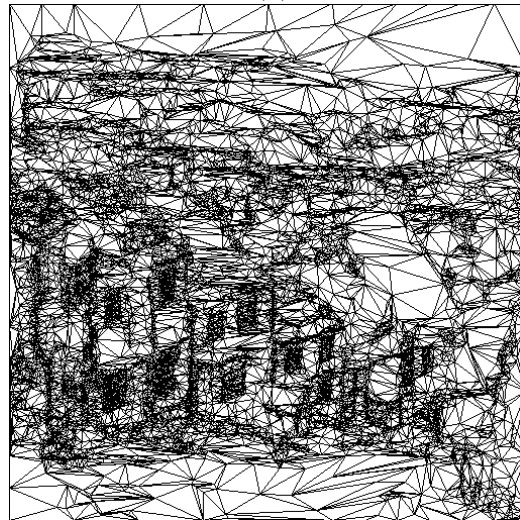
(g)



(h)



(i)



(j)

Figure 6.18 (continued) - Reconstructed image of Goldhill. Reconstructed image and corresponding triangular meshes from (g-h) level=0, and (i-j) level=-2.

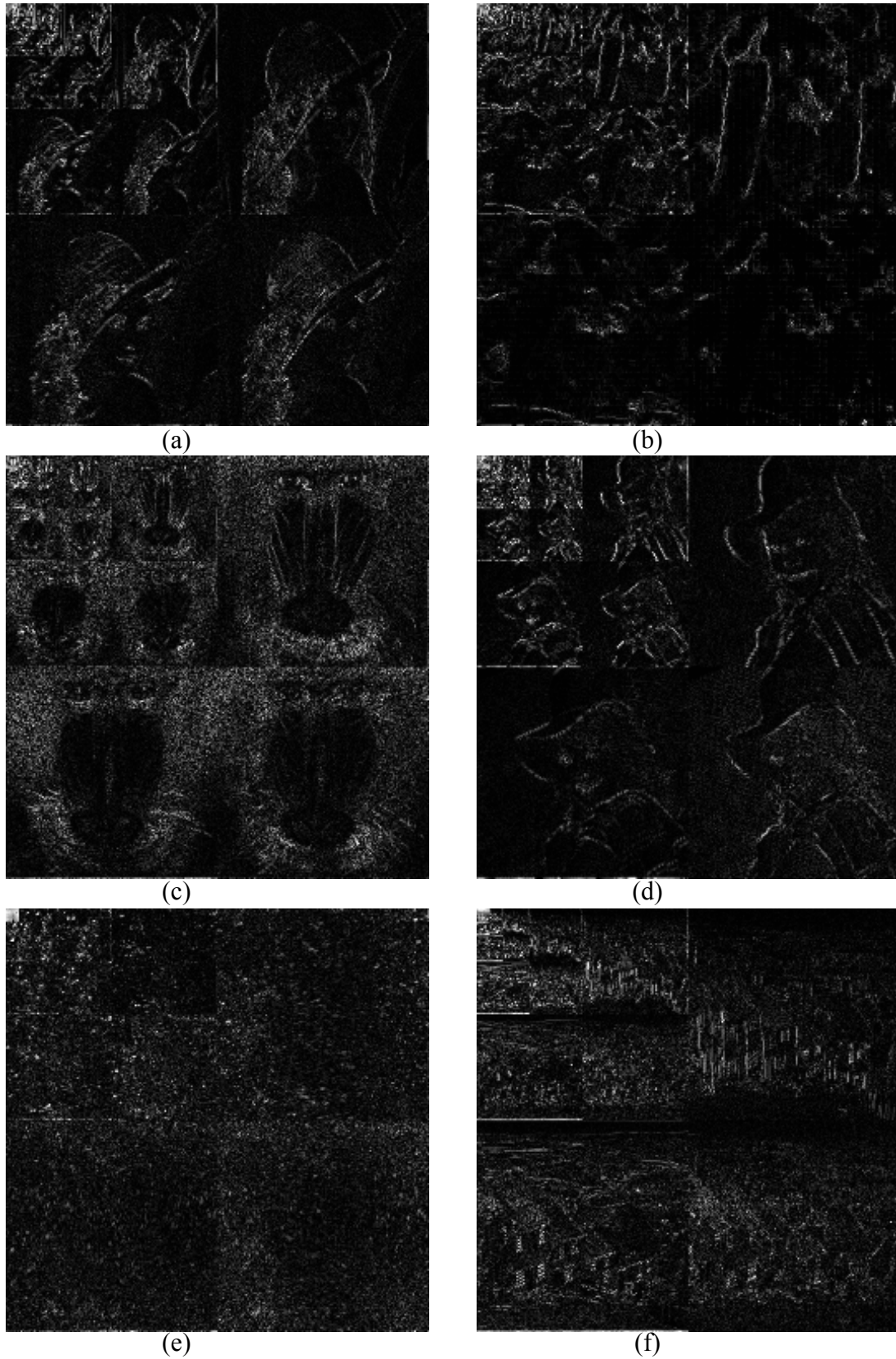


Figure 6.19 – Discrete wavelet transform results (a) – (f) Discrete wavelet transform at level 5 from (a) Lena, (b) Peppers, (c) Mandrill, (e) Elaine, (f) Moon and (g) Goldhill images.

One of the main characteristics of the Lena and Peppers images is that they contain many regions of approximately the same intensity value. Since there is a lot of this redundancy in data, the algorithm could gain a high compression ratio without substantially degraded image quality. It is found that, with less than nine thousand triangles, the triangular mesh can successfully represent the Peppers image at quality of higher than 30 PSNR. Furthermore, at the data compression ratio of 1:32, the number of required vertices is less than two percent of the original data.

Notice that the bodies of Lena and the Peppers are represented by many big triangles, while the edges are approximated by many long, sliver triangles. By comparing the reconstructed image and wavelet coefficients image, it can be observed that most of these vertices that lie on the location where large wavelet coefficients can be found. Likewise, the hat in the Elaine image is represented by large triangles instead of long, sliver triangles that are usually used to approximate the edges.

Figures 6.20 and 6.21 compare the PSNR and the number of triangles of all the test images. It can be observed that the Mandrill image is one of the hardest images for the triangulation mesh to represent well. This is because it contains a lot of high frequency components, which is difficult to approximate with linear segments. Furthermore the animal has a long thin beard that requires many refined triangles to approximate these curves well. Unfortunately, it is found that as many as ninety thousand triangles can only achieve 25.47 dB in PSNR, which is the lowest among the test images. However, the approximation algorithm still gains some vertex reduction. It requires only 17.25 percent of original vertices to reconstruct the Mandrill image at this quality. It can be imagined that the worst-case scenario for the approximation algorithm would be an image with alternating black and white values throughout the image.

Like the Lena image, a triangular mesh with less than five thousand vertices can represent the Elaine image at higher than 30 dB in PSNR. Most of the triangles are used in refining the hair, eyes, mouth and the edges of both images. Figures 6.20 and 6.21 show that at approximately the same compression ratio, the triangulation meshes can achieve about the same image quality in PSNR for both images.

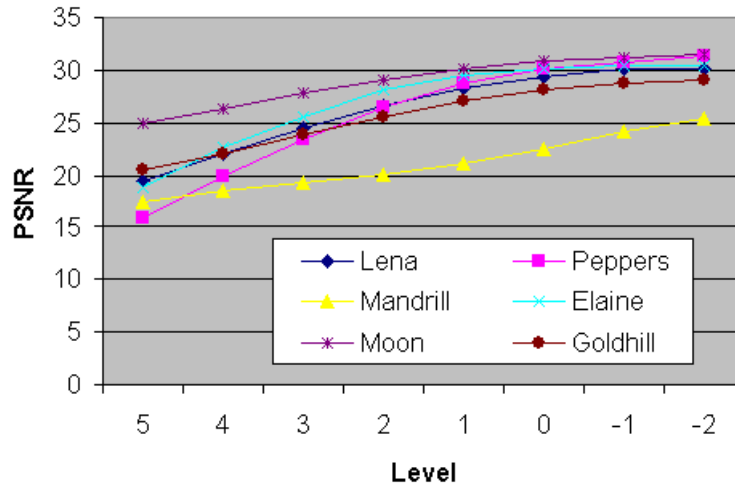


Figure 6.20 - PSNR comparison on all images at different level.

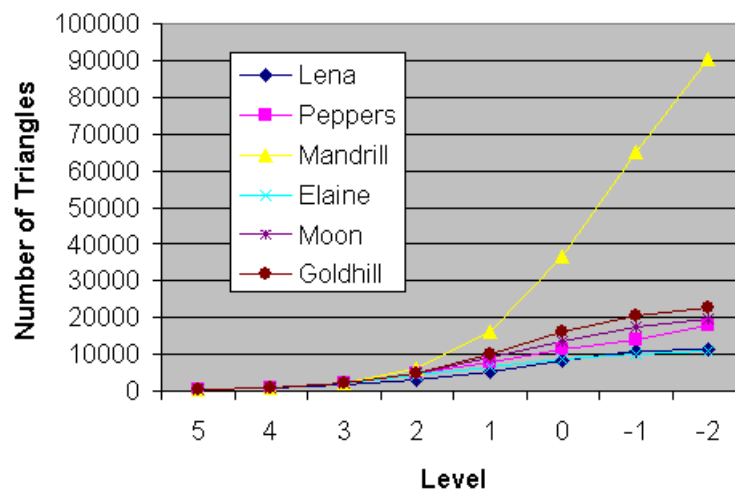


Figure 6.21 – Number of triangles comparison on all images at different levels.

After some analysis, it is found that, unlike other test images, most intensity values for are clustered near the middle range. This can be verified from the histogram of all test images shown in Figure 6.22. It can be clearly seen that 80% of the intensity values lie in the range of 90 to 160. The red line shows the range of this area. With this narrow dynamic range, the Moon image is expected to result in a good compression ratio as well as good PNSR because most of the absolute error will be very small. Since PSNR is computed using the  $L^2$  norm, the reconstructed image is usually influenced by the large absolute error rather by many small magnitude errors. Another observation for the Moon image is, unlike the Lena and Peppers images, the vertex locations are more scattered over the parameter domain than clustered at specific locations. For the Lena and Elaine images, the majority of the vertices can be found near the hair, eyes, mouth and edges. However, the Moon image does not show many obvious edges, but many impulses, which are small in area, but large in amplitude.

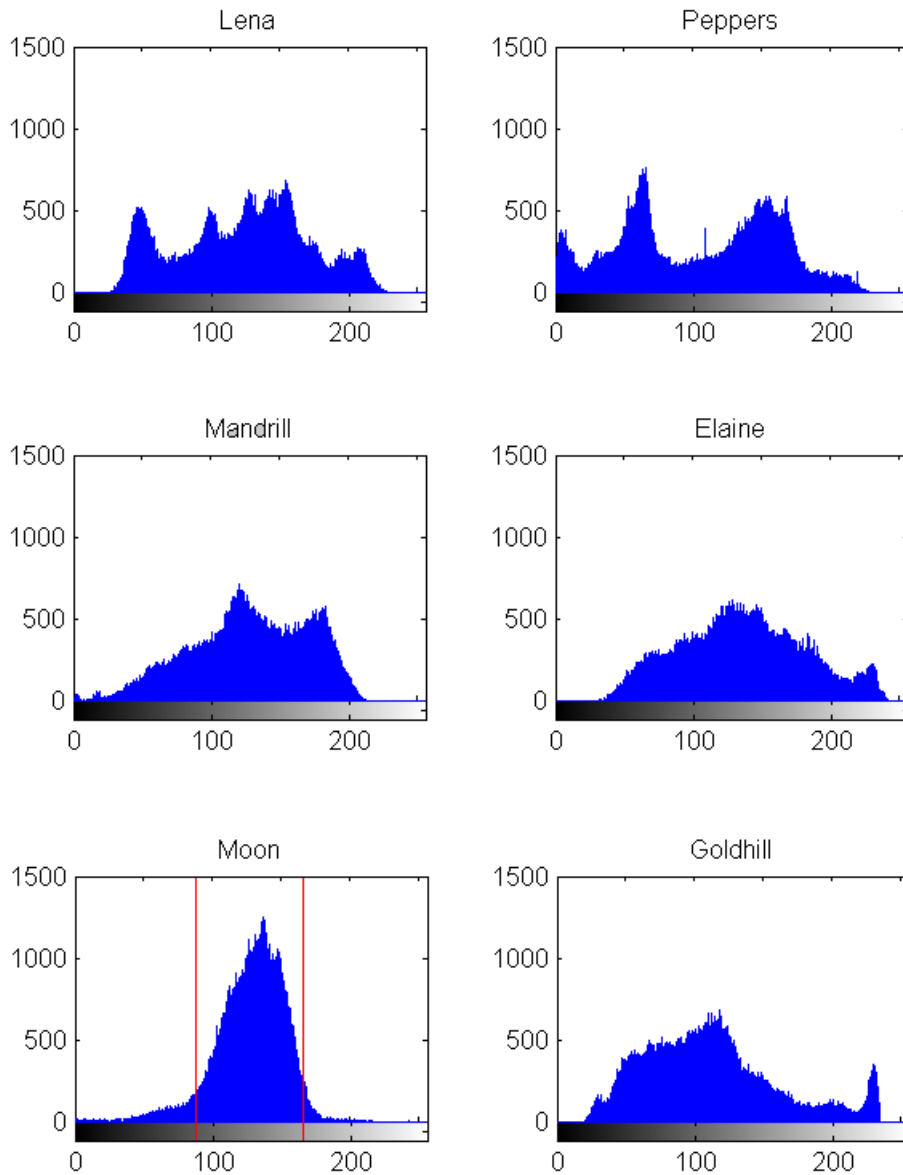


Figure 6.22 – Histogram of the test images

Therefore the resulting triangulation is filled mostly with more regular triangles than those found in the Lena or Peppers images. Although it is expected that these regularly scattered triangles tend to produce poorer results than those with a combination of fat and thin triangles, the Moon image still achieves a good PSNR value because of its low dynamic range.

The last test image is the Goldhill image, whose low-frequency component can be found in the sky and the road while high frequency details are located in the houses and trees. Like the Lena, Peppers and Elaine images, most of the vertices are found along the prominent edges. However, like the Mandrill image, the algorithm did not successfully achieve high

image quality and compression ratio. At level 1, it reached 27 dB in PSNR with about ten thousand triangles. However, at level -2, it reached only 29 dB in PSNR with twice the number of triangles. One of the reasons is, as the triangles get smaller, each triangle refinement reduces the sum of squared error only slightly. Unlike the early iterations where the triangles are still large and each refinement could still gain large advantages, later iterations only result in a small reduction in error.

## 6.4 Applications

There are many applications that can take advantage of the triangular mesh representation. In addition to its multiresolution property, some of the advantages of using triangular meshes are region-of-interest refinement, object detection and segmentation, which will be discussed in Sections 6.4.1, 6.4.2 and 6.4.3 respectively.

### 6.4.1 Region of Interest (ROI) Refinement

One of the other advantages of using triangular meshes is the ability to perform region-of-interest (ROI) refinement. This property helps the application to adjust the level of control only on some desired regions. For example, in the face recognition application where only face details are important, it might be more favorable to represent face regions with higher quality than the entire image scene. Given some example masking as shown in Figure 6.23. These masks were obtained manually. Figure 6.24 and 6.25 show some examples of ROI refining on Lena and Peppers image.



Figure 6.23– Masking for the Lena and Peppers images.

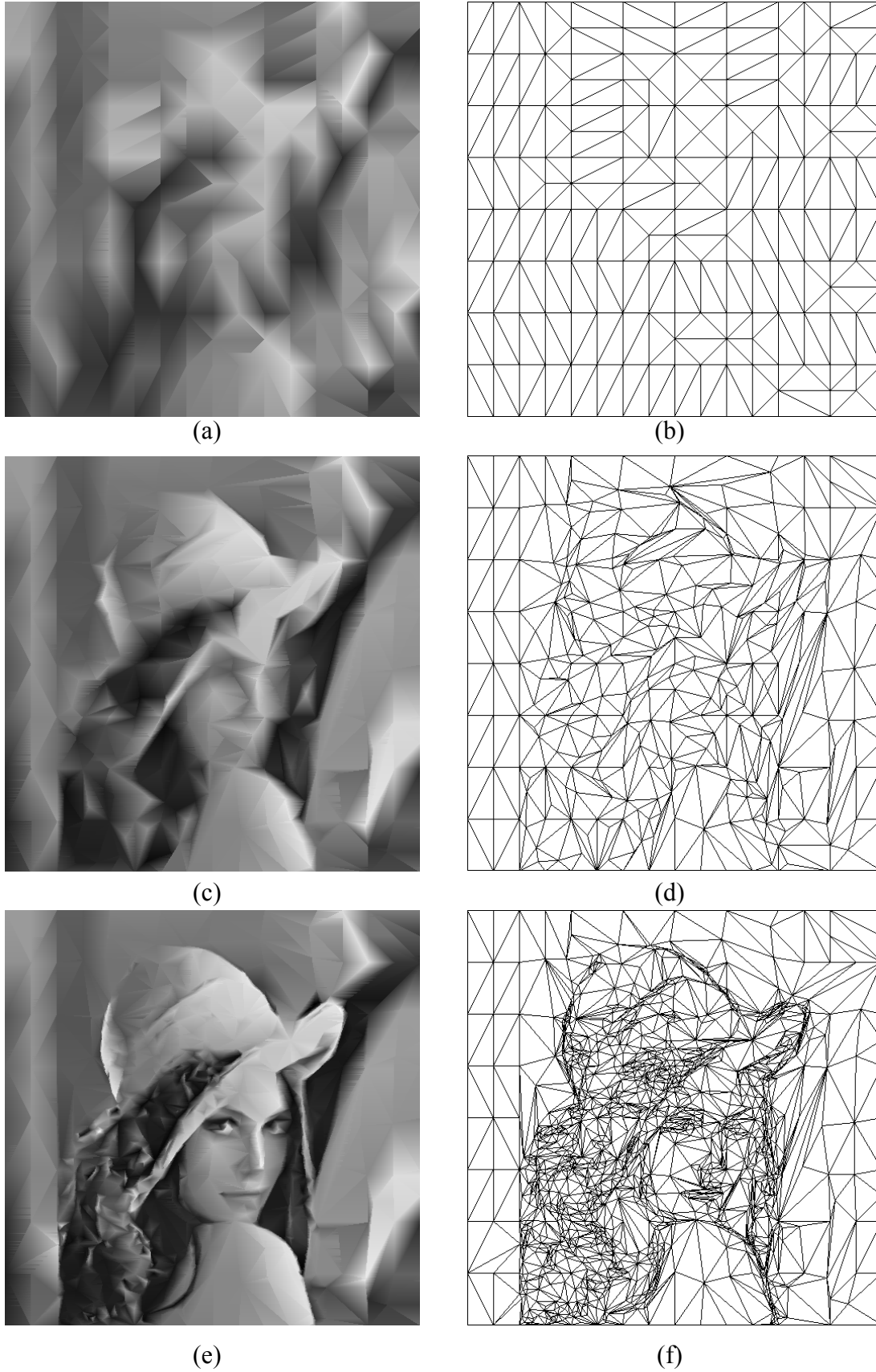


Figure 6.24 – Triangulation-based refinement based on assumed masking.



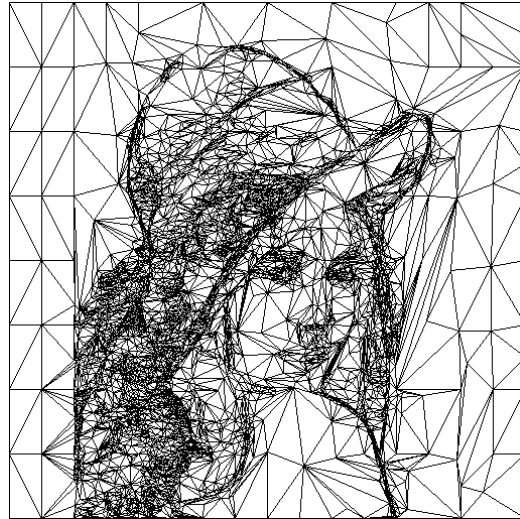
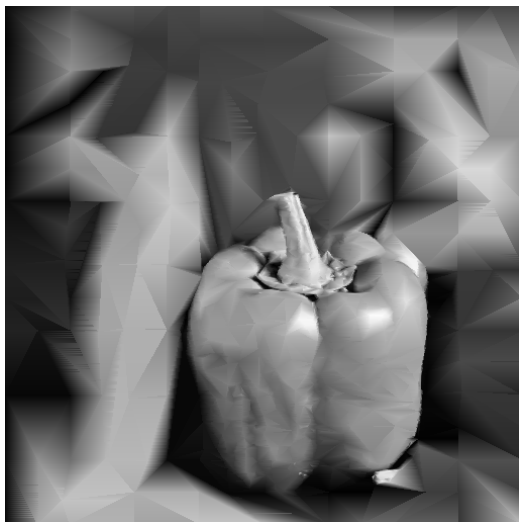
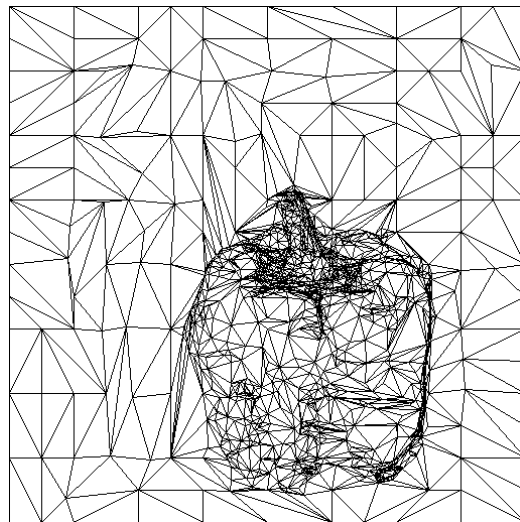


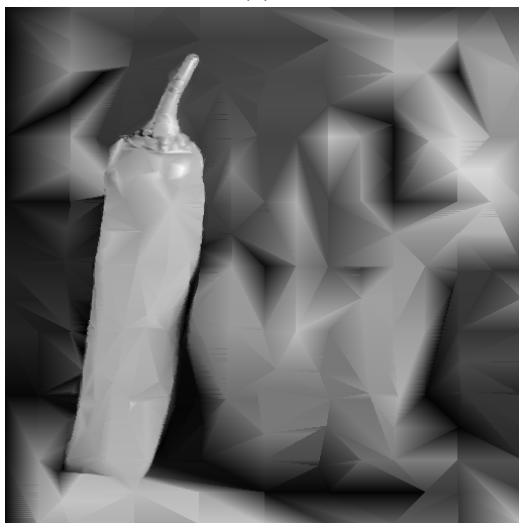
Figure 6.24 (Continued) – Triangulation refinement bases on masking shown in Figure 6.23.



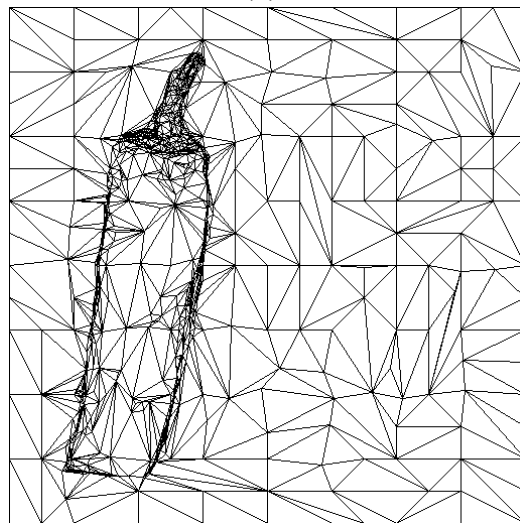
(a)



(b)



(c)



(d)

Figure 6.25 – Triangulation-based refinement of two different Peppers using the assumed masking.

### 6.4.2 Eye Detection

Other application that can take advantage of triangular mesh representation is object detection. For example, to recognize a face, it is desirable to detect the location of the eyes so that further information about the eyes can be extracted. Basing on a simple algorithm shown in Figure 6.26, the algorithm takes less than 10 seconds to detect the eyes.

First, the input image is converted into a triangular mesh representation. Next, since the eyes usually contain high frequency components, which are normally represented by small triangles, the algorithm can take advantage of this property. The algorithm can use size filtering to screen out the large triangles from the possibility. Furthermore the high contrast regions of the eyes are usually represented by black and white triangles. Therefore, triangles, which are not located near to any of its contrast triangles, will be filtered out. Assuming that the two eyes are located within some expected distance, the algorithm can use this property to screen out isolated triangles that do not have any triangles located nearby. The results at this point are shown in Figure 6.27. The red and blue triangles represent the white and black regions respectively.

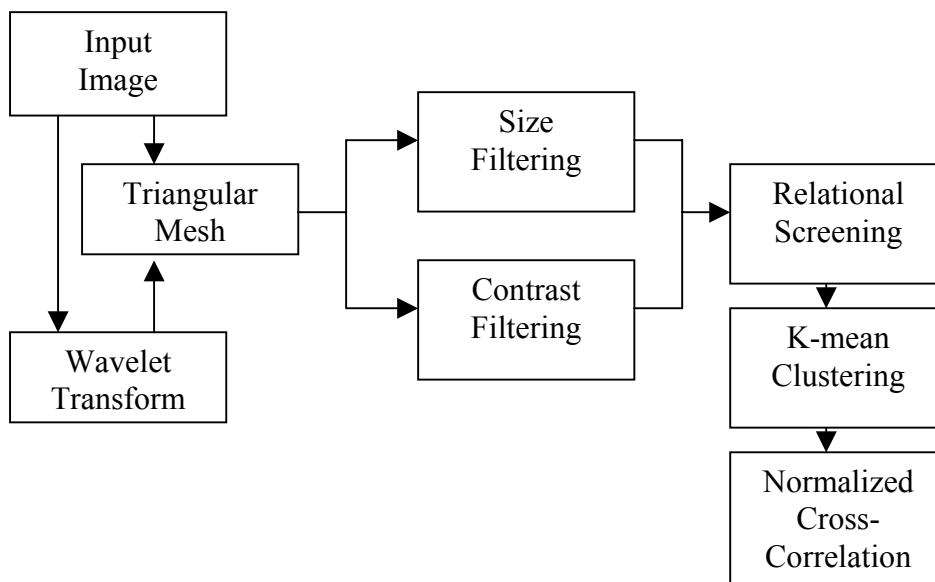


Figure 6.26 – Eye detection algorithm using triangular mesh representation as input.

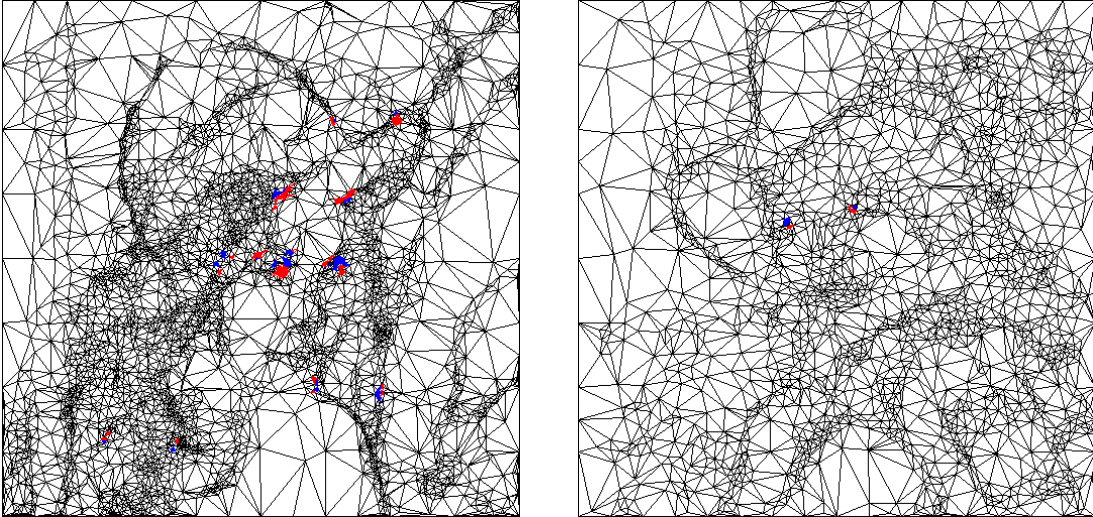


Figure 6.27– Results of eye detection on the Lena and Elaine images after relational screening.

Next, to confirm that the regions are almost identical, normalized cross-correlation is needed to prove their similarity. However, since the current results contain too many triangles, K-means clustering [Bovik 00] is used to cluster these triangles into regions of interest so that less calculation is required. The normalized cross-correlation formula used to compare two regions is

$$\gamma(u, v) = \frac{\sum_{x,y} [f(x, y) - \bar{f}_{u,v}] [t(x-u, y-v) - \bar{t}]}{\left\{ \sum_{x,y} [f(x, y) - \bar{f}_{u,v}]^2 \sum_{x,y} [t(x-u, y-v) - \bar{t}]^2 \right\}^{\frac{1}{2}}} \quad (6.1)$$

where  $f(x, y)$  are the possible regions of the eyes and  $t(x, y)$  are the eye templates chosen from the possible regions of the eyes. It is found that the size of the template is also one of the important factors to match the eyes. It is better to compare only the white and black regions of the eyes instead of the regions that include the eyelash and eyebrow. These regions are shown in Figure 6.28. Figure 6.29 shows the final results of the eye from four different images. The result is very accurate except for the Lisa image. This is because the lip and teeth in Lisa image have so much similarity that it produces false alarms.

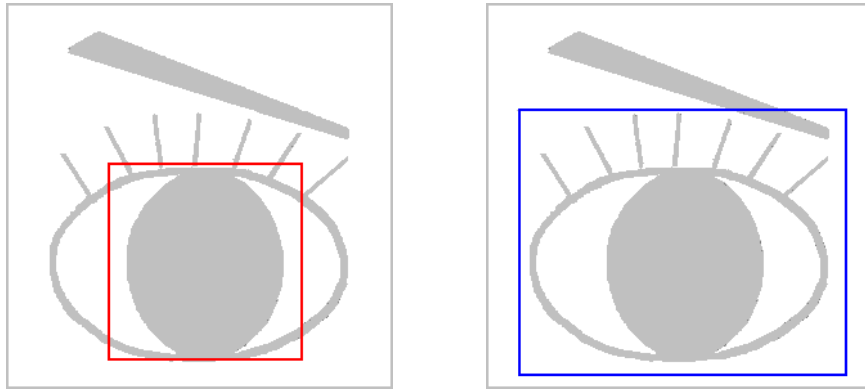


Figure 2.8 – Comparison of the eye templates. (a) Template that includes black and white regions of the eyes. (b) Template that includes eyelashes and eyebrow.



Figure 6.29 – Results of eye detection. The possible locations of the eyes obtained from the images of (a) Lena, (b) Elaine, (c) Zelda, and (d) Lisa.

### 6.4.3 Segmentation

The other application of triangulation mesh is segmentation. Figure 6.30 below shows the result of segmentation of Peppers image obtained from some simple algorithm for triangulation mesh. First, for each triangle, the average intensity is calculated. Next this value is compared to some three thresholds so that each triangle will be assigned to one of the four possible intensity region. The results are obtained with different mesh regularities,  $\alpha$ , of 0.0, 0.1, 0.2 and 1.0.

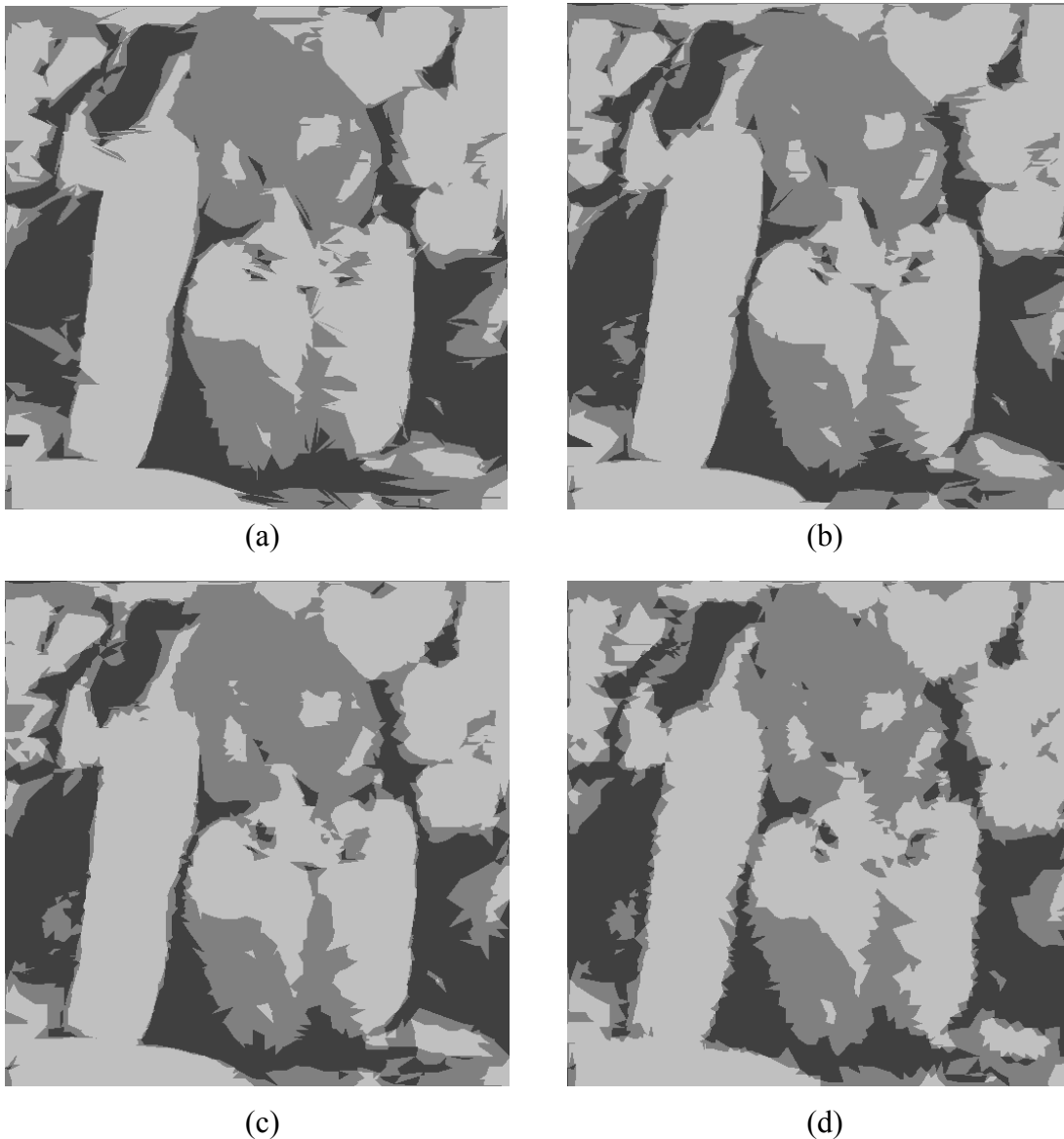


Figure 6.30 – Segmentation of Peppers image. Mesh regularity value of (a)  $\alpha = 0.0$ , (b)  $\alpha = 0.1$ , (c)  $\alpha = 0.20$ , and (d)  $\alpha = 1.00$  respectively.

It can be observed that, at  $\alpha = 0.0$ , the segmented regions contain many inappropriate sharp spikes at the location of edges. The reason for this is that it is possible for the triangulation in the early stage to choose the thin sliver triangles with minimum wavelet or error energy, which cannot be further divided in the next levels. These spikes can be found less in the results from  $\alpha = 0.1$  and  $0.2$ . For the case of Delaunay triangulation, where  $\alpha = 1.0$ , it can be easily noticed that there are a lot of thick spikes along the edges. This is because the Delaunay rule does not allow thin, sliver triangles to exist in the triangulation at all. Therefore, if the vertices do not well lie along these edges, these zigzag divisions will occur along region boundaries.