Comparison of Square-Pixel and Hexagonal-Pixel Resolution in Image Processing

Girish Tirunelveli[1], Richard Gordon[1, 2] and Stephen Pistorius[2, 3, 4]
1 Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Manitoba, R3T2N2, Canada
2 Department of Radiology, University of Manitoba, Winnipeg, Manitoba, R3T2N2, Canada
3 Medical Physics, CancerCare Manitoba, Winnipeg MB R3E 0V9, Canada
4 Department of Physics and Astronomy, University of Manitoba, Winnipeg, Manitoba, R3T2N2, Canada
E-mail: tggirish@mts.net, GordonR@ms.umanitoba.ca, Stephen.Pistorius@CancerCare.mb.ca

Abstract

In most applications of image processing data is collected and displayed in square pixels. Hexagonal pixels offer the advantage of greater rotational symmetry in addition to close packed structure and a nearly circular pixel. We compared the image quality of images using square pixels with that of images employing hexagonal pixels. The comparison was done using various images, each considering a different aspect of geometry (i.e., lines at different angles, curves, etc.). The square pixel images were constructed using the average of a square area of smaller square pixels. Hexagonal pixel images were constructed using two techniques. The first one was called the "two-template approach", wherein two different templates were used to create a close packed hexagonal image from smaller square pixels. The second approach was called the "six-neighbor approach" which creates a rectangular template using the six neighbors of a hexagonal pixel. An Euclidean distance measure was used to compare the square pixel and hexagonal pixel images. A brief explanation of the algorithm and the results are provided in the paper. Based on our results obtained using the Euclidean distance as a quality measure, we conclude that contrary to our intuition and their widespread use in nature (retinas and ommatidia), hexagonal pixels do not offer any advantage over conventional square pixels.

Keywords: Square pixel; hexagon pixel; Euclidean; resolution; two template; six neighbor.

1. INTRODUCTION

For most modern display devices the shape of the pixels are square. It is due to this fact that in most applications of image processing, including computed tomography, data is gathered and arranged in square pixels [1]. The compound eye of insects and crustaceans is made of smaller, simple eye units, called ommatidia. The rhabdome is the common area where light is transmitted to the reticular cells. Each of these cells is connected to an axon and since each ommatidium consists of seven or eight reticular cells, there are these numbers of axons, which form a bundle from each ommatidium. Each ommatidium passes information about a single point source of light. The eyes of strepsipteran insects are very unusual among living insects. Externally they differ from the usual "insect plan" by presenting far fewer but much larger lenses. Beneath each lens is its own independent retina. Anatomical and optical measurements indicate that each of these units is image forming, so that the visual field is subdivided into and represented by "chunks," unlike the conventional insect compound eye that decomposes the visual image in a point wise manner. This results in profound changes in the neural centers for vision and implies major evolutionary changes [2]. The total image formed therefore is a sum of the ommatidia fired. This resultant image can be thought of as a series of dots, just like a computer image is composed of a series of discreet dots (pixels). The more pixels, the better the picture. The ommatidia are more hexagonal than square shaped. It is this natural occurrence that motivated us to hypothesize that hexagonal pixels would provide a better image quality than square pixels.

2. METHODS AND MATERIALS

2.1 Platform

The experiments were done on a Windows98 PC with 96MB RAM and having a single AMD-K6 450MHz processor. The image processing programs were written in Matlab and later converted to Java to permit distributed processing. The images that were used for comparison purposes were assorted test patterns and not partial to any particular geometry. Some test patterns were mathematically created to observe and verify the accuracy of the image comparison algorithms. All images were 256 by 256 hexagonal voxels. Euclidean distance was the image quality used for comparison.

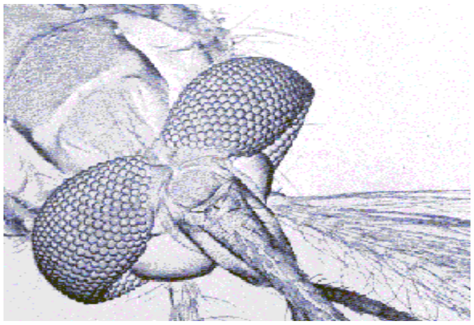

Figure 1. The eyes of an insect such as a mosquito have hexagonally arranged ommatidia. [3]

Figure 2. University of Manitoba Administration Building (courtesy of Prof. W. Lehn, University of Manitoba, reproduced from his Digital Image Processing class).

2.2 Euclidean Distance

Euclidean Distance is defined as the straight-line distance between two points. In a plane with point p1 at (x1, y1) and point p2 at (x2, y2), it is $((x1 - x2)^2 + (y1 - y2)^2)^{1/2}$ [4]. For comparing the difference between two images, the Euclidean Distance is calculated as the square root of the sum of the difference of the squares of pixels. For example if x1, x2, x3… are the pixel values of image1 at position p1, p2, p3… respectively and y1, y2, y3… are the pixel values of image2 at the same positions then the per-pixel normalized Euclidean distance for an n x n picture is calculated as:

$$n^{-2}((x1-y1)^2 + (x2-y2)^2 + (x3-y3)^2 + \ldots)^{1/2}$$

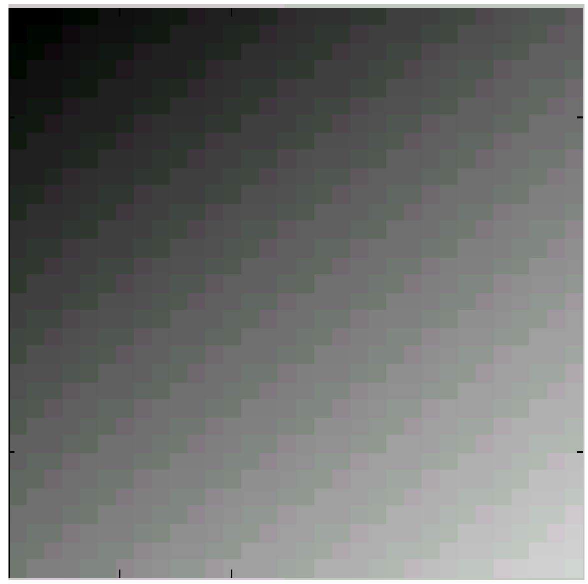

Figure 3. Regular Square Image. This image is constructed by having one value for all voxels in an 8x8 square.

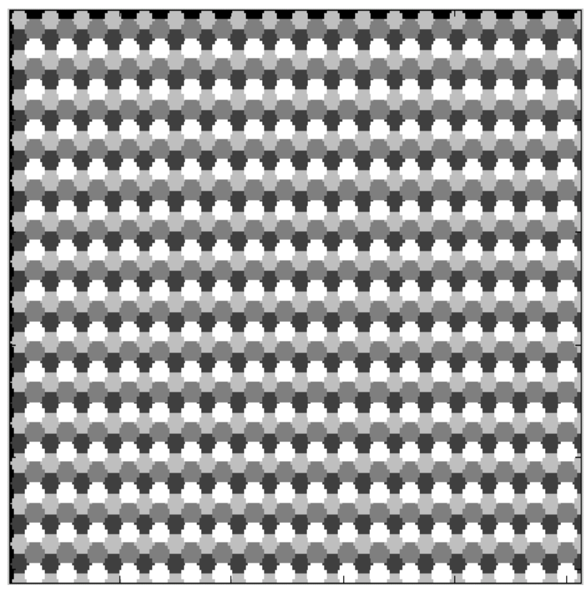

Figure 4. Regular Hexagon Image. This image is constructed by having one value for all voxels in a hexagon of length 4.5.

2.3 Test Images and Process Steps

The images that were used for carrying out the image quality analysis are shown in Figures 2, 3 and 4. These images were reconstructed into squares of 64 pixels and hexagons of 62 pixels in size. The manner in which this is done is explained below.

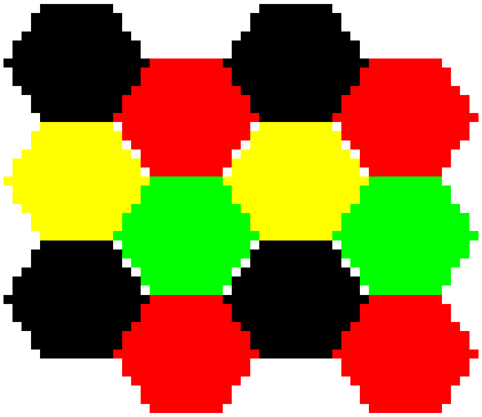

Figure 5. Illustration of the gaps and overlaps which can occur in hexagonal pattern created using a single definition of a hexagon.

The original image shown in Figure 2 was broken into smaller hexagons and each hexagon pixel was given the average value of the voxels that fall in the hexagon. In order to ensure that no two hexagons overlap and no gaps exist between two hexagons (see Figure 5) the hexagons were created using the following two approaches:

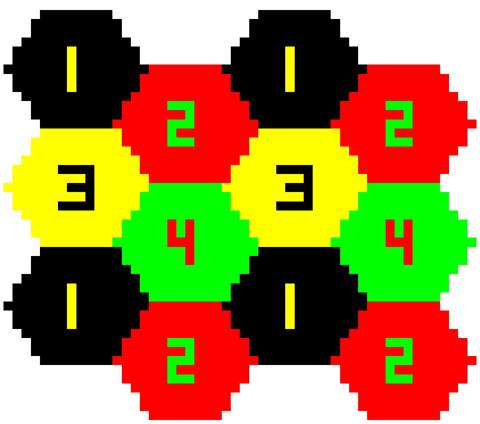

Figure 6. Hexagonal Packed Structure using the two-template approach.

a) Two -Template Approach

In the two-template approach, the hexagons numbered 1 were created first, the hexagons numbered 3 were constructed later using the same formula as that of the hexagons numbered 1 and were vertically displaced by the height of the hexagon. The hexagons numbered 1 and 3 were called odd layered hexagons. Once the entire image was filled with odd layered hexagons, the hexagons numbered 2 were constructed such that they resemble very closely to the hexagons numbered 1 and do not include any voxel already taken by the odd layered hexagons and would include all voxels not considered by the odd layered hexagons. The hexagons numbered 4 were constructed similarly and were displaced by length equal to the hexagon's height from the hexagons numbered 2. The hexagons numbered 2 and 4 were called even layered hexagons. The odd layered hexagons form one template, where as the even layered hexagons form another template. Hence this approach was called the two-template approach.

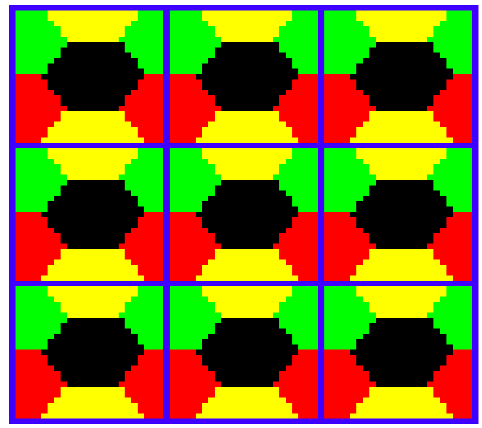

Figure 7. Hexagonal Packed Structure using the six-neighbor approach. Note that the grid is shown only to clearly identify the rectangular template.

b) Six -Neighbor Approach

The six-neighbor approach used a rectangular template by considering one hexagon and part of its six neighbors as shown in Figure 7. The rectangular template was then replicated to tile the entire image.

The two-template approach algorithm in itself ensured that no two hexagons overlapped and that no voxel was left out. However it was computationally too cumbersome. The six-neighbor approach on the other hand offered us the advantage of being computationally and programmatically efficient but required additional logic to ensure that all the voxels in the image were accounted exactly once.

Using both the approaches described above the original images were converted into hexagon pixels. The original images (Figures 2, 3 and 4) were then converted into square pixels of size 8 x 8 and each square pixel was given a value equal to the average of the voxel values that fell in the square. For Figure 8, which is equivalent to Figure 2, the results are shown in Figures 9 and 10 for the square and hexagon pixels respectively.

2.4 Evaluation of the Image Quality

The image quality was evaluated using the Euclidean distance approach. Euclidean distance was calculated between Figure 9 and Figure 8 (original image). This was the square pixel Euclidean distance. Similarly Euclidean distance was calculated between Figure 10 and Figure 8. This was the hexagon pixel Euclidean distance.

The original image was then rotated by 5 degrees. Square pixel image and hexagon pixel image were generated for this new rotated image and their Euclidean distances (by comparing to the rotated image) were calculated as before. This operation was carried out by rotating the original image in increments of 5 degrees up to 90 degrees. A plot of square and hexagon Euclidean distance vs. rotation was generated as shown in Figure 11.

The process was repeated by using Figures 3 and 4 as the original images. Their respective plots of square and hexagon Euclidean distance vs. rotation are shown in Figures 12 and 13.

Figure 8. University of Manitoba Administration Building (courtesy of Prof. W. Lehn, University of Manitoba, reproduced from his Digital Image Processing class)

3. RESULTS AND DISCUSSION

The results of the Euclidean distance evaluation are shown for all three input figures in Table 1. The results shown for the hexagon Euclidean distance are based on hexagon pixels generated using the two-template approach. The results for hexagon pixels generated using the six-neighbor approach were similar to that of the two-template approach.

Figure 9. The 256 x 256 test image shown in Figure 8 was broken into square pixels each of size 8 x 8 (64 pixels).

Figure 10. The 256 x 256 test image shown in Figure 8 is broken into hexagon pixels each of length 4.5 (62 pixels). The above figure was constructed using the two-template approach but even the six-neighbor approach gives the same result.

Contrary to our intuition, the plots shown in Figure 11 of square and hexagon Euclidean distance (inverse to image quality) were very close to each other. We were expecting that the hexagon image quality would be better than the square image quality especially at the rotation angles of 30 degrees and 60 degrees, but as the center of rotation was (128, 128), which was not necessarily the center of the middle hexagon, the image quality was not affected.

Rotation in Degrees	Figure 2		Figure 3		Figure 4	
	Square-Pixel Euclidean Distance	Hexagon Pixel Euclidean Distance	Square-Pixel Euclidean Distance	Hexagon Pixel Euclidean Distance	Square-Pixel Euclidean Distance	Hexagon Pixel Euclidean Distance
0	0.0773	0.0782	0.0000	0.0065	0.1657	0.0000
5	0.0825	0.0838	0.0308	0.0308	0.1694	0.1678
10	0.0943	0.0943	0.0564	0.0565	0.1783	0.1772
15	0.1070	0.1069	0.0787	0.0786	0.1896	0.1888
20	0.1186	0.1190	0.0967	0.0967	0.2001	0.2001
25	0.1282	0.1288	0.1112	0.1111	0.2091	0.2097
30	0.1361	0.1364	0.1218	0.1217	0.2168	0.2176
35	0.1415	0.1415	0.1284	0.1284	0.2213	0.2221
40	0.1447	0.1451	0.1332	0.1331	0.2257	0.2252
45	0.1457	0.1458	0.1346	0.1348	0.2271	0.2269
50	0.1452	0.1449	0.1332	0.1331	0.2262	0.2250
55	0.1413	0.1413	0.1285	0.1283	0.2232	0.2224
60	0.1361	0.1363	0.1220	0.1219	0.2185	0.2170
65	0.1285	0.1283	0.1114	0.1114	0.2108	0.2154
70	0.1187	0.1185	0.0969	0.0969	0.2010	0.1988
75	0.1065	0.1064	0.0788	0.0787	0.1896	0.1887
80	0.0942	0.0932	0.0565	0.0565	0.1789	0.1772
85	0.0829	0.0818	0.0309	0.0309	0.1693	0.1680
90	0.0773	0.0755	0.0000	0.0064	0.1657	0.1634

Table 1. Summarized result of Euclidean distance measurements using square and hexagon pixels for images shown in Figures 2, 3 and 4.

As the plot shown in Figure 12 was generated based on the square image (Figure 3), it was expected that the Euclidean distance of the square pixel would be 0 at 0 degrees and 90 degrees rotations as there was no difference between the original image and the square pixel image. Since the image was partial to square pixels, we expected that the image quality of square pixels would be better than the image quality of its hexagon pixel counterpart. However our hypothesis turned out to be incorrect. The square pixel and hexagon pixel Euclidean distances do not differ significantly for most angles.

As the plot shown in Figure 13 was generated based on the test hexagon pixels (Figure 4), we were expecting that the hexagon image Euclidean distance at 0 degrees and 60 degrees would be 0. Note that the hexagon pixel Euclidean distance was 0 at 0 degrees, but it was not 0 at 60 degrees. This was due to the fact that the rotation algorithm used, rotates the image at the coordinate (128, 128). However this point was not the center of the centermost hexagon. Hence it was not 0 at 60 degrees. One unanticipated observation noted was the fact that the hexagon Euclidean distance at 65 degrees was almost the same as that at 60 degrees. We were not able to explain this behavior. Further investigation of this discrepancy is needed. Notice the jump in the Euclidean distance in the 0 to 5 degrees range of Figures 12 and 13. This jump is more prominent in Figure 13, which is partial to hexagon pixels than Figure 12, which is partial to square pixels. This shows that the hexagon pixels have an advantage in case of small rotations but that after that there is little difference between the two. The hexagon pixel was cumbersome to construct and computationally inefficient. Also we had given more weight to hexagon pixels in our experiments because one hexagon pixel covers 62 pixels in the original image whereas the square pixel covers 64 pixels.

4. CONCLUSION

Based on our results attained by using Euclidean distance, we have shown that for small angles of rotation hexagon pixels represents square images better than square pixels are able to represent hexagonal images. In spite of their widespread use in nature (retinas and ommatidia), our results which use the Euclidean distance as a measure of image quality do not show hexagonal pixels to offer an advantage in terms of accuracy of representation of an image over the conventional square pixels. The only other regular tessellation of the plane is into equilateral triangles. Future work on comparing triangular pixels with square pixels may be warranted.

Acknowledgements

We would like to thank Dr. Brian Hennen, Dean of Medicine, and the Organ Imaging Fund of the Department of Radiology, the University of Manitoba, for grant support. We would like to thank Dr. Miroslaw Pawlak for his critical comments.

References

[1] J. Foley, A. Van Dam and J. Hughes, *"Computer Graphics, Principles and Practice (Second Edition)"*. Addison Wesley, Sydney (1990).
[2] E. Buschbeck, B. Ehmer and R. Hoy, "Chunk versus point sampling: visual imaging in a small insect". *Science* **286**(5442), 1178-1180 (1999).
[3] D. Scharf, *"Magnifications: Photography with the Scanning Electron Microscope"*. Schocken Books, New York (1977).
[4] P. Agarwal and M. Shamir, "Efficient Algorithms for Geometric Optimization". *ACM Computing Surveys* **30**(4): 412-458 (1998).
[5] D. Alexander, P. Sheridan and P. Bourke, "An Algebraic-Geometric Model of the Receptive Field Properties of the Macaque Striate Cortex". *Proceedings Australian Neuroscience Society* vol. 8, February 1997.

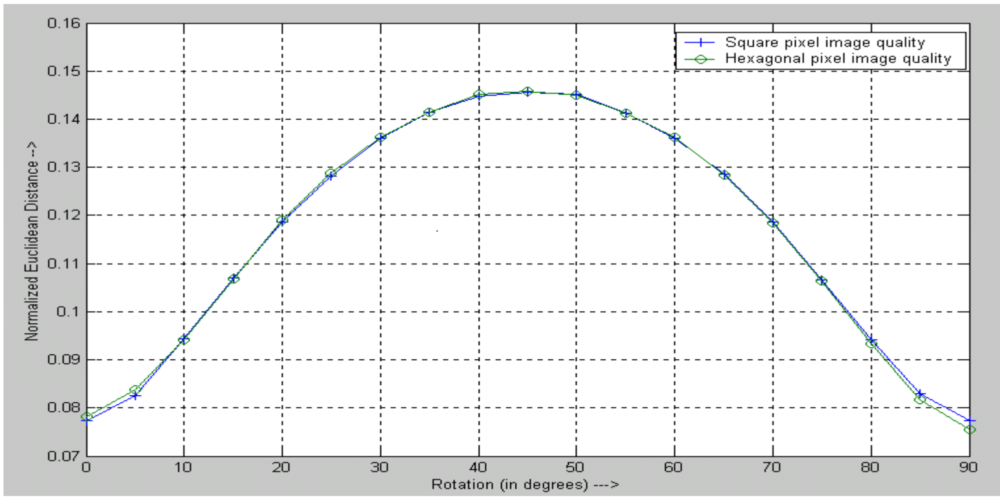

Figure 11. Square and hexagon Euclidean distance vs. rotation (in degrees) for the test image shown in Figure 2.

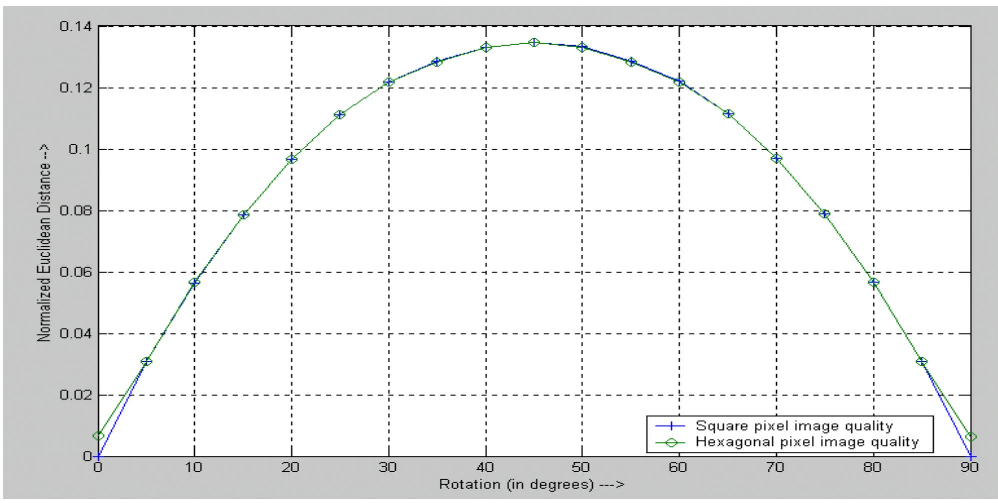

Figure 12. Square and hexagon Euclidean distance vs. rotation (in degrees) for the test image shown in Figure 3.

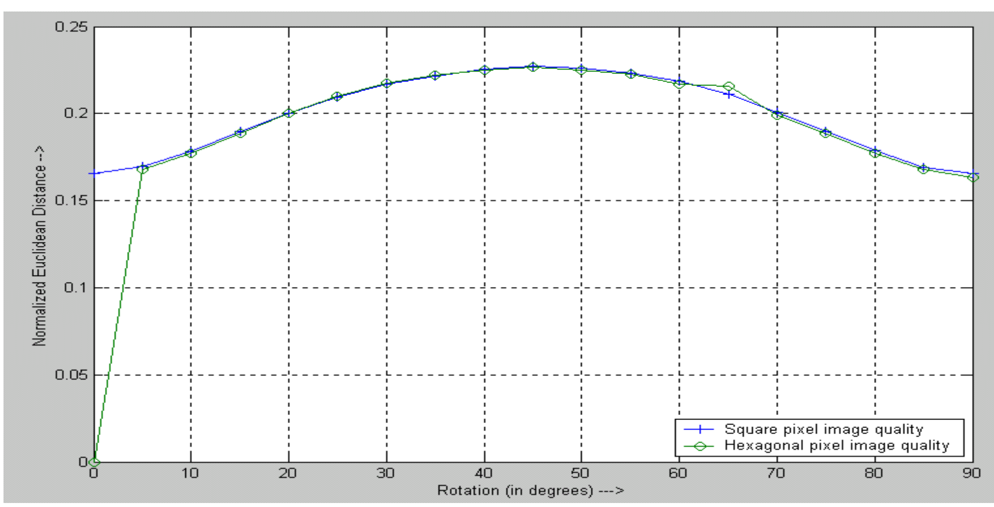

Figure 13. Square and hexagon Euclidean distance vs. rotation (in degrees) for the test image shown in Figure 4.