HBComposer - User Guide
Version 1.4.4 rev.2 - © 2025 Marco Ristuccia

Contents

List of Figures

Introduction

HBComposer is an unofficial tool for pre-processing *Hasselblad* RAW files. It is able to perform the following *3FR* to *3FR* operations in the RAW domain:

- Frame-Average
- Frame-Median
- Advanced Flat-Field correction

Since the produced files are still raw files, they can be further processed in any RAW developer, especially in *Hasselblad Phocus* **without losing the advantage of the *Hasselblad Natural Color Solution*** (*HNCS*). The advanced flat-field correction is able to fix the infamous ***Hasselblad* PDAF banding** issue and any **sensor tiling** artifacts generated by digital backs when used in conjunction with technical cameras and shifted wide-angle lenses.

****** DISCLAIMER *** I am in no way connected to Hasselblad, nor does Hasselblad endorses this project. This tool is the result of my personal initiative and the files produced are not guaranteed to be 100% OEM compliant. Moreover, Hasselblad may change their RAW file specs in any moment causing the tool to stop working.***

Product installation and execution

Getting HBComposer

HBComposer is offered for free. You can download it from my web site under the following link:

https://photography.marcoristuccia.com/hbcomposer-an-hasselblad-raw-file-composer/

Support my work *Please consider donating to allow me spend more time in improving it and fix any bugs. Thank you!*

Installing HBComposer

The *HBComposer* installation file is a typical *Apple DMG* archive. Once downloaded, just double click on it, a window will show up (*Figure 1*).

Just drag the *HBComposer.app* icon onto the *Applications* one. The application will be installed, if a previous version is already present, *macOS* will ask whether to overwrite the old version.

Installing Java runtime

To run *HBComposer*, a *Java* runtime needs to be installed on your system. The minimum supported version is ***Java SE 11***, although it is highly recommended to install ***Java SE 21 LTE*** (Long Time Support) or ***Java SE 24*** (the latest version to date).
You can download Java from the following link:

https://www.oracle.com/java/technologies/downloads/?er=221886#jdk21-mac

Running HBComposer

Once installed, you will find *HBComposer* among the *macOS* applications. Just click on it. Please be patient, the application startup may take a few seconds since *macOS* needs to verify the application and instantiate the Java runtime environment.

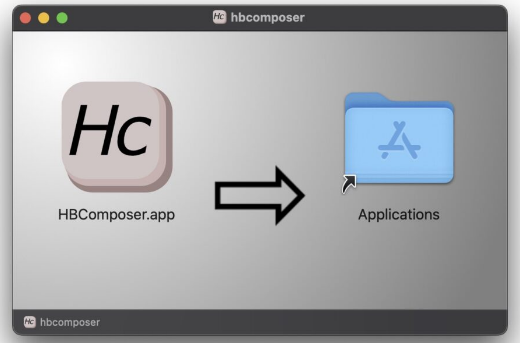

Figure 1: HBComposer installation window

Since I'm officially part of the Apple Developer Program, I have digitally signed and notarized *HBComposer* as per *Apple* recommendations. Security wise, *macOS* should not complain about opening it, apart from warning you that the application has been downloaded from *Internet* and asking for your authorization to proceed.

Since the application will not be downloaded from the official *Apple Store* however, you may need to slightly relax the security settings by allowing applications from "**Apple Store & Known Developers**".
Such an option can be found under **Settings -> Privacy & Security** as shown in *Figure 2*.

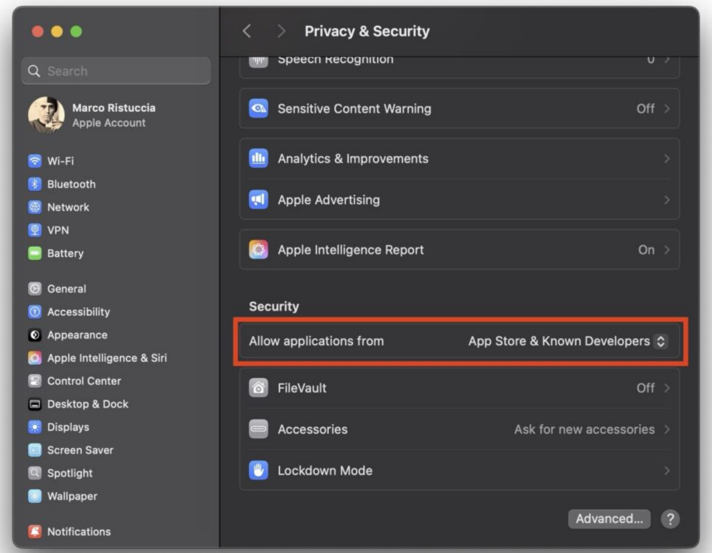

Figure 2: macOS Privacy & Security settings

User interface overview

In *Figure 3*, a screenshot of the graphical user interface (GUI) is shown.

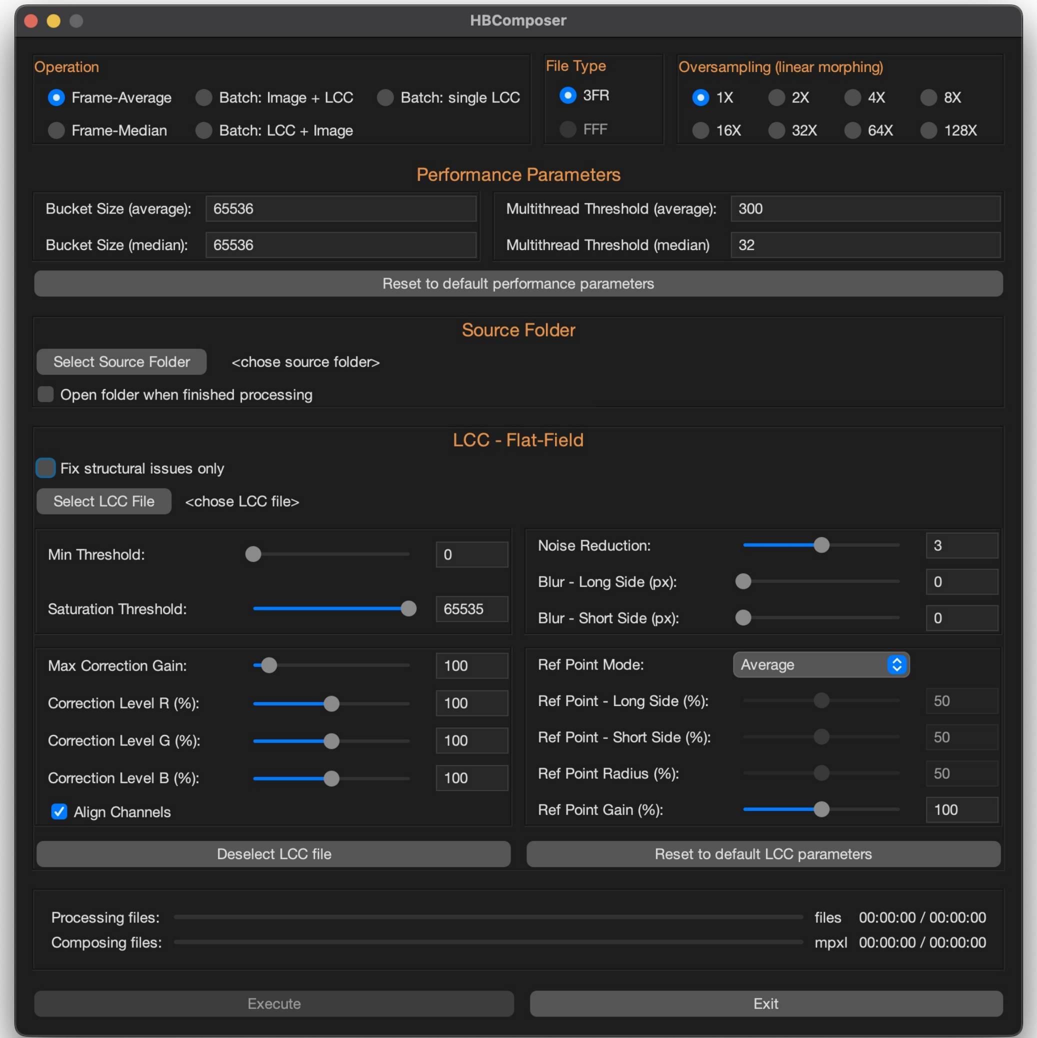

Figure 3: HBComposer - User interface

In the following sections, the general configurations are described first. Then, each operation will be fully explained in dedicated sections.

File Type

The tool has been initially designed to operate on both *3FR* and *FFF Hasselblad* RAW files.
Due to technical reasons though, only *3FRs* are supported at the moment. For such reason, the *FFF* option is grayed out and not selectable.

Performance parameters

These parameters only apply to the *frame-average* and *frame-median* operations.
There is a couple of them for each of the two operations:

- **Bucket Size**
  Represents the sliding memory window which buffers the part of the image that is kept in memory while processing the frame compositions. It is represented in bytes. The default value is a good compromise for the majority of systems. If your system happens to have a large amount of RAM (> 36GB) you can try increasing this value (keep it a multiple of 4096). This may speed up the composition operation. Be aware however that you may experience memory over-allocations, in such case the application may slow down or even crash with an unexpected exception. If this happens, just restart *HBComposer* and reset to the default performance values. If you decide to venture into changing this parameter, you'll need to find the sweet spot.

- **Multithread Threshold**
  Represents the maximum number of parallel processing threads. The default value is a good compromise for the majority of systems. If your system has a large amount of CPU cores (> 14) you can try increasing it. This may speed up the composing operation. If you decide to venture into changing this parameter, you'll need to find the sweet spot.

NOTE: ***all performance parameters are remembered*** *after closing and reopening the application. They can be reset to their default values by clicking the dedicated button.*

Source Folder

A user selectable folder where all operations are performed. All images to be processed must reside there. The processing is not recursive, therefore any sub-folder won't be included. You have the option to automatically open such folder after the processing is finished.

Processing progress & statistics

In the bottom area of the application window there are progress bars that are updated while processing. The composition operations will process files first, then compose them, so both progress bars are updated. The batch *LCC* operations only update the first progress bar.

Operations

Frame-Average

This operation will perform a composition of all images found in *Source Folder*, producing a single RAW image which represents a pixel-by-pixel average of all processed images. The output file will take the name ***FA.3FR*** and is automatically ignored by all operations, so you can keep it there and re-run *frame-average* or *frame-median* without it to be part of the composition. It will be simply rewritten.

In a typical situation, you place your camera on a tripod and use the featured internal intervalometer to shoot a series images of the same scene at a specified time cadence.

Figure 4 presents an example of the results that can be achieved with *frame-average*.

(a) (b)

Figure 4: (a) Single frame (b) Frame-average of 160 RAW images

It shows the result of a frame-averaged composition of 160 frames, each one shot at ISO 64, f8 and 0.8s exposure time without any ND filter. It produced a RAW image which is roughly equivalent to a 2-minutes long exposure taken with an 8-stops ND filter on.

Frame-Average may be useful for the following purposes:

- **It reduces the overall image noise, especially in the shadows.**
  Noise is reduced by the square root of the number of frames averaged. So, if you average 4 images, noise will be reduced to 1/2 of the noise you'd have in a single frame.

- **On moving subjects, it produces an effect similar to a long-exposure.**
  An advantage to a real long-exposure is that you can operate in daylight without the need of an ND filter.
  A disadvantage is that *Hasselblad* cameras do not currently allow shooting images at a speed greater than one frame every two seconds. Thus, **fast moving objects may not be represented as a continuous trail** in the final composed image. To try to partially compensate for this issue, an **oversampling** function has been added to the tool (see below for more info on this).

Frame-Median

This operation will perform a composition of all images found in the *Source Folder*, producing single RAW image which represents a pixel-by-pixel median of all processed images. The output file will take the name ***FM.3FR*** and is automatically ignored by all operations, so you can keep it there and re-run *frame-median* or *frame-average* without it to be part of the composition. It will be simply rewritten.

In a typical situation, you place your camera on a tripod and use the featured internal intervalometer to shoot a series images of the same scene at a specified time cadence.

Figure 5 presents an example of the results that can be achieved with *frame-median*.

(a) (b)

Figure 5: (a) Single frame (b) Frame-median of 34 RAW images

It shows the result of a frame-median composition of 34 frames, each one shot at ISO 64, f8 and 1/10s exposure time without any ND filter.

As shown in the sample image, on scenes containing moving objects like **cars, persons, etc...** *frame-median* may be useful to **cancel them out from the final image**. A crowded place where many people pass by can be turned into an empty space. This can be useful for example in architecture photography.

Like with *frame-average*, oversampling could help in smoothing out some gaps, like on moving clouds for example, although it will increase the calculation time (see an example of oversampling results below in the dedicated section).

Oversampling You'll find this option in the top-right area of the application window. This feature adds "fake" intermediate frames where each pixel is created as a linear interpolation between each couple of real frames. Another way of seeing it is that it weights the frames differently in the composing operation, giving more weight to the ones in the middle of the sequence. It does that with the goal of smoothing out some gaps that may arise in presence of moving objects in the scene when the time interval between captured frames is long.
In *Figure 6* you'll find a result comparison based on a *frame-median* composition of **100 images shot at a frequency of 1 every 10 seconds**. Without oversampling (left), gaps in the clouds are visible in the upper and central right part of the image. An oversampling of 16X succeeded in smoothing things out (right).

(a) (b)

Figure 6: (a) Frame-median with no oversampling – gaps are present in the clouds (b) Frame-median with 16X oversampling – gaps are in the clouds are smoothed out

Flat-Field Correction

HBComposer features a *flat-field* correction operation and offers an advanced set of fine-tune parameters.

This technique consists in taking an additional frame, called *flat-field* or *LCC*, while holding a thick peace of plexiglas in front of the lens. This special image should be acquired under the very same conditions as the photo it will correct. Aperture, focus, shift/tilt levels, and light conditions need to match. Shutter duration can be adjusted to get a correct exposure. A good *LCC* exposure is key to the effectiveness of this technique.

From now on, the words *flat-field* and *LCC* will be used alternatively to indicate this special frame.

Figure 7 shows a typical histogram of a well exposed *flat-field* image.

Figure 8 shows a typical example of a flat-field image taken with a technical camera and a shifted (raised) lens.

Figure 9 explains how the *LCC* correction process works. The *flat-field* capture is applied to the original image to normalize it.

Goals of flat-field correction

Flat-field correction allows the reduction and often the full elimination of the following image issues:

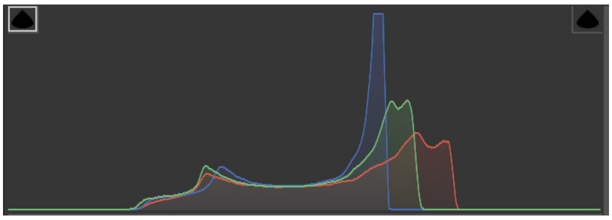

Figure 7: Histogram of a well exposed flat-field (LCC) image

Figure 8: Example of Flat-Field (LCC) image

- **Color cast** when taking images with **shifted lenses**. The issue is particularly evident on non-BSI sensors when used in conjunction with symmetrical wide angle and ultra wide-angle lenses, such as the *Schneider-Kreuznach* technical ones, and some *Rodenstocks* as well.

- **Vignetting** when taking images with **shifted lenses**. The issue is particularly evident if using symmetrical wide angle and ultra wide-angle lenses, such as the *Schneider-Kreuznach* technical ones, and some *Rodenstocks* as well.

- **PDAF sensor banding**, an issue that has been discovered by *GetDPI* forum member *@diggles* while using use the most recent *Hasselblad CFV-100c* digital back together with a *Schneider-Kreuznach Apo-Digitar 5.6/35 XL* lens. It shows up when shifting this lens, especially on uniform areas like the sky. In *Figure 10* an example of PDAF banding fix performed by *HBComposer*'s *flat-field* correction is presented.
  You can follow the forum discussion under the following link (click on it):

  https://www.getdpi.com/forum/index.php?threads/hasselblad-100c-and-35xl.75781/

  Figure 10 shows an example of PDAF banding fix performed by HBComposer's flat-field correction.

Figure 9: Flat-field application process

(a) (b)

Figure 10: (a) Original image - horizontal PDAF banding and vignetting present (b) Processed image - horizontal PDAF banding and vignetting corrected

- **Sensor tiling**, an issue which makes evident in the final image the different tiles a digital sensor is typically composed of. The issue is particularly evident on non-BSI sensors when used in conjunction with symmetrical wide angle and ultra wide-angle lenses, such as the *Schneider-Kreuznach* technical ones, and some *Rodenstocks* as well. In *Figure 11* an example of sensor tiling fix performed by HBComposer's *flat-field* correction is presented.

- **Dust spots** present on the sensor glass which are visible in the final image as darker circular or elongated shapes. In *Figure 12* an example of dust spot fix performed by HBComposer's *flat-field* correction is presented.

IMPORTANT NOTE: unfortunately, in case of extreme vignetting or CA the flat-filed algorithm will not be able to correct everything. High noise and residual color casts need to be corrected later on in the image processing chain, in the RAW developer or in post-production.

(a) (b)

Figure 11: (a) Original image - sensor tiling, color cast and vignetting present (b) Processed image - sensor tiling, color cast and vignetting corrected

(a) (b)

Figure 12: (a) Original image - dust spot and vignetting present (b) Processed image - dust spot and vignetted corrected

Operations and flat-field correction

You can apply a *flat-field* correction to the following operations:

- **Frame-Average and Frame-Median**
  In such cases, the *flat-field* correction is applied to the final composed image before saving it to disk. Select the *LCC* file and adjust all the parameters before executing the composition.

**** NOTE: the LCC file must not be placed in the Source Folder together with the images to be composed, If it is put there, it will be included in the composition as an additional frame. You can place it in a sub-folder instead so that it won't be composed but only applied at the end.*

- **Batch LCC process on image couples**
  In this case, images in *Source Folder* need to be arranged in couples so that, when alphabetically sorted, each *LCC* frame will immediately follow (Image + LCC) or precede (LCC + Image) its target image.
  The corrected images produced by *HBComposer* will keep the original name with the additional ***__HBC*** suffix.

- **Batch image process with a common LCC image**
  A single *LCC* is applied to all files present in *Source Folder*. In such case, you need to manually select the *LCC* image as you would do with *frame-average* and *frame-median*.
  The corrected images produced by *HBComposer* will keep the original name with the additional ***__HBC*** suffix.

Flat-field correction modes

Starting from version *1.3.0b*, *HBComposer* offers two ways of applying the *LCC* to a RAW image:

- **Correction of structural anomalies only**
  It only tries to fix PDAF banding, sensor tiling and dust spots. Usually, such anomalies are not addressed by the *flat-field* correction implemented in RAW developers like *Hasselblad Phocus* and *Adobe Lightroom*. After you've performed this step through *HBComposer*, you'll need to re-apply the *LCC* through such applications to correct color casts and vignetting, as you would do in normal circumstances. This allows for maximum flexibility.
  With this correction mode a large blur radius close to 64 or even greater is recommended. A bit of noise reduction may also help but usually it is not necessary (see below for more details on noise reduction).

- **Full LCC correction**
  It tries to fix all anomalies: PDAF banding, dust spots, sensor tiling, color cast and vignetting. After this step *LCC* correction is fully done and you won't need any further assistance from external RAW developers to complete the correction.
  With this correction mode, a blur radius close to 0 is recommended to fix PDAF banding and sensor tiling. Since the recover of vignetting increases the noise level, a bit of noise reduction (level 2 or 3) is also suggested to get clean results (see below for more details).

Flat-field correction settings

HBComposer offers a set of fine-tuning parameters for *flat-field* correction. Normally, you would leave everything at default values, with perhaps the exception of *Ref Point* selection.
Here below a brief description of each setting is given:

Fix structural issues only As described above, this flag selects the flat-field mode.
When enabled, it restricts the correction process to PDAF banding, sensor tiling, and small dust spots only. Color cast and vignetting need to be fixed through an external RAW developer and its *LCC* correction abilities.
This flag is disabled by default, so *HBComposer* will normally perform a full LCC correction.

NOTE: when enabled, this option will increase the execution time due to the large blurring radius that needs to be applied.

Noise Reduction Applies a bilateral noise reduction to the final RAW image which is proportional to the level of vignetting recovery performed by the *flat-field* correction. This clever way of applying noise reduction will be more aggressive in the areas that need it most, while excluding the areas untouched by *LCC* correction.
A radius of 2 or 3 is usually enough for light to medium degraded LCC images. A value larger than this may be necessary for extremely degraded ones.
This setting is mostly needed when using the full *flat-field* correction mode (flag *"Fix structural issues only"* deselected) since in this mode vignetting is recovered by *HBComposer*. If you only fix structural issues, then noise reduction should be performed by the RAW developer used to complete the *flat-field* correction.

Min Threshold Sets the minimum threshold value to consider a pixel of the *LCC* image valid for correction (0-65535). A value below this threshold leaves the matching pixel of the final image uncorrected. Normally, *Hasselblad* RAW files have a black level set to 4096, so anything below that would perform a full correction. You may need to increase it above 4096 if the sensor has some defective pixels that would create problems in the final image. This is really rare though.
If unsure, leave it at its default value (0).

Saturation Threshold Sets the maximum threshold value to consider a pixel of the *LCC* image valid for correction (0-65535). A value above this threshold leaves the matching pixel of the final image uncorrected. Normally *Hasselblad* RAW files have a maximum pixel level set to 65535. You may need to decrease this setting below that if the sensor has some defective pixels that would create problems in the final image. This is really rare though.
If unsure, leave it at its default value (65535).

Max Correction Gain The *flat-field* correction algorithm calculates an amplification factor for each pixel of the target image. This setting defines its maximum (peak). The algorithm will cap the amplification factor to this value. Normally, it can be kept high. You may need to lower it down in case of clipping.
If unsure, leave it at its default value (100), or even 1000.

Correction Levels (R, G, B) (%) Sets the percentage of *LCC* correction to be applied on each color channel. Normally it should be kept at 100%. You may increase it together with the other channels to over-correct, or decrease to under-correct. If the three sliders are disjoined and set differently from each other, this correction can help in fine-tuning any global color cast, especially if the chosen reference point is not perfectly neutral (see below).
If you still see some banding or sensor tiling, especially when only fixing structural issues, you may want to increase the correction percentage.
The *Max Correction Gain* caps the maximum amplification factor resulting from this setting.
If unsure, leave the three levels aligned and at their default value (100%).

Blur (long / short side) Sets the pixel radius of *LCC* blur along each side of the frame. This setting has two different meanings and consequent behaviors depending on the selected *LCC* correction mode:

- **"Fix structural issues only" option disabled**
  Increasing it will reduce the noise in the final image, but it will reduce the effectiveness of any correction of structural image issues, like dust spots, sensor tiling and PDAF banding. **For PDAF banding correction, the default value of (0, 0) is recommended**. A slight blur along the long side (1-2px) may still work and also reduce the noise. It will vary from case to case.

- **"Fix structural issues only" option enabled**
  Sets the blur pixel radius of the *LCC* image when applied to itself before being used to fix the RAW image. As opposed to the preceding case, here **the blur value must be high (64 pixels or larger) to correct PDAF banding and sensor tiling**.

Ref Point Mode The *flat-field* correction algorithm needs a reference value taken from the *LCC* image which will be used to normalize the final image in terms of colors and brightness. It sets the values (one for each RGGB channel) where amplification factor is considered to be 1. To over-simplify the matter, you can consider it as the white balance point. This setting allows to choose different ways of calculating it:

- **Average**
  The average of the full *LCC* image is used.
  In presence of high vignetting, it may produce a dark image, which you can try to counter balance by using the *Ref Point Gain* setting (see below),
  When a high color cast is present in the *LCC* frame, the full average may leave some residual global color cast in the final image. In this eventuality, you can disjoin the *Correction Levels (R, G, B) %* sliders and try to counter-balance the color cast.
  If some (under) clipping in the final image happens (usually pink areas), try to increase the *Ref Point Gain* (see below).
  In presence of any of the above problematic cases however it is advisable to switch to *Custom Selection* mode and manually choose a good neutral LCC point/area.
  This is the default selection.

- **Custom Selection**
  Offers the option of selecting a specific point of the *LCC* (*Ref Point Radius* = 0) or a circular averaged area (*Ref Point Radius* is > 0). This kind of selection works best in cases where a significant degradation in a zone of the LCC image is present. Just place the reference point/area so that it keeps the degraded zone excluded (see *Ref Point (long/short side) (%)* and *Ref Point Radius (%)* below).

Ref Point (long / short side) (%) These sliders are active when *LCC Custom Selection* mode is selected. They help positioning the reference point along the X-Y plane in percentage values, like shown in *Figure 13*.

If unsure, leave the *Ref Points* to the default value (50%, 50%). This places the reference point at the very center of the *LCC* frame, which works well on average use-cases that usually only have a slight amount of peripheral aberrations.

NOTE: unfortunately, the tool does not feature a live preview of the reference point/area placement nor of the LCC correction results. Multiple attempts may be needed to nail the right spot.

Ref Point Radius (%) When *Ref Point Mode* is set to *Custom Selection*, *HBComposer* offers the possibility of increasing the area around the chosen reference point. The area will cover a circle of a specific radius (percentage) around it. A radius of 0% corresponds to the very single reference point. The maximum radius (100%) corresponds to a circle around the reference point having a radius of half the diagonal of the *flat-field* image. Values within the area will be averaged.
If unsure, leave the radius at its default value (50%).

Ref Point Gain (%) In some situations, usually in presence of a highly degraded *flat-field* image, and especially when including in the *LCC* reference area the black corners caused by reaching the extreme end of a lens IC, the produced image may result dark or even under-clipped (pink color areas). Increasing it may help bring the produced image to the right brightness level and avoid under-clippings as well. This parameter shouldn't be used normally, as it would be better to define a proper *LCC* area.

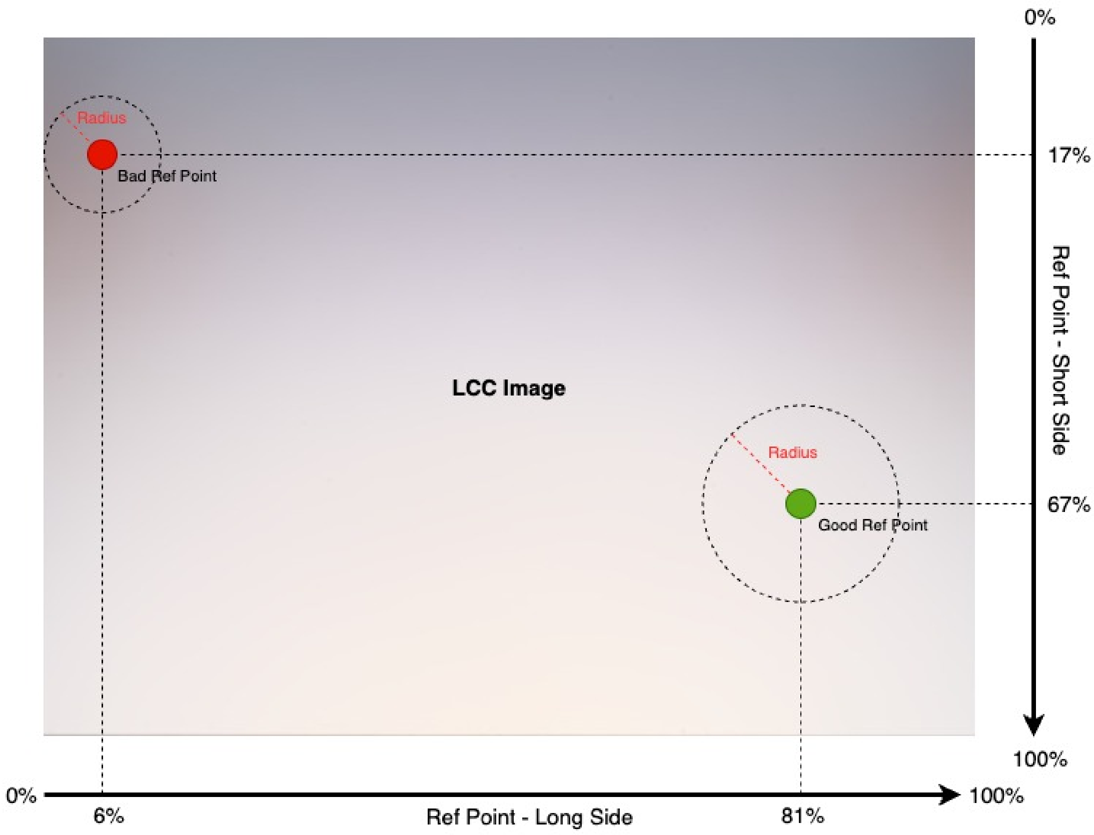

Figure 13: A scheme indicating how to use the Ref Point sliders to place the LCC reference point.

If unsure, leave it at its default value (100%) and use the *Custom Selection* mode to select a good zone of the *LCC* which excludes such bad areas.

*NOTE: Please be aware that **all flat-field parameters are remembered** after closing and reopening the application. You should always check their values when reopening HBComposer for a new project. They can be reset to their default values by clicking the dedicated button.*