

Removing Internal Reflection Halos With PixInsight

Tutorial by David Ault

Alnitak Image Provided by Shawn Harrison

Overall Steps

- Measure the circles
 - Measure the extents
 - Measure the intensities
- Determine the center and radius for each circle
- Create a new image for the halo model
- Use PixelMath to re-create each circle at the measured intensities
- Subtract the model from the original image
- Clean up any remaining artifacts with CloneStamp tool

Measure the Circles - Extents

- The goal here is to measure the extents of each of the circles
- I use PixelMath to insert lines at different x & y locations (see Figure 1)
- Each line (R/K, G & B) checks to see if the current location has an x value equal to the variable xLoc which has been assigned the value of 599
- For the R/K (red channel) equation it will set the pixel to 1 (pixels intensities range from 0 to 1 in PixInsight)
- The \$T variable tells PixelMath to use the value of the active window
- Since the G & B (green and blue channels) get a value of 0 while R/K gets 1, the resulting line should be red

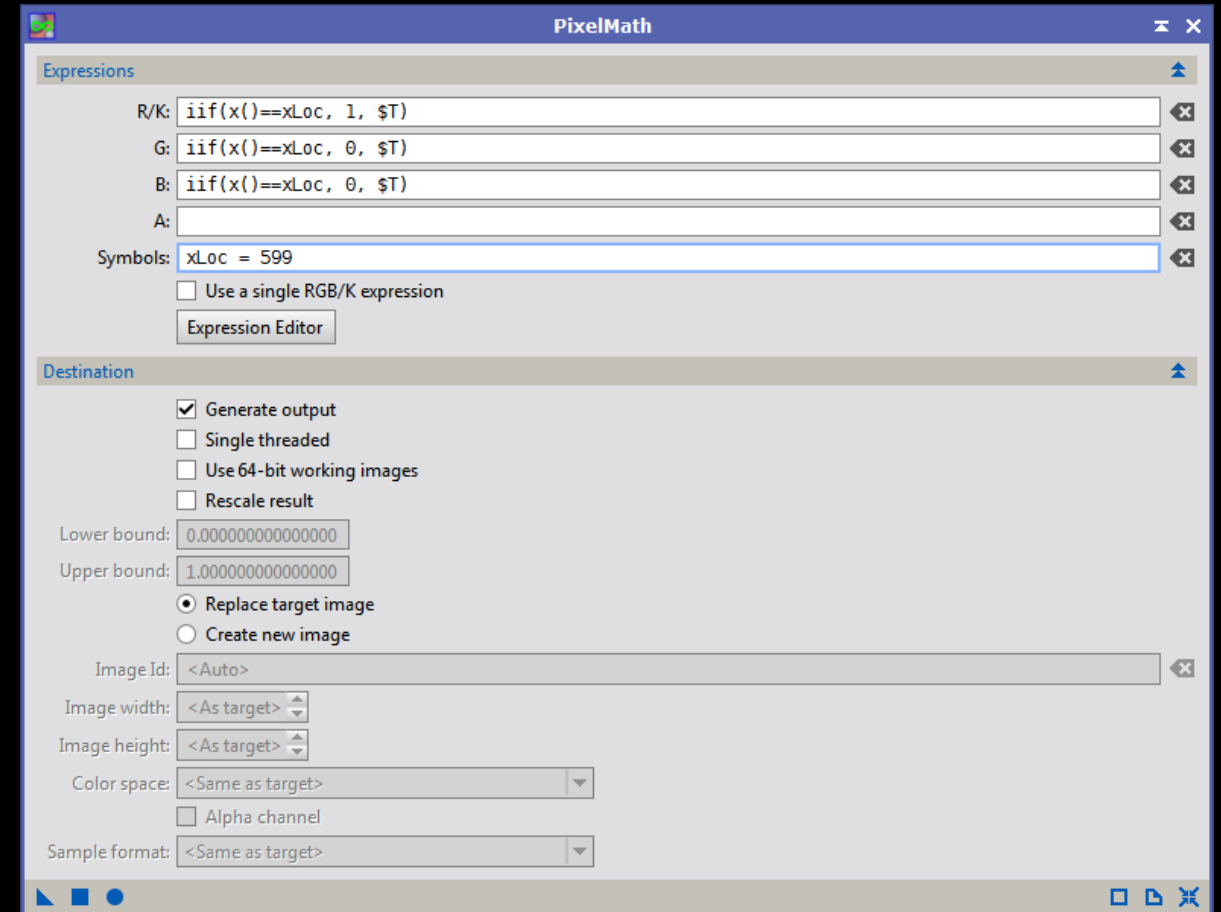


Figure 1: Drawing a Vertical Line

Measure the Circles - Extents

- Typically I have the mouse over the image to get an approximate x/y location
- Then I draw a line at that location using the PixelMath operation in Figure 1 by modifying the xLoc variable
- Here I've used this process with xLoc values of 599 and 1329
- You can either make a copy of the image to draw on, or just make the changes and then undo them when your measurements are complete

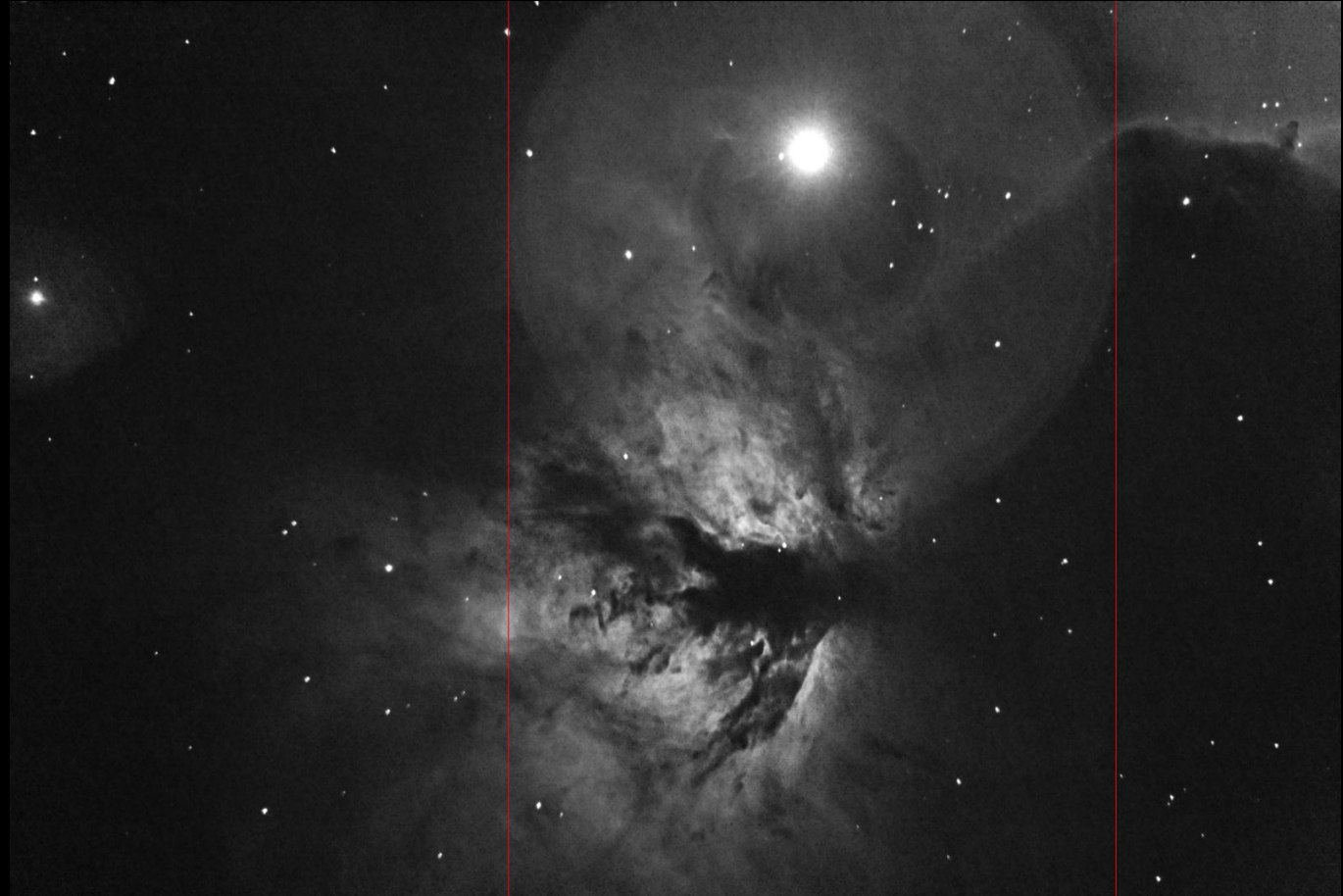


Figure 2: X Bounds of the Largest Circle

Measure the Circles - Extents

- In this example it is difficult to measure the vertical extents for the larger circles, so instead I guess at the center of each circle (the blue lines in Figure 3)
- Here I've marked all the approximate edges of the circles (green for the vertical)
- Keeping track of each of the x and y locations will help with the next step, especially if organized by circle
- You can change the colors by switching the 0/1 values in the R/K, G & B channels equations

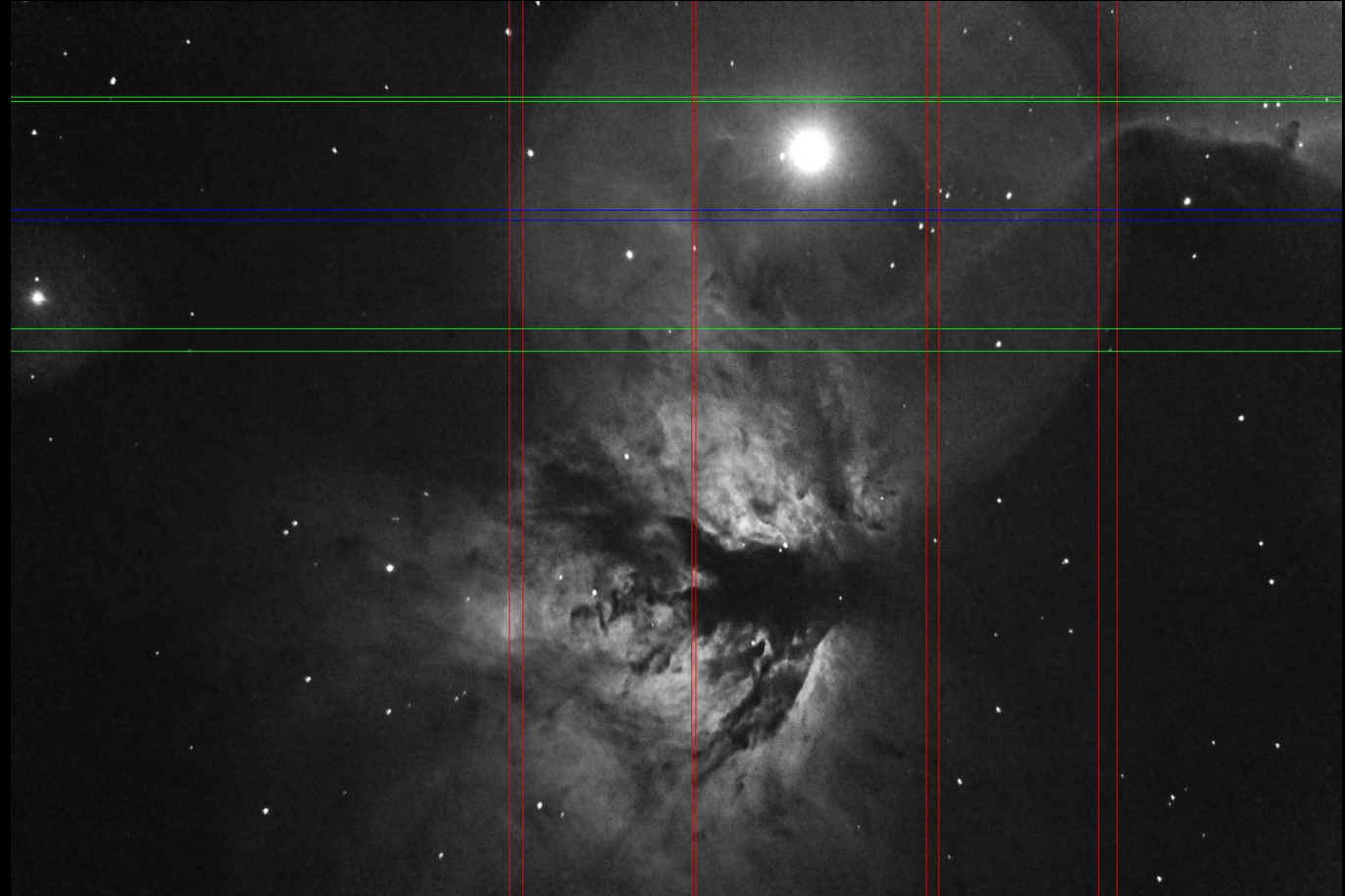


Figure 2: X Bounds of the Largest Circle

Measure the Circles - Intensities

- Instead of just hovering over each circle and reading the value from the PixInsight status bar, we are going to get a better estimate by including more pixels
- This can be done several ways, but I am going to use previews so I have better control over the sampling



Figure 4: Sampling Location Previews

Measure the Circles - Intensities

- To add a preview to a window click on the highlighted 'New Preview Window Mode' (the left of the two icons highlighted in yellow)
- Then click and drag in the image to create the preview box (see Figure 4)
- If you don't like the position or size of a preview you can change it using the second icon which in enables 'Preview Edit Mode'
- Create a preview to sample the values in each circle as well as just outside (see Figure 4)
- Try not to include any stars or nebulosity

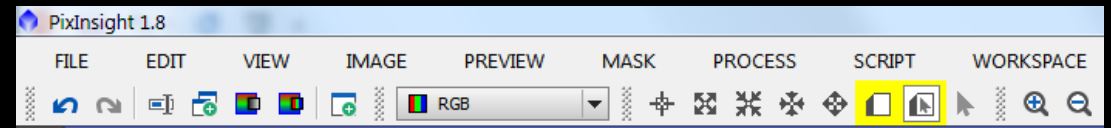


Figure 5: Preview Controls

Measure the Circles - Intensities

- Next we use the Statistics tool to measure the mean (or median if you have some small stars in the preview) of each preview
- Here Preview01 represents the largest circle and Preview02 the background outside the circles
- We want to know the delta from the background to all the circles
- In this case the difference between the means for each preview is **0.0550763**
- This value will be the intensity value we use for this circle later on

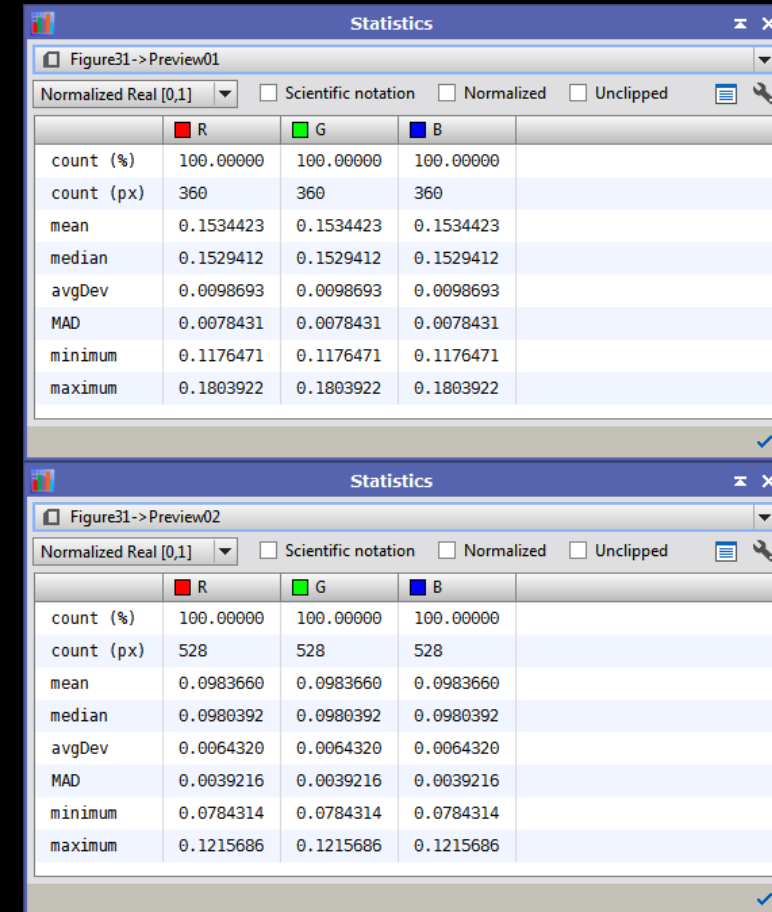


Figure 6: Statistics Read-out

Determine the Center and Radius for Each Circle

- For the circles that you measured the extents of, you can calculate the center as:
$$(x_{\text{High}} + x_{\text{Low}})/2$$
- And the radius as:
$$(x_{\text{High}} - x_{\text{Low}})/2$$
- If you estimated the center, just use that value
- You can do the same for the vertical extents
- We will use the center and radius values for the PixelMath equations later

Create a New Image for the Halo Model

- This can be done with the menu option: **Image->New** (Ctrl+N)
- I prefer to do tasks like this in grayscale mode so if you have color channels it may be easier separate them before doing halo removal
- The original image was 1600x1077 so I set the same size here
- I initialize it with black: $k=0.0$
- Click the circle icon to create the new image

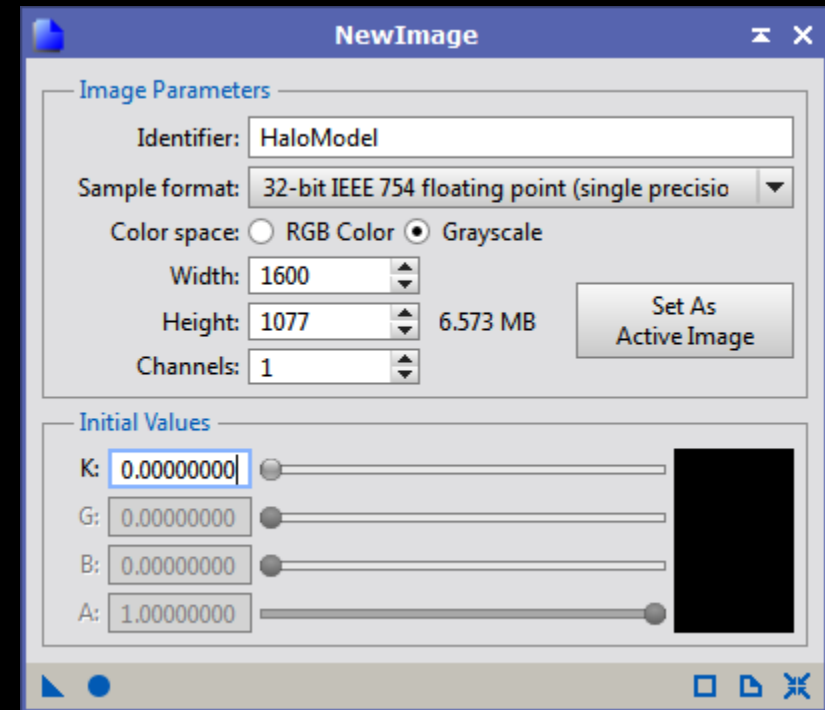


Figure 7: NewImage Tool

Using PixelMath to Create the Halo Model

- This next part seems complicated but it really just boils down to simple algebra

- The equation for a circle about the origin in a Cartesian coordinate system is:

$$r^2 = x^2 + y^2 \quad (\text{Equation1})$$

- This can be extended to place the circle about an arbitrary center: (cx,cy)

$$r^2 = (x-cx)^2 + (y-cy)^2 \quad (\text{Equation2})$$

- We are going to use the center and radius information we measured to see if we are within a particular radius:

$$r = \sqrt{(x-cx)^2 + (y-cy)^2} \quad (\text{Equation3})$$

- Figure 8

- I create several Symbols (variables):
 - r: radius
 - rm: max radius
 - cx, cy: center x & y
 - i: intensity
- The first part of the equation is simply Equation3 above
- The second part is an inline-if. It checks to see if r is less than the max radius. If it is, the value of 'i' is stored at that location, otherwise the existing value is used.

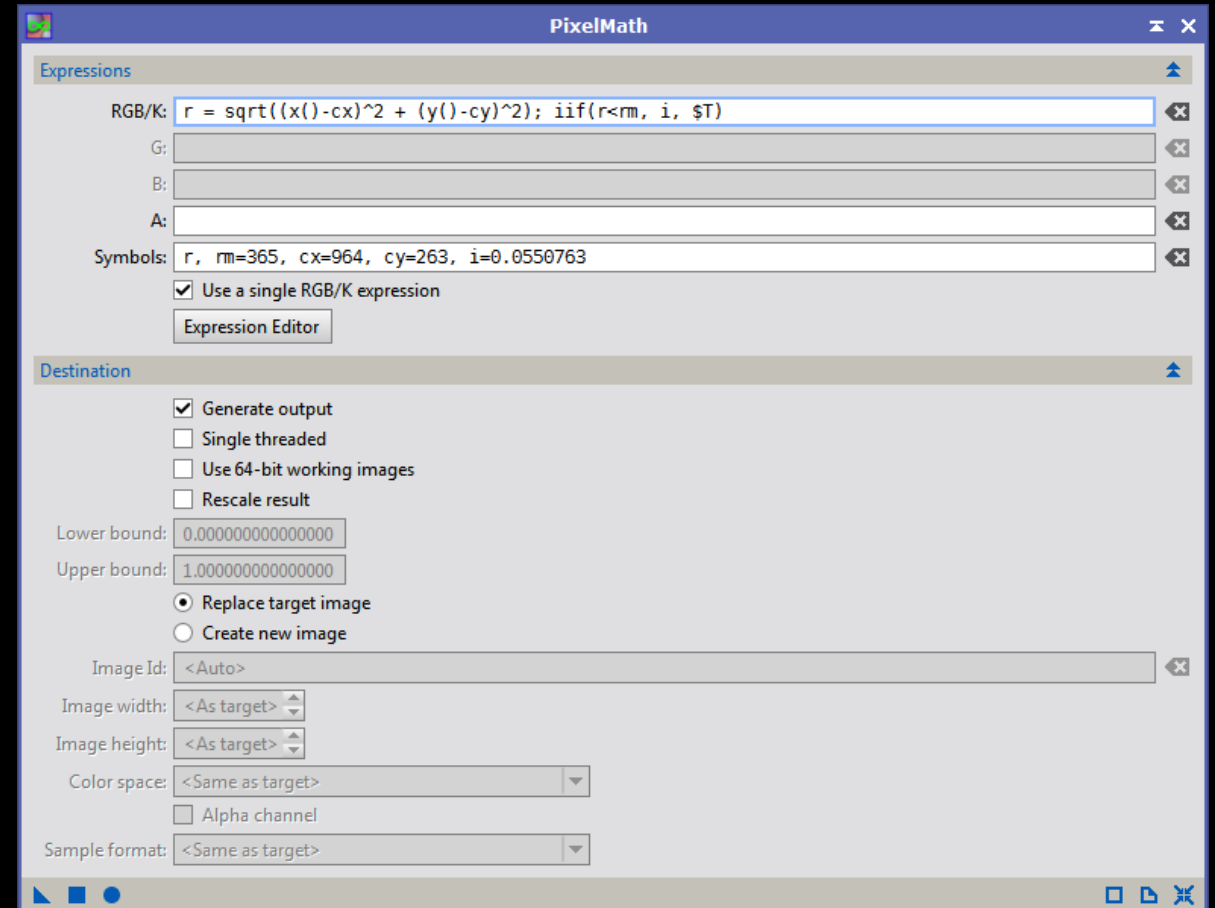


Figure 8: Creating a Filled Circle With PixelMath

Using PixelMath to Create the Halo Model

- Repeat Figure8 for each circle from the largest to the smallest
- This keeps you from having to check interior and exterior radii
- After the model is complete do a minor convolution to soften the edges (Figure 10)
- The final image should look similar to the halo you are trying to remove (Figure 9)



Figure 9: The Halo Model

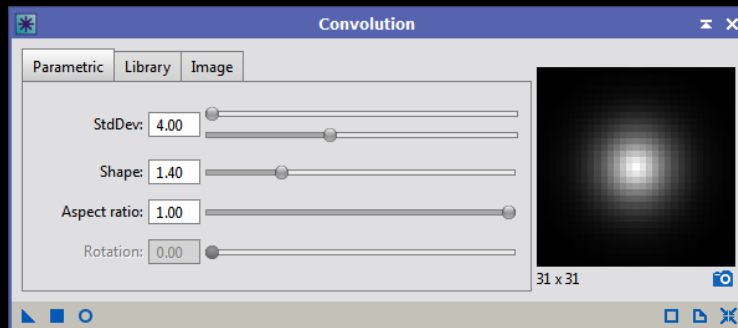


Figure 10: Convolution (blurring)

Subtract the Model from the Original Image

- Subtracting the HaloModel from the image is easy with PixelMath (see Figure 9)
- Just drag the triangle icon to the original image and the model will be subtracted
- As you can see in Figure 13 below the model is not perfect

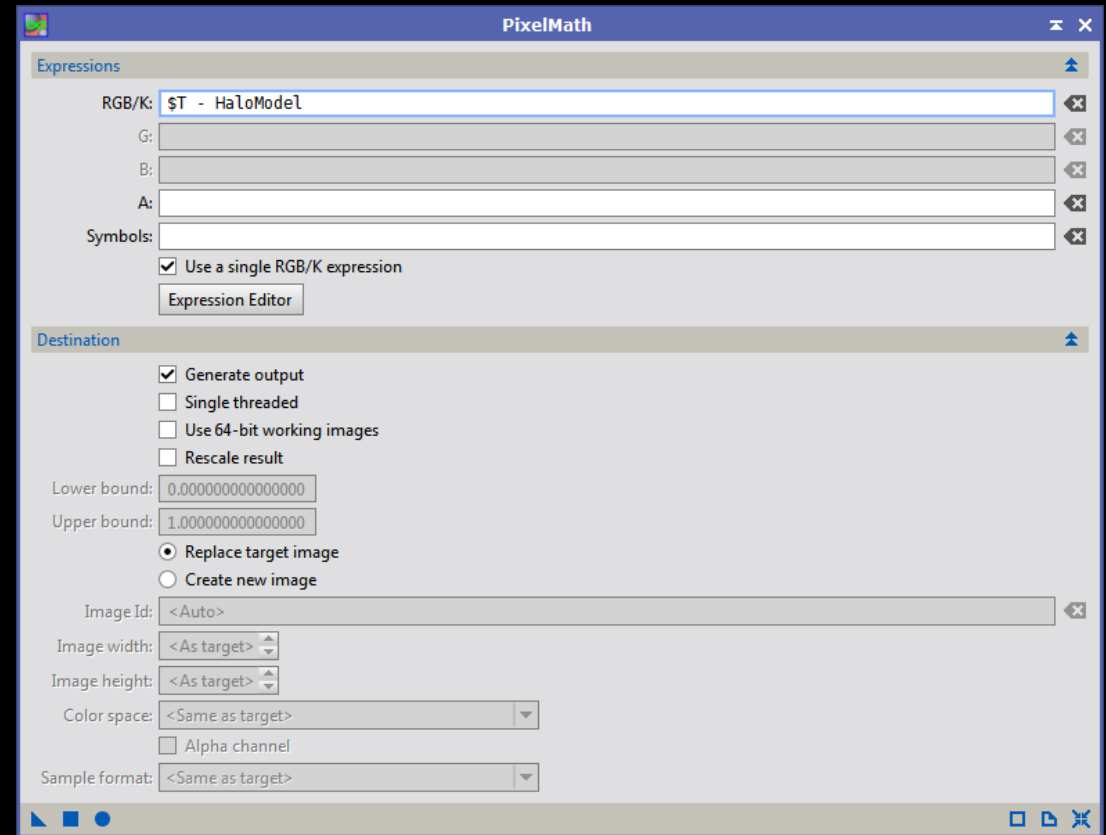


Figure 11: Halo Subtraction

Subtract the Model from the Original Image



Figure 12: The Original Image



Figure 13: The Halo Subtracted Image

Clean-up Remaining Artifacts

- You should try to refine the model until it looks like the image will be easily cleaned up with the CloneStamp tool
- The PixInsight CloneStamp tool is very similar the ones in any other image editing program
- You select a reference location and paint it over the existing data
- The Opacity setting can be used to blend the new data with the old

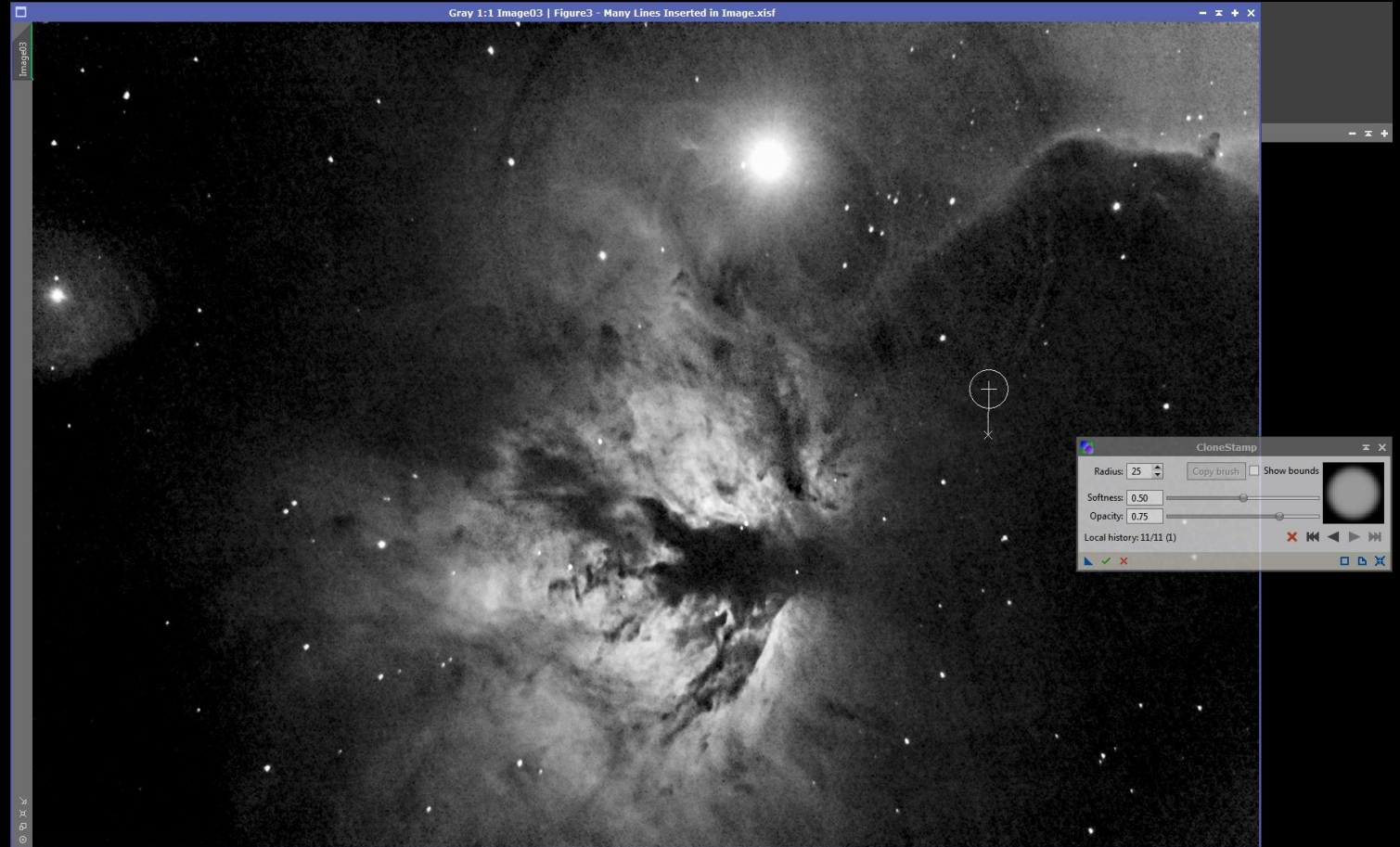


Figure 13: The Halo Subtracted Image

Final Image

