

## **CSIT 691 Independent Project**

### **A comparison of Mean Average Error (MAE) Based Image Search for Hexagonally and Regularly Structured Pixel Data**

**Student: Sijing LIU**

**Email: [sijing@ust.hk](mailto:sijing@ust.hk)**

**Supervisor: Prof. David Rossiter**

## Index

<b>1. Introduction</b> .....	<b>4</b>
1.1 Aims and Goals .....	4
1.2 The project .....	4
<b>2. Hexagonal Structured Image</b> .....	<b>4</b>
2.1 Hexagonal structure .....	4
2.2 Advantages.....	5
<b>3. Mean Average Error</b> .....	<b>6</b>
3.1 Overview .....	6
3.2 MAE in regular square pixel-based images.....	7
3.3 MAE in hexagonal pixel-based images .....	8
<b>4. Results</b> .....	<b>10</b>
<b>5. Conclusion</b> .....	<b>14</b>
<b>Reference</b> .....	<b>16</b>

## List of Figures, Tables and Equations:

Figure 1: Three Approaches to Image Representation .....	5
Figure 2: A honeycomb .....	5
Figure 3: Comparison of angular resolution between square image and hexagonal image.....	6
Figure 4: Reference block and search area in an image .....	7
Figure 5: The procedure in normal MAE .....	7
Figure 6: Hexagonal pixel structure .....	9
Figure 7: MAE results in regularly structured image and hexagonally structured image.....	12
Figure 8: MAE results in regularly structured image and hexagonally structured image.....	12
Figure 9: MAE results in regularly structured image and hexagonally structured image .....	13
Figure 10: MAE results in regularly structured image and hexagonally structured image.....	13
Table 1: Pseudo code of MAE in square pixel-based images .....	8
Table 2: Pseudo code of MAE in hexagonal pixel-based images .....	10
Table 3: Comparison of MAE in square pixel-based images and hexagonal pixel-based images .....	14
Equation 1: MAE algorithm for square pixel-based images.....	8
Equation 2: MAE algorithm for hexagonal pixel-based images .....	9
Equation 3: MAE algorithm for hexagonal pixel-based images.....	9
Equation 4: MAE algorithm for hexagonal pixel-based images.....	9

## **1. Introduction**

### **1.1 Aims and Goals**

The aims of this project are to generate a hexagonal structured image and a regular square structured image and then implement the MAE algorithm to both the square structured image and the hexagonal structured image. The comparison between MAE in the two kinds of images will then be made.

### **1.2 The project**

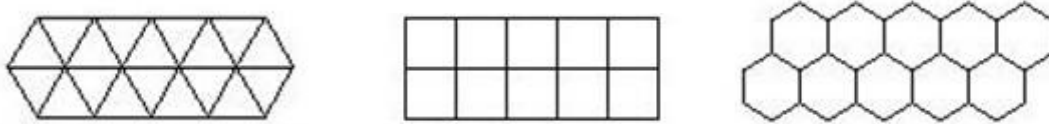
In the field of image processing, square structure has been dominant for many years and mature techniques based on square structure have been developed by researchers. However, hexagonal lattices, which emerged at the same time as digital images, have attracted many scientists' interest. During the past 40 years' study, various experiments comparing performance on hexagonal structured images and square structured images have been conducted and advantages of hexagonal structured images over square structured images have been accepted by most scientists and researchers.

This project focuses on the performance of MAE in motion tracking, which can be widely applied in video processing, in both square pixel-based images and hexagonal pixel-based images. The result will be analyzed as preliminary work for possible further study on hexagonal pixel-based images applied in video processing.

## **2. Hexagonal Structured Image**

### **2.1 Hexagonal Structure**

The repetition of a pattern is called tessellation. From observing the paintings and drawings from ancient civilizations, it is easily found that most of them have used regular triangular, square, and hexagonal tessellation. In fact, those three tessellations are proved to be the only three regular planar tessellations since their interior angle can exactly be divisor of 360 degrees and fill up each vertex in a plane (shown as Figure 1). Other types of polygons will result in a sensory configuration where pixel are either at different distances from their neighbors or involve gaps and overlaps.



*Figure 1. Three Approaches to Image Representation*

Among all the three tessellations, square has held the dominant position in image representation because it is conceptually easy to develop 2D processing algorithms from the 1D case. At the same time, square lattice permits the use of natural coordinates such as Cartesian and also it supports a separable discrete Fourier transformation (DFT).

However, people got inspiration from the nature that hexagons may be the best choice to partition an image. For example, the honeycomb, known as the best 2D example of hexagonal lattice in nature, has led to the famous honeycomb conjecture which states that a hexagonal grid is the best way to divide a surface into regions of equal area with the least total perimeters [1]. The honeycomb conjecture was proposed by Pappus of Alexandria [2] and proved by mathematician Thomas C. Hales elegantly in 1999[3].



*Figure 2. A honeycomb*

## **2.2 Advantages of hexagonal structure**

In this way, hexagonal structure has come to the attention of image processing and been researched ever since. After nearly 40 years' study, now at least six advantages of

hexagonal structured images over square structured images are generally accepted by scientists [4]. For example, the angular resolution of hexagonal structure is much better than square structure which can be easily seen from representing a curved structure (as shown in Figure 3).



(a) Hexagonal pixel-based image

(b) Square pixel-based image

Figure 3. Comparison of angular resolution between Square pixel-based image and hexagonal pixel-based image

Other advantages include both spatial and spectral advantages: high efficiency which was proved by Peterson and Middleton in 1962 [5], uniform connectivity, perfect isoperimetry and equidistance and high symmetry.

### 3. Mean Average Error

#### 3.1 Overview

Motion tracking is an important application in video processing. By tracking motion, we can detect the presence of moving object, track and analyze the motion of object and even make prediction of the next step of a selected object. Motion tracking can be applied to facial recognition, intelligent transportation system, medical imaging and so on. The basic idea of motion tracking is to set a reference block in the former frame and compare it with every block having the same size within a selected searching area in the next frame. The one with the least difference would be the matching one which means the tracking object has moved to this position.

In this project, for simplicity, we just search the reference block in the same image. That is to say, we define the reference block and the search area, and then start to search the reference block in the search area.

### 3.2 MEA in regular square pixel-based images

In traditional images, pixel is represented by square. The MAE algorithm for these types of images is explained as below.

**Step 1:** Suppose we have an image (shown in Figure 4) in which the reference block is represented by white and the search area is represented by black.

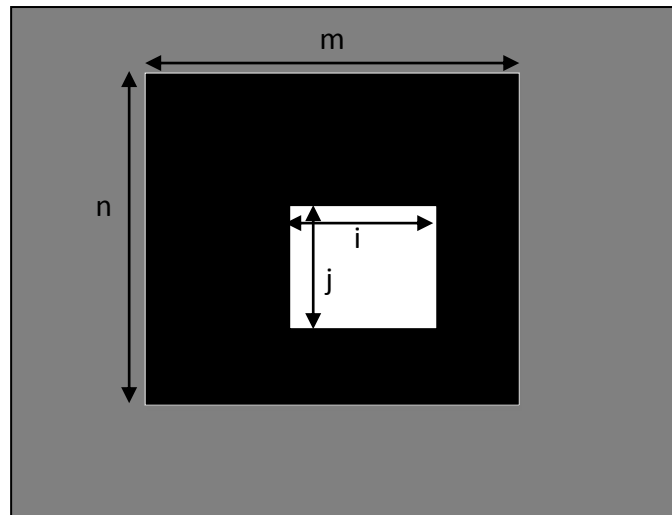


Figure 4. Reference block and search area in an image

As can be seen from figure 4, the size of the search area is  $m * n$  and the size of the reference block is  $i * j$ .

**Step 2:** In the search area (shown as the red rectangle in Figure 5), we take one  $i*j$  block as a unit. From the first unit whose start point is the start point of the search area to the last unit whose end point is the end point of the search area, calculate every unit's mean average error compared with the reference block (shown in figure 5).

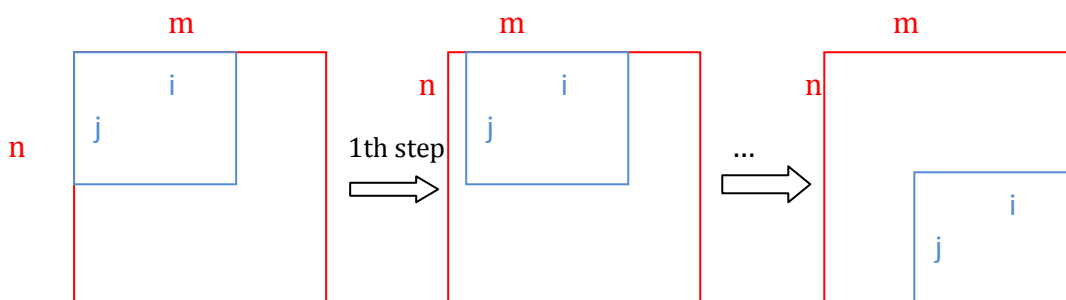


Figure 5. The procedure in normal MAE

The method to calculate the mean average error is very straightforward. For each pixel

in the reference block, subtract the value of its corresponding pixel in the assessed block within the search area and take absolute value of the difference. Sum every pair of pixels' difference and divide it by the size of the reference block. The formula to calculate the mean average error positioned at  $(x, y)$  is shown below:

$$\text{MAE}(x,y) = (1/i*j)*\text{abs}(\left(\sum_{k=0}^{i-1} \sum_{l=0}^{j-1} A(x+k,y+l)-R(k,l)\right)) \quad (1)$$

Here we store the  $i*j$  reference block and the  $m*n$  search area in two separate images which can be more convenient to do the calculation.  $A(x+k,y+l)$  represents the currently assessed block of which the start point is  $(x, y)$ .  $R(k,l)$  presents the reference block which always starts from its origin point  $(0,0)$  and ends at  $(i-1, j-1)$ .

**Step 3:** Find out the smallest MAE and the corresponding position of this block. This is the current position of the tracking object.

Pseudo Code of MAE in regular square pixel-based images:

```

for i from 0 to effective height of search area by i++
  for j from 0 to effective width of search area by j++
    for x from 0 to reference block height by x++
      for y from 0 to reference block width by y++
        calculate the current MAE
        add it to the total MAE
        store the MAE of this current block to the list of MAE
for i from 0 to the length of the MAE list by i++
  find out the smallest MAE and its corresponding position of the block
mark the block and display it

```

*Table 1. Pseudo code of MAE in square pixel-based images*

### 3.3 MAE in hexagonal pixel-based images

In hexagonal pixel-based images, the basic idea of MAE algorithm is the same as the one in regular square pixel-based images. The only difference between them is the



different structure of square structure and hexagonal structure. In a hexagonal pixel-based image, the structure is shown in figure 6.

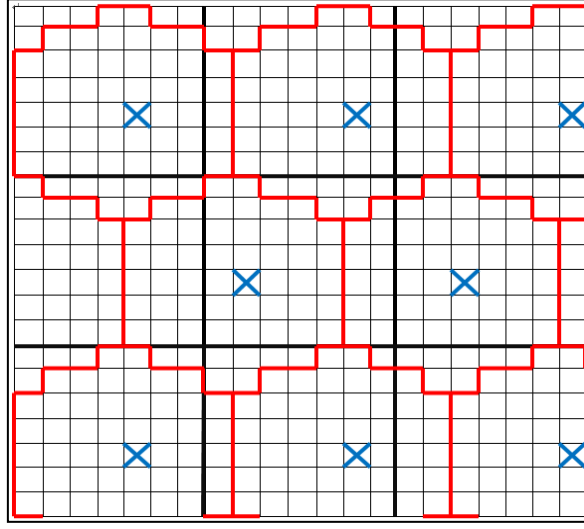


Figure 6. Hexagonal pixel structure

The process of MAE in hexagonal pixel-based images is the same as that in regular square pixel-based images except the method to calculate mean average error. As can be seen from figure 6, the pixels in even row and the pixels in odd row have different pattern from each other. So the mean average error in hexagonal pixel-based images consists of two sub mean average error – one based on the even row pixels and the other based on the odd row pixels.

Thus we have three formulas to calculate the mean average error in hexagonal pixel-based images.

$$MAE_{\text{even-row}}(x, y) = (1/((i/8)*(j/14))) * \text{abs} \left( \sum_{k=0}^{i/8} \sum_{l=0}^{j/14} A_{\text{even-row}}(x+k, y+l) - R_{\text{even-row}}(k, l) \right) \quad (2)$$

$$MAE_{\text{odd-row}}(x, y) = (1/((i-4)/8*(j-7)/14)) * \text{abs} \left( \sum_{k=0}^{(i-4)/8} \sum_{l=0}^{(j-7)/14} A_{\text{odd-row}}(x+k, y+l) - R_{\text{odd-row}}(k, l) \right) \quad (3)$$

$$MAE(x, y) = MAE_{\text{even-row}}(x, y) + MAE_{\text{odd-row}}(x, y) \quad (4)$$

Here we store the  $i*j$  reference block and the  $m*n$  search area in two separate images which can be more convenient to do the calculation. Based on the hexagonal pixel structure, the number of even columns in the reference block is  $i/8$ , the number of odd columns is  $(i-4)/8$ , the number of even rows is  $j/14$  and the number of odd rows is  $(j-7)/14$ .  $A_{\text{even-row}}(x+k, y+l)$  and  $A_{\text{odd-row}}(x+k, y+l)$  represent the currently assessed block

of which the start point is  $(x, y)$ .  $R_{\text{even-row}}(k,l)$  and  $R_{\text{odd-row}}(k,l)$  presents the reference block which always starts from its origin point  $(0,0)$  and ends at  $(i-1, j-1)$ .

Pseudo code of MAE in hexagonal pixel-based images:

<p>The pixels in even row and odd row are stored in two vectors separately.</p> <pre> for i from 0 to effective height of even row in search area by i++   for j from 0 to effective width of odd row in search area by j++     for m from 0 to length of reference block even row       calculate the MAE       add it to the total even row MAE of the block     store the MAE to the even row MAE list for i from 0 to effective height of odd row in search area by i++   for j from 0 to effective width of odd row in search area by j++     for m from 0 to length of reference block odd row       calculate the MAE       add it to the total odd row MAE of the block     store the MAE to the odd row MAE list for every MAE in the even row MAE list and odd row MAE list   add the corresponding even row MAE and odd row MAE   store the sum to a final MAE list for every MAE in the final MAE list   find out the smallest MAE   find out the corresponding position of the block mark the block and display it </pre>
--

*Table 2. Pseudo code of MAE in hexagonal pixel-based images*

## 4. Results

Currently, there are no hexagonal camera sensors commercially available and therefore hexagonal pixel-based images are created via resampling. The resampling technique we

use is the 56 sub-pixel approach in [6] which is based on the technique of Wuthrich [7]. In order to create a sub-pixel effect to enable the sub-pixel clustering, each pixel of the original rectangle pixel based image is represented by a 7\*7 pixel block of equal intensity in the new image. This creates a resized image of the same resolution as the original image with the ability to display each pixel as a group of  $n*n$  sub pixels and limits the loss of image resolution. Another motivation for image resizing is to enable the display of sub pixels, which is not otherwise possible. With this structure, a cluster of 56 sub pixels in the new image, closely representing the shape of a hexagon, can be created that represents a single hexagonal pixel in the resized image. At the same time, in order to get a good visualization, we resize the original image by 7 times which then can be as the same size as the resampled hexagonal pixel-based image.

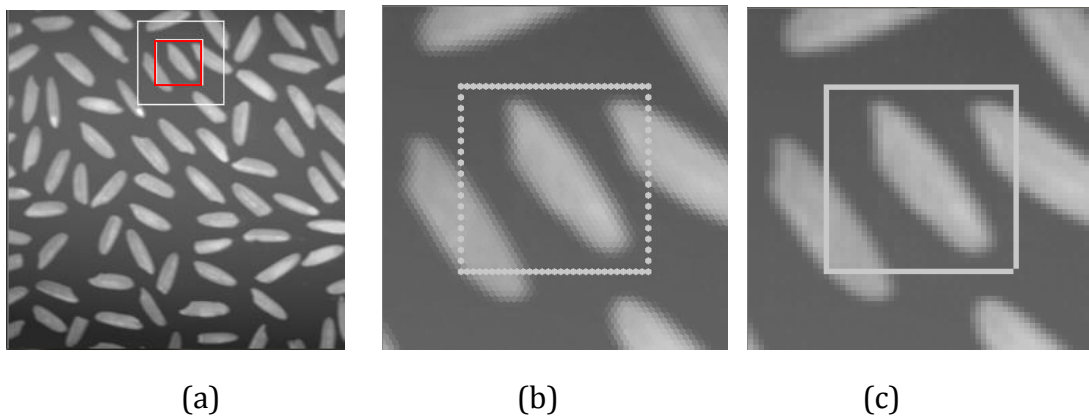
On the other hand, in order to make this project more original and scientific, we also present another method to create hexagonal pixel-based image. In this method, we generate a hexagonal pixel-based image by drawing some simple shapes based on hexagonal pixels. The advantage of generating a hexagonal pixel-based image instead of resampling is that the shape is purer and is easier to collect research data.

Using those two above approaches, we can compare the proposed hexagonal pixel-based MAE technique with the traditional MAE technique applied to square pixel-based images.

For comparison, the visual responses from each of the two kinds of images are illustrated in Figure 7, Figure 8, Figure 9 and Figure 10.

The first two figures (Figure 7 and Figure 8) show the MAE algorithm applied in a square pixel-based image and a hexagonal pixel-based image which is resampled from the former. Figure 7 displays the situation that the object in the reference block also exists in the search area. It can be seen both the two kinds of images perform well and the exact reference block is found in the search area. While in Figure 8, the search area does not contain the searched object but has a similar object. In this situation, both of

these two kinds of images successfully identified the similar object. But there is a tiny difference between them. Although MAE in both of these two kinds of images can identify the searched object in the search area, the exact positions of the searched reference block in the search area are not always the same. As can be seen from Figure 8(b) and 8(c), the searched reference block in the hexagonal pixel-based image is a little higher than that in the square pixel-based image.

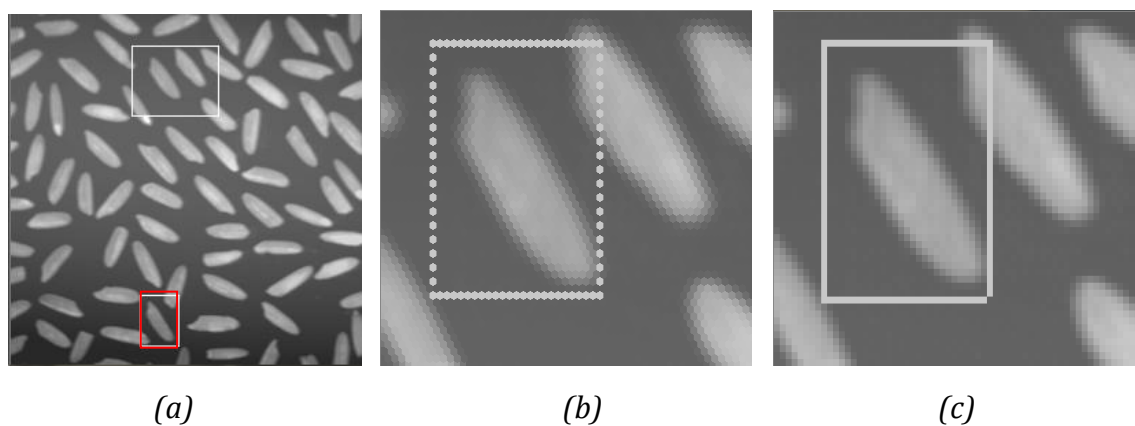


*Figure 7. Searching for an object which exists in the search area*

*(a) The test image in which red rectangle represents reference block  
and white rectangle represents search area*

*(b) MAE search result in hexagonal pixel-based image*

*(c) MAE search result in square pixel-based image*



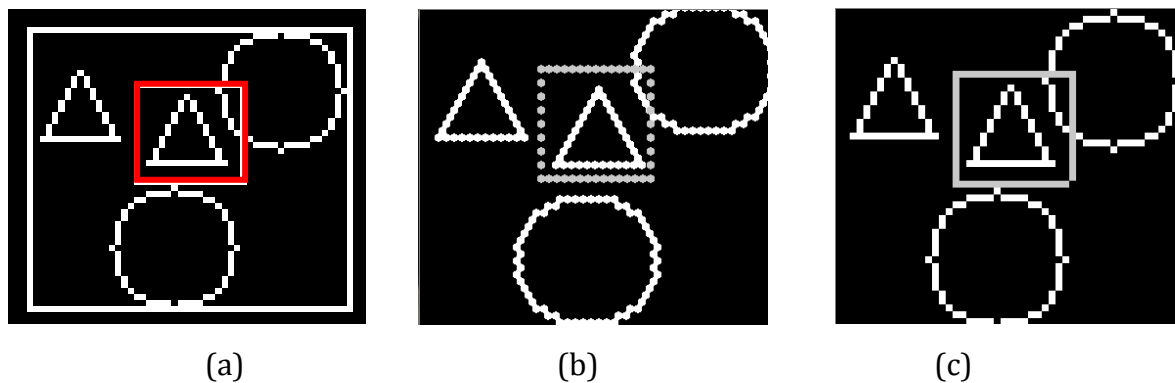
*Figure 8. Searching for an object in a search area which contains a similar object*

*(a) The test image in which red rectangle represents reference block  
and white rectangle represents search area*

*(b) MAE search result in hexagonal pixel-based image*

*(c) MAE search result in square pixel-based image*

The following two figures (Figure 9 and Figure 10) indicate the MAE algorithm applied in a square pixel-based image and a hexagonal pixel-based image which is generated by drawing shapes instead of resampling from an existing square pixel-based image. Both of the two images have the same contents: two identical triangles and two identical circles. Figure 9 shows the result of search an object already exists in the search area. In this situation, both the two images found out the right object and have nearly the same identified area. In figure 10, the search object no longer exists in the search area which contains a similar object. As we can see, although the searched object is detected in both images, hexagonal pixel-based image identified a more precise area than the square pixel-based image did.



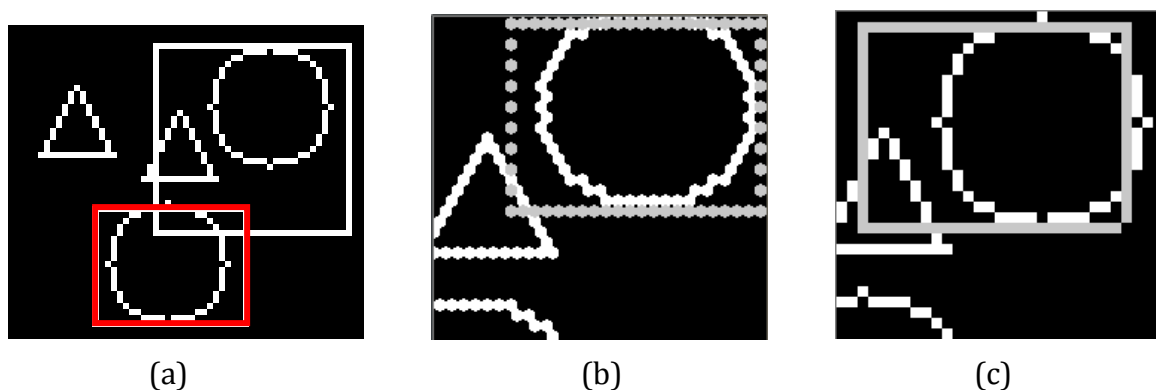
*Figure 9. Searching for an object which exists in the search area*

*(a) The test image in which the red rectangle represents the reference block*

*and the white rectangle represents the search area*

*(b) MAE search result in a generated hexagonal pixel-based image*

*(c) MAE search result in a square pixel-based image*



*Figure 10. Searching for an object in a search area which contains a similar object*

(a) *The test image in which the red rectangle represents the reference block  
and the white rectangle represents the search area*

(b) *MAE search result in a generated hexagonal pixel-based image*

(c) *MAE search result in a square pixel-based image*

Based on the four results, a summary table can be generated to give us an overview of the performance of MAE applied in both hexagonal pixel-based image and square pixel-based image.

	Search the right object	Effective pixels (pixel of the searched object)	Detected effective pixels	Detected pixels	Percentage of detected effective pixels
MAE in square pixel-based image(photo etc)	YES	N/A	N/A	N/A	N/A
MAE in Hexagonal pixel-based image(resampled)	YES	N/A	N/A	N/A	N/A
MAE in square pixel-based image(generated)	YES	56	46	57	80.7%
MAE in hexagonal pixel-based image(generated)	YES	50	50	54	92.59%
MAE in square pixel-based image (generated)	YES	42	42	42	100%
MAE in hexagonal pixel-based image(generated)	YES	40	40	40	100%

*Table 3. Comparison of MAE in square pixel-based images and hexagonal pixel-based images*

It is clear from Table 3 that in all situations including resampled hexagonal pixel-based image and generated hexagonal pixel-based image, the MAE algorithm can be successfully implemented. The main object in the reference block can be identified in the search area in both square pixel-based images and hexagonal pixel-based images. But the performances of these two kinds of image data structure are not the same. For

example, when searching for an object in a search area containing a similar object, in a generated square pixel-based image, the number of effective pixels in the reference block is 56, and the number of detected pixels in the search area is 57 including 46 effective pixels. While in a generated hexagonal pixel-based image, the number of effective pixels in the reference block is 50, and the number of detected pixels in the search area is 54 including 50 effective pixels. Thus, the percentage of detected effective pixel in the hexagonal pixel-based image is 92.59% which is higher than that in square pixel-based image. But the superiority of hexagonal pixel-based image does not always exist, in another situation, when searching for an object which exists in the search area, both the square pixel-based image and the hexagonal pixel-based image can achieve 100% percentage of detected effective pixels.

## 5. Conclusion

Motion tracking plays an important role in video processing which can be widely applied to various fields in human society. Since the current camera sensors generate square pixel-based images, video processing techniques based on square pixel-based image have been developed maturely. However, with people's rising interests toward hexagonal pixel-based image and its application in image processing and real-time vision, kinds of experiments based on hexagonal pixel-based images have been conducted to compare the performance with traditional square pixel-based images.

In this project, we present an approach to implement MAE algorithm in hexagonal pixel-based images. Based on the results of the project, it can be concluded that the accuracy of detected effective pixels in hexagonal pixel-based image is high and the performance of hexagonal pixel-based images is more stable than that of square pixel-based image which suggests that hexagonal pixel-based images may have a future application in video processing. However the limitation of this project is that we used fixed image instead of moving images. Since there are no commercial camera sensors currently generate hexagonal pixel-based images, the time spent on resampling will be the biggest obstacle in video processing based on hexagonal pixel-based images.

## Reference

- [1] Wikipedia, [http://en.wikipedia.org/wiki/Honeycomb\\_conjecture](http://en.wikipedia.org/wiki/Honeycomb_conjecture), retrieved on 17<sup>th</sup> May 2011.
- [2] Weisstein, Eric W. "Honeycomb Conjecture", Mathworld, <http://mathworld.wolfram.com/HoneycombConjecture.html> , Retrieved 17th May 2011.
- [3] Hales, Thomas C. "The Honeycomb Conjecture". Discrete and Computational Geometry 25: 1-22(2001).
- [4] Xiangjian He, Wenjing Jia, Hexagonal structure for Intelligent Vision, p. 53-55.
- [5] Peterson, D. P. and D. Middleton, Sampling and Reconstruction of Wave-Number-Limited Functions in NDimensional Euclidean Spaces. Information and Control, 1962.5: p. 279-323.
- [6] Middleton, L. and Sivaswamy, J., "Hexagonal Image Processing; A Practical Approach", Springer 2005.
- [7] Wuthrich, C.A, Stucki, P. "An algorithmic Comparison Between Square- and Hexagonal-based Grid", CVGIP: Graphical Models and Image Processing, Vol. 53, pp. 324-339, 1999.