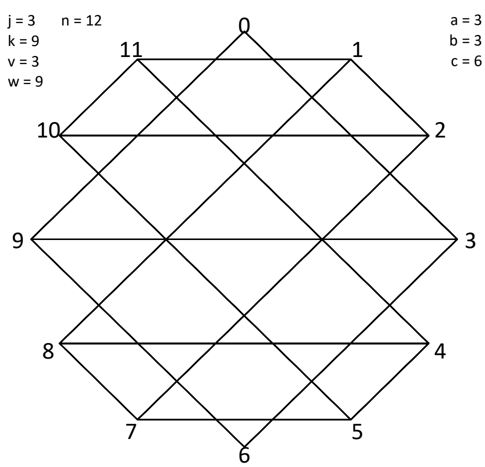
An Introduction to General Triangles Images

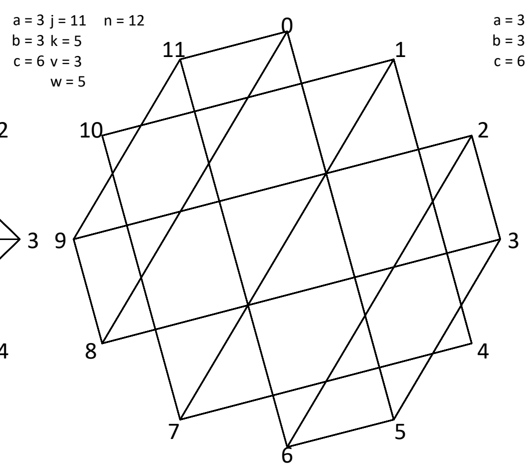
There are a couple of goals for the final triangles chapters of **PwP**. This chapter, by necessity, must be suggestive rather than exhaustive in nature. This follows the progression started in the second sharpest triangles Chapter 5, following the more exhaustive treatment of sharpest triangles images in Chapters 3 and 4. The next chapter deals with interior concurrences since concurrences disrupt counting patterns in some situations but not others.

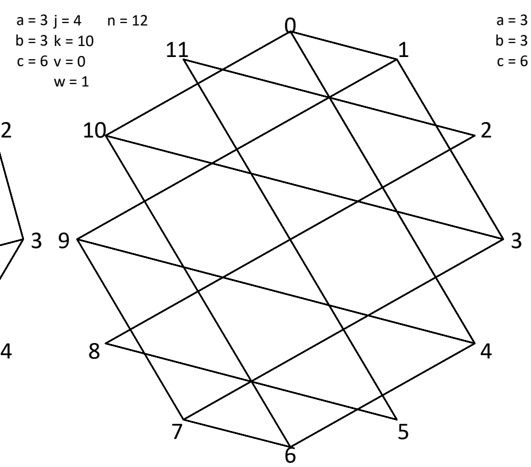
The *General Triangles* model allows for exploration of triangles images in an open-ended fashion because the types of images that you can examine is only subject to your ability to describe the rule used to create the three directions for parallel lines from ***n***-gon vertices to the size of ***n***.

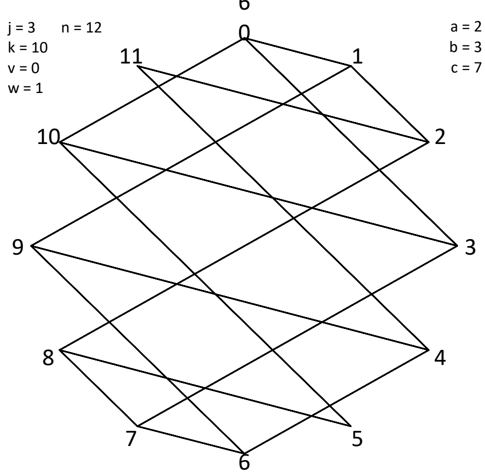
As a result, we will first focus attention on using the *General Triangles* file and include a couple of elementary examples using the file. Next, we will explore some of the general patterns seen in triangles images. Many of these patterns were initially uncovered in earlier chapters, but we examine them here in a more systematic fashion. Finally, we provide additional examples of analyzing triangles images, much like we did with second sharpest images in Chapter 5.

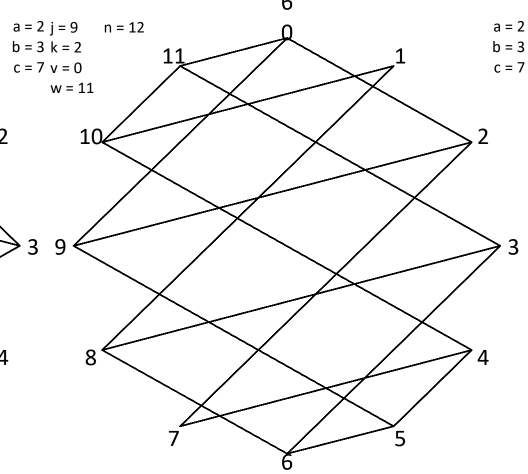
A Second Look at the *General Triangles* file. Images are created in the *General Triangles Excel* file using ***n*** and four parameters, ***J***, ***K***, ***V***, and ***W***, that control the directions of the three sets of parallel lines. These lines create similar triangles whose angular measure is ***a***, ***b***, and ***c*** times 180/***n*** degrees. As discussed in the notes in M10:X14, an image can be rotated by adjusting ***J, K, V,*** and ***W*** without affecting ***a***, ***b***, and ***c***. We initially examined the *General Triangles* file in Section 2.5 which provided a solution for obtaining isosceles right triangles images. Such images by necessity require ***n*** to be a multiple of 4 because the angles required must be of the form ***a*** = ***b*** and ***c*** = 2***a*** with ***a***+***b***+***c*** = ***n*** so that ***n*** = 4***a***. Both rows show ***n*** = 12 images. The first three are isosceles right triangles. The left was shown in 2.5, the middle is a rotated version (120° clockwise) of the same image, but the right is different because here, none of the triangles is a vertex triangle. The second three are scalene 2, 3, 7 images with the first two showing that we can obtain reflected images by replacing ***J, K, V,*** and ***W*** with ***n***-***J***, ***n***-***K***, ***n***-***V***, and ***n***-***W*** but the right is different because once again, none of the triangles are vertex triangles. Rotation and reflection do not affect the number of triangles in an image.

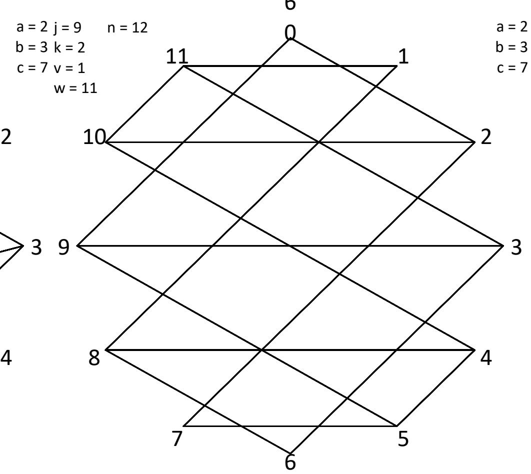






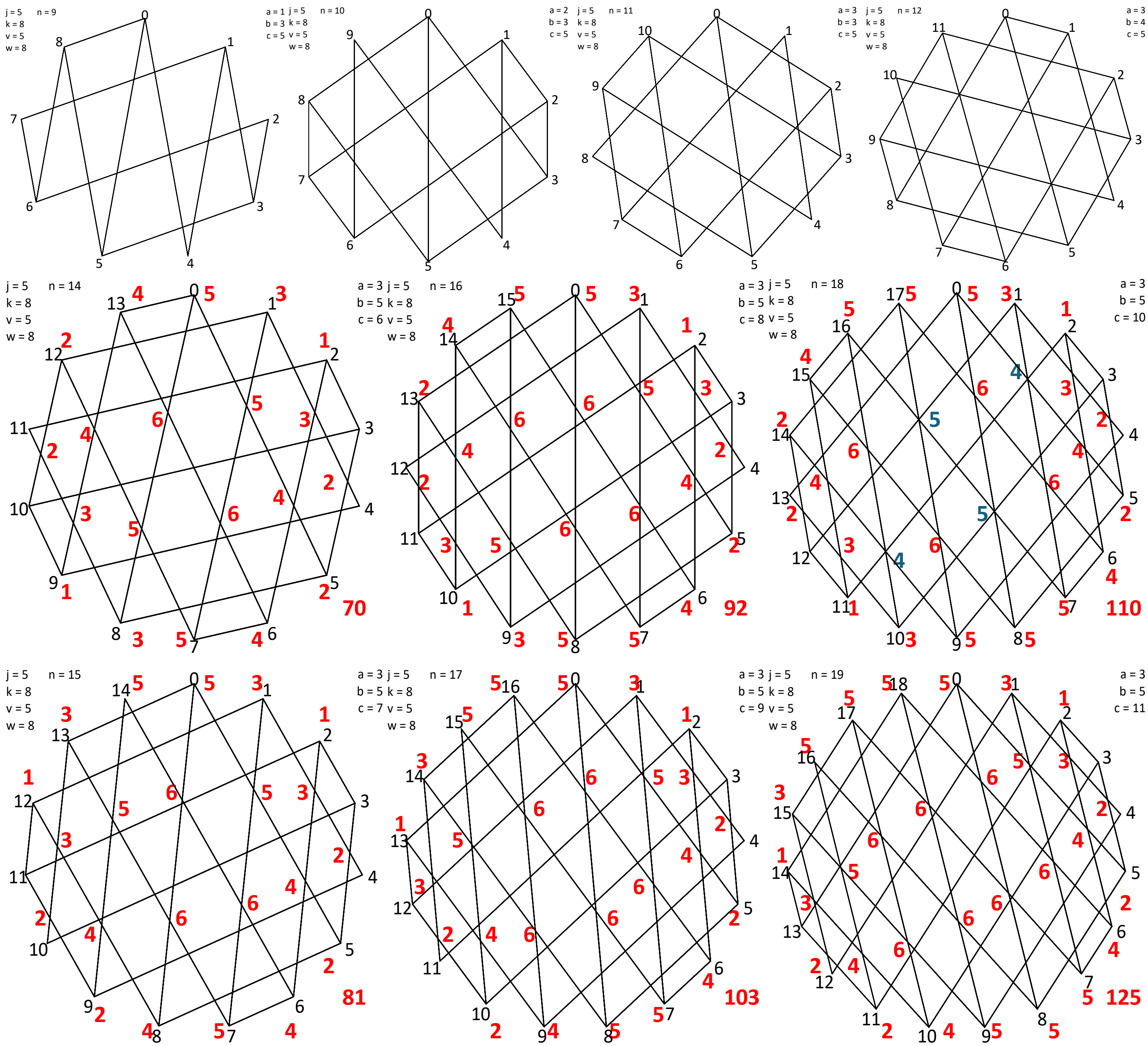






For counting purposes, when ***n*** is even there are two distinct image types but only one distinct image type when ***n*** is odd.

Analyzing a Fixed Set of Parameters as *n* Changes: 0-5-8 Images. The simplest images to analyze in this more general setting use fixed values for ***JKVW*** and analyze what happens as ***n*** changes. Here we show a simple extension of the 0-5-7 analysis done in the previous chapter. It is worth recalling that two sections were devoted to this analysis, and it is useful to review both before undertaking this extension which changes 7 to 8, thereby having a smallest angle span 3 vertices so that there are two internal apex arcs. The top row shows the progression from ***n*** = 9 to 12 in which the two smallest angles are not 3 and 5. The next, ***n*** = 13, is an odd isosceles triangles image which will be taken up in more general form later in this chapter. Here we focus on ***n*** > 13 which produces scalene triangles images. As with 0-5-7, note that there are distinct patterns for even and odd ***n*** apex counts in the middle and bottom row.



Even n. Let ***n*** = 2***k***. The ***n*** = 14, ***k*** = 7 has 1+2+3+4+5 = Δ_5 at top and bottom vertices and two interior arcs of 2+3+4+5+6 = Δ_6-1 or 70 total. Subsequent even ***n*** adds an additional 5 at top and bottom and 6 on each arc in the middle for a total of 22 more so that ***n*** = 16 has 92 and ***n*** = 18 would have 114 except for the 4 concurrent points noted in **blue** for a total count of 110. Ignoring concurrent points (the next even ***n*** with concurrent points is ***n*** = 28 with 2).

$$T_{even}(n = 2k) = 2\Delta_5+2(\Delta_6-1)+22(k-7) = 2(5\cdot6)/2+2((6\cdot7)/2-1)+22(k-7) = 30+42-2+22(k-7) = 70+22(k-7).$$

Odd *n*. Let ***n*** = 2***k***+1. Compare the two left images, ***n*** = 14 and ***n*** = 15. The location of the apex counts (on the left half of the image) change, but the same sets of triangular numbers appear: The perimeter for ***n*** = 15 has $2\Delta_5+5$ (since there are three **5**s on the perimeter rather than two) and interior apex counts are $2(\Delta_6-1)+6$ (three interior **6**s rather than two) or **11** more than ***n*** = 14 and subsequent odd ***n*** increase by **22** just like even ***n***.

$$T_{odd}(n = 2k+1) = 2\Delta_5+2(\Delta_6-1)+11+22(k-7) = 81+22(k-7).$$

Even and Odd Together. In this situation, we see that although the perimeter and arc patterns switch with one another for even versus odd, the overall process can be thought of as simply being a function of ***n***. After ***n*** = 14 with T(14) = **70**, each additional ***n*** adds 11 triangles. This equation works for ***n*** ≥ 14 except for images with concurrent points.

$$T(n) = 2\Delta_5+2(\Delta_6-1) + 11(n-14) = 70 + 11(n-14).$$

a* = 3, *b* = 5, Even *n* with no Vertex Triangles Images.** The final three images show the second version of even ***n images with small and medium angles of 3 and 5. This follows the strategy discussed in the second vertex triangles section later in this chapter. A quick comparison of these three images with the three in the second row shows that the patterns are reversed left to right, but the same total apex counts result, subject to the disruptions created by concurrent points noted in **blue**. As a result, only ***n*** = 16 is the same apex count for both versions.