Comet composite in PixInsight

V 1.1, 28/04/2024

Introduction

This write-up is the transcription of a webinar by Edoardo Radice on the processing of comet images in such a way that both the comet and the background stars appear "still" in the final composite. The webinar (in Italian only) is available at the following link:

https://www.youtube.com/watch?v=RmgjvC8c_yc

For the sake of brevity, this document contains almost no images, mostly just plain text.

Prerequisites

Given the (intentional) lack of images in this text, it is assumed that the reader has good familiarity with PixInsight, its underlying approach and its processes, namely: ScreenTransferFunction (STF), StarAlignment, CometAlignment, Convolution, ImageAlignment, ImageIntegration, AutomaticBackgroundExtractor, RangeSelection and PixelMath.

It is also assumed that:

- The subframes were guided on the stars with an exposure time short enough not to cause any visible drift on the comet's false nucleus (2-3 pixels at most);
- The raw frames have already been calibrated, e.g., with WBPP or a similar tool;
- All intermediate images are assumed to be saved in PixInsight's default format (.xisf).

The contents have been updated to the latest PI version available (1.8.9-2).

Description of procedure

1. Take the calibrated frames and register them with StarAlignment (default parameters).
2. From the set of star-aligned frames, use CometAlignment to produce a set of comet-aligned frames. The alignment reference image must be around halfway in the total exposure run. Save the relevant process icon.
3. To get rid of the stars, we run the set of comet-aligned images through ImageIntegration with the following settings:
    a. Quite aggressive pixel rejection[1] (*Sigma high* = 1.5);
    b. *Image Integration*: "Average", *Normalization:* "No Normalization", *Weights*: "Don't care";
    c. Run the process;
    d. Save the resulting stack as "comet[.xisf]"[2].

Now, we would like to use the image from the above step to separate the comet from the background stars, but we cannot use it "as-is" since this would also erase any background objects (e.g., nebulae) and would likely produce artifacts around the brightest stars. Therefore, our goal is to "mask" the comet and even out the background, which is not relevant in the comet-only image.

4. Clone the "comet" image into "comet_clone" and create an equally-sized solid white image with PixelMath:
    a. *RGB/K* = 1
    b. *Destination*: "create new image" to create a new solid-white image
5. Open CloneStamp, set the size to 30-50 pixels, and apply it to "comet_clone"; then CTRL+left click on the solid white image and start "painting" the comet in white. When done, click on CloneStamp's green tick to finalize the changes. This step will turn "comet_clone" into a mask, getting the white color from the solid white image previously created with PixelMath.
6. Apply RangeSelection to "comet_clone", fine-tuning *lower limit* parameter so that only the white mask is selected. We need to crank up the *Smoothness* (around 30) so that the mask has soft edges.
7. Further blur the comet mask by using Convolution with *StdDev* at 30 or 40.
8. Now we need to "map" the brightness gradient of the comet-only image. To this end, go back to the "comet" image and run it through AutomaticBackgroundExtractor with default settings and *Function degree* = 1 as we just have to extract the background sky including any residual gradients.
9. Rename the resulting gradient to "G".
10. Apply the range mask from step 7 to the "comet" image and invert it, as we have to protect the comet and not the background sky.
11. Use PixelMath to even out the background sky and remove any gradients:

---

[1] The most suitable pixel rejection algorithm depends on the number of frames. For example, Winsorized Sigma Clipping works particularly well with large image sets.

[2] If the integration result contains black areas after an STF screen stretch, turn *Sigma low* 4 (default) down to 3 or even much less (tested down to 0.5) and re-run the process. It could also be necessary to remove from the stack frames having the brightest background, for example because they were taken during twilight.

a. *RGB/K* = "med($T)+G-med(G)" so that the background sky does not change;

b. Tick *Replace target image*, otherwise the range mask won't work;

c. Run the process[3].

12. Save the resulting image as "comet_no_sky.xisf".

13. Re-open the CometAlignment process icon from step 2, which contains the star-aligned frames and the same alignment reference frame:

a. In the *Subtract* section: *Operand Image* = "comet_no_sky.xisf";

b. *Operand Type* = "Comet aligned";

c. Untick *Linear Fit* and tick *Normalize*;

d. *Output directory*: create a new folder named "stars_only";

e. Run the process and put the relevant process icon aside.

14. Back to ImageIntegration:

a. Pick the images from the "stars_only" folder;

b. Set *Image Integration:* "average", *Normalization:* "No normalization", *Weights*: "Don't care";

c. *Sigma high*: back to its default value of 3;

d. Run the process[4];

e. Once satisfied with the result, save the image as "stars_only.xisf".

15. To remove the stars from the comet image, bring up the CometAlignment icon from step 13:

a. In the *Subtract* section, pick "stars_only.xisf" as *Operand Image* (step 14.e);

b. *Operand Type* = "Stars aligned";

c. Untick *Linear Fit* and tick *Normalize*;

d. *Output directory*: a new folder called "comet_only"

e. Run the process.

16. Back to ImageIntegration:

a. Pick the image set from the "comet_only" folder;

b. Other settings as in step 14, *Image Integration*: "Average", *Normalization:* "No normalization", *Weights*: "Don't care", *Sigma high = 3*;

c. Run the process and save the result as "comet_only.xisf".

17. Bring up or open the following images (if they were previously closed down):

a. "comet_only.xisf" and rename its it id as "comet";

b. "stars_only.xisf"and rename its id as "sky".

18. Bring up the "comet" image and use AutomaticBackgroundExtractor:

a. *Function degree* = 1;

b. *Correction* = "Subtraction";

c. *Normalize* ticked;

d. *Replace Target Image* ticked;

e. Run the process.

19. Select the "sky" image, which becomes our target.

20. Open PixelMath:

---

[3] If, after applying PixelMath, a residual halo from the comet's coma or tail is still visible, it might be necessary to go back to step 5 e draw a bigger white mask around the comet (repeat steps 5 to 7).

[4] If the result contains dark patches especially around the brightest stars, it might be necessary to re-run the process with a lower *Sigma low* value (3 or even less, down to 0.5).

     a. *RGB/K* = "$T+comet-med(comet)"[5];

     b. Run the process on the "sky" image.

21. **Optional**: we can also perform step 20 after applying the range mask from step 7. Using the mask has the advantage of not adding noise from the comet-only image, but could produce "seaming" problems between the comet (superimposed via PixelMath) and the star-only image, which could be containing some residual halo from the comet. This being the case, it's better not to use the mask: rather, it is advisable to run the whole composite through some selective noise reduction tool such as NoiseXterminator.

The above procedure will yield an image that is still linear and should contain both the comet and the stars appearing still. Subsequently, one can go on with the usual workflow, for example:

- Gradient removal;
- SPCC;
- Deconvolution;
- Delinearization with one of the many stretching tools available in PixInsight.

---

[5] If we want to enhance the comet's brightness and the background sky is even enough, a multiplicative coefficient can be applied, e.g., "$T+1.5*(comet-med(comet))". Try not to overdo it, anyway the most appropriate value of the coefficient must be found by trial and error.
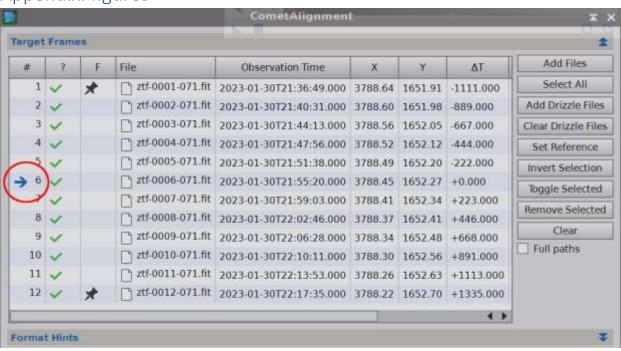
Appendix: figures

Figure 1 – The CometAlignment process window where an alignment reference frame has been selected about halway in the sequence (red circle). In this case, the number of frames is even (12), so the most reasonable candidates are frame 6 or 7.