OSC Dual-Band to the SHO Palette

Applies to: one-shot-colour data through a dual-band filter (L-Ultimate, L-eNhance, L-eXtreme), stacked elsewhere (APP, WBPP, Siril) and brought into PixInsight as a linear master. **Status:** working reference. Every step has been checked against current vendor documentation, and anything that departs from vendor guidance is flagged in the **Check** boxes.

The workflow at a glance

STAGE 1 LINEAR REPAIR — linear

1. **Load master + icon set**
2. **STF autostretch** — screen only, survey gradient and edges
3. **DynamicCrop** — every stacking artefact gone
4. **GraXpert background extraction** — show background model, verify it
5. **BlurXTerminator** — defaults, linear only

optional calibrated route: BXT (Correct Only) → SPCC narrowband → BXT (sharpen)

STAGE 2 THE HINGE — linear to nonlinear

6. **Stretch** — HT Stretch Linked RGB, then reset STF

STAGE 3 CLEANUP + SPLIT — nonlinear

7. **NoiseXTerminator** — never before step 5
8. **StarXTerminator** — generate star image + unscreen
9. **GHS tweak on stars image** — optional, knocks back bright cores

STAGE 4 COLOUR — starless only

10. **NarrowbandNormalization** — palette on HO, small OIII boost
11. **Global saturation curve** — mask prep, not a look decision
12. **Build yellow + blue masks** — one MaskBlur pass each
13. **Blue mask → curves** — OIII to teal: blue up, green back
14. **Yellow mask → curves** — Ha to copper and gold, drop green shadow

STAGE 5 FINISH — recombine

15. **Contrast** — HistogramTransformation, or gentle S-curve
16. **Recombine stars** — screen blend, never addition
17. **Final NoiseXTerminator** — catches chroma noise the curves raised

the stars image parked at step 8 rejoins the chain at step 16

The whole logic in one line: everything that mathematically requires linear data happens before step 6, everything that is an aesthetic decision happens after it. If you remember one thing, remember where the stretch sits.

NON-NEGOTIABLE

- BXT runs on linear data only, after crop and gradient removal, before any noise reduction.
- Crop stacking artefacts first. Ragged white borders break the BXT network outright.
- NarrowbandNormalization needs a stretched, starless image to give you its full toolset.
- Unscreen at extraction must pair with a screen blend at recombination.
- Palette stays on HO. It describes the data you hold, not the look you want.

JUDGEMENT CALL

- Whether to insert SPCC in narrowband mode between two BXT passes.
- How hard to push the OIII boost at step 10.
- Whether the bright stars need the GHS knockdown at step 9.
- Histogram versus S-curve for the contrast at step 15.
- How far to run saturation before the image starts to look manufactured.

Stage 1: Linear repair (steps 1 to 5)

1. Load the master and the toolset

Drag the integrated FITS and my process icon set into an open PixInsight workspace. The icon set carries the PixelMath icons used later (stretch, masks, rescreen). On anything below 4K, right-click → Arrange Icons to get them all visible.

2. STF autostretch

Open ScreenTransferFunction, hit the nuke icon. This is a display stretch only, the underlying data is untouched. Use it to survey the frame before you touch anything:

- Where is the gradient, and which way does it run?
- How bad are the stacking artefacts at the edges?

Naming those two problems now is what lets you verify the fix later.

3. DynamicCrop

Pull in from all four corners until every stacking artefact is gone. Accept the plate-solve warning, since the astrometric solution is discarded by the crop.

> **Check: this step is not optional.** RC Astro's own notes flag white or corrupt image borders as a known failure mode: they cause the BlurXTerminator neural network to overflow and output white for any tile touching the border. Crop before deconvolution, not after.
>
> If you use SPCC (see the box at step 5), do not accept the plate-solve loss casually. Re-solve with ImageSolver after cropping, or crop before solving.

4. GraXpert background extraction

Scripts → Toolbox → GraXpert. Defaults are the right starting point. **Tick "Show background model".**

The background model is your evidence. Compare it against the gradient you identified at step 2. If the model looks like the gradient you saw, the correction is honest. If it contains structure that looks like your target, the tolerance is wrong and it is eating nebulosity.

> **Check: confirmed.** GraXpert's own script documentation states gradient removal should always be the first processing step after cropping, and notes that because the image is not yet colour calibrated at this point the script applies an **unlinked** STF to the result. A linked STF is the more common habit here. Both are only display choices, but unlinked is the vendor default for a reason: on uncalibrated data a linked stretch can hide a residual colour cast.

5. BlurXTerminator

Defaults, applied to the whole image. With GPU acceleration this is seconds even on APS-C. The gain is strongest on longer focal lengths; at short focal lengths expect a modest tightening rather than a transformation.

> **Check: the two hard rules from RC Astro.**
>
> 1. **BXT requires linear data.** From AI4 onward this is a hard rule, not a preference. A stretched image processed by BXT is not deconvolution in any meaningful sense and the result will be sub-optimal.
> 2. **No noise reduction before deconvolution, ever.** BXT's network is trained to recover detail in the presence of noise, and most noise reduction destroys exactly the low-contrast fine-scale information it needs. The ordering below (BXT at step 5, NXT at step 7) is correct, and reversing the two would be a genuine mistake.
>
> **Divergence: colour calibration is missing from this workflow.** RC Astro's documented ideal position for BXT is right after integration, channel combination, **and perhaps colour calibration and gradient removal**. Their recommended pattern for best colour is: run BXT in **Correct Only** mode, then run precision colour calibration (SPCC), then run BXT again with your sharpening settings. In their AI4 testing this gave more consistent SPCC fits.
>
> As written, this sequence skips calibration entirely and lets NarrowbandNormalization at step 10 carry the colour balance. That is defensible for dual-band data heading to a synthetic palette, but it is a shortcut, not a rule. For the calibrated route, insert after step 4:
> `GraXpert → BXT (Correct Only) → SPCC in narrowband mode → BXT (sharpening)`

Stage 2: The stretch (step 6)

Apply the **HT Stretch Linked RGB** PixelMath icon. Then reset the STF, because the data is now genuinely nonlinear and no longer needs a display stretch.

Any stretch you are comfortable with substitutes here: GHS, HistogramTransformation, MultiscaleAdaptiveStretch. The specific tool matters far less than the position in the chain.

Stage 3: Nonlinear cleanup and separation (steps 7 to 9)

7. NoiseXTerminator

Defaults. You are looking for the salt-and-pepper chroma mottling in the background to go smooth in one pass. Only reach for the denoise/detail sliders if you can see a specific problem.

> **Check: position is fine, timing is flexible.** RC Astro state NXT processes linear data just as easily as nonlinear: internally it stretches, denoises, then precisely reverses the stretch, so the network barely cares. Running it after the stretch, as here, is perfectly valid. The rule that actually binds is the one already satisfied, never before deconvolution.

8. StarXTerminator

Tick **Generate star image**. Tick **Unscreen**.

> **Check: unscreen is correct here, and only here.** RC Astro are explicit: if you extract stars from a **linear** image, do **not** select Unscreen, because that option is for nonlinear data, and simple subtraction gives the best star colour accuracy on linear data. Because this workflow removes stars after stretching, Unscreen is the right choice, and it pairs with the screen-blend recombination at step 16.
>
> **Divergence worth knowing.** RC Astro's general advice is to use StarXTerminator as early as possible, ideally right after integration while still linear, since anything that alters star profiles degrades the extraction. This workflow removes stars late, after BXT, stretching and denoise. BXT tightens star profiles rather than smearing them, so the extraction still works well, but if you ever see poor star separation, moving SXT earlier (linear, no Unscreen, subtractive) is the fix.

9. Optional GHS tweak on the stars image

Aimed at long subs where the brightest stars are blown out and pull the eye off the nebula. On the **stars** image:

1. Draw a small preview over a bright star cluster.
2. Open GeneralizedHyperbolicStretch, sample a black background area, **Send to SP** (symmetry point).
3. Click **Invert**.
4. Raise the stretch factor slightly. Inverted, this *reduces* the stretch and takes the edge off the brightest cores.

Purely an artistic call. With 3-minute subs you may not need it at all; with 5-minute subs it is usually worth doing.

Stage 4: Colour (steps 10 to 14)

This is where the image is made, so it is worth being clear about what is actually happening. The colour here is not being corrected, it is being constructed. Dual-band data gives you two emission lines: Ha, which lands in the red channel, and OIII, which lands across green and blue. Every move in this stage exists to do one of two things: separate those two signals cleanly by hue, or push each one to a chosen colour once separated.

One structural point that makes all of it safe: the stars came out at step 8. Everything below happens to the starless image only. You can be far more aggressive with saturation and channel curves than you would ever dare on a full frame, because star colour is sitting in a separate file and returns unchanged at recombination.

10. NarrowbandNormalization

Process → ColorCalibration → NarrowbandNormalization, with the stretched starless image selected.

Open the real-time preview, then reset the tool. The preview often will not populate until you do. Easy to assume the tool is broken at this point when it is only waiting.

Leave the palette on HO. The temptation is to select SHO because that is the look you are chasing. Resist it. The dropdown describes the data you have, not the destination. You have two real emission lines, not three, and telling the tool otherwise asks it to normalise a channel that does not exist.

Apply a small OIII boost if the oxygen looks flat. A gentle drag lifts the OIII regions relative to Ha.

The judgement to make here is not "does this look finished". It is **"are the two emission regions now clearly two different colours?"** You are looking for a clean two-tone split. If OIII is still sitting inside the same hue family as Ha, the masks at step 12 will grab both regions at once and the entire rest of the workflow falls apart. This step is the gate everything after it passes through.

Apply, reset the tool, close the preview, close the process.

> **Check: confirmed, and the prerequisites matter.** The documented usage is to start with a stretched, starless image, which is exactly where this sits. The process does technically accept linear images, but the Lightness and Adjustments sections do not apply to them, so you lose most of the tool. Running it starless is also the documented recommendation, so step 8 has to come first.

11. Global saturation curve

A small saturation lift in CurvesTransformation, on the S channel. Deliberately subtle.

This is not a look decision, it is mask preparation, and the reason is worth knowing. Hue-based masks select on hue and chroma together. Weakly saturated pixels sit near the centre of the colour wheel, where the hue value is numerically unstable and a small amount of noise flips a pixel from one hue to another. Pushing saturation moves those pixels out towards the rim, where the hue reading is decisive. The result is a mask with firmer edges and less speckle.

Keep it light. Anything heavy-handed here will still be visible after two more rounds of saturation later.

12. Building the masks

Drag-and-drop mask icons are the quick route. The stock PixInsight tools do the same job.

> **Check: know which tool you are on.** ColorMask (Script → Utilities → ColorMask, by Rick Stevenson) ships with PixInsight and includes its own “Mask blur: layers to remove” control, so the separate mask-blur pass below is doing the same thing by another route. Note that the ColorMask script repository is only maintained for PixInsight 1.9.3 and earlier. **On 1.9.4 and later, use the ColourMask process instead** (Process → MaskGeneration, from Cranfield and Emchenko), which adds a real-time preview so you can tune the hue range while watching the mask update.
>
> Also worth knowing: the hue angles behind the primary colour buttons are documented as not isolating the named colours as cleanly as you would expect. A “green” mask, for example, spills into yellow and magenta. Independent testing has found the PixelMath vector masks produce effectively the same result as the script, so the choice of route makes little difference. Treat any named colour button as a starting point and always inspect the mask before trusting it. This is precisely why the fallback below exists.

One pass of MaskBlur on every mask. Raw hue masks are computed per pixel and inherit the image’s noise, which gives them a grainy, hard-edged appearance. Applied unblurred, the boundary of the mask becomes a visible boundary in the colour work, because the curve simply stops at the mask edge. One pass feathers that transition so the adjustment blends into the surrounding data.

The fallback when a mask comes out weak. If a mask (typically the blue one) is too thin to be useful:

1. Clone the working image.
2. On the clone only, push saturation hard, and put an S-curve through the blue channel to spread the blue values apart.
3. Generate the mask from the clone.
4. Apply that mask to the real image.

The clone is a mask factory. It is never edited, never saved, never seen. This works because the mask only needs to know **where** a hue is, not what the image looks like, so you are free to distort the clone into whatever makes the selection easiest.

13. Blue mask: OIII to teal

Apply the blue mask, drag its tab onto the working image, open CurvesTransformation with a preview.

1. **Raise the blue midtones.** On its own this drifts purple.
2. **Add a small amount of green.** This is why it drifts: in dual-band OSC data the OIII signal is spread across both green and blue, so boosting blue alone rotates the hue towards purple. Green rotates it back. Too much and you overshoot into a cold clinical cyan, so ease off until it settles on teal.
3. **Small saturation lift** to finish.

The oxygen can go further for a punchier image. Stopping deliberately short of that is a taste call, not a technical one, and worth recognising as such rather than chasing a fixed target.

14. Yellow mask: Ha to copper and gold

Same routine: yellow mask, MaskBlur, drag onto the image, open curves. Three moves, and the order matters because each one fixes what the previous one broke.

1. **Raise the red midtones.** Flat yellow moves towards copper. The target is copper, not orange, so do not run away with it.
2. **Raise green.** This looks like a step backwards, since it pushes the colour back towards the yellow you just left. The reason is the bright rims. After move 1 the edges of the pillars come out a light salmon-copper, which reads as washed out. Green pulls those rims to gold, and gold on the rims is a large part of what makes an image read as Hubble palette.
3. **Drag the green shadow point down.** This is the move that does the real work. It strips green from the darker end of the masked range only, leaving the bright end alone. In one gesture you get gold on the rims, copper through the midtones and deep red in the transition zones.

Finish with saturation, and a very slight red lift if it needs more punch.

Why step 3 matters so much: it is what makes two channels of data look like three. A genuine SHO image gets its colour variety from an actual sulphur channel. Here the variety comes from **tonal position inside a single Ha mask**, which costs you nothing in acquisition and is the closest honest equivalent.

Reset the tool, close the preview, and **remove the mask before you move on.** Leaving a mask attached is the standard way to ruin the next step and spend twenty minutes wondering why nothing is working.

Stage 5: Finish (steps 15 to 17)

15. Contrast, one of two ways

The reasoning behind doing anything here: after the masked colour work the image usually has unused range at the shadow end. That is headroom, and using it is free contrast.

Option A, S-curve. CurvesTransformation with RGB/K selected, a gentle S. More dramatic, and more obviously processed.

Option B, HistogramTransformation. The more natural-looking of the two, and my default.

- Look at the gap between the left edge of the histogram and where the data actually starts. That gap is shadow range you are not using.
- Bring the black point up to just short of the data. **Never into it.** Judge this on the histogram, not on the preview, because the preview will keep looking better for a while after you have started clipping.
- Then nudge the midtone slider slightly left to lift everything else back up a touch.

The reason B reads as more natural is that it repositions tones without compressing midtone contrast the way an S-curve does. The S-curve buys punch by flattening the middle of the range.

16. Recombine the stars

Starless icon onto the starless image, stars icon onto the stars image, rescreen icon onto either. The Cosmic Photons **ImageBlend** script or a `~((~starless)*(~stars))` PixelMath expression do the same job.

Why screen rather than add: screen blending cannot exceed 1.0, so bright star cores do not clip flat and gain a hard rim, and it correctly inverts the unscreen that was applied when the stars were extracted at step 8. Simple addition would return the stars at roughly double brightness and blow the cores.

> **Check: confirmed.** RC Astro's recombination guidance is to stretch both images first and use screen blending in PixelMath, which is precisely what the unscreen option at step 8 was setting up.

17. Optional final NoiseXTerminator pass

Worth doing because of what came before it. Two rounds of saturation and several channel curves amplify chroma noise that was genuinely invisible in the flat, unsaturated version at step 10. A light final pass cleans that up without touching structure. Then save.

What to carry forward

- **The stretch is the hinge.** Crop, gradient, deconvolution before it. Denoise, star removal, colour, contrast after it.
- **Deconvolution before noise reduction is a rule, not a style.** Everything else in this chain has legitimate alternative orderings.
- **HO in NarrowbandNormalization, SHO by hand.** The palette comes from mask work, not from a dropdown.
- **Steps 10 and 11 exist to serve steps 13 and 14.** Normalisation and the saturation lift are not look decisions, they are what make the masks accurate. Judge them on hue separation, not on beauty.
- **The green shadow drag is the single most transferable idea here.** It manufactures three colour zones from one mask, which is what stands in for the sulphur channel you never captured.
- **Aggression is safe here only because the stars are out.** Put the stars back before the colour work and none of this would be usable.
- **The gap still to close:** SPCC in narrowband mode, sandwiched between two BXT runs. As written, the colour is judged entirely by eye. Adding calibration costs one extra pass and gives a defensible starting point rather than a taste-based one.

Sources checked

- RC Astro, BlurXTerminator Technical Manual and AI4 release notes: linear-only requirement, no denoise before decon, Correct

Only + SPCC pattern, white-border failure mode

- RC Astro, NoiseXTerminator 2 / AI3 User Manual: linear or nonlinear both fine, never before deconvolution
- RC Astro, StarXTerminator Usage Notes: Unscreen is for nonlinear data only, screen blending for recombination, extract early where possible
- GraXpert script documentation: gradient removal first after crop, unlinked STF on uncalibrated data
- NarrowbandNormalization (Blanshan / Cranfield) usage documentation: stretched starless input, limited function on linear data
- ColorMask script and ColourMask process documentation (Stevenson; Cranfield and Emchenko): built-in mask blur control, 1.9.4 version split, known hue-angle caveat on the primary colour buttons