Computational Efficiency of Hexagonal Arrays in Compound Eyes

Jessica White and David C. Tam
Dept. of Biological Sciences
University of North Texas
Denton, TX 76203
(940) 565-3261
(940) 565-4136 (FAX)
dtam@unt.edu

Summary

1. Introduction

Hexagonal arrays are commonly found in compound eyes of insects forming an array of ommatidia for light detection. Such hexagonal arrays are more commonly found than other geometric arrays such as rectilinear orthogonal arrays (see Fig. 1). In this paper we explore the computational efficiency using hexagonal arrays as compared to rectilinear arrays.

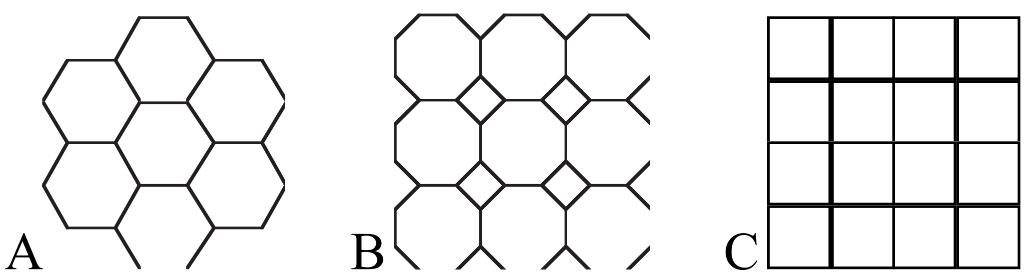

Figure 1. Diagram showing the differences between various hexagonal and rectilinear orthogonal arrays.

2. Geometric Considerations

Rectilinear arrays are often used in spatial computing because of the orthogonal nature and the ease of implementation in Cartesian coordinate system. Yet, rectilinear arrays do not necessarily provide the most symmetrical geometric configuration for efficient computation in 2-D spatial dimensions, when computation between adjacent neighboring cells is involved.

Motion detection in compound eyes can be computed based on the local difference of light intensity between adjacent neighboring cells [2, 4, 5] of the ommatidia [1] using lateral inhibitions [2, 3].

Square array is a good candidate geometric array for computing the local differences in light intensity between adjacent cells. Yet, because the array is orthogonal to each other, the direction (and angle) of motion detection is limited to 90º. Each cell shares 4 contagious neighbors in a square array. If 45º motion detection is needed, computation between corner cells is needed, with does not provide contagious boundary sharing between adjacent cells (see Fig. 1C). This can be overcome by forming a octagonal array (see Fig. 1B). In an octagonal array, the 45º motion detection can be computed between contagious cells lying at the corners of the octagon.

Octagonal arrays suffer from one drawback in terms of different number of neighbors. The center octagonal cell has 8 contagious neighbors whereas the corner diamond cell has only 4 neighbors. Although this configuration resolves the 45º motion detection computation using contagious neighboring cell light intensity differences, it breaks the symmetry of having two classes of cells – those with 8 contagious neighbors and 4 contagious neighbors. Thus, it requires two sets of algorithms for computing the local differences that maps to the direction of motion angle.

In contrast, hexagonal array provides the best candidate for contagious neighboring cell computation of local light intensity difference between adjacent cells without the complexity of an octagonal rectilinear array. The first-order computation of light intensity differences can be computed using the contagious neighboring cells at 60º angles (see Fig. 1A).

Using this hexagonal configuration, finer angular differences, such as 30º angles, can be obtained using the second-order adjacent cells. Thus, this provides a means for symmetric computation using one set of computational algorithm, each having exactly 6 contagious neighbors.

The significance of this symmetrical geometry arrangement can be appreciated when the computation of light intensity difference between second-order and higher-order adjacent cells is required, such as the computation needed for finer angular increments, and velocity and acceleration computation (time-derivatives of second-order and third-order differences).

These geometric constraints have significance in the physical implementation and packing of ommatidium in the compound eyes that require local computation involving adjacent neighboring cells.

3. Comparison of computational methods for motion detection using hexagonal array and rectilinear array

We compare the computation required for motion detection in the primary (first-order) and secondary (second-order) contagious neighbors for each type of array – hexagonal and rectilinear arrays. We further compare the computation required for detecting the velocity and acceleration of motion. Velocity is computed by the first-order derivative of the light intensity difference between adjacent cells, and acceleration is computed by the second-order derivative of the light intensity difference between adjacent cells. We found that the complexity of computation for rectilinear array is much greater than the hexagonal array in motion detection; thus increasing the computational efficiency using a symmetric, repeatable configuration.

References

[1] Blum M, Labhart T. Photoreceptor visual fields, ommatidial array, and receptor axon projections in the polarisation-sensitive dorsal rim area of the cricket compound eye. J Comp Physiol [A] (2000) 186:119-128

[2] Coleman BD, Renninger GH. Theory of delayed lateral inhibition in the compound eye of limulus. Proc Natl Acad Sci U S A (1974) 71:2887-2891

[3] Laughlin S. A simple coding procedure enhances a neuron's information capacity. Z Naturforsch [C] (1981) 36:910-912

[4] Schultz AM, Wilcox MJ. Parallel image segmentation using the L4 network. Biomed Sci Instrum (1999) 35:117-121

[5] Srinivasan MV, Bernard GD. The effect of motion on visual acuity of the compound eye: a theoretical analysis. Vision Res (1975) 15:515-525