An In-depth Description of the Enhancement Routines used in this Facility

M. Patrick Martin

February 13, 2015

Update: June 2024

This online Cometary Coma Image Enhancement Facility provides five different enhancement routines for the users. Other less sophisticated techniques are available in standard image processing/enhancing packages and are therefore not included here. While the help page offers a short overview of the routines, this write-up is meant to provide an in-depth description of the above five techniques. Additional information on these techniques as well as the pros and cons of different techniques can be found in Samarasinha & Larson (2014)[1] and references therein (click here[2] for a preprint). We highly recommend users be familiar with the different enhancement techniques to facilitate their proper usage.

Unenhanced cometary images suffer from the very aspect that makes comets so amazing to the average person: their brightness distribution. The structure of the coma is obfuscated by its brightness distribution, and many individual features may not be easily recognizable. This makes analyzing unenhanced cometary images very difficult, as important structural information is contained both near the bright optocenter (the brightest point of the coma, assumed to be the location of the nucleus) and in the dimmer outer regions of the coma. Enhancement techniques therefore attempt to bring out the structures in both of these areas, through different methods.

1 Common Information

Some user-input fields in the web Facility are shared among all enhancement algorithms. The first of these is the file to upload. This needs to be a FITS image file (with .fits or .fit extension) of the comet you wish to enhance.

Second is the (x,y) coordinates of the nucleus/optocenter. This is the brightest point of the comet, which sometimes is nearly in the middle of the coma, but that is not guaranteed. You can find the optocenter (nucleus) by adjusting the brightness floor of the image until there are only a few pixels that are highlighted – these are the brightest pixels of the comet in the image, so choose the one with the highest brightness. Some programs (like IRAF's imcntr task) are capable of finding the optocenter given a point reasonably close to it.

[1] http://www.sciencedirect.com/science/article/pii/S0019103514002814

[2] http://arxiv.org/abs/1406.0033

Third is the region of the image to enhance. This can be the entire image, but may be scaled down. Large images take longer to enhance, and if there is only a specific region you are interested in, then you may wish to restrict the routines to that region. Do note, however, that restricting the region in this manner may alter the optocenter's position in the resultant image.

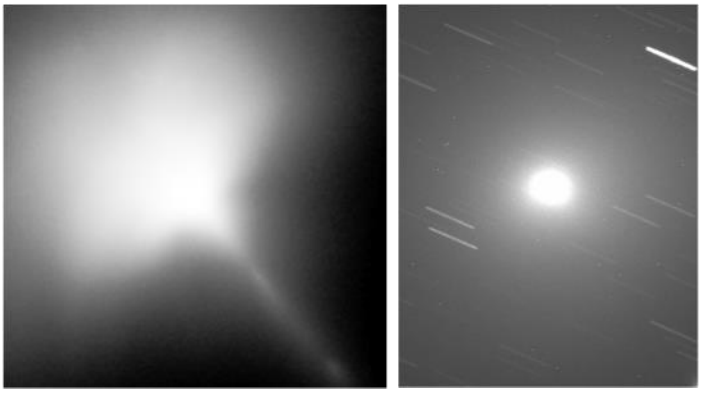


Figure 1: The unenhanced images used in later figures. Left image is from D. Schleicher and L. Woodney.

Further, when using the Radially Variable Spatial Filtering method, you may have to select the three kernel terms by trial and error to find the values that work best with your image. Additionally, there is a 10.8 Megapixel size limit to each image upload, which can be subverted by choosing a region smaller than 10.8 million pixels in area.

To help the explanations, we will provide the outcomes from different enhancement algorithms of the unenhanced images, shown in Figure 1.

All of the routines will add keywords to the image headers of the output images indicating the relevant parameters chosen.

2 Division by Azimuthal Average

Division by Azimuthal Average works by dividing each pixel's brightness by the average brightness at that radial distance from the nucleus. To do this, the image is first converted into ρ-θ (polar) coordinates with the optocenter at the origin – this is the "Unenhanced ρ-θ image" in the output. Moving along the vertical axis, ρ, moves you radially away from the optocenter, while moving horizontally moves you around the optocenter in a circle, the azimuth. In this format, each row of pixels represents different azimuths; however, because a given (ρ,θ) pixel likely does not exactly correspond to an (x,y) pixel in the original x-y image, each pixel is divided into 100 (10x10) subpixels, which are used to determine the value of those pixels. The size of these ρ-θ images is determined by the values for the ρ- and θ-axes that the user specifies. To assist in choosing these values, the program calculates the radii and circumferences of the largest circle completely contained in the image and of the largest circle partially contained in the image (see Figure 2). For practical reasons, the ρ-axis value cannot exceed the radius of the largest circle partially contained in the image. The θ-axis has a set maximum of 2500 pixels for image size reasons; this can be larger than the circumference of the largest circle partially contained in the image because each azimuth's pixel values are spread across the θ-axis anyway. For example, even though there are four times as many (x,y) pixels in the azimuth when ρ=4 than when ρ=1, both azimuths must spread across the entire θ range.

This routine then finds the average value of each row, excluding pixels that are outliers – these can be stars, cosmic rays, or other imperfections in the image – and divides each pixel's brightness value by that average. Notice that the "spreading" done to fit pixels at each ρ value along the θ-axis does not affect the average, as all pixels for a given ρ are spread equally. Once all the pixels have been adjusted, the resultant image is the "Enhanced ρ-θ image" in the output. Finally, this is reconverted to rectangular (x,y) coordinates to create the "Enhanced x-y image" in the output.

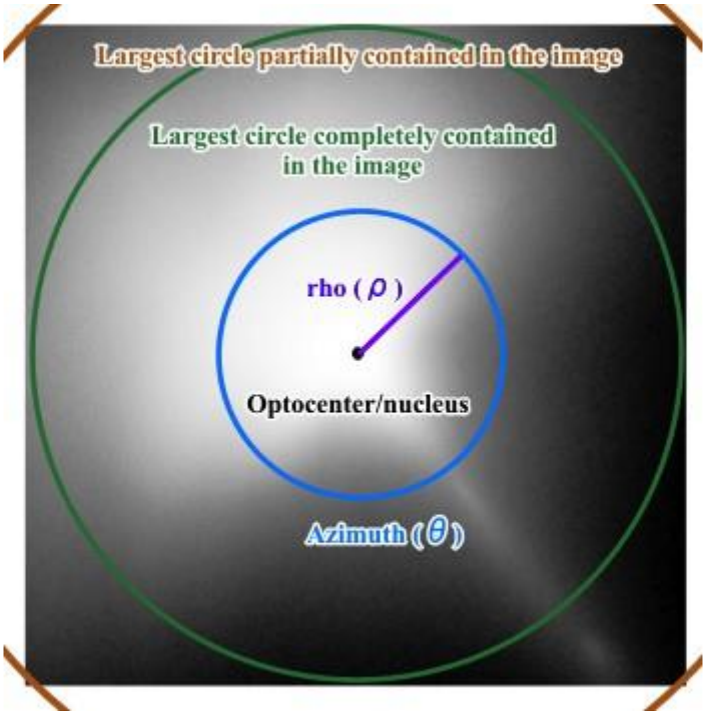

Largest circle partially contained in the image
Largest circle completely contained in the image
rho (ρ)
Optocenter/nucleus
Azimuth (θ)

Figure 2: Visual description of the parameters relevant to Division by Azimuthal Average.

Unless the ρ-axis value is set to be the radius of the largest circle partially contained in the image, there will be pixels in the initial image that are not considered in the routine. This shows up in the final enhanced image as a circle of enhanced pixels surrounded by zero-valued pixels (e.g. Figure 3). This distinct boundary may not show up immediately in the output PNG images displayed on the website, rather you may have to download the FITS image and adjust the brightness and contrast settings.

Even though the Division by Azimuthal Average routine excludes outlier pixels, it still has weaknesses when it comes to images with many stars or imperfections. This is especially problematic when an image has imperfections along the edge. Restricting the region of the image to enhance (to exclude those imperfections) can mitigate this.

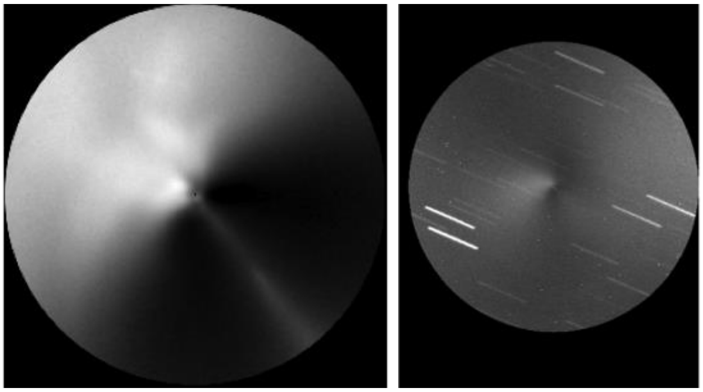


Figure 3: Comet images enhanced by Division by Azimuthal Average. The black background represents pixels that were excluded during the conversion to ρ-θ coordinates.

3 Division by Azimuthal Median

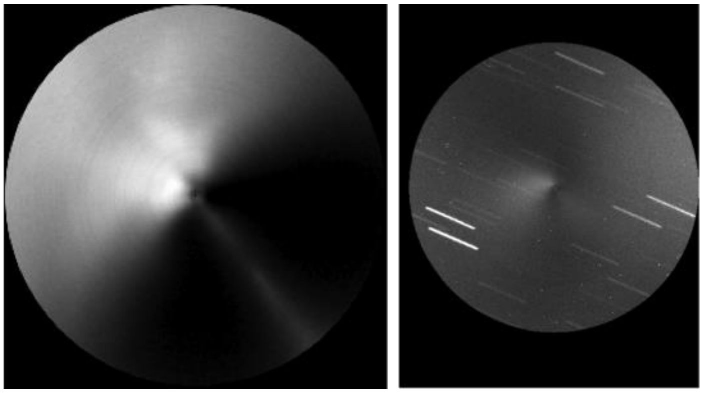


Figure 4: Comet images enhanced using Division by Azimuthal Median. Note the extreme similarity to the enhanced images shown in Figure 3.

Division by Azimuthal Median works by dividing each pixel's brightness by the median brightness at that radial distance from the nucleus. The routine is very similar to Division by Azimuthal Average, with the only differences being what value is used to divide the pixels at a given ρ, and there is no pixel exclusion in Division by Azimuthal Median. This is because finding the median is much less susceptible to outliers than finding the average. Therefore, Division by Azimuthal median is preferred over Division by Azimuthal Average for images with many stars or imperfections. In practice, however, there is no significant difference in enhanced images produced by these two routines; compare Figures 3 and 4 to see the output of both routines on the same images.

4 Azimuthal Renormalization

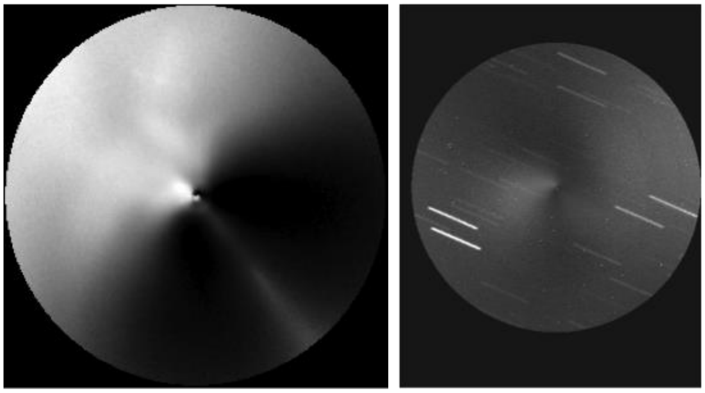


Figure 5: Comet images enhanced using Azimuthal Renormalization.

Azimuthal Renormalization adadjusts the brightness of pixels based on the maximally bright and minimally bright pixels along the azimuth for a given radial distance from the nucleus. Similar to the last two routines, it converts the image to ρ-θ coordinates, but instead of calculating the mean or median, it finds the maximum and minimum pixel values. Because this is very susceptible to outliers on both ends, the routine disregards any pixels that are outside of some number of standard deviations (e.g. 3) from the mean. Then, using these values, it scales the brightness of each pixel along the azimuth relative to the brightness of these two values. This means that for each different ρ, non-outlier pixels will all have the same brightness range.

This routine as well as Division by Azimuthal Median and Division by Azimuthal Average are all qualitatively similar enhancement techniques; Azimuthal Renormalization can be thought of as "Division by Azimuthal Range", which is precisely the case when the minimum non-outlier brightness in the azimuth is 0. Compare Figures 3, 4 and 5 to see the differences in the three. The question then arises: when should one technique be used over another?

In general, Division by Azimuthal Median has a clear advantage in that it is not as susceptible to outliers as the Average or Range – using this technique doesn't even require you to specify a rejection sigma – and Division by Azimuthal Average is less susceptible than Azimuthal Renormalization.

5 Division by $1/\rho$

Division by $1/\rho$, where ρ is the cometocentric distance in the sky plane, takes advantage of the property that comets are brightest at the optocenter, and dimmer further away. To counteract this, it multiplies the brightness of each pixel by its distance from the optocenter – i.e. it divides by the inverse ρ, which means the same mathematically. This wording, however, is preferred because brightness decays as $1/\rho$ in an ideal spherically symmetric uniform outflow, which can be used to roughly model the brightness distribution of dust in cometary comae, but not the brightness distribution of gas species. Because of this, gas images will not be properly enhanced by this routine, instead yielding an undesirable outcome, as seen in Figure 6 (right). This image is a CN image, and as such cannot be modeled with a $1/\rho$ profile. The left image in Figure 6 shows the outcome of the routine when applied to the dust (continuum) image.

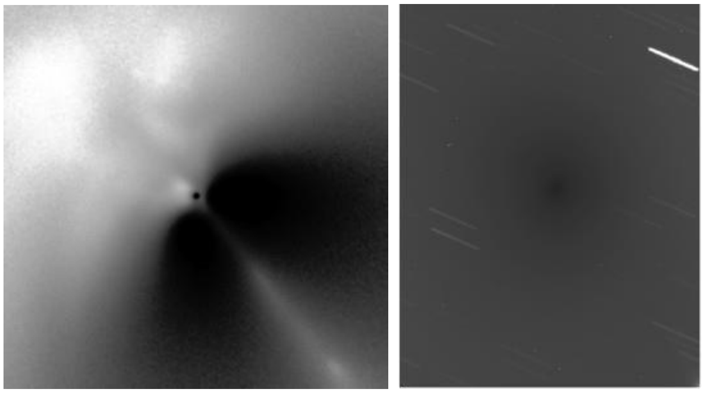


Figure 6: Comet images enhanced using Division by $1/\rho$.

This routine produces a characteristic "hole" around the optocenter, which is clearly visible in the left image of Figure 6. This is caused by the multiplication of the optocenter brightness by ρ=0 at the optocenter.

While this routine requires input values for the number of pixels in the ρ- and θ-axes, the entire region of the image will be enhanced regardless of the values chosen. This is because, unlike the previous routines that work by altering a pixel's values based on all the pixels at a given radial distance, this routine only looks at a pixel's distance from the nucleus/optocenter. However, these numbers of pixels in the ρ- and θ- axes do impact the size of the unenhanced and enhanced ρ-θ images. If those two images are not important to you, feel free to use the default values.

Additionally, because this routine enhances each pixel individually, as opposed to comparing them to surrounding pixels like the earlier routines, this routine is unaffected by outliers.

6 Radially Variable Spatial Filtering

Radially Variable Spatial Filtering is based on the property that features get larger the further they are from the optocenter, which occurs due to velocity dispersions in the coma. This routine adjusts the brightness value of each pixel by considering pixels on the boundary of a region, hereafter called the kernel, around that pixel. The kernel grows larger as it moves away from the optocenter, which provides a finer filter when working in the inner coma, where features are smaller, and a coarser one when in the outer coma, where features are larger.

For each pixel, this routine creates a square having a certain kernel width centered at that target pixel with 8 relevant pixels on that square: the four corners, and the four midpoints for the edges (Figure 7). Then the target pixel's brightness value is adjusted by the following formula:

$$I_{out}(x, y) = 1024 \cdot I(x, y) - 192\,(\Sigma I_{ep}) - 64(\Sigma I_{cp})$$

where I_{out}, $I(x, y)$, I_{ep}, and I_{cp} are the brightness values of the output pixel, the input pixel, an edge pixel, and a corner pixel respectively. To understand why this is useful, consider a region with flat (or near flat) brightness. In this case, $I(x, y) = I_{ep} = I_{cp}$, so the output value is

$$\begin{aligned} I_{out}(x, y) &= 1024 \cdot I(x, y) \\ &\quad - 192(4 \cdot I(x, y)) \\ &\quad - 64(4 \cdot I(x, y)) \\ &= 1024 \cdot I(x, y) \\ &\quad - 1024 \cdot I(x, y) \\ &= 0 \end{aligned}$$

which means that pixels with similar brightness levels to their surroundings will be darkened, while those that are different will stand out, with the brighter values being those highlighted.

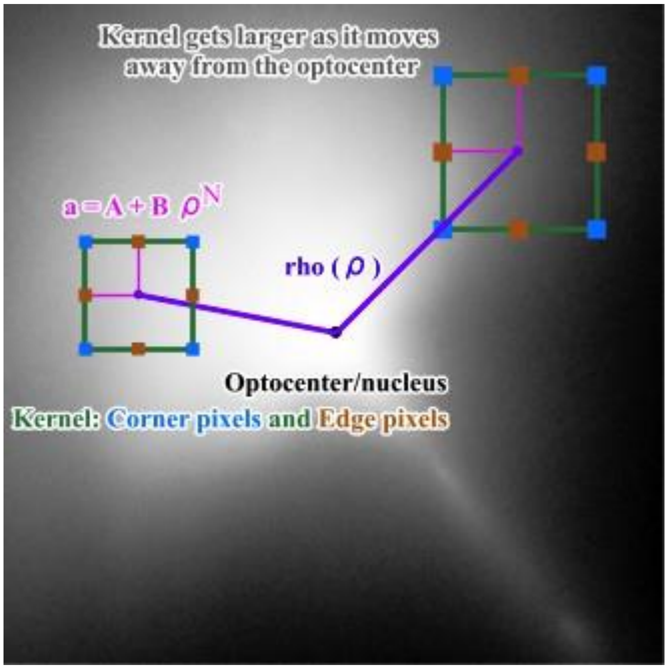


Figure 7: Visual description of some parameters required for running Radially Variable Spatial Filtering.

The Facility also provides the option to take the logarithm of each pixel value before applying the above formula. If the image contains any negative pixel values, the logarithm will obviously fail, and thus the presence of any negative pixel values will prevent the "log option" from being applied. To avoid this, one should ensure there are no negative pixels; one possible remedy is to uniformly increase all the pixel values by an amount larger (in absolute value) than the most negative pixel value. The header of the output enhanced image will reflect whether the “log option” was successful, based on the presence of absence of the keyword “logarithmic image”.

To run this routine, one needs kernel values A, B, and N to determine how the kernel grows as it moves away from the optocenter. The width of the kernel, $2a$, is computed using these values and ρ, the distance from the optocenter to the current pixel, according to the formula

$$a = A + B \cdot \rho^N$$

This means that A, B, and N affect the size of the kernel in different ways. Close to the nucleus/optocenter, $(B \cdot \rho^N)$ is not very large, so increasing the A term significantly increases the size of the kernel. As ρ increases, however, A’s contribution to the kernel width is quickly overshadowed by the exponential term.

As the N term is an exponential term, its influence grows as ρ increases; so increasing N provides the greatest effect on the size of the kernel in the outer coma. Closer to the nucleus/ optocenter, ρ^N does not change much with changes in N, as ρ is close to 1.

Finally, the B term is a linear term, which impacts the size of the kernel at all values of ρ.

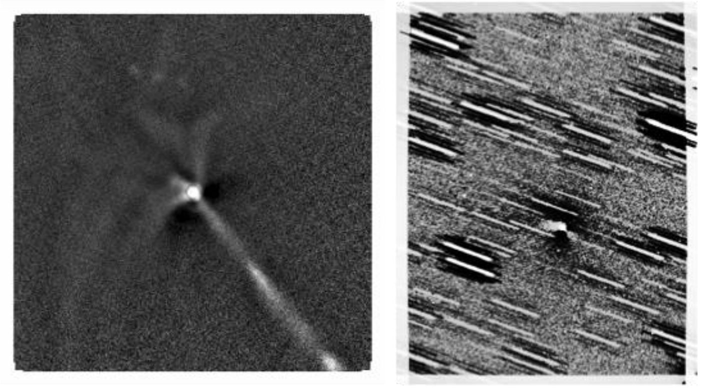


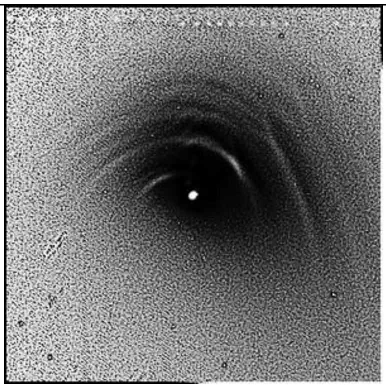
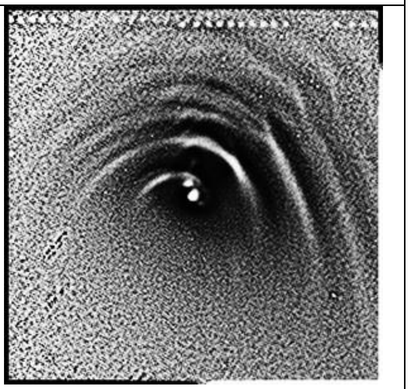
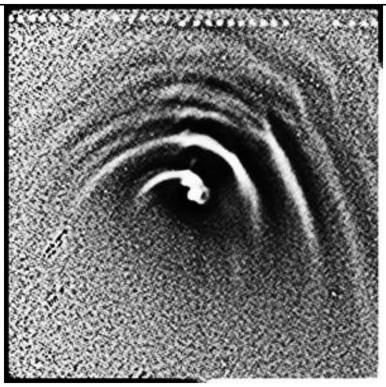
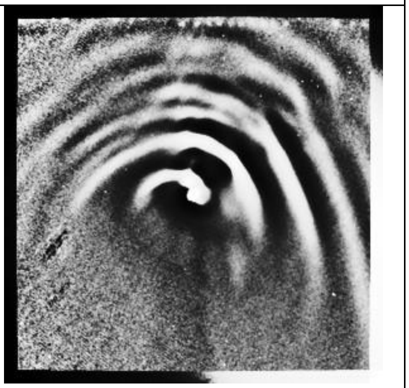
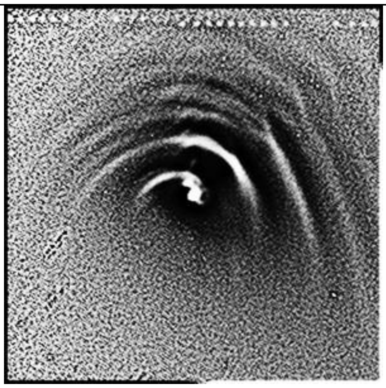
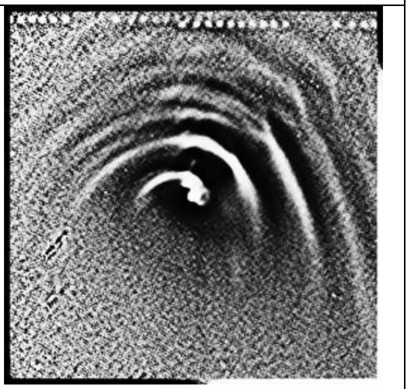
Figure 8: Comet images enhanced using Radially Variable Spatial Filtering. The left image used A = 0.2, B = 2, N = 0.3 with log option; in the right image A = 5, B = 5, N = 0.3 without log option.

This allows us to manipulate the size of the kernel at all distances from the optocenter: if the kernel is too large in the outer coma, reduce the kernel N term; if it is too large near the nucleus/optocenter, reduce the kernel A term; and if it is too large overall, reduce the kernel B term.

In general, the A and B terms should be on the order of 1 and the N term should be around 0.2; the N term in particular should be less than 1, or else the kernel will grow far beyond the size of the image. However, these values need to be adjusted for each individual image through trial and error.

Because this routine only compares each pixel to 8 surrounding pixels, outliers only affect the pixels in a box around them, producing dark shadows around the outliers, which can be seen in the right image in Figure 8 as dark outlines around the white streaks. Figure 9 shows how varying the values of A, B, and N affect the resultant image.

Figure 9: Image of comet Hale-Bopp enhanced with Radially Variable Spatial Filtering with different values of A, B, and N chosen.

		$B = 1$	$B = 4$
$A = 1$	$N = 0.1$		
	$N = 0.3$		
$A = 4$	$N = 0.1$		
	$N = 0.3$