An Introduction to PixInsight

By Carlos Milovic, Core PTeam Member

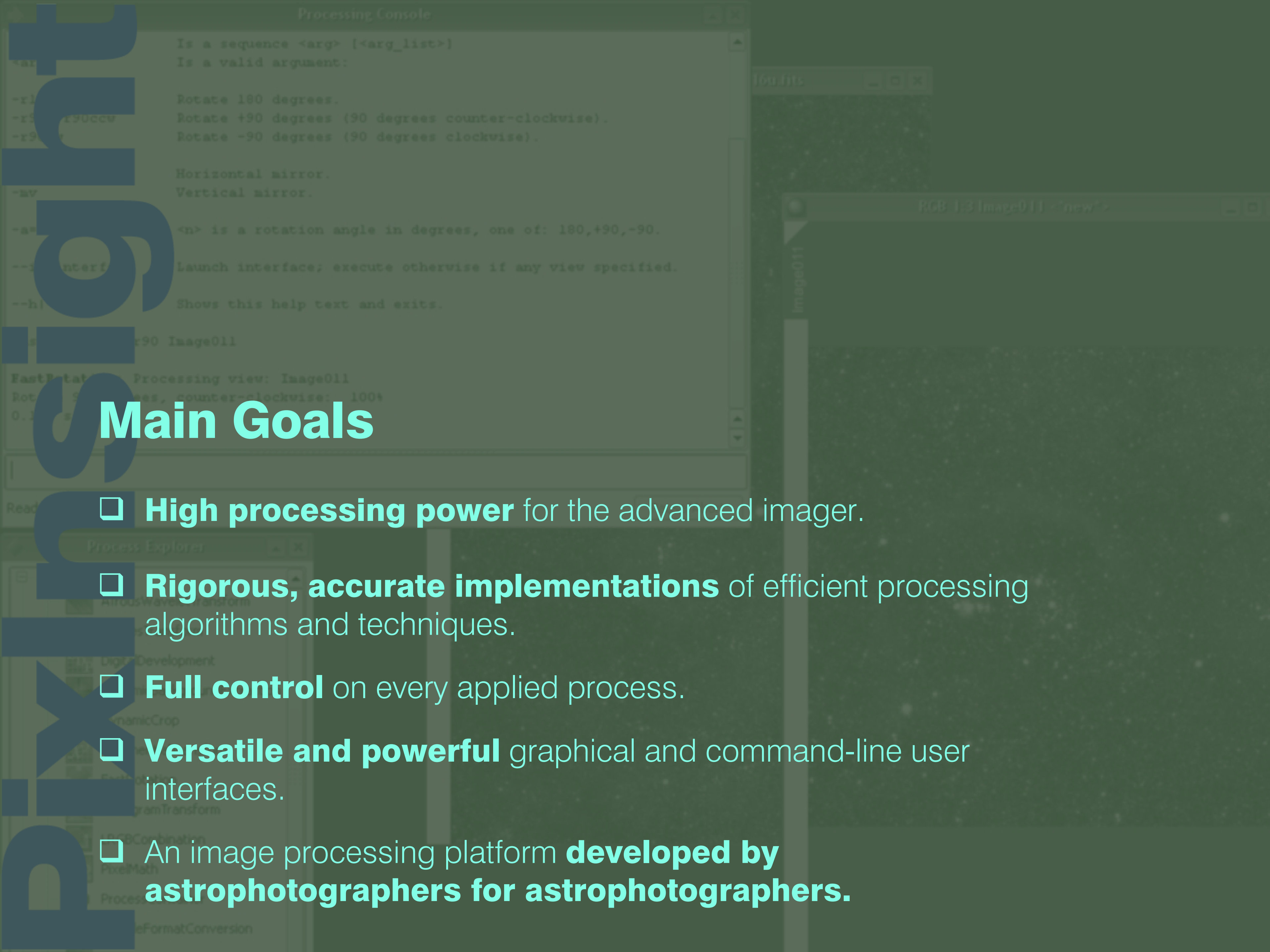
East Coast Conference on Astronomical Imaging / August 2006

Cover image: Deep Southern Orion, by Carlos Milovic (PTeam).
Three 25-minute exposures on Provia 400F film +2, with a 135 mm Canon FD lens @ f/4.
Fully processed in PixInsight.

Back cover image: NGC 7000, by José Luis Lamadrid and Vicent Peris (PTeam).
Five hours exposure with a modified Canon 20D camera on a Takahashi FS102.
Fully processed in PixInsight.

The Timeline of PixInsight

- ❑ Created by Juan Conejero, a Spanish professional software developer and astrophotographer.
- ❑ A predecessor: The **SGBNR** application for noise reduction, first published in 2001.
- ❑ The PixInsight project started in 2003.
- ❑ **PixInsight LE** (freeware limited edition) was released in 2004/2005.
- ❑ **PixInsight Standard** (commercial full edition, open/modular architecture) is currently in its final development stages.



```
Processing Console

          Is a sequence <arg> [<arg_list>]
          Is a valid argument:

-r1       Rotate 180 degrees.
-r9  r90ccw  Rotate +90 degrees (90 degrees counter-clockwise).
-r9       Rotate -90 degrees (90 degrees clockwise).

          Horizontal mirror.
-mv       Vertical mirror.

-a=       <n> is a rotation angle in degrees, one of: 180,+90,-90.

--i  nterf   Launch interface; execute otherwise if any view specified.

--h       Shows this help text and exits.

        r90 Image011

FastRotat     Processing view: Image011
Rot        ees, counter-clockwise: 100%
0.1
```

Main Goals

- ❑ **High processing power** for the advanced imager.
- ❑ **Rigorous, accurate implementations** of efficient processing algorithms and techniques.
- ❑ **Full control** on every applied process.
- ❑ **Versatile and powerful** graphical and command-line user interfaces.
- ❑ An image processing platform **developed by astrophotographers for astrophotographers.**

Advanced Image Processing

State-of-the-art implementations of avant garde processing techniques.
A few examples:

❑ **Multiscale processing.** *À trous* wavelet transform, morphological wavelet transform. Planned implementations of curvelet and ridgelet transforms.

❑ **Background modelization.** Manual and automatic background extraction tools for high precision vignetting and gradient correction.

❑ **PixelMath** interface with proprietary expression parser/interpreter, including more than 35 built-in functions and a comprehensive set of arithmetic, logical, bitwise and relational operators.

❑ **Noise reduction.** High-performance multiscale and contrast-driven adaptive algorithms.

❑ **Regularized deconvolution.** Richardson-Lucy and Van Cittert regularized deconvolution algorithms.

What Is PixInsight?

Quick answer: An **image processing software** application.
However, it is **not just another image processing application:**

- □ **Modular, open architecture:** The entire processing functionality and file format support are provided by external, installable modules.

- □ **Free and readily available software development framework** to build PixInsight modules: **PixInsight Class Library (PCL).**

- □ **Highly portable** to Microsoft Windows, Linux/UNIX, and Mac OS X.

- □ **Five data types** supported transparently (available to all processes): 8, 16, and 32-bit integers, plus 32 and 64-bit floating point.

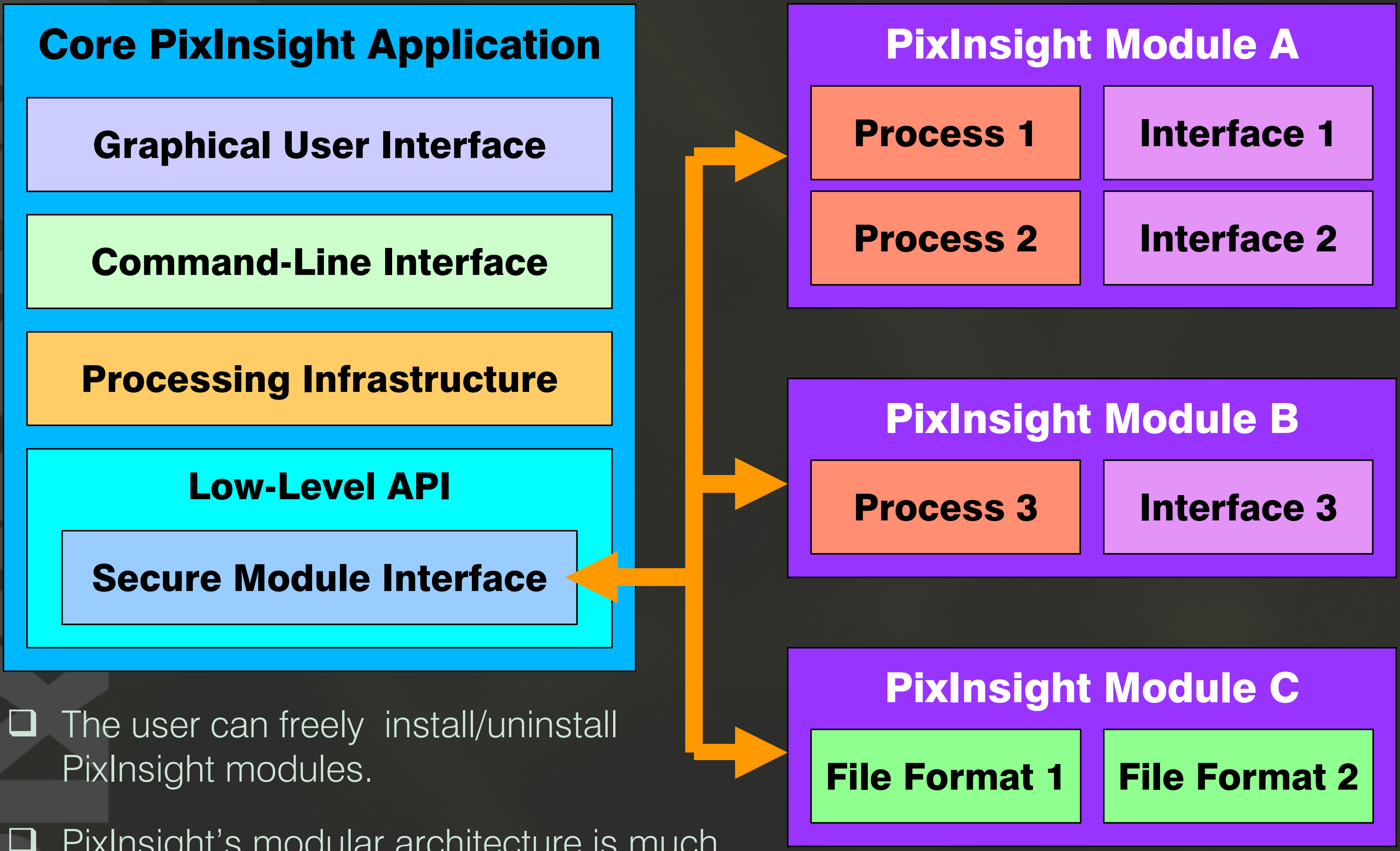
- □ Advanced **masking system.**

- □ **Object-oriented graphical interface** with **multiple previews, real-time preview,** and **color management** through ICC profiles.

- □ **Command-line interface** with scripting support.

- □ **Parallel processing:** Advanced support for multi-processor, multi-core and HyperThreading technologies.

Open, Modular Architecture



- ❑ The user can freely install/uninstall PixInsight modules.
- ❑ PixInsight's modular architecture is much more versatile, powerful and flexible than a traditional plug-ins system.

Portability of the PixInsight/PCL Platform

- □ **Microsoft Windows** 2000/2003/XP/Vista
- □ **Linux** and main **UNIX** variants
- □ **Macintosh OS X**
- □ **64-bit versions of all supported operating systems**

The keys of PixInsight's high portability:

- □ **Core application** based on **PCL** and Trolltech's **Qt framework.**
- □ **PCL is fully hardware and O.S. independent** code.

Five Data Types

Use the most adequate data format for each processing task:

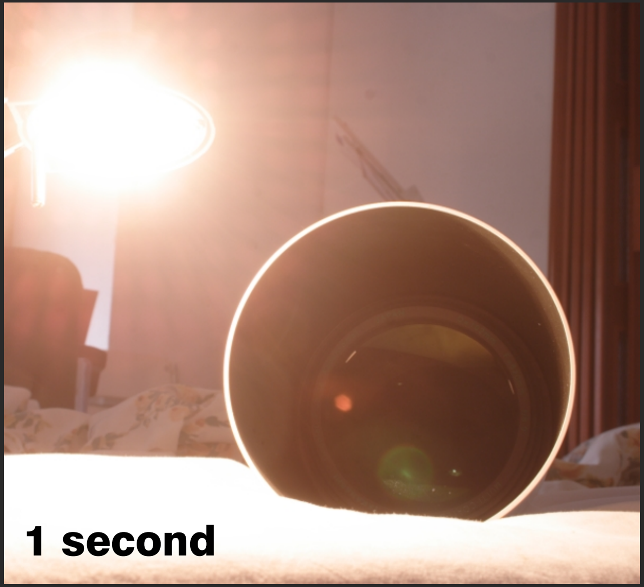
- □ **Unsigned Integers: 8-bit, 16-bit, 32-bit**
- □ **IEEE 754 Floating Point: 32-bit and 64-bit**
- □ **PCL support for complex-valued floating-point images**

The 64-bit floating point format provides a huge working space of 10^{15} discrete values, ideal to handle extremely large dynamic ranges.

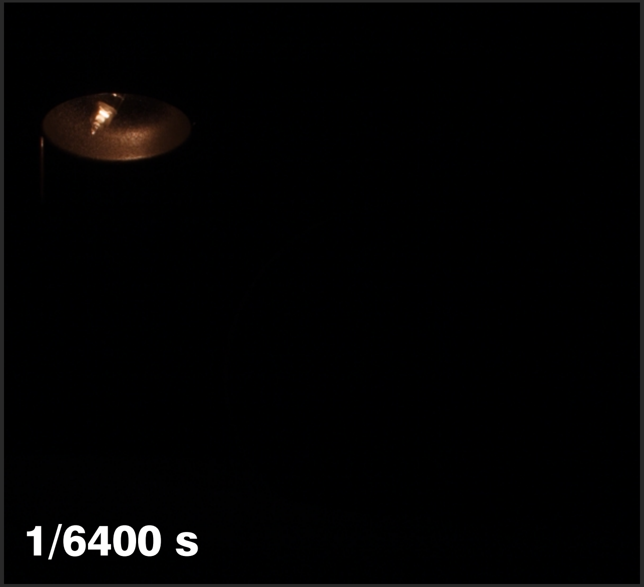
Transparent data type support:

- □ **All processes can work with all data types** without distinction. The only limits are imposed by roundoff errors inherent to each numerical format.
- □ From the developer's perspective, **a single source code works for all data types,** thanks to PCL's advanced template support.

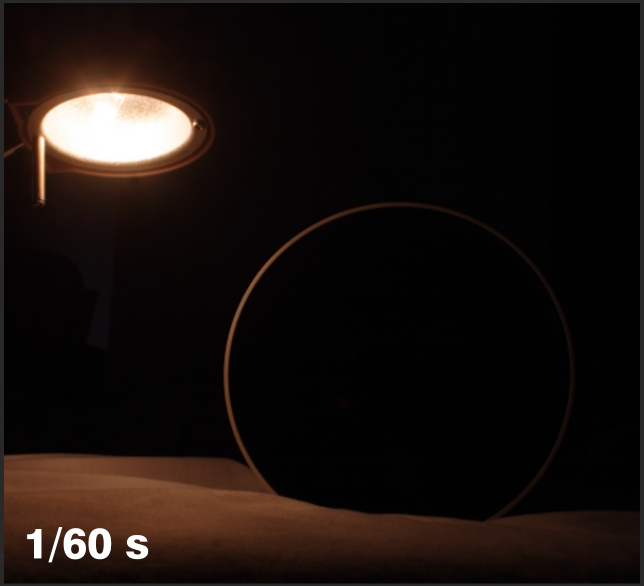
A 64-bit High Dynamic Range Experiment



1 second



1/6400 s



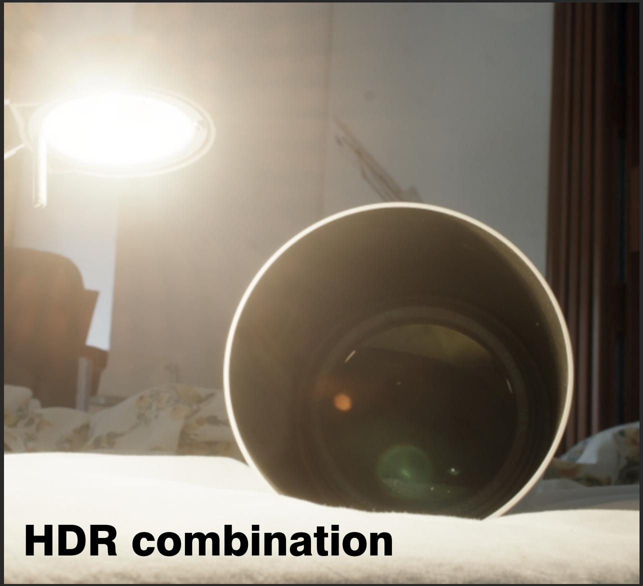
1/60 s

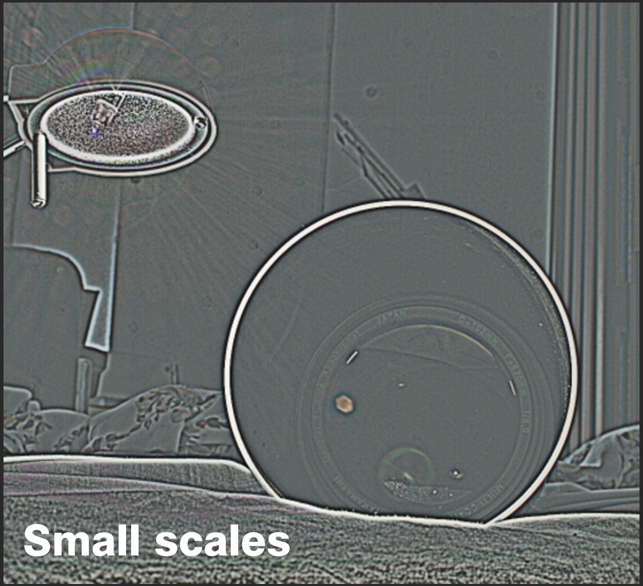
The scene shows a 20-watt halogen lamp and a Takahashi FS102 objective.

A sequence of exposures of 1, ¼, 1/15, 1/60, 1/200, 1/400, 1/1600 and 1/6400 seconds were taken with a modified Canon 20D camera @ 200 ISO through a 28 mm lens working at f/13.

Experiment conducted by Vicent Peris, core PTeam member.

A 64-bit High Dynamic Range Experiment

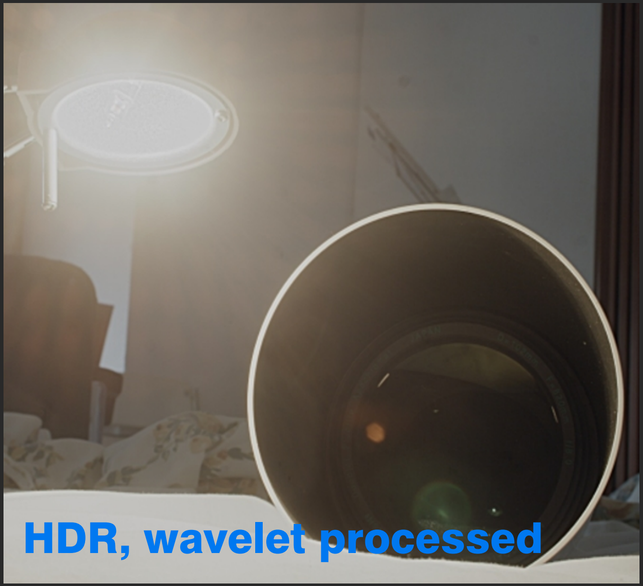

HDR combination


Small scales

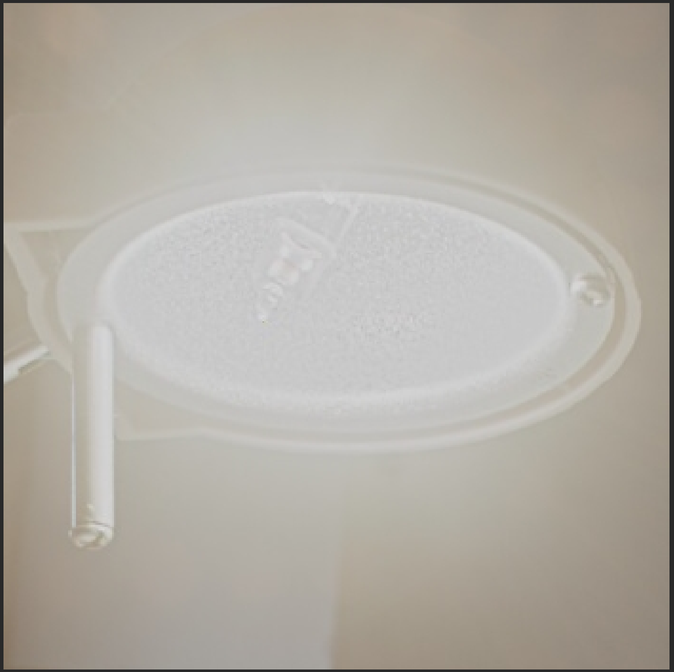
Upper left: The whole set of exposures from 1 to 1/6400 seconds were integrated linearly as a 64-bit floating-point image, and the result was stretched with a non-linear histogram transform (0.00001 midtones balance).

Upper right: A wavelet transform was applied to extract small-scale structures.

Lower right: small scales were enhanced and reinserted to form a combined image representing all image structures throughout the whole original dynamic range.


HDR, wavelet processed

A 64-bit High Dynamic Range Experiment



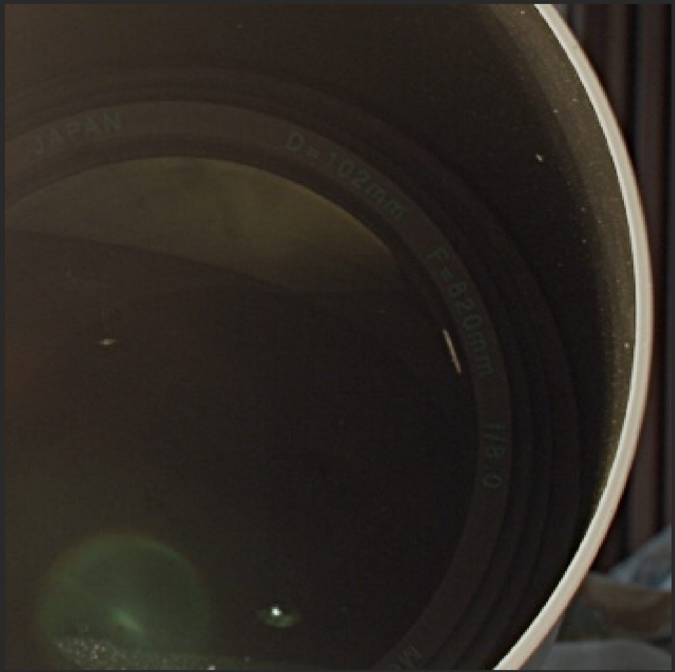
Two detail crops of the processed high-dynamic range, 64-bit floating point combined image.

Note the visibility of details on the lamp and the readable text over the closing ring of the FS102 objective.

To combine the whole set of exposures without data loss, at least 25,000,000 discrete sample values are required.

The 32-bit floating point format provides no more than 10,000,000 sample values.

This example requires 64-bit floating point or 32-bit integers. Both formats are transparently supported by PixInsight.



Masking System

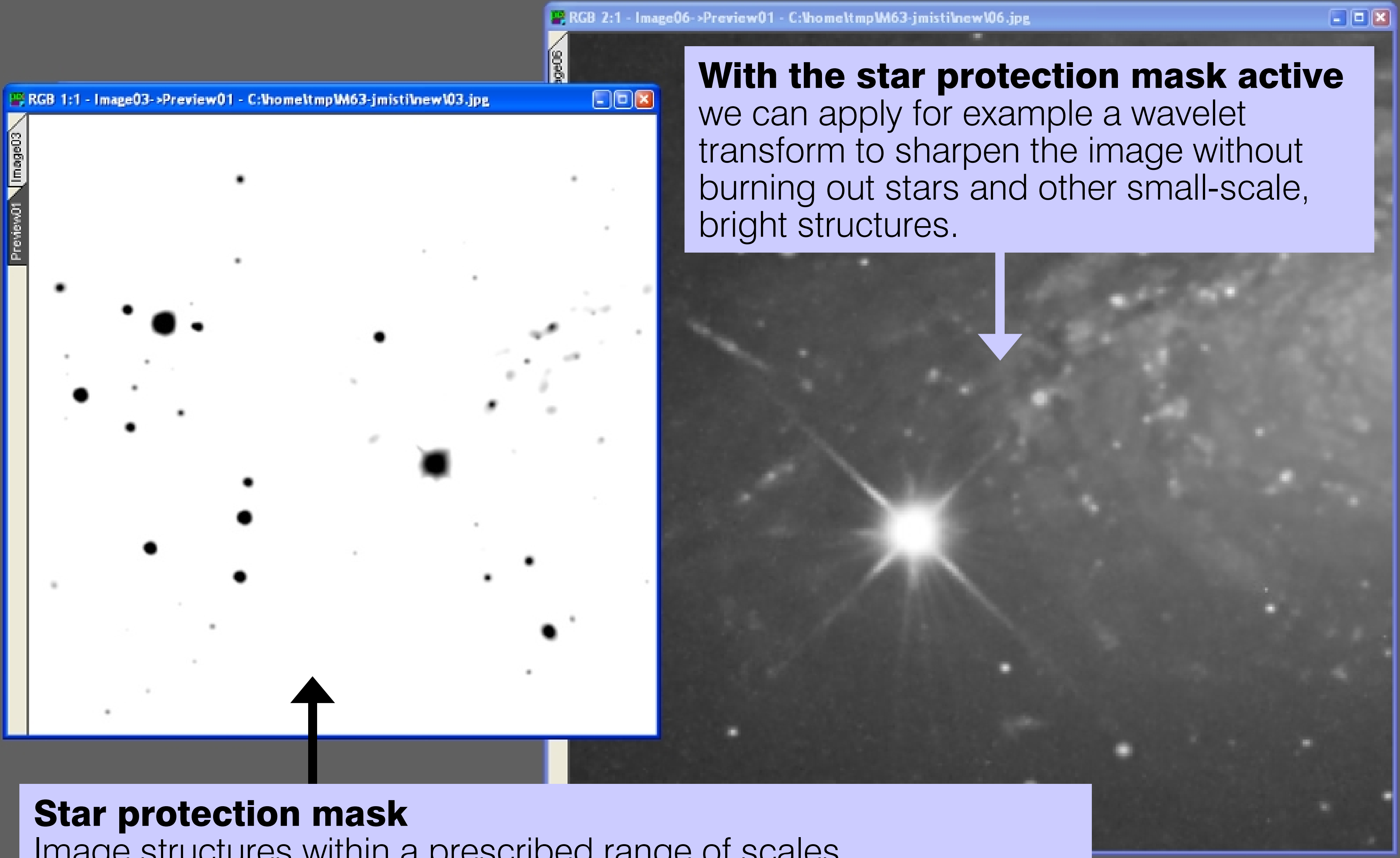
Masks play a key role in the advanced image processing workflow. A mask can be used to:

□ **Apply processes selectively** to image structures of interest.

□ **Protect selected image structures** from the adverse effects of some processes.

In PixInsight, any image can work as a mask for an unlimited number of images: the simplest, most efficient and versatile masking system.

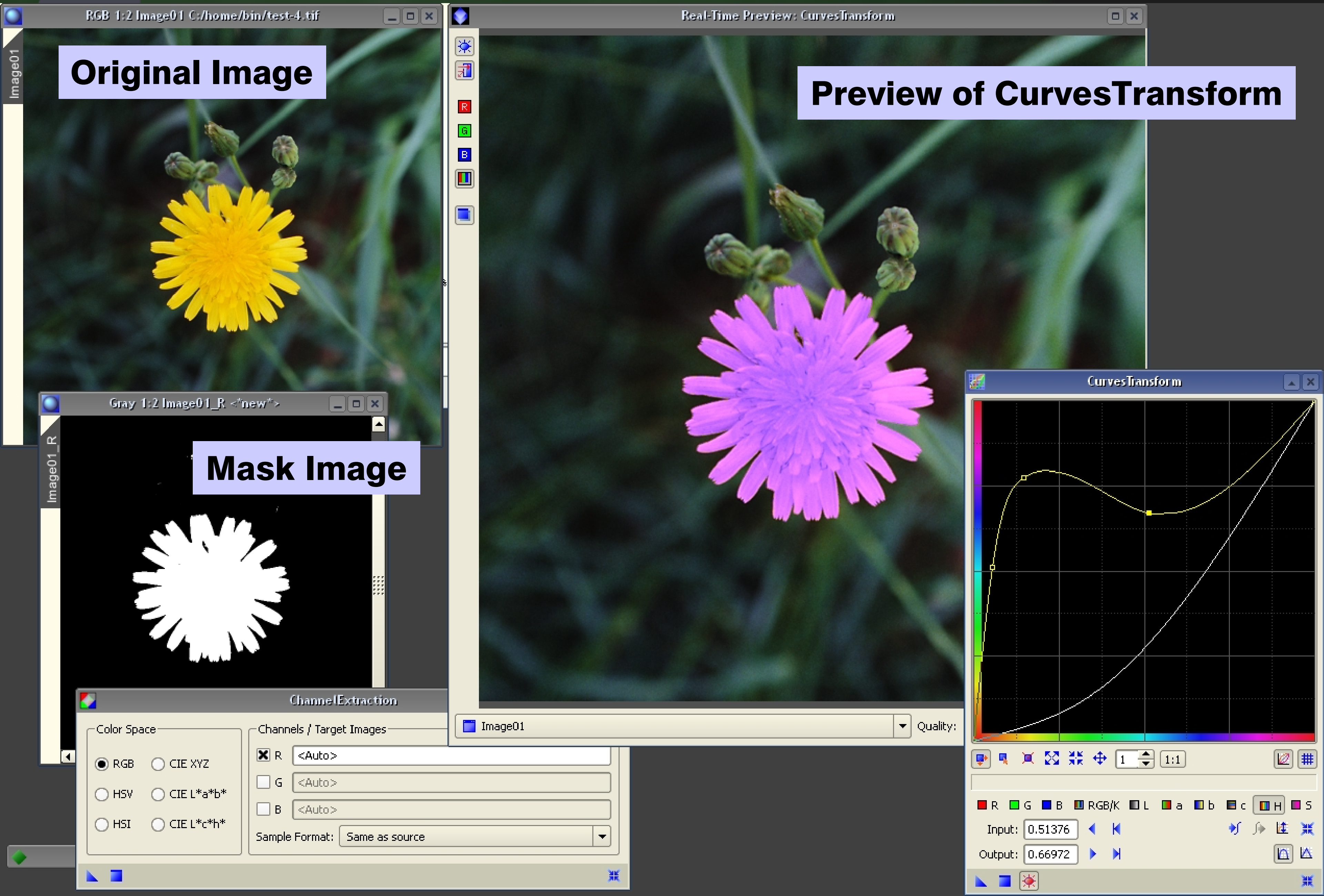
Masking Example: Star Protection Masks


RGB 2:1 - Image06->Preview01 - C:\home\tmp\M63-jmisti\new\06.jpg
age06
RGB 1:1 - Image03->Preview01 - C:\home\tmp\M63-jmisti\new\03.jpg
Image03
Preview01

With the star protection mask active we can apply for example a wavelet transform to sharpen the image without burning out stars and other small-scale, bright structures.

Star protection mask
Image structures within a prescribed range of scales have been isolated by means of a special wavelet-based technique.

Masking Example: Selective Hue/Contrast


Original Image

Preview of CurvesTransform

Mask Image

RGB 1:2 Image01 C:/home/bin/test-4.tif

Real-Time Preview: CurvesTransform

Gray 1:2 Image01_R <*new*>

ChannelExtraction

Color Space

● RGB ○ CIE XYZ

○ HSV ○ CIE L*a*b*

○ HSI ○ CIE L*c*h*

Channels / Target Images

☒ R <Auto>

☐ G <Auto>

☐ B <Auto>

Sample Format: Same as source

Image01 Quality:

CurvesTransform

R G B RGB/K L a b c H S

Input: 0.51376

Output: 0.66972

Object-Oriented Graphical User Interface

A novel, flexible user interface paradigm:

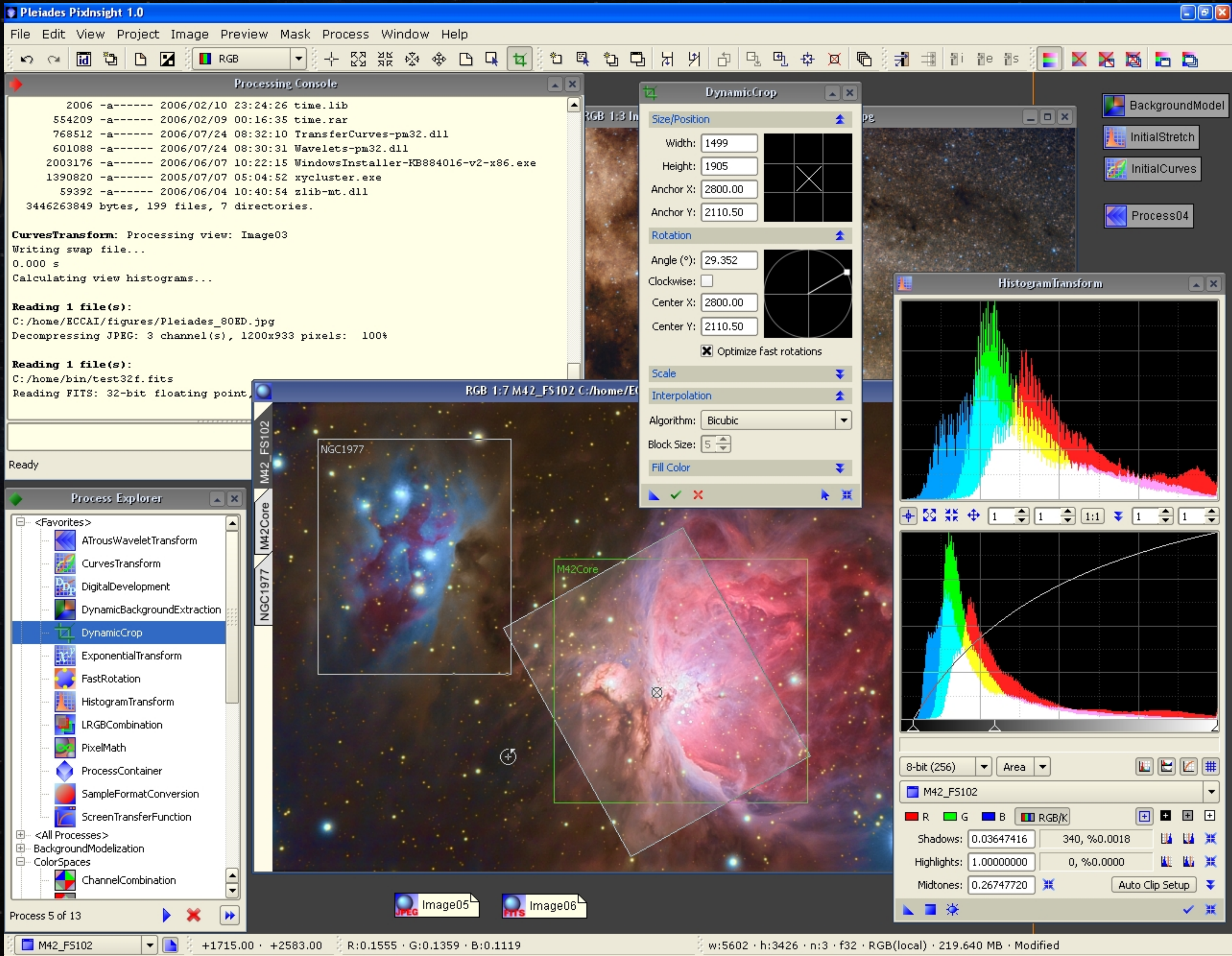
□ Graphical interface composed by **independent, self-sufficient and self-contained elements** (objects) with mutual interaction.

□ **Images and processes** are living objects that **can be handled and managed independently** through specific interface resources.

□ **Processes can be defined independently of images.** To define a process, there is no need to have an image opened.

□ PixInsight follows a **strict object-oriented design not just internally, but also *externally*,** in its user interface.

PixInsight's Graphical User Interface



Pleiades PixInsight 1.0

Processing Console

```
       2006 -a------ 2006/02/10 23:24:26 time.lib
     554209 -a------ 2006/02/09 00:16:35 time.rar
     768512 -a------ 2006/07/24 08:32:10 TransferCurves-pm32.dll
     601088 -a------ 2006/07/24 08:30:31 Wavelets-pm32.dll
    2003176 -a------ 2006/06/07 10:22:15 WindowsInstaller-KB884016-v2-x86.exe
    1390820 -a------ 2005/07/07 05:04:52 xycluster.exe
      59392 -a------ 2006/06/04 10:40:54 zlib-mt.dll
 3446263849 bytes, 199 files, 7 directories.

CurvesTransform: Processing view: Image03
Writing swap file...
0.000 s
Calculating view histograms...

Reading 1 file(s):
C:/home/ECCAI/figures/Pleiades_80ED.jpg
Decompressing JPEG: 3 channel(s), 1200x933 pixels:  100%

Reading 1 file(s):
C:/home/bin/test32f.fits
Reading FITS: 32-bit floating point,
```

Ready

DynamicCrop

Size/Position
Width: 1499
Height: 1905
Anchor X: 2800.00
Anchor Y: 2110.50

Rotation
Angle (°): 29.352
Clockwise: ☐
Center X: 2800.00
Center Y: 2110.50
☒ Optimize fast rotations

Scale

Interpolation
Algorithm: Bicubic
Block Size: 5

Fill Color

HistogramTransform

8-bit (256) · Area · M42_FS102

Shadows: 0.03647416 · 340, %0.0018
Highlights: 1.00000000 · 0, %0.0000
Midtones: 0.26747720 · Auto Clip Setup

Process 5 of 13

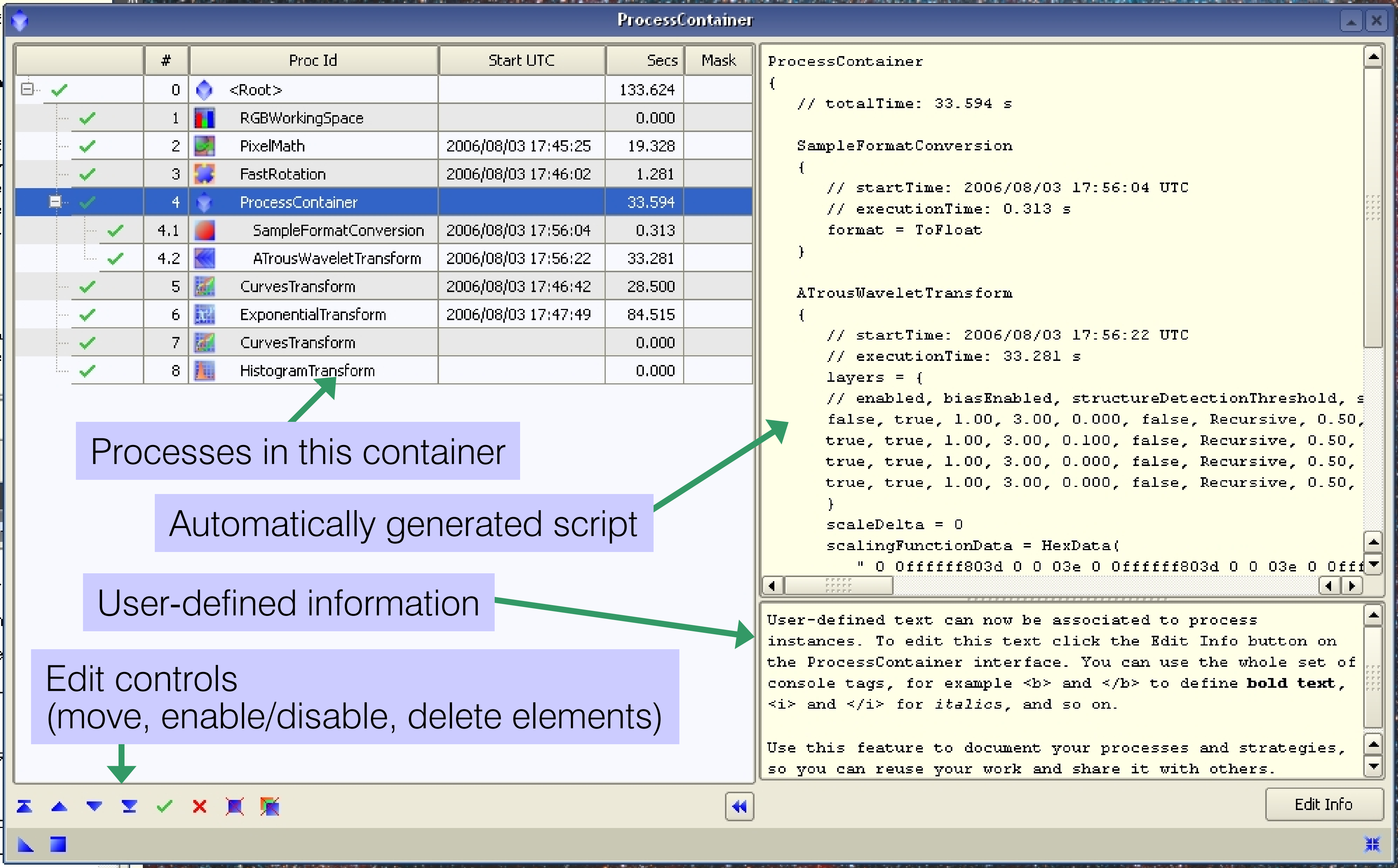
M42_FS102 · +1715.00 · +2583.00 · R:0.1555 · G:0.1359 · B:0.1119 · w:5602 · h:3426 · n:3 · f32 · RGB(local) · 219.640 MB · Modified

Process Containers

- □ **Ordered sequences of processes.**
- □ **A process container is also a process,** so a process container can contain other process containers, forming a **tree structure.**
- □ **Process containers can be freely edited:** individual processes can be added, removed, extracted and rearranged.
- □ Individual processes can be assigned arbitrary textual information. This permits to **document an entire processing workflow.**

Process containers are versatile objects to store, manage and organize processing strategies. They make it extremely easy to share our procedures with others, and to reuse them in multiple projects.

The ProcessContainer Interface



ProcessContainer

	#	Proc Id	Start UTC	Secs	Mask
✓	0	<Root>		133.624	
✓	1	RGBWorkingSpace		0.000	
✓	2	PixelMath	2006/08/03 17:45:25	19.328	
✓	3	FastRotation	2006/08/03 17:46:02	1.281	
✓	4	ProcessContainer		33.594	
✓	4.1	SampleFormatConversion	2006/08/03 17:56:04	0.313	
✓	4.2	ATrousWaveletTransform	2006/08/03 17:56:22	33.281	
✓	5	CurvesTransform	2006/08/03 17:46:42	28.500	
✓	6	ExponentialTransform	2006/08/03 17:47:49	84.515	
✓	7	CurvesTransform		0.000	
✓	8	HistogramTransform		0.000	

```
ProcessContainer
{
   // totalTime: 33.594 s

   SampleFormatConversion
   {
      // startTime: 2006/08/03 17:56:04 UTC
      // executionTime: 0.313 s
      format = ToFloat
   }

   ATrousWaveletTransform
   {
      // startTime: 2006/08/03 17:56:22 UTC
      // executionTime: 33.281 s
      layers = {
      // enabled, biasEnabled, structureDetectionThreshold, s
      false, true, 1.00, 3.00, 0.000, false, Recursive, 0.50,
      true, true, 1.00, 3.00, 0.100, false, Recursive, 0.50,
      true, true, 1.00, 3.00, 0.000, false, Recursive, 0.50,
      true, true, 1.00, 3.00, 0.000, false, Recursive, 0.50,
      }
      scaleDelta = 0
      scalingFunctionData = HexData(
         " 0 0ffffff803d 0 0 03e 0 0ffffff803d 0 0 03e 0 0fff
```

User-defined text can now be associated to process instances. To edit this text click the Edit Info button on the ProcessContainer interface. You can use the whole set of console tags, for example <b> and </b> to define **bold text**, <i> and </i> for *italics*, and so on.

Use this feature to document your processes and strategies, so you can reuse your work and share it with others.

Processes in this container

Automatically generated script

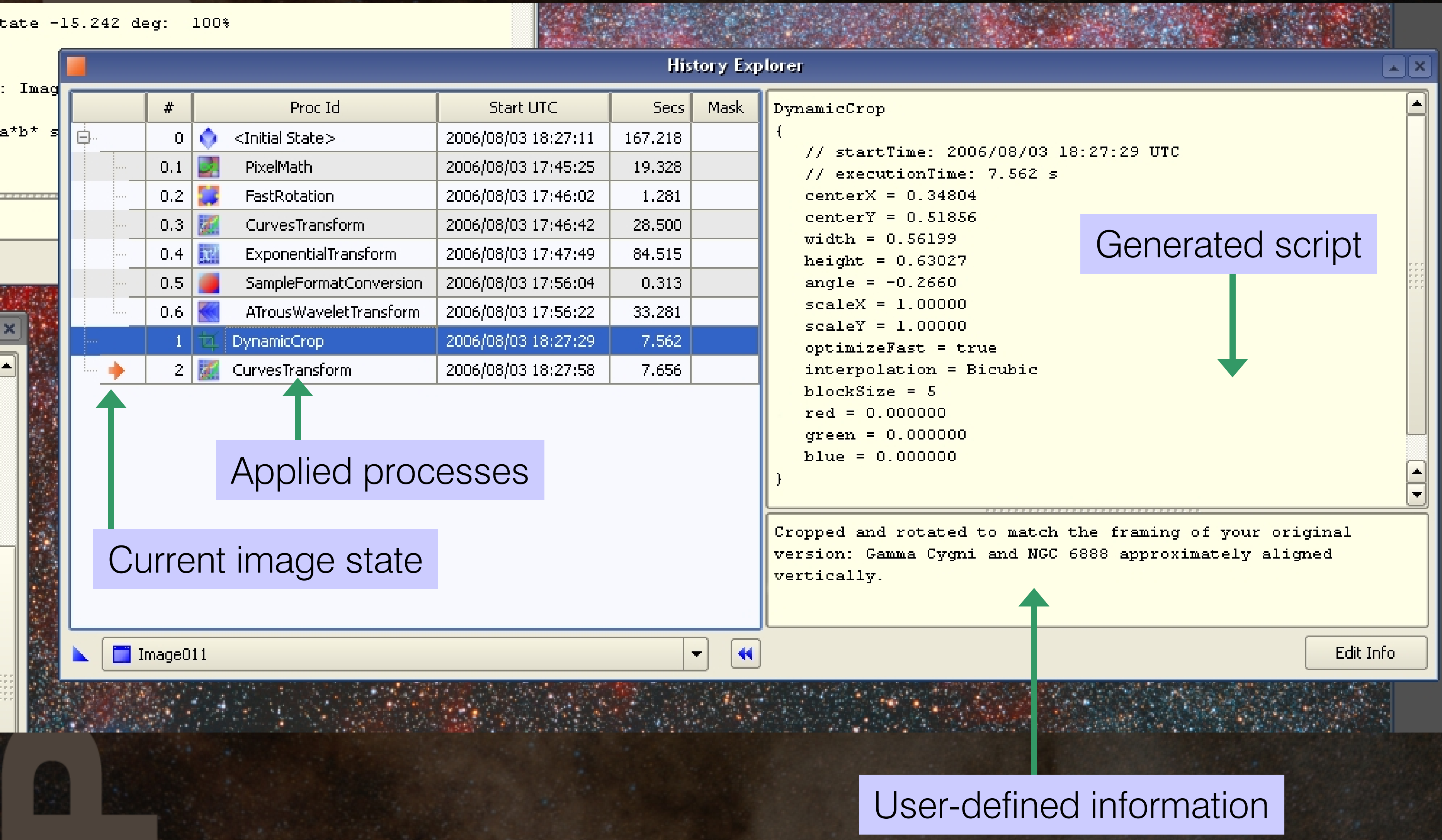
User-defined information

Edit controls
(move, enable/disable, delete elements)

Processing Histories

- ❑ **Every image has an associated processing history** in PixInsight.
- ❑ A processing history is a **special, read-only process container.**
- ❑ A processing history can be traversed arbitrarily. The user has **random access to all processing steps,** and the platform provides **unlimited undo/redo capacity;** the only limit is the available hard disk space.
- ❑ Processing histories can be converted into normal, editable process containers. Thanks to this feature, **any procedure applied to an image is fully reusable in PixInsight,** and the user has **full control over every step** of the processing workflow.

The History Explorer Window



	#	Proc Id	Start UTC	Secs	Mask
⊟	0	<Initial State>	2006/08/03 18:27:11	167.218	
	0.1	PixelMath	2006/08/03 17:45:25	19.328	
	0.2	FastRotation	2006/08/03 17:46:02	1.281	
	0.3	CurvesTransform	2006/08/03 17:46:42	28.500	
	0.4	ExponentialTransform	2006/08/03 17:47:49	84.515	
	0.5	SampleFormatConversion	2006/08/03 17:56:04	0.313	
	0.6	ATrousWaveletTransform	2006/08/03 17:56:22	33.281	
	1	DynamicCrop	2006/08/03 18:27:29	7.562	
➔	2	CurvesTransform	2006/08/03 18:27:58	7.656	

```
DynamicCrop
{
   // startTime: 2006/08/03 18:27:29 UTC
   // executionTime: 7.562 s
   centerX = 0.34804
   centerY = 0.51856
   width = 0.56199
   height = 0.63027
   angle = -0.2660
   scaleX = 1.00000
   scaleY = 1.00000
   optimizeFast = true
   interpolation = Bicubic
   blockSize = 5
   red = 0.000000
   green = 0.000000
   blue = 0.000000
}
```

Cropped and rotated to match the framing of your original version: Gamma Cygni and NGC 6888 approximately aligned vertically.

Generated script

Applied processes

Current image state

User-defined information

Image Containers

- □ PixInsight Standard introduces a new class of container objects: **Image containers are ordered sequences of references to images.**

- □ The elements of an image container can be references to either **disk files or opened images.**

- □ **Any process**, including of course process containers, **can be applied to an image container.** Applied processes execute sequentially on each member of an image container.

- □ As disk images are being processed in an image container, **resulting images are written to prescribed destination folders and file names,** which can be fully customized.

- □ Image containers provide a flexible, easy-to-use and intuitive way to implement **batch processing** in PixInsight.

Process Icons

- □ A process icon **encapsulates a process** under a graphical envelope that can be **freely managed** within PixInsight's core graphical environment.

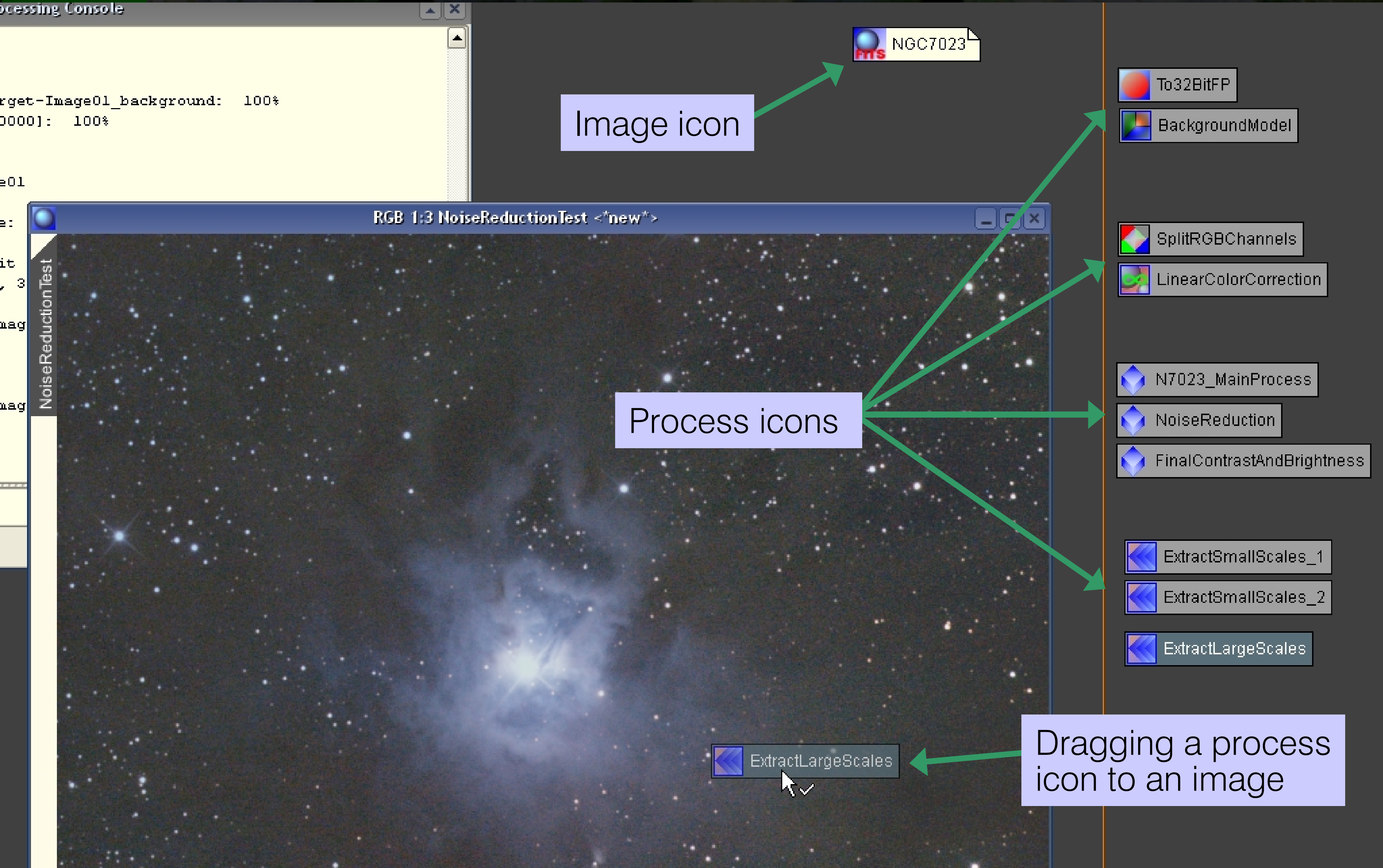
- □ Process icons are **living objects pertaining to the core application's workspace.**

- □ Process icons can be **dragged, organized, renamed, copied, deleted, applied to images and image containers, and saved to special disk files (PSM files).**

- □ PSM files are the main way **PixInsight users can share and reuse their processing work.**

The implementation of process icons in PixInsight has been widely acclaimed as a fresh, innovative contribution to graphical interfaces for imaging applications.

Process Icons and Image Icons



```
rget-Image01_background:   100%
0000]:   100%

e01

e:

it
, 3

mag

mag
```

Image icon

Process icons

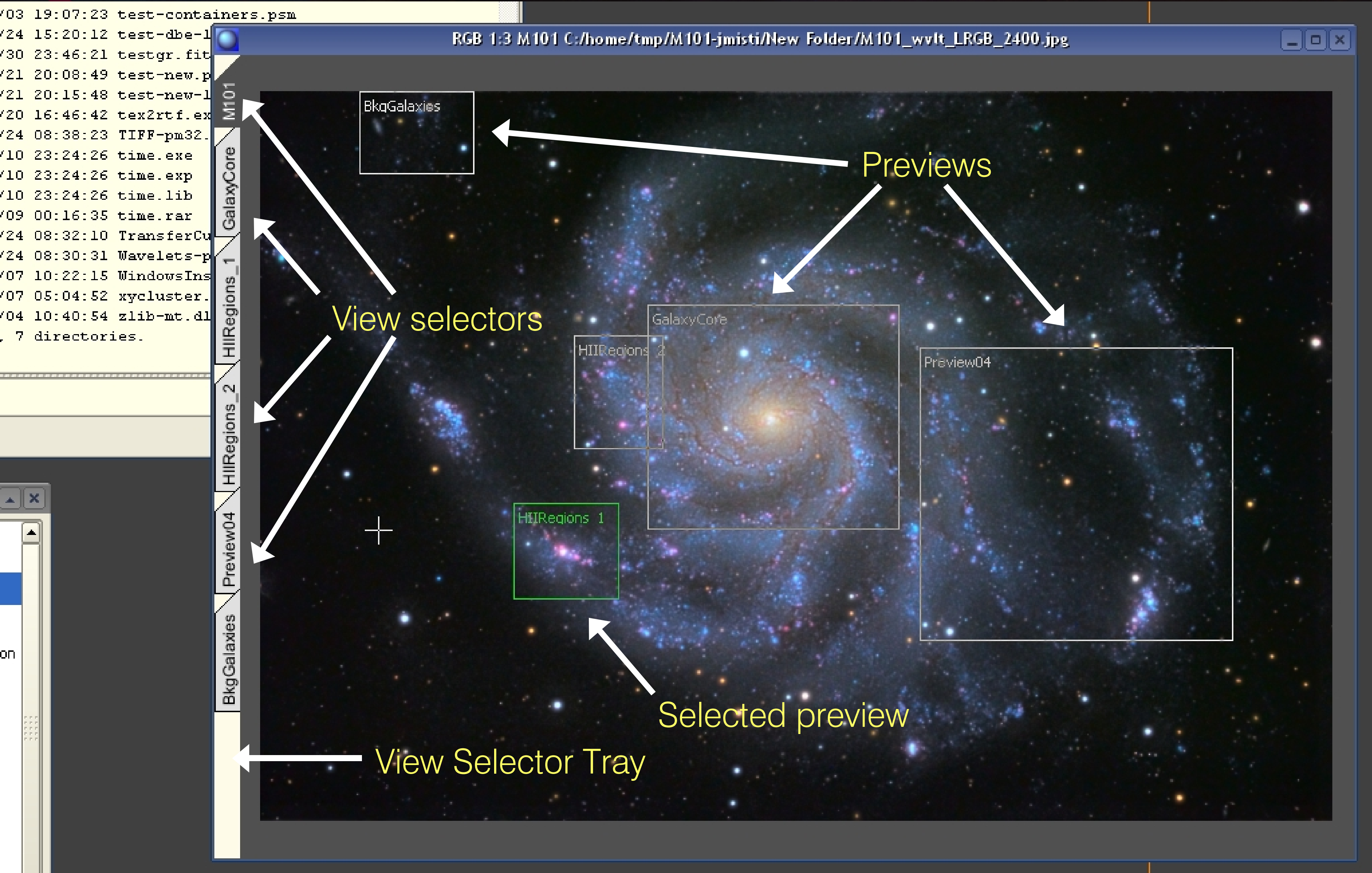
Dragging a process icon to an image

Previews

- ❑ Previews are **temporary subimages** that the user can freely define over any image in PixInsight. An **unlimited number of previews** can be defined for any image.
- ❑ Previews are mainly used to **try out** any number of **processes without modifying their parent images.**
- ❑ Previews have **their own processing histories,** exactly like independent images, which can be extracted as process containers.
- ❑ When a process is applied to a preview, **the whole work is performed in RAM without accessing swap disk files,** which is **extremely fast.**
- ❑ Previews can be easily **converted into independent images.**

By defining relatively small previews over regions of special interest, the intense trial/error work required for virtually any nontrivial image processing task can be carried out quickly and with a high degree of flexibility and accuracy.

Previews



```
/03 19:07:23 test-containers.psm
/24 15:20:12 test-dbe-l
/30 23:46:21 testgr.fit
/21 20:08:49 test-new.p
/21 20:15:48 test-new-l
/20 16:46:42 tex2rtf.ex
/24 08:38:23 TIFF-pm32.
/10 23:24:26 time.exe
/10 23:24:26 time.exp
/10 23:24:26 time.lib
/09 00:16:35 time.rar
/24 08:32:10 TransferCu
/24 08:30:31 Wavelets-p
/07 10:22:15 WindowsIns
/07 05:04:52 xycluster.
/04 10:40:54 zlib-mt.dl
, 7 directories.
```

RGB 1:3 M101 C:/home/tmp/M101-jmisti/New Folder/M101_wvlt_LRGB_2400.jpg

Previews

View selectors

Selected preview

View Selector Tray

PixInsight

Real-Time Preview

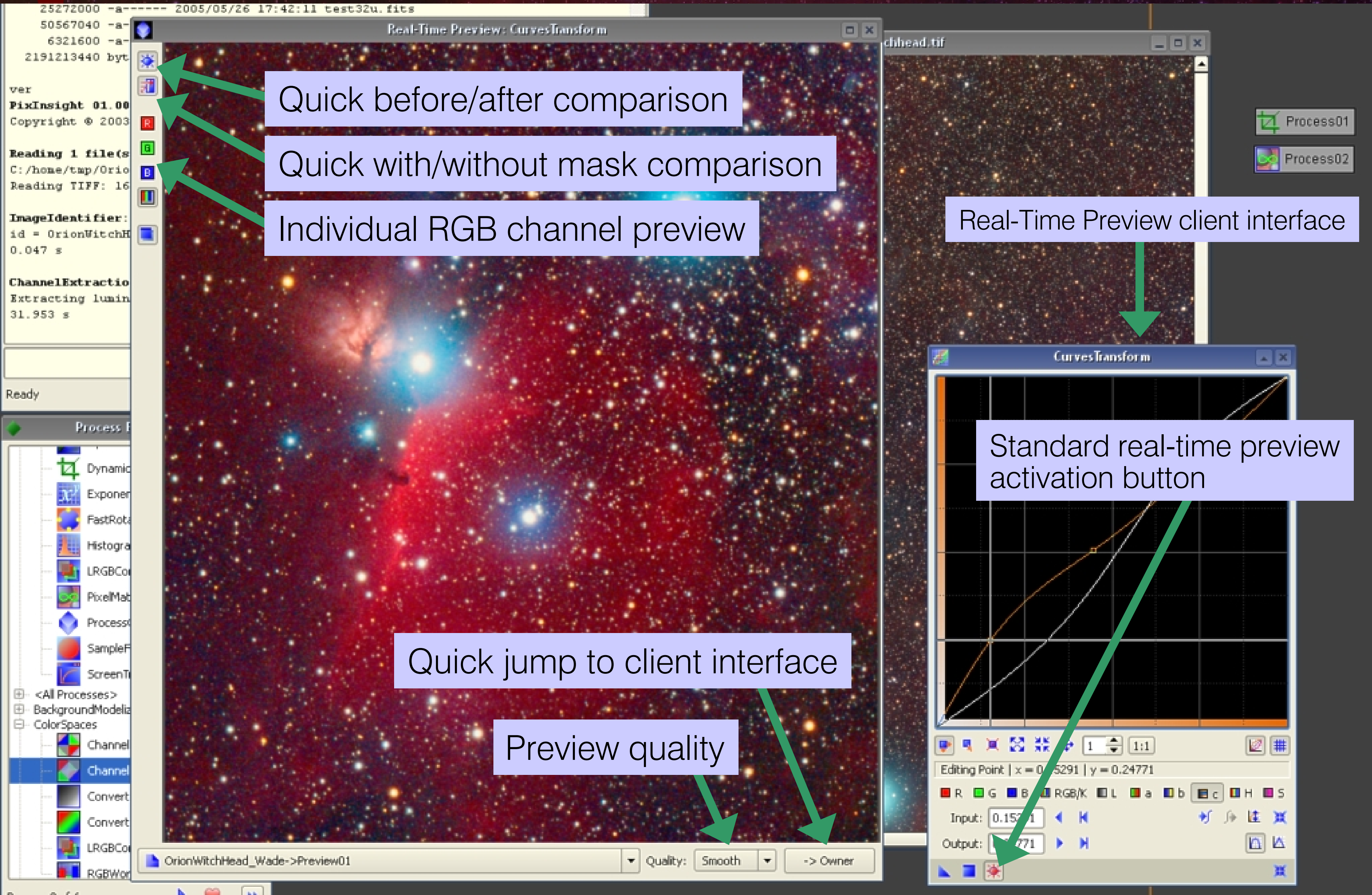
- ❑ PixInsight’s Real-Time Preview is a sophisticated GUI resource that provides **instant feedback** while the user **adjusts parameters in a processing interface.**

- ❑ The Real-Time Preview interface allows **quick before/after comparisons without having to recalculate a complex process.** Calculate once, preview as many times as you want, quickly and easily.

- ❑ When previewing masked processes, you can easily **toggle between previewing the effects of a process with and without the mask active.** Again, this is done **without requiring recalculation.**

- ❑ Developers can optimize real-time previewing functionality for their newly authored processes with an easy-to-use, efficient interface provided by the PCL.

The Real-Time Preview Window



Quick before/after comparison

Quick with/without mask comparison

Individual RGB channel preview

Real-Time Preview client interface

Standard real-time preview activation button

Quick jump to client interface

Preview quality

```
   25272000 -a------ 2005/05/26 17:42:11 test32u.fits
   50567040 -a-
    6321600 -a-
 2191213440 byt

ver
PixInsight 01.00
Copyright © 2003

Reading 1 file(s
C:/home/tmp/Orio
Reading TIFF: 16

ImageIdentifier:
id = OrionWitchH
0.047 s

ChannelExtractio
Extracting lumin
31.953 s
```

Ready

Editing Point | x = 0.?5291 | y = 0.24771

Input: 0.15??1

Output: ??771

OrionWitchHead_Wade->Preview01

Quality: Smooth

Command-Line Interface

□ **The best of two worlds:** A powerful command-line interface coexisting with a high-end GUI.

□ A large set of **emulated UNIX commands** (*cd, mkdir, ls, cpy, alias,* and so on) available on all supported platforms.

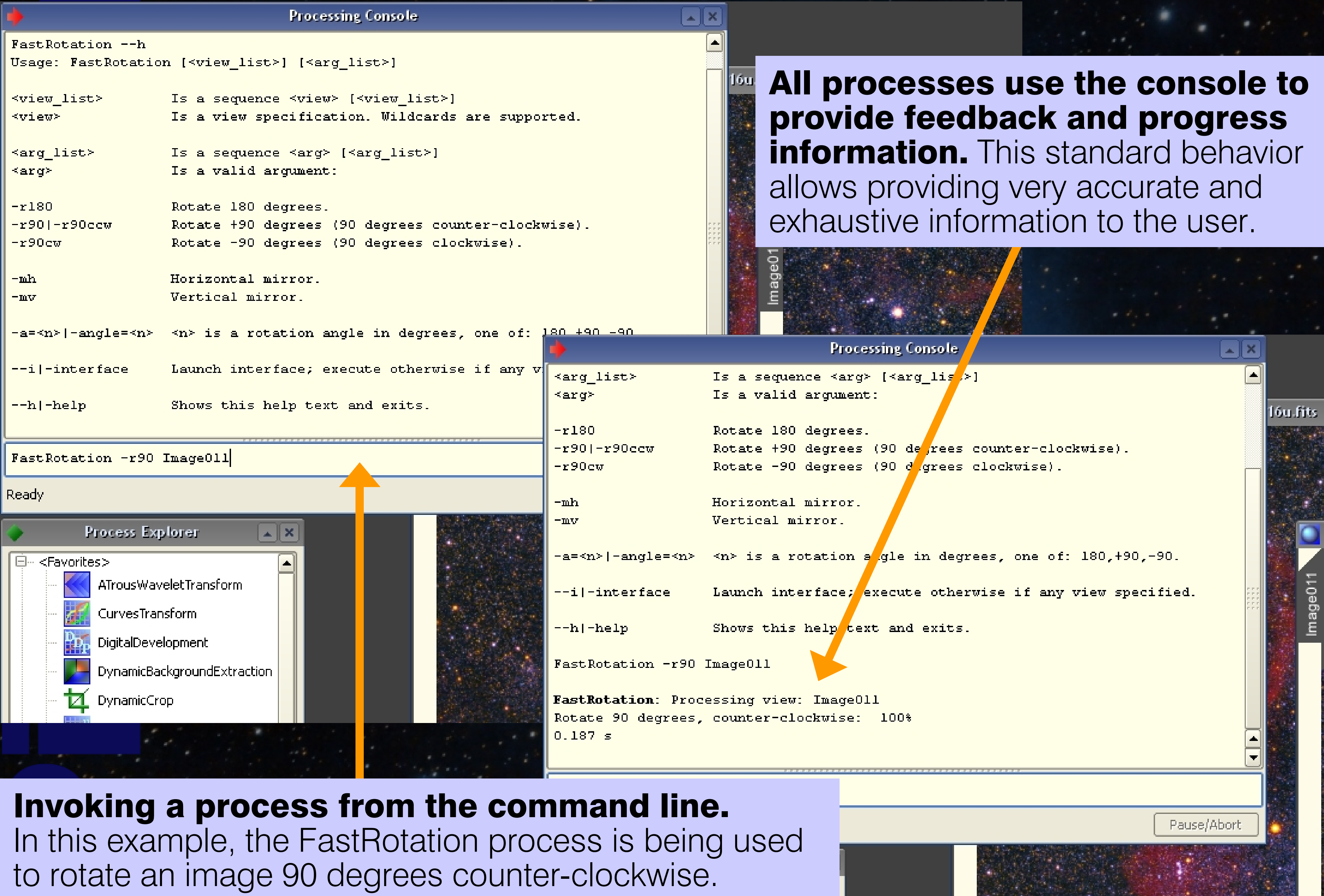
□ Comprehensive set of **internal commands giving access to the entire core application functionality.** Examples: *new, open, close, setid, duplicate, newpreview, exit,* etc.

□ **Every** installed **process can be invoked from the command line.** This functionality is automatically provided by the core PixInsight application.

□ Developers can (and are encouraged to) implement **specialized command-line functionality for newly authored processes.**

□ **Support for script files.** A proprietary scripting language can be used to define processes and sequences of command-line actions.

The Processing Console Window



```
FastRotation --h
Usage: FastRotation [<view_list>] [<arg_list>]

<view_list>        Is a sequence <view> [<view_list>]
<view>             Is a view specification. Wildcards are supported.

<arg_list>         Is a sequence <arg> [<arg_list>]
<arg>              Is a valid argument:

-r180              Rotate 180 degrees.
-r90|-r90ccw       Rotate +90 degrees (90 degrees counter-clockwise).
-r90cw             Rotate -90 degrees (90 degrees clockwise).

-mh                Horizontal mirror.
-mv                Vertical mirror.

-a=<n>|-angle=<n>  <n> is a rotation angle in degrees, one of: 180,+90,-90

--i|-interface     Launch interface; execute otherwise if any v

--h|-help          Shows this help text and exits.
```

```
FastRotation -r90 Image011
```

```
<arg_list>         Is a sequence <arg> [<arg_list>]
<arg>              Is a valid argument:

-r180              Rotate 180 degrees.
-r90|-r90ccw       Rotate +90 degrees (90 degrees counter-clockwise).
-r90cw             Rotate -90 degrees (90 degrees clockwise).

-mh                Horizontal mirror.
-mv                Vertical mirror.

-a=<n>|-angle=<n>  <n> is a rotation angle in degrees, one of: 180,+90,-90.

--i|-interface     Launch interface; execute otherwise if any view specified.

--h|-help          Shows this help text and exits.

FastRotation -r90 Image011

FastRotation: Processing view: Image011
Rotate 90 degrees, counter-clockwise:   100%
0.187 s
```

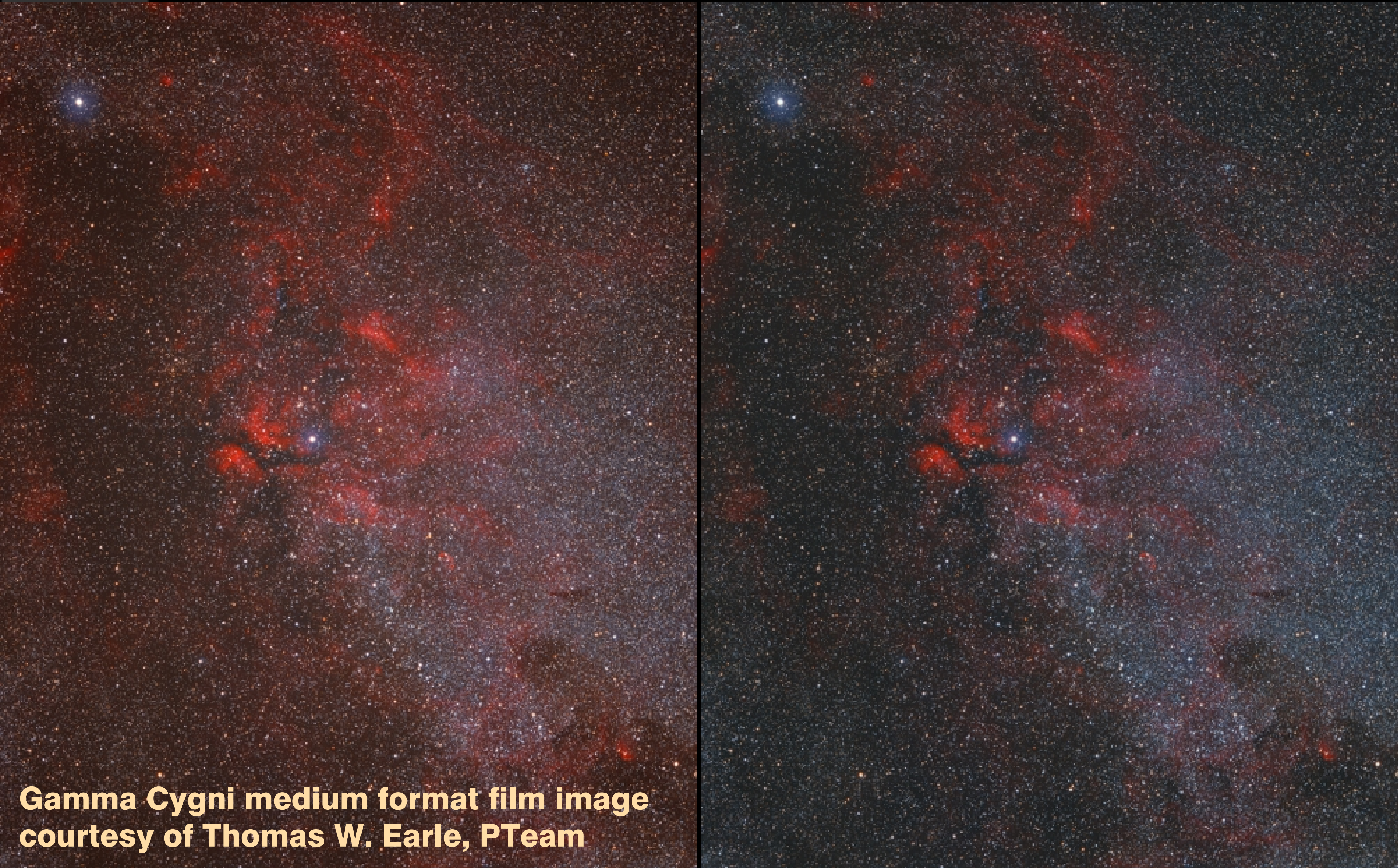
All processes use the console to provide feedback and progress information. This standard behavior allows providing very accurate and exhaustive information to the user.

Invoking a process from the command line. In this example, the FastRotation process is being used to rotate an image 90 degrees counter-clockwise.

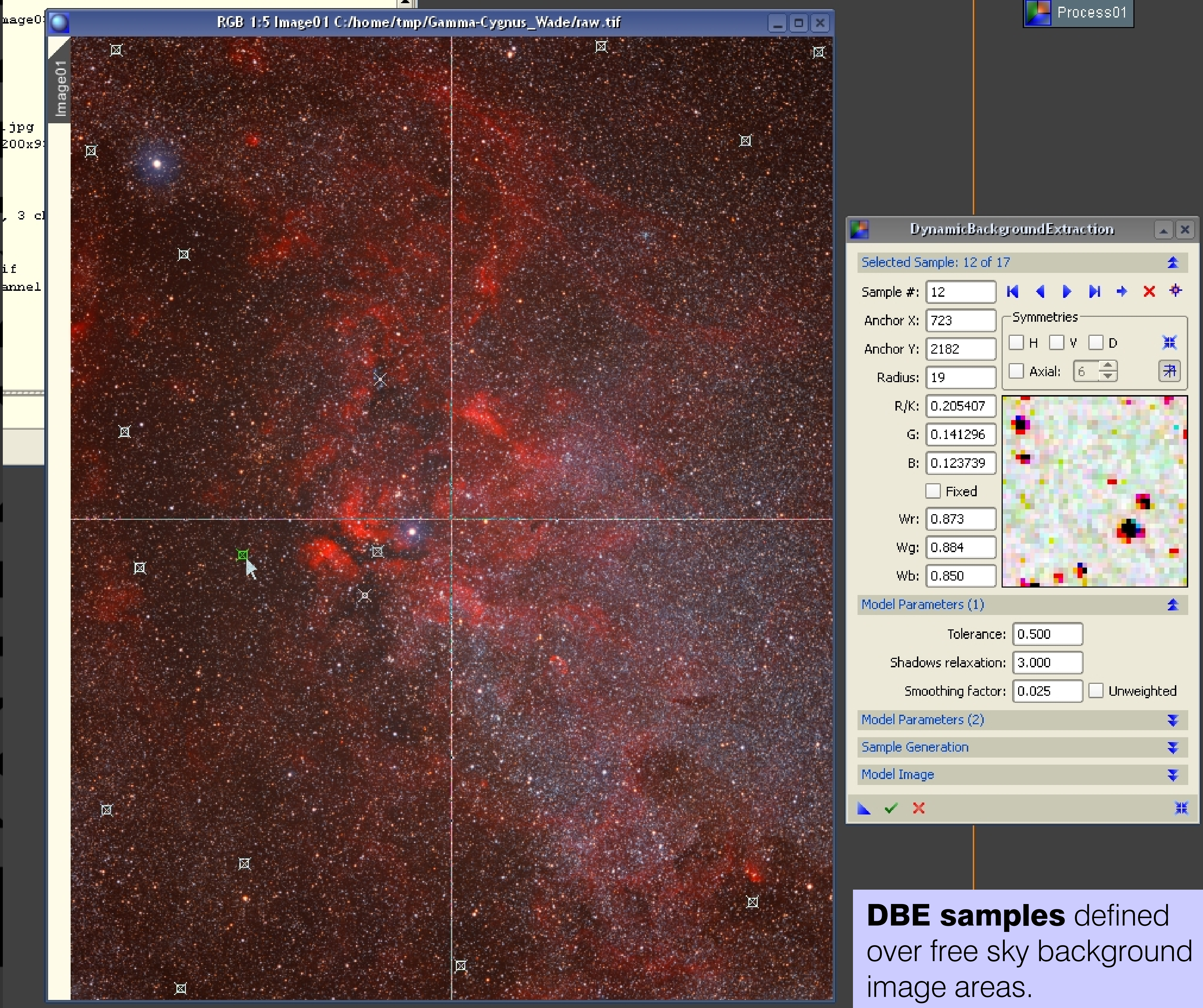
Processing Example

Vignetting and Sky Gradient Correction

DynamicBackgroundExtraction and PixelMath in PixInsight Standard



Gamma Cygni medium format film image
courtesy of Thomas W. Earle, PTeam



RGB 1:5 Image01 C:/home/tmp/Gamma-Cygnus_Wade/raw.tif

DynamicBackgroundExtraction

Selected Sample: 12 of 17

Sample #: 12

Anchor X: 723

Anchor Y: 2182

Radius: 19

R/K: 0.205407

G: 0.141296

B: 0.123739

☐ Fixed

Wr: 0.873

Wg: 0.884

Wb: 0.850

Symmetries: ☐ H ☐ V ☐ D

☐ Axial: 6

Model Parameters (1)

Tolerance: 0.500

Shadows relaxation: 3.000

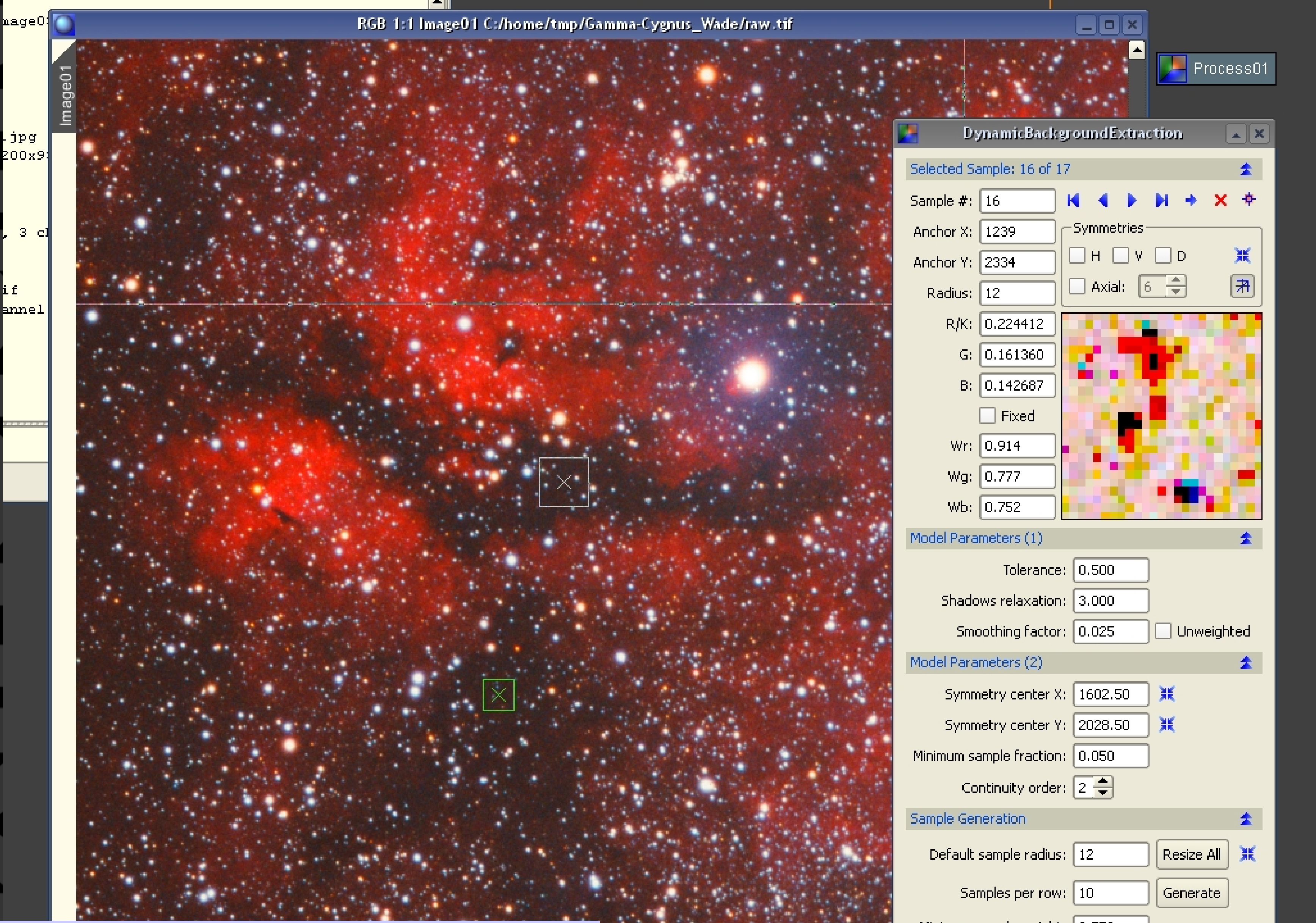
Smoothing factor: 0.025 ☐ Unweighted

Model Parameters (2)

Sample Generation

Model Image

DBE samples defined over free sky background image areas.



RGB 1:1 Image01 C:/home/tmp/Gamma-Cygnus_Wade/raw.tif

DynamicBackgroundExtraction

Selected Sample: 16 of 17

Sample #: 16

Anchor X: 1239

Anchor Y: 2334

Radius: 12

R/K: 0.224412

G: 0.161360

B: 0.142687

□ Fixed

Wr: 0.914

Wg: 0.777

Wb: 0.752

Symmetries: □ H □ V □ D

□ Axial: 6

Model Parameters (1)

Tolerance: 0.500

Shadows relaxation: 3.000

Smoothing factor: 0.025 □ Unweighted

Model Parameters (2)

Symmetry center X: 1602.50

Symmetry center Y: 2028.50

Minimum sample fraction: 0.050

Continuity order: 2

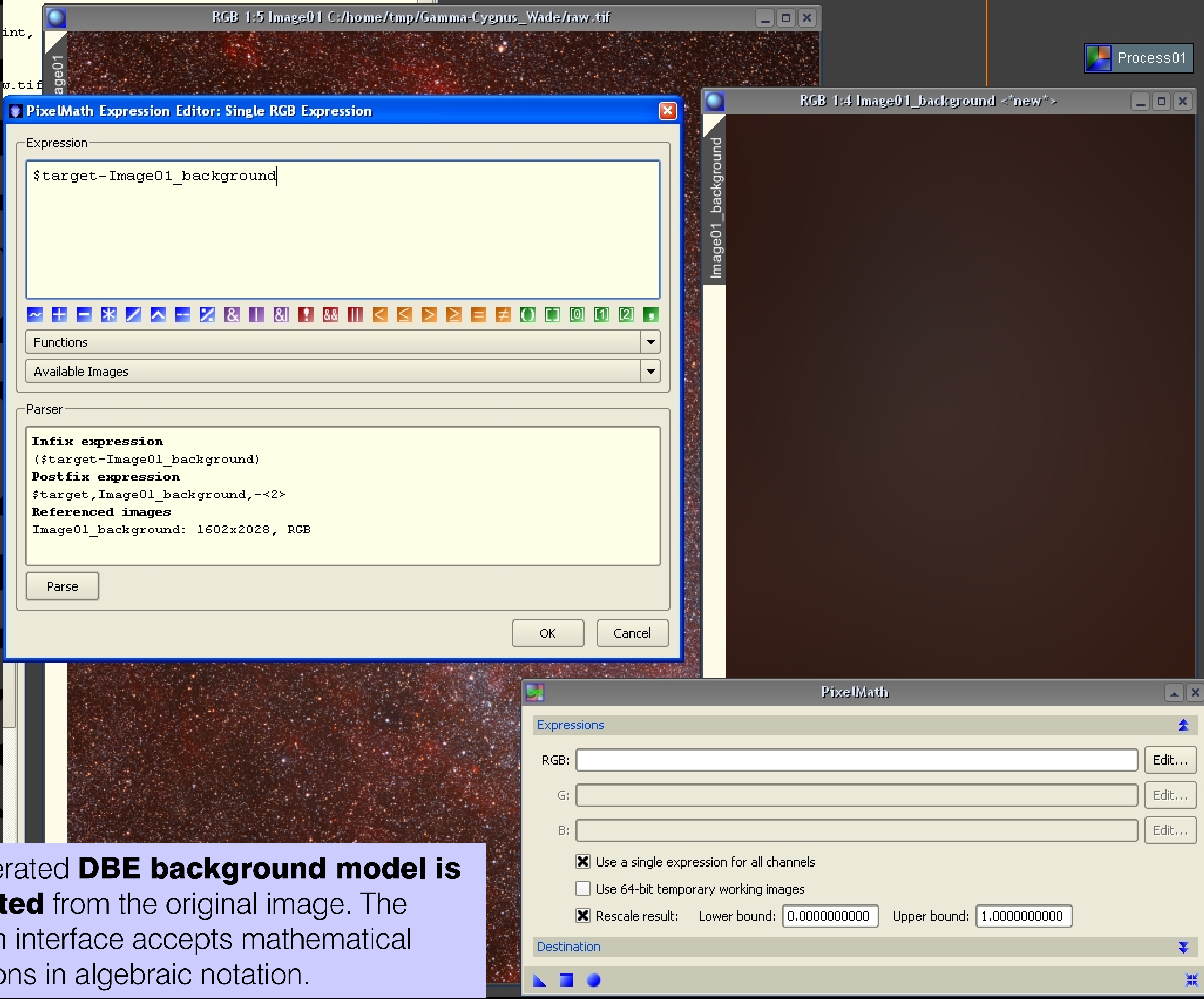
Sample Generation

Default sample radius: 12

Samples per row: 10

Minimum sample weight: 0.750

Samples must be carefully defined on difficult images. A sophisticated rejection algorithm avoids the contributions of stars and other outliers to the generated background model.


RGB 1:5 Image01 C:/home/tmp/Gamma-Cygnus_Wade/raw.tif
RGB 1:4 Image01_background <*new*>
Image01_background
Process01
PixelMath Expression Editor: Single RGB Expression
Expression
Functions
Available Images
Parser
Parse
OK
Cancel
PixelMath
Expressions
Destination

```
$target-Image01_background
```

Infix expression

```
($target-Image01_background)
```

Postfix expression

```
$target,Image01_background,-<2>
```

Referenced images

```
Image01_background: 1602x2028, RGB
```

RGB: Edit...

G: Edit...

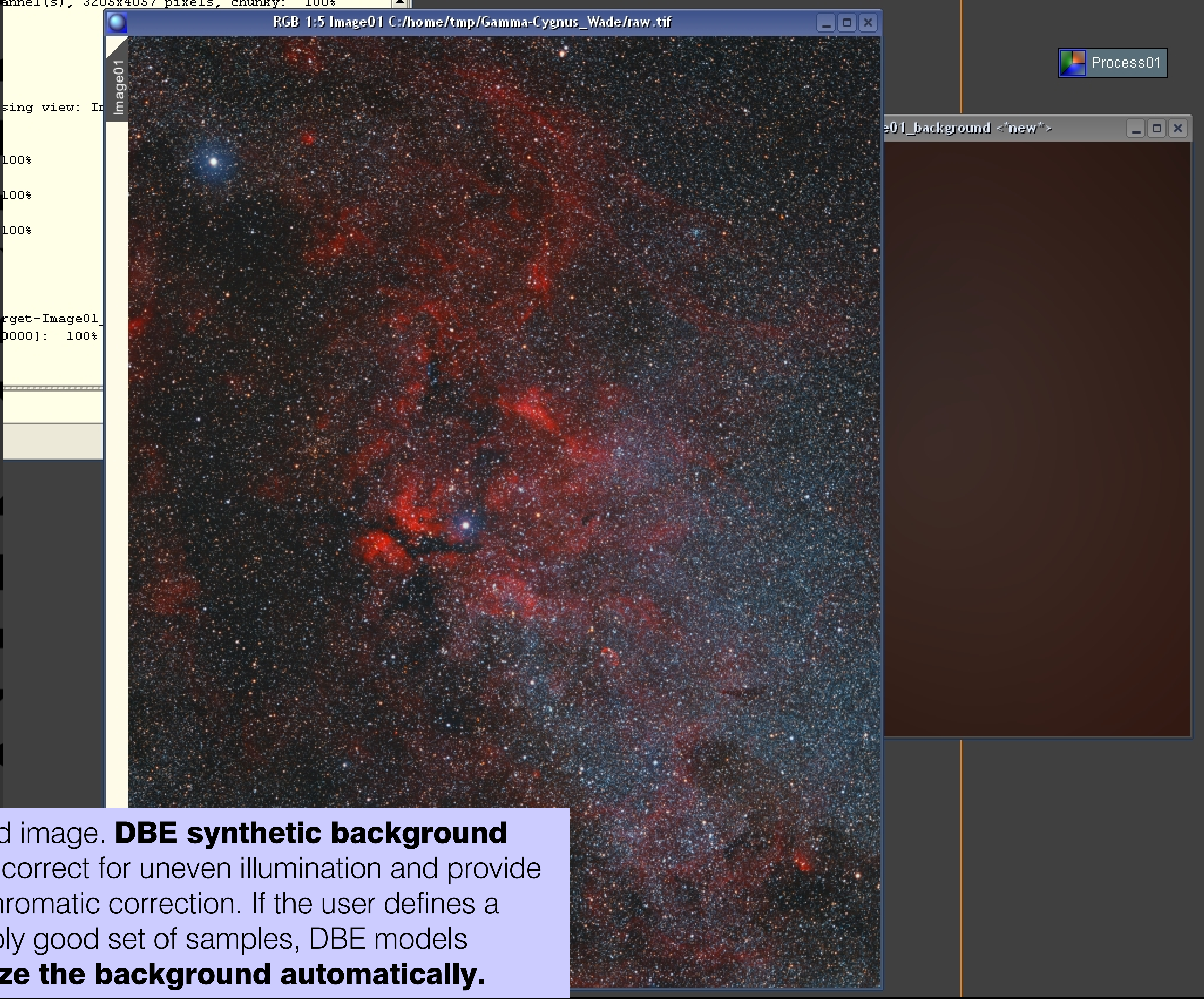
B: Edit...

☒ Use a single expression for all channels

☐ Use 64-bit temporary working images

☒ Rescale result: Lower bound: 0.0000000000 Upper bound: 1.0000000000

The generated **DBE background model is subtracted** from the original image. The PixelMath interface accepts mathematical expressions in algebraic notation.

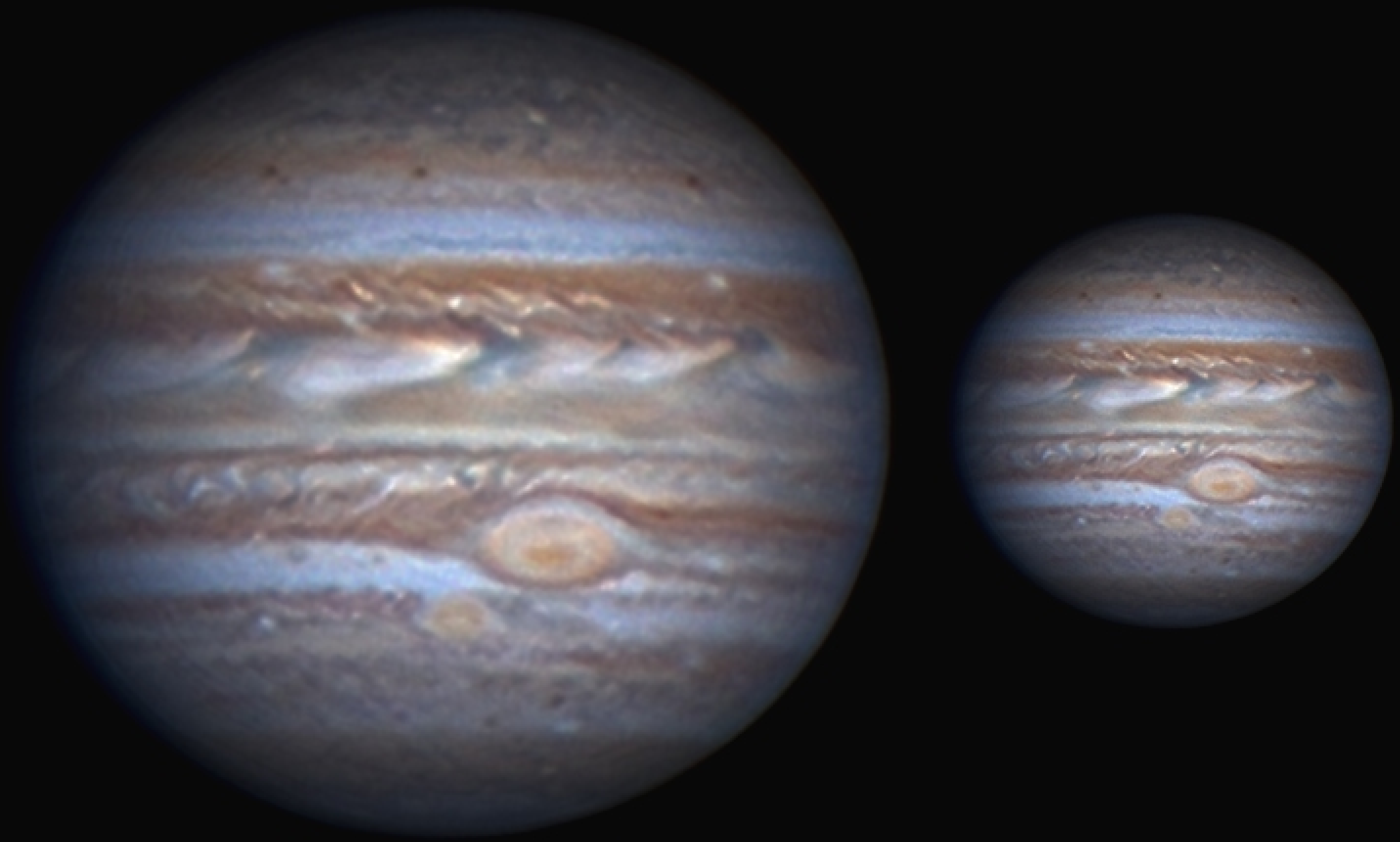

```
annel(s), 3203x4037 pixels, chunky:  100%

sing view: Im

100%

100%

100%

rget-Image01_
0000]:  100%
```

Corrected image. **DBE synthetic background models** correct for uneven illumination and provide robust chromatic correction. If the user defines a reasonably good set of samples, DBE models **neutralize the background automatically.**

Processing Example

Processing a High-Resolution Jupiter Image

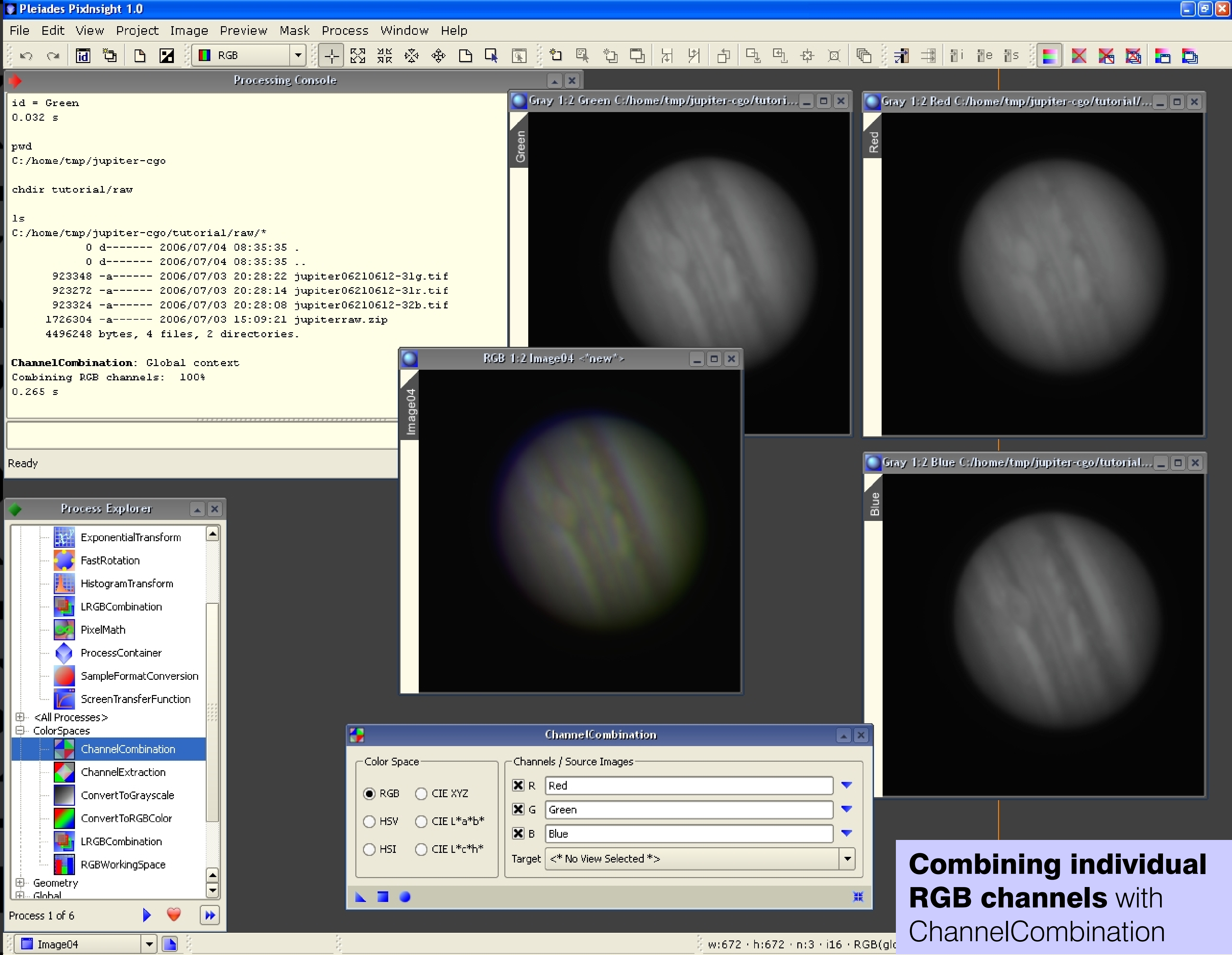
ATrousWaveletTransform in PixInsight Standard



Raw Jupiter image data courtesy of Christopher Go

The complete tutorial is available on our website:

http://pleiades-astrophoto.com/tutorials/



```
id = Green
0.032 s

pwd
C:/home/tmp/jupiter-cgo

chdir tutorial/raw

ls
C:/home/tmp/jupiter-cgo/tutorial/raw/*
           0 d------- 2006/07/04 08:35:35 .
           0 d------- 2006/07/04 08:35:35 ..
      923348 -a------ 2006/07/03 20:28:22 jupiter06210612-31g.tif
      923272 -a------ 2006/07/03 20:28:14 jupiter06210612-31r.tif
      923324 -a------ 2006/07/03 20:28:08 jupiter06210612-32b.tif
     1726304 -a------ 2006/07/03 15:09:21 jupiterraw.zip
     4496248 bytes, 4 files, 2 directories.

ChannelCombination: Global context
Combining RGB channels:  100%
0.265 s
```

Ready

ChannelCombination

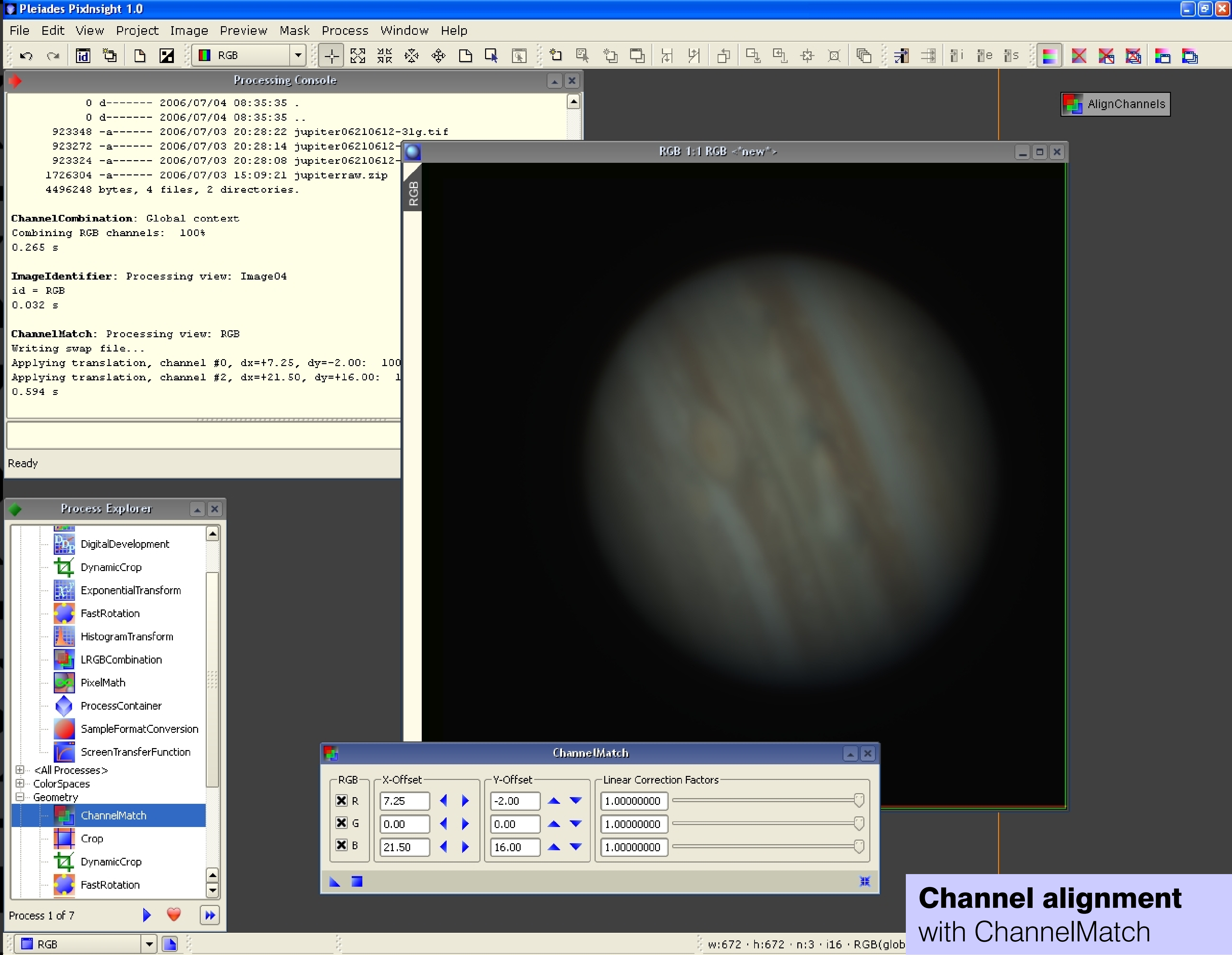
Color Space

● RGB ○ CIE XYZ
○ HSV ○ CIE L*a*b*
○ HSI ○ CIE L*c*h*

Channels / Source Images

☒ R Red
☒ G Green
☒ B Blue
Target <* No View Selected *>

Combining individual RGB channels with ChannelCombination



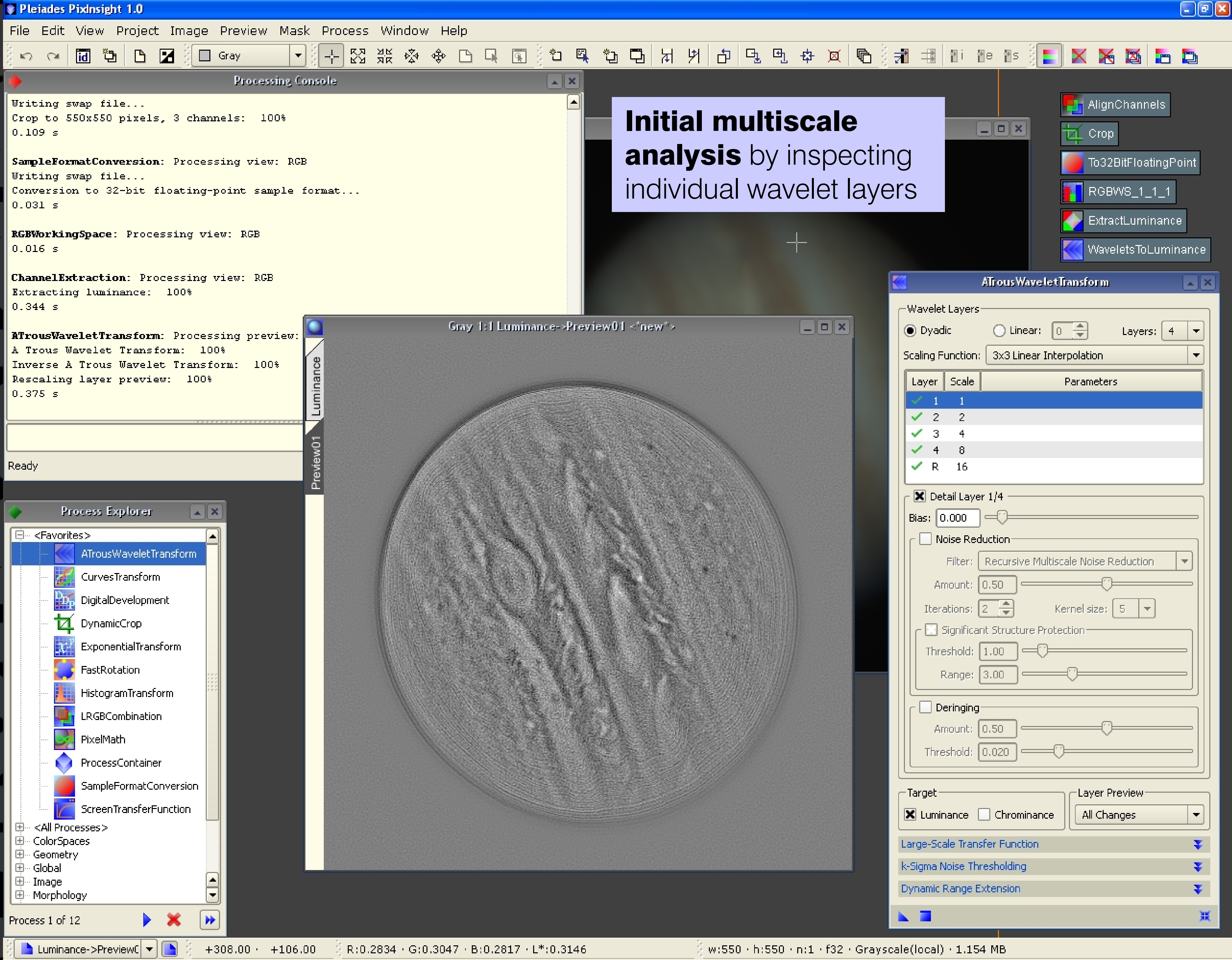
```
           0 d------- 2006/07/04 08:35:35 .
           0 d------- 2006/07/04 08:35:35 ..
      923348 -a------ 2006/07/03 20:28:22 jupiter06210612-31g.tif
      923272 -a------ 2006/07/03 20:28:14 jupiter06210612-
      923324 -a------ 2006/07/03 20:28:08 jupiter06210612-
     1726304 -a------ 2006/07/03 15:09:21 jupiterraw.zip
     4496248 bytes, 4 files, 2 directories.

ChannelCombination: Global context
Combining RGB channels:  100%
0.265 s

ImageIdentifier: Processing view: Image04
id = RGB
0.032 s

ChannelMatch: Processing view: RGB
Writing swap file...
Applying translation, channel #0, dx=+7.25, dy=-2.00:  100
Applying translation, channel #2, dx=+21.50, dy=+16.00:  1
0.594 s
```

Ready

RGB	X-Offset	Y-Offset	Linear Correction Factors
☒ R	7.25	-2.00	1.00000000
☒ G	0.00	0.00	1.00000000
☒ B	21.50	16.00	1.00000000

Channel alignment
with ChannelMatch

Initial multiscale analysis by inspecting individual wavelet layers



Processing Console

```
Writing swap file...
Crop to 550x550 pixels, 3 channels:  100%
0.109 s

SampleFormatConversion: Processing view: RGB
Writing swap file...
Conversion to 32-bit floating-point sample format...
0.031 s

RGBWorkingSpace: Processing view: RGB
0.016 s

ChannelExtraction: Processing view: RGB
Extracting luminance:  100%
0.344 s

ATrousWaveletTransform: Processing preview:
A Trous Wavelet Transform:  100%
Inverse A Trous Wavelet Transform:  100%
Rescaling layer preview:  100%
0.375 s
```

Ready

Process Explorer

- <Favorites>
  - ATrousWaveletTransform
  - CurvesTransform
  - DigitalDevelopment
  - DynamicCrop
  - ExponentialTransform
  - FastRotation
  - HistogramTransform
  - LRGBCombination
  - PixelMath
  - ProcessContainer
  - SampleFormatConversion
  - ScreenTransferFunction
- <All Processes>
- ColorSpaces
- Geometry
- Global
- Image
- Morphology

Process 1 of 12

ATrousWaveletTransform

Wavelet Layers

● Dyadic ○ Linear: 0 Layers: 4

Scaling Function: 3x3 Linear Interpolation

Layer	Scale	Parameters
✓ 1	1	
✓ 2	2	
✓ 3	4	
✓ 4	8	
✓ R	16	

☒ Detail Layer 1/4

Bias: 0.000

□ Noise Reduction

Filter: Recursive Multiscale Noise Reduction

Amount: 0.50

Iterations: 2 Kernel size: 5

□ Significant Structure Protection

Threshold: 1.00

Range: 3.00

□ Deringing

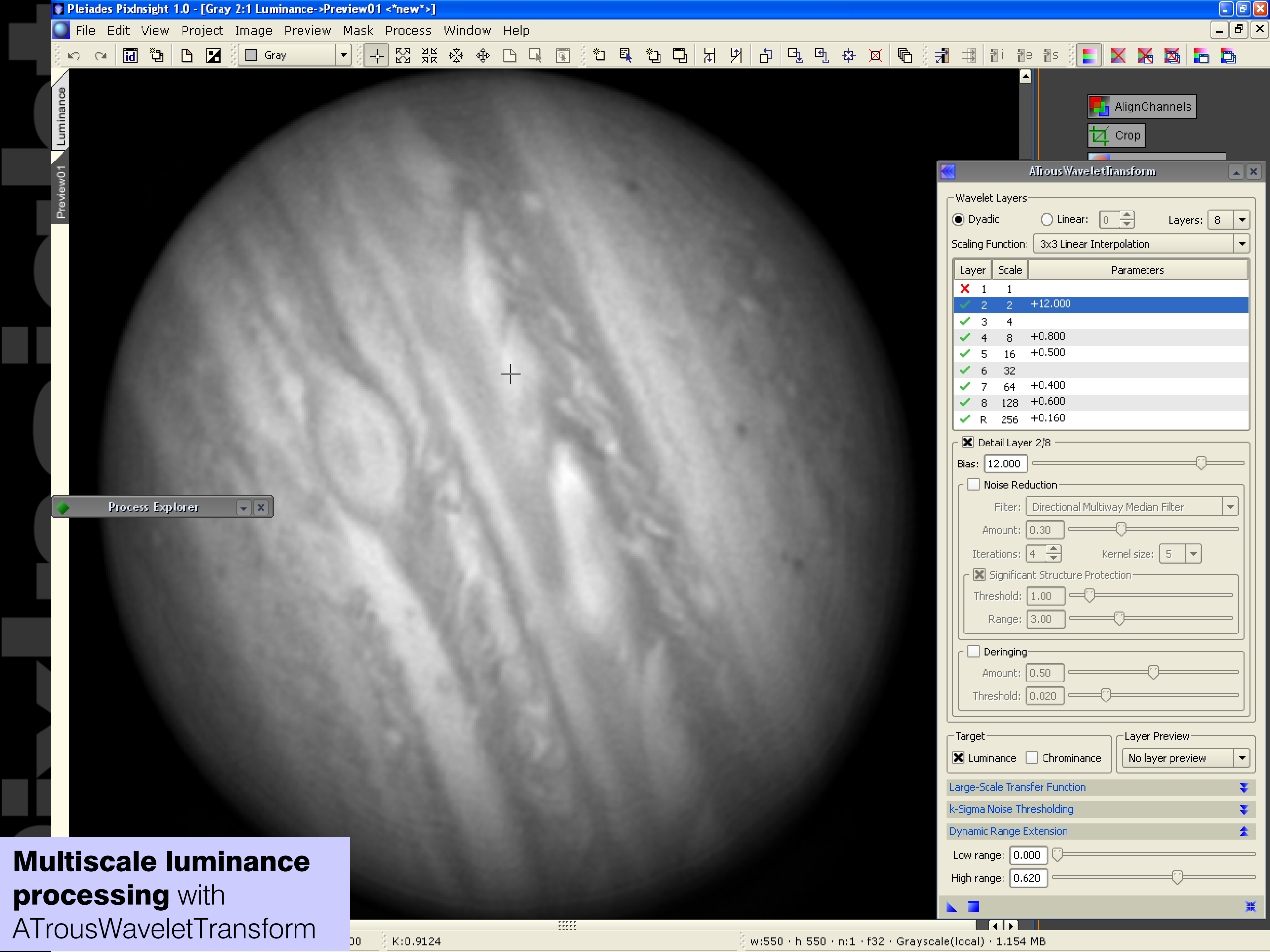
Amount: 0.50

Threshold: 0.020

Target: ☒ Luminance □ Chrominance

Layer Preview: All Changes

Luminance->Preview0 +308.00 · +106.00 R:0.2834 · G:0.3047 · B:0.2817 · L*:0.3146 w:550 · h:550 · n:1 · f32 · Grayscale(local) · 1.154 MB



ATrousWaveletTransform

Wavelet Layers

● Dyadic ○ Linear: 0 Layers: 8

Scaling Function: 3x3 Linear Interpolation

	Layer	Scale	Parameters
✗	1	1	
✓	2	2	+12.000
✓	3	4	
✓	4	8	+0.800
✓	5	16	+0.500
✓	6	32	
✓	7	64	+0.400
✓	8	128	+0.600
✓	R	256	+0.160

☒ Detail Layer 2/8

Bias: 12.000

□ Noise Reduction

Filter: Directional Multiway Median Filter

Amount: 0.30

Iterations: 4 Kernel size: 5

☒ Significant Structure Protection

Threshold: 1.00

Range: 3.00

□ Deringing

Amount: 0.50

Threshold: 0.020

Target: ☒ Luminance □ Chrominance

Layer Preview: No layer preview

Large-Scale Transfer Function

k-Sigma Noise Thresholding

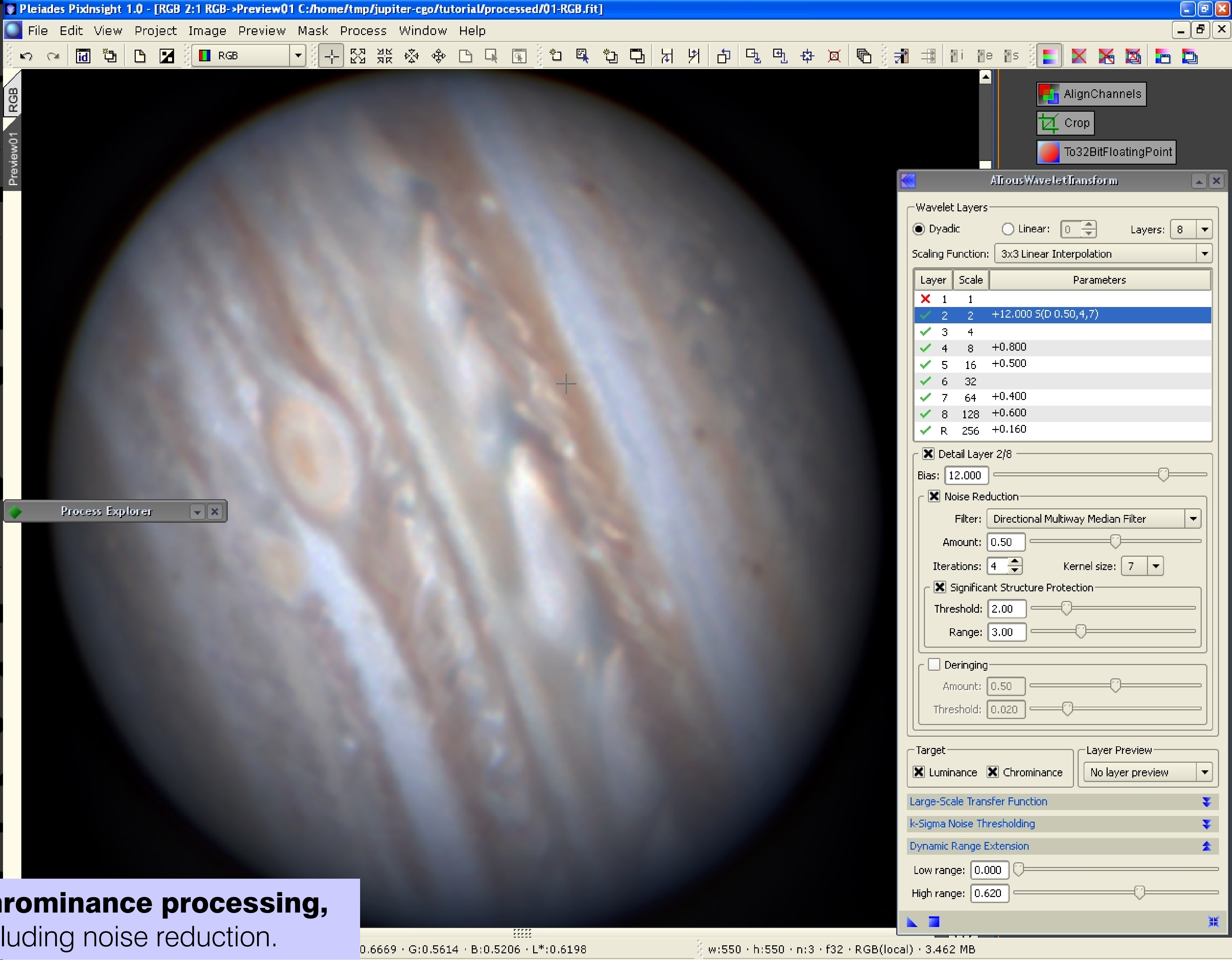
Dynamic Range Extension

Low range: 0.000

High range: 0.620

00 K:0.9124 w:550 · h:550 · n:1 · f32 · Grayscale(local) · 1.154 MB

Multiscale luminance processing with ATrousWaveletTransform



ATrousWaveletTransform

Wavelet Layers

● Dyadic ○ Linear: 0 Layers: 8

Scaling Function: 3x3 Linear Interpolation

	Layer	Scale	Parameters
✗	1	1	
✓	2	2	+12.000 S(D 0.50,4,7)
✓	3	4	
✓	4	8	+0.800
✓	5	16	+0.500
✓	6	32	
✓	7	64	+0.400
✓	8	128	+0.600
✓	R	256	+0.160

☒ Detail Layer 2/8

Bias: 12.000

☒ Noise Reduction

Filter: Directional Multiway Median Filter

Amount: 0.50

Iterations: 4 Kernel size: 7

☒ Significant Structure Protection

Threshold: 2.00

Range: 3.00

☐ Deringing

Amount: 0.50

Threshold: 0.020

Target: ☒ Luminance ☒ Chrominance

Layer Preview: No layer preview

Large-Scale Transfer Function

k-Sigma Noise Thresholding

Dynamic Range Extension

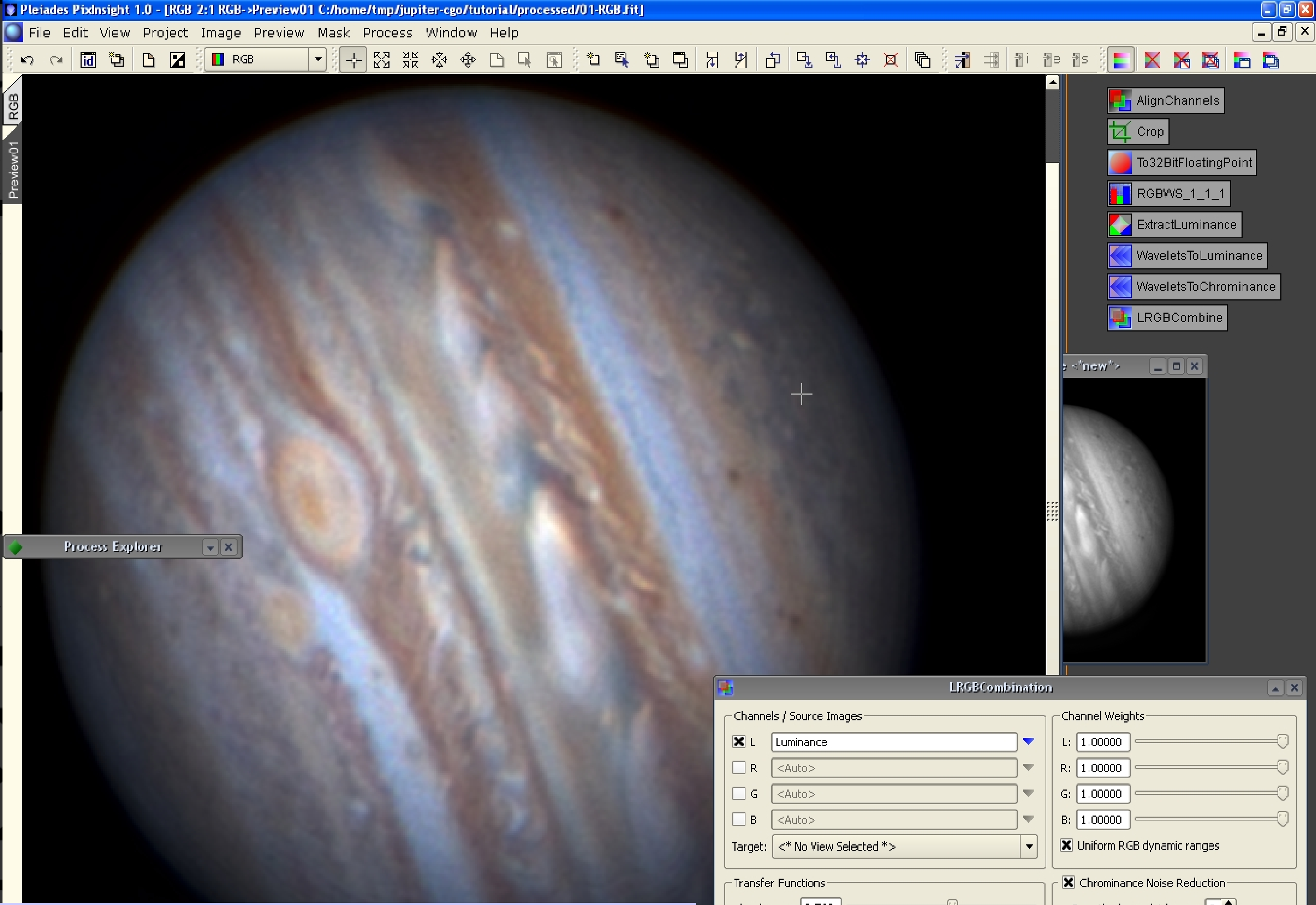
Low range: 0.000

High range: 0.620

0.6669 · G:0.5614 · B:0.5206 · L*:0.6198

w:550 · h:550 · n:3 · f32 · RGB(local) · 3.462 MB

Chrominance processing,
including noise reduction.



LRGBCombination

Channels / Source Images

☒	L	Luminance
□	R	<Auto>
□	G	<Auto>
□	B	<Auto>

Target: <* No View Selected *>

Channel Weights

L:	1.00000
R:	1.00000
G:	1.00000
B:	1.00000

☒ Uniform RGB dynamic ranges

Transfer Functions

Luminance: 0.560

Saturation: 0.340

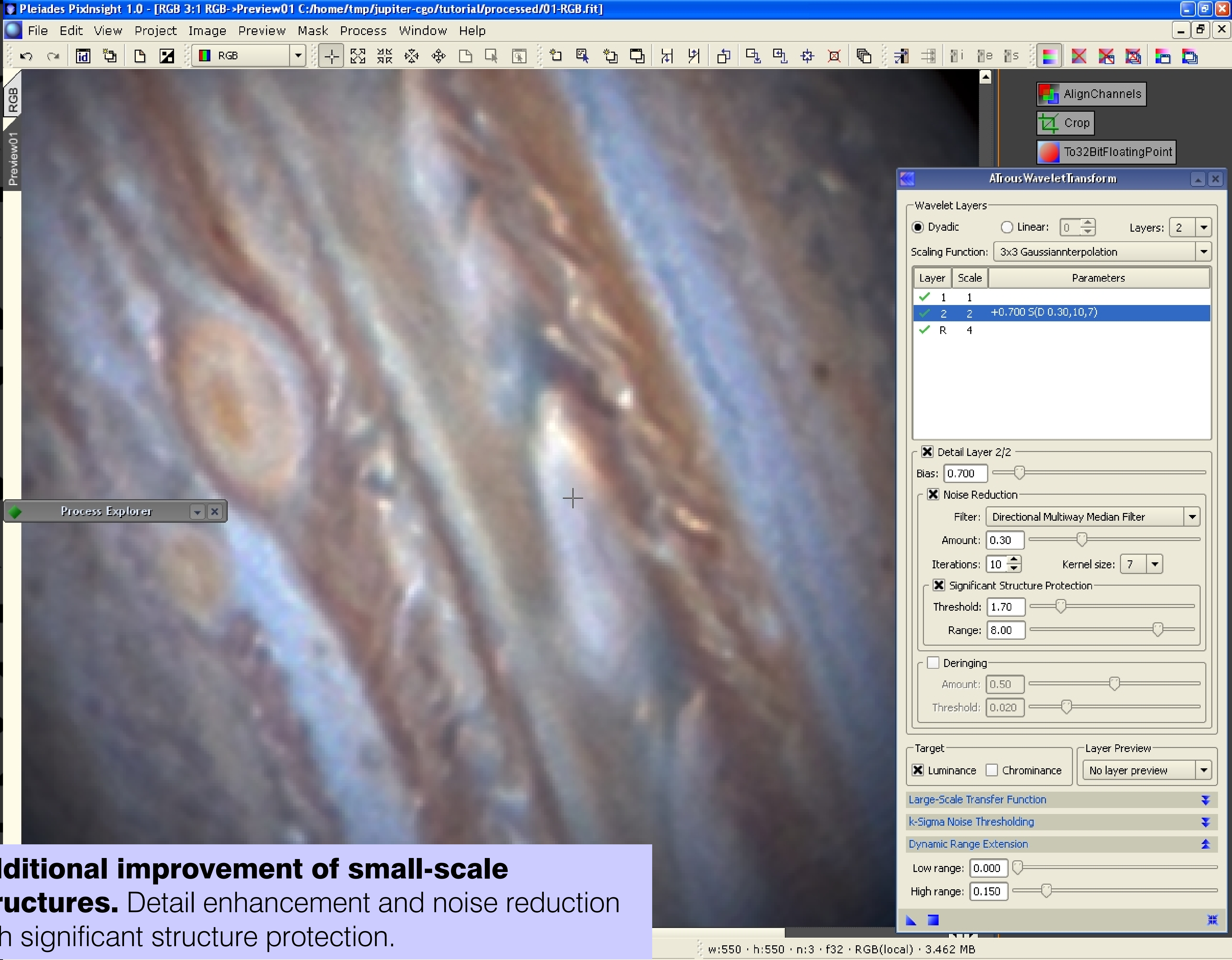
☒ Chrominance Noise Reduction

Smoothed wavelet layers: 2

Protected wavelet layers: 1

w:550 · h:550 · n:3 · f32 · RGB(local) · 3.462 MB

LRGB combination of the processed luminance and chrominance, including color saturation enhancement plus additional chrominance noise reduction.



Pleiades PixInsight 1.0 - [RGB 3:1 RGB->Preview01 C:/home/tmp/jupiter-cgo/tutorial/processed/01-RGB.fit]

ATrousWaveletTransform

Wavelet Layers

● Dyadic ○ Linear: 0 Layers: 2

Scaling Function: 3x3 GaussianInterpolation

	Layer	Scale	Parameters
✓	1	1	
✓	2	2	+0.700 S(D 0.30,10,7)
✓	R	4	

☒ Detail Layer 2/2

Bias: 0.700

☒ Noise Reduction

Filter: Directional Multiway Median Filter

Amount: 0.30

Iterations: 10 Kernel size: 7

☒ Significant Structure Protection

Threshold: 1.70

Range: 8.00

□ Deringing

Amount: 0.50

Threshold: 0.020

Target: ☒ Luminance □ Chrominance

Layer Preview: No layer preview

Large-Scale Transfer Function

k-Sigma Noise Thresholding

Dynamic Range Extension

Low range: 0.000

High range: 0.150

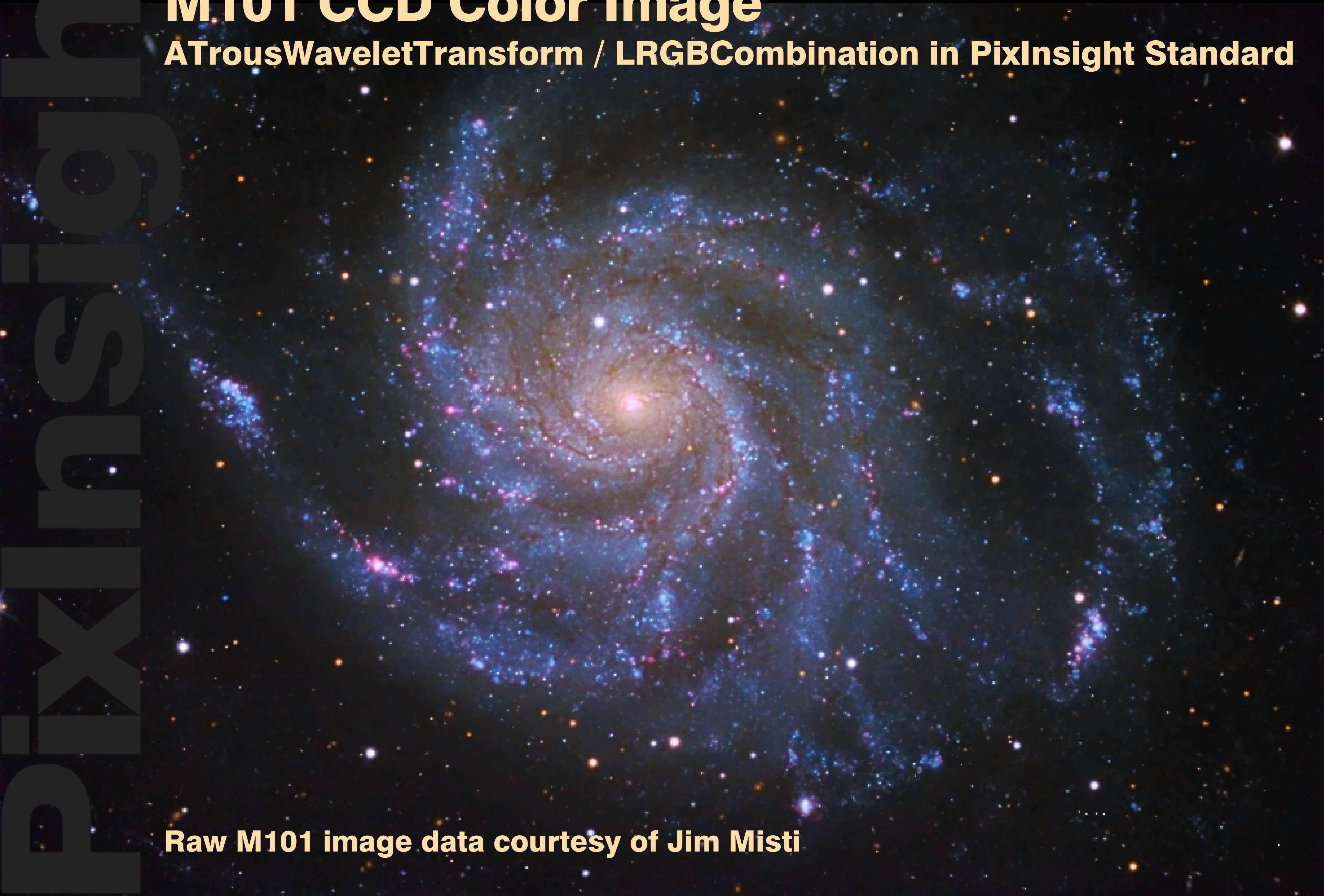
w:550 · h:550 · n:3 · f32 · RGB(local) · 3.462 MB

Additional improvement of small-scale structures. Detail enhancement and noise reduction with significant structure protection.

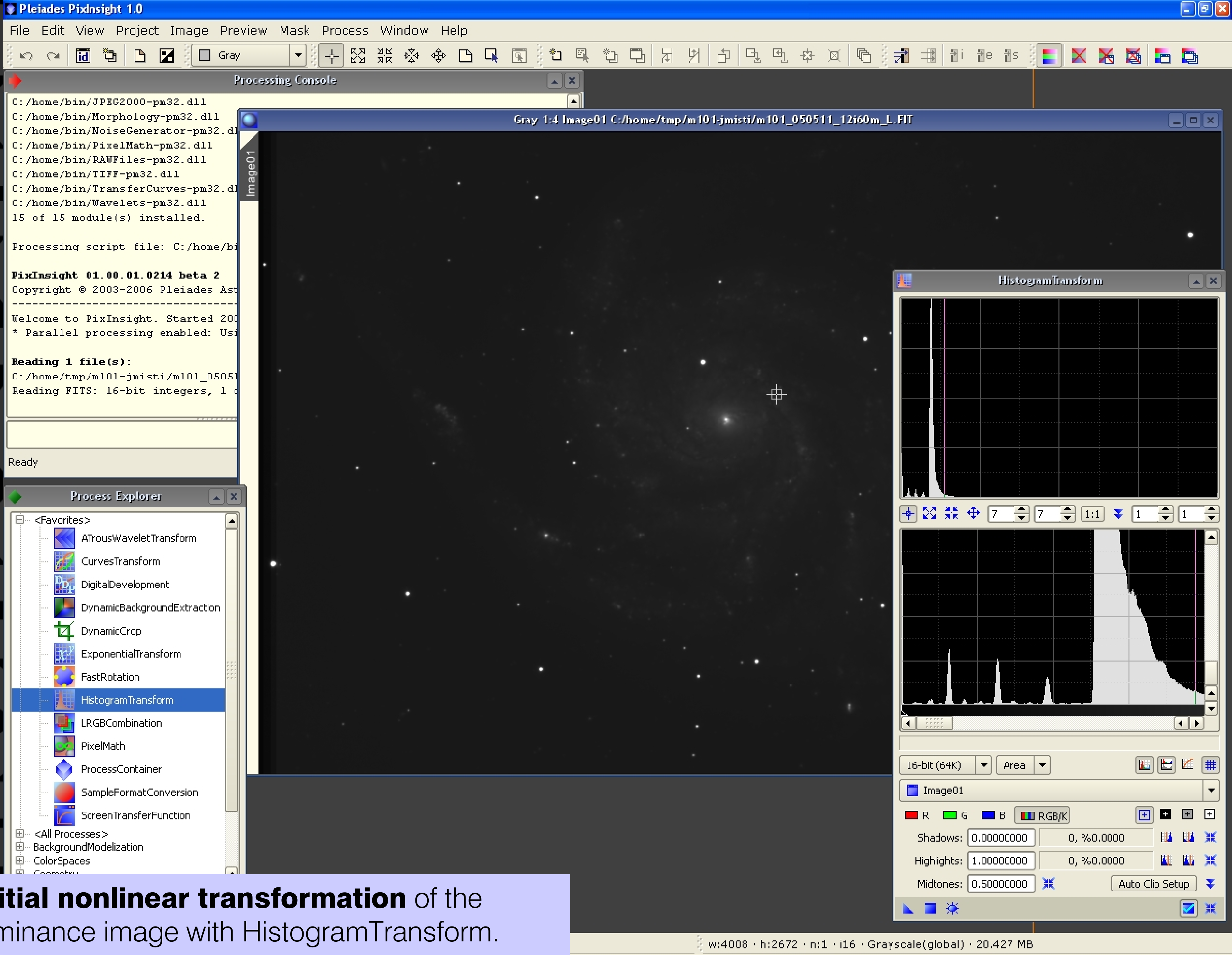
Processing Example

M101 CCD Color Image

ATrousWaveletTransform / LRGBCombination in PixInsight Standard



Raw M101 image data courtesy of Jim Misti



```
C:/home/bin/JPEG2000-pm32.dll
C:/home/bin/Morphology-pm32.dll
C:/home/bin/NoiseGenerator-pm32.dl
C:/home/bin/PixelMath-pm32.dll
C:/home/bin/RAWFiles-pm32.dll
C:/home/bin/TIFF-pm32.dll
C:/home/bin/TransferCurves-pm32.dl
C:/home/bin/Wavelets-pm32.dll
15 of 15 module(s) installed.

Processing script file: C:/home/bi

PixInsight 01.00.01.0214 beta 2
Copyright © 2003-2006 Pleiades Ast
----------------------------------
Welcome to PixInsight. Started 200
* Parallel processing enabled: Usi

Reading 1 file(s):
C:/home/tmp/m101-jmisti/m101_05051
Reading FITS: 16-bit integers, 1 c
```

Ready

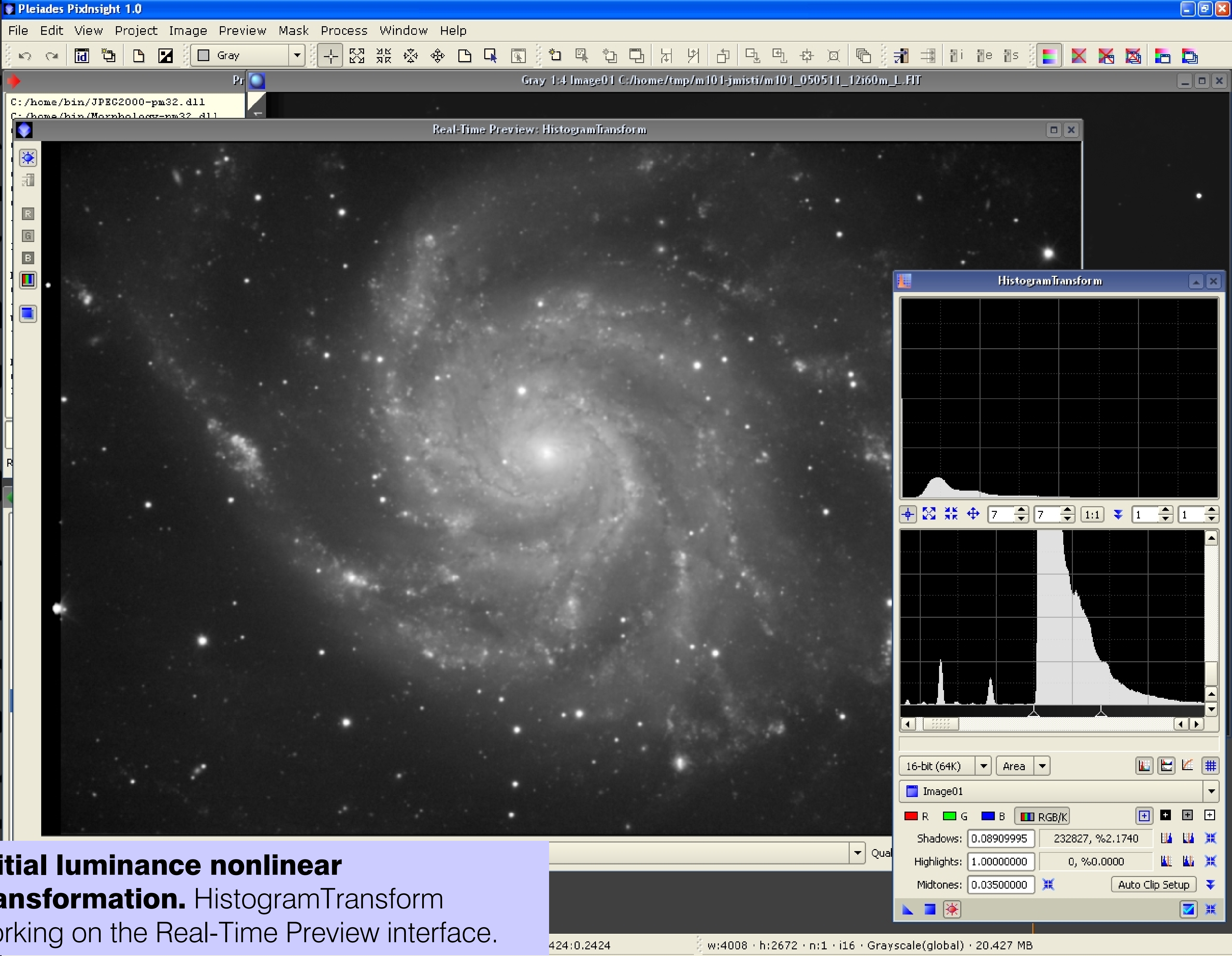
Shadows: 0.00000000 0, %0.0000

Highlights: 1.00000000 0, %0.0000

Midtones: 0.50000000

w:4008 · h:2672 · n:1 · i16 · Grayscale(global) · 20.427 MB

Initial nonlinear transformation of the luminance image with HistogramTransform.



Gray 1:4 Image01 C:/home/tmp/m101-jmisti/m101_050511_12i60m_L.FIT

```
C:/home/bin/JPEG2000-pm32.dll
C:/home/bin/Morphology-pm32.dll
```

Real-Time Preview: HistogramTransform

HistogramTransform

16-bit (64K) · Area

Image01

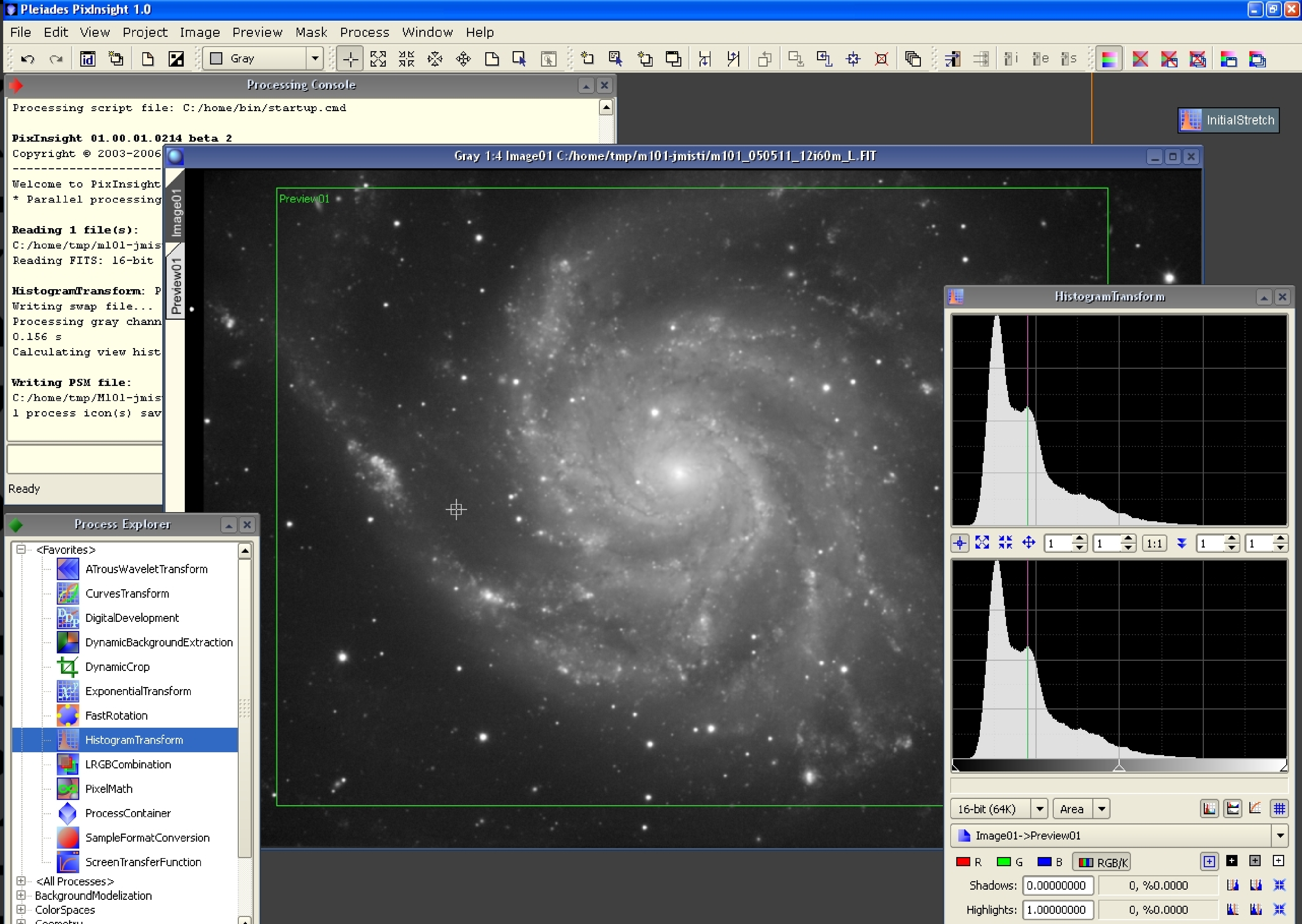
Shadows: 0.08909995 · 232827, %2.1740

Highlights: 1.00000000 · 0, %0.0000

Midtones: 0.03500000

```
424:0.2424   w:4008 · h:2672 · n:1 · i16 · Grayscale(global) · 20.427 MB
```

Initial luminance nonlinear transformation. HistogramTransform working on the Real-Time Preview interface.



```
Processing script file: C:/home/bin/startup.cmd

PixInsight 01.00.01.0214 beta 2
Copyright © 2003-2006
--------------------
Welcome to PixInsight
* Parallel processing

Reading 1 file(s):
C:/home/tmp/m101-jmis
Reading FITS: 16-bit

HistogramTransform: P
Writing swap file...
Processing gray chann
0.156 s
Calculating view hist

Writing PSM file:
C:/home/tmp/M101-jmis
1 process icon(s) sav
```

Ready

16-bit (64K) | Area

Image01->Preview01

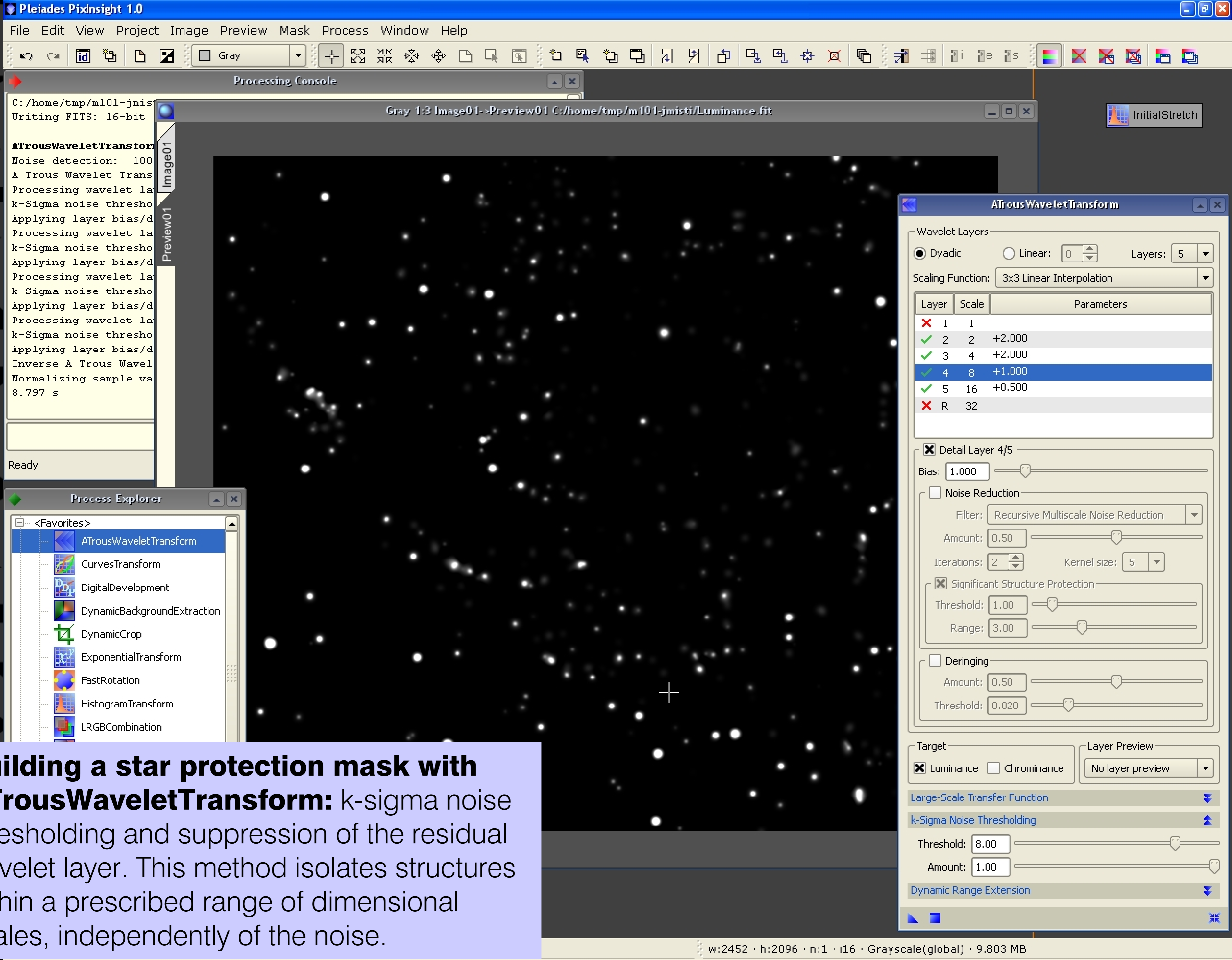
Shadows: 0.00000000 | 0, %0.0000

Highlights: 1.00000000 | 0, %0.0000

Midtones: 0.50000000

w:4008 · h:2672 · n:1 · i16 · Grayscale(global) · 20.427 MB · Modified

Verifying the resulting histogram on a preview defined to avoid spurious data.



```
C:/home/tmp/m101-jmist
Writing FITS: 16-bit

ATrousWaveletTransfor
Noise detection:  100
A Trous Wavelet Trans
Processing wavelet la
k-Sigma noise thresho
Applying layer bias/d
Processing wavelet la
k-Sigma noise thresho
Applying layer bias/d
Processing wavelet la
k-Sigma noise thresho
Applying layer bias/d
Processing wavelet la
k-Sigma noise thresho
Applying layer bias/d
Inverse A Trous Wavel
Normalizing sample va
8.797 s
```

ATrousWaveletTransform

Wavelet Layers

● Dyadic ○ Linear: 0 Layers: 5

Scaling Function: 3x3 Linear Interpolation

	Layer	Scale	Parameters
✗	1	1	
✓	2	2	+2.000
✓	3	4	+2.000
✓	4	8	+1.000
✓	5	16	+0.500
✗	R	32	

☒ Detail Layer 4/5

Bias: 1.000

☐ Noise Reduction

Filter: Recursive Multiscale Noise Reduction

Amount: 0.50

Iterations: 2 Kernel size: 5

☒ Significant Structure Protection

Threshold: 1.00

Range: 3.00

☐ Deringing

Amount: 0.50

Threshold: 0.020

Target: ☒ Luminance ☐ Chrominance

Layer Preview: No layer preview

Large-Scale Transfer Function

k-Sigma Noise Thresholding

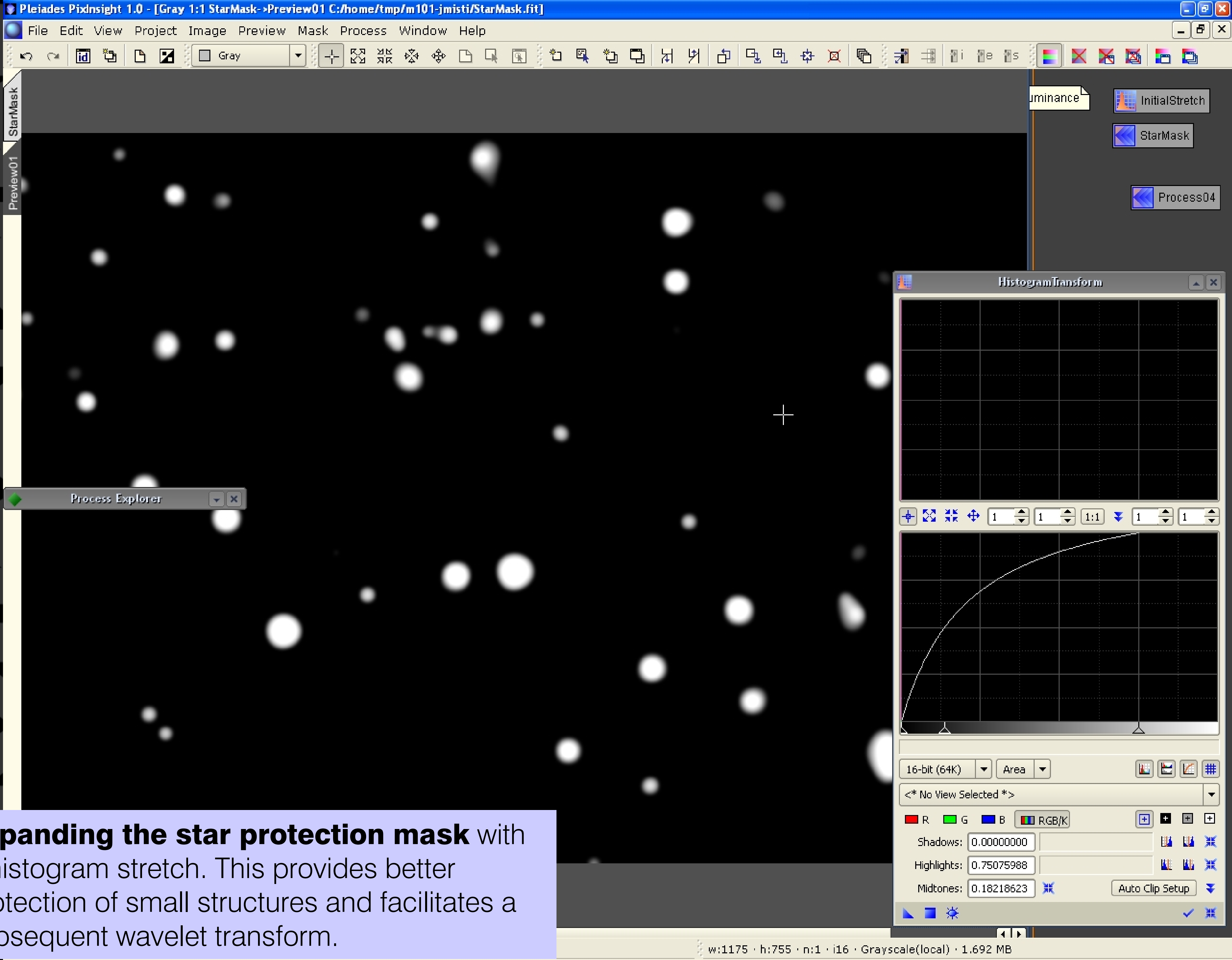
Threshold: 8.00

Amount: 1.00

Dynamic Range Extension

w:2452 · h:2096 · n:1 · i16 · Grayscale(global) · 9.803 MB

Building a star protection mask with ATrousWaveletTransform: k-sigma noise thresholding and suppression of the residual wavelet layer. This method isolates structures within a prescribed range of dimensional scales, independently of the noise.



Pleiades PixInsight 1.0 - [Gray 1:1 StarMask->Preview01 C:/home/tmp/m101-jmisti/StarMask.fit]

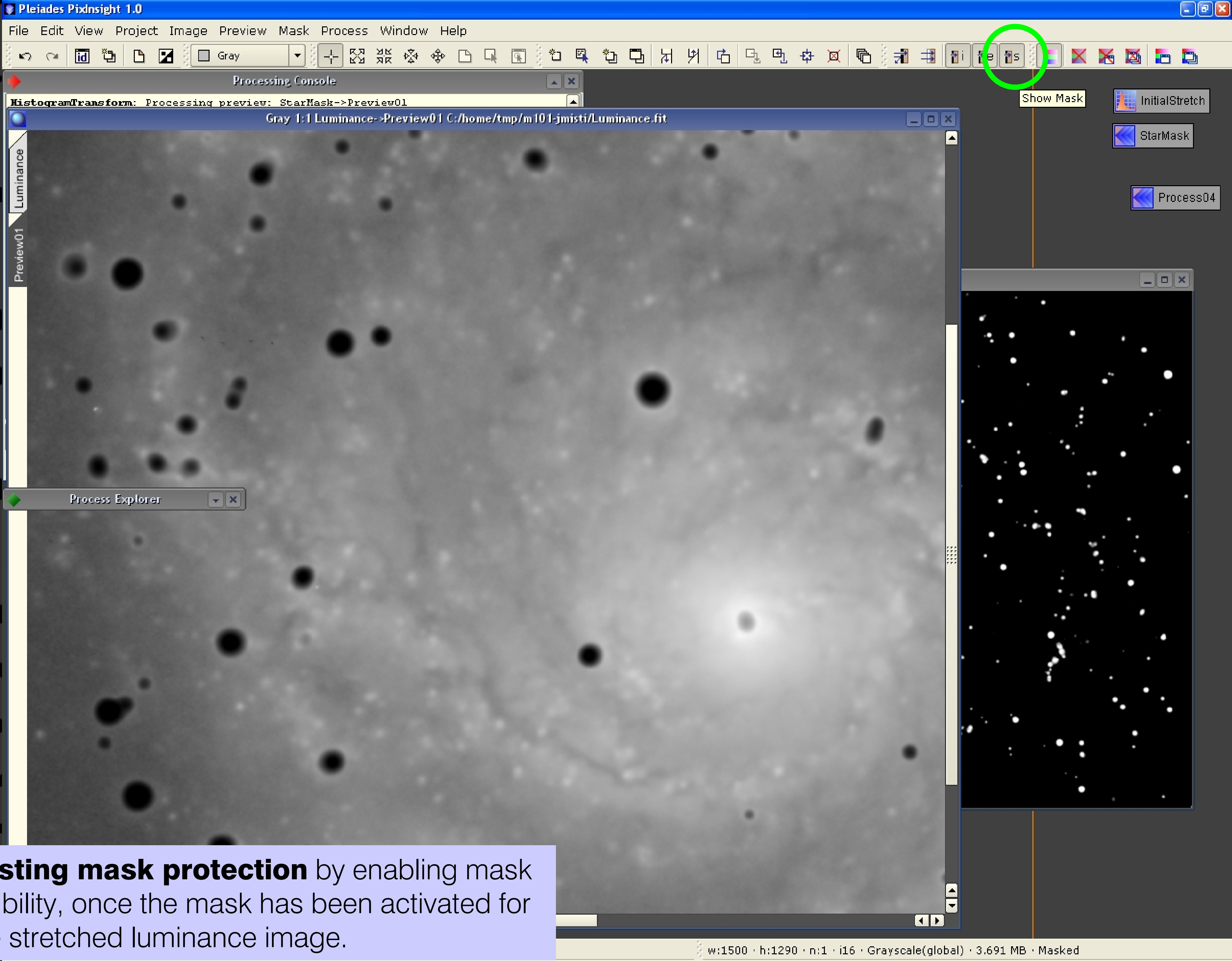
Shadows: 0.00000000

Highlights: 0.75075988

Midtones: 0.18218623

w:1175 · h:755 · n:1 · i16 · Grayscale(local) · 1.692 MB

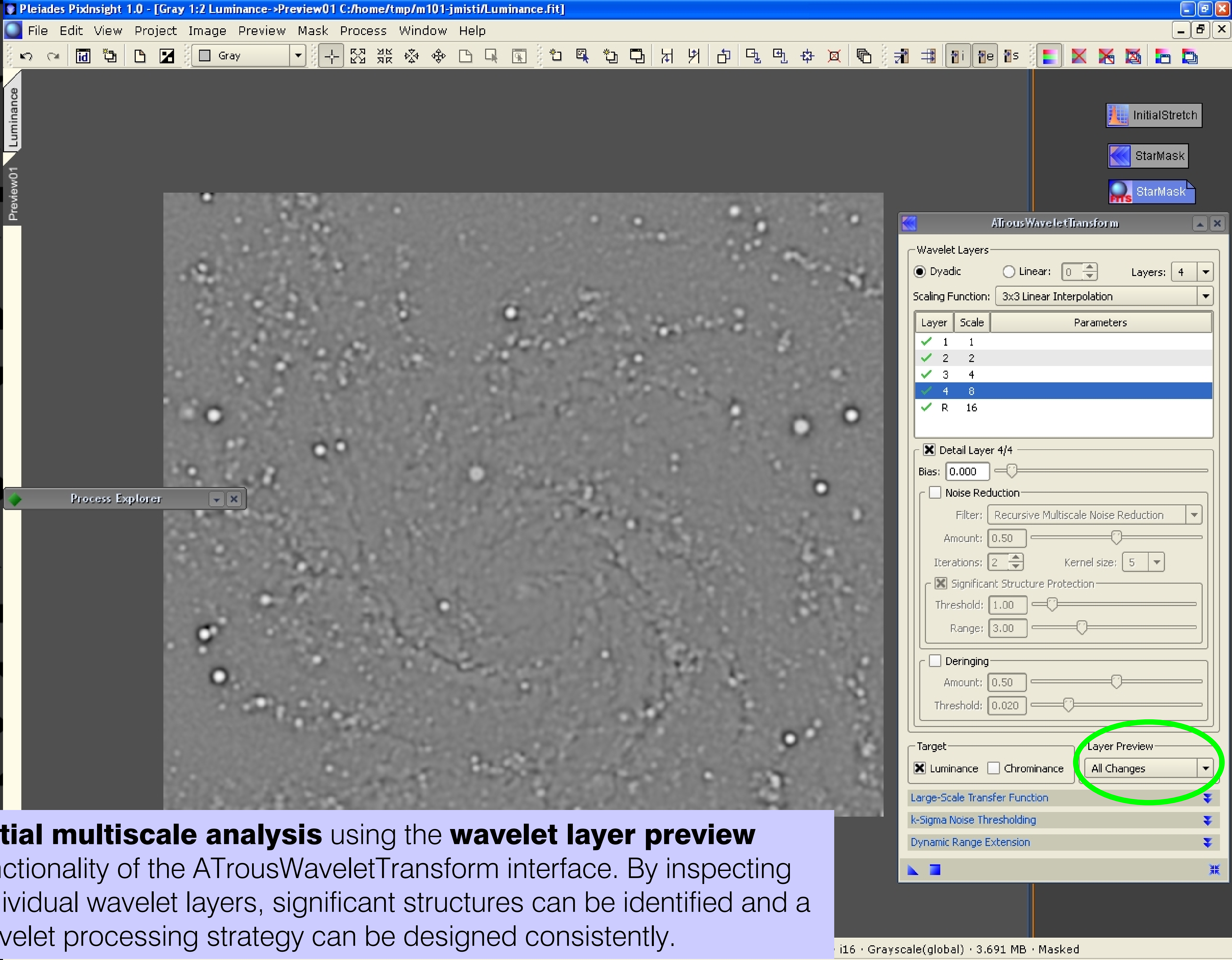
Expanding the star protection mask with a histogram stretch. This provides better protection of small structures and facilitates a subsequent wavelet transform.



```
HistogramTransform: Processing preview: StarMask->Preview01
```

```
w:1500 · h:1290 · n:1 · i16 · Grayscale(global) · 3.691 MB · Masked
```

Testing mask protection by enabling mask visibility, once the mask has been activated for the stretched luminance image.



Pleiades PixInsight 1.0 - [Gray 1:2 Luminance->Preview01 C:/home/tmp/m101-jmisti/Luminance.fit]

ATrousWaveletTransform

Wavelet Layers

● Dyadic ○ Linear: 0 Layers: 4

Scaling Function: 3x3 Linear Interpolation

	Layer	Scale	Parameters
✓	1	1	
✓	2	2	
✓	3	4	
✓	4	8	
✓	R	16	

☒ Detail Layer 4/4

Bias: 0.000

□ Noise Reduction

Filter: Recursive Multiscale Noise Reduction

Amount: 0.50

Iterations: 2 Kernel size: 5

☒ Significant Structure Protection

Threshold: 1.00

Range: 3.00

□ Deringing

Amount: 0.50

Threshold: 0.020

Target

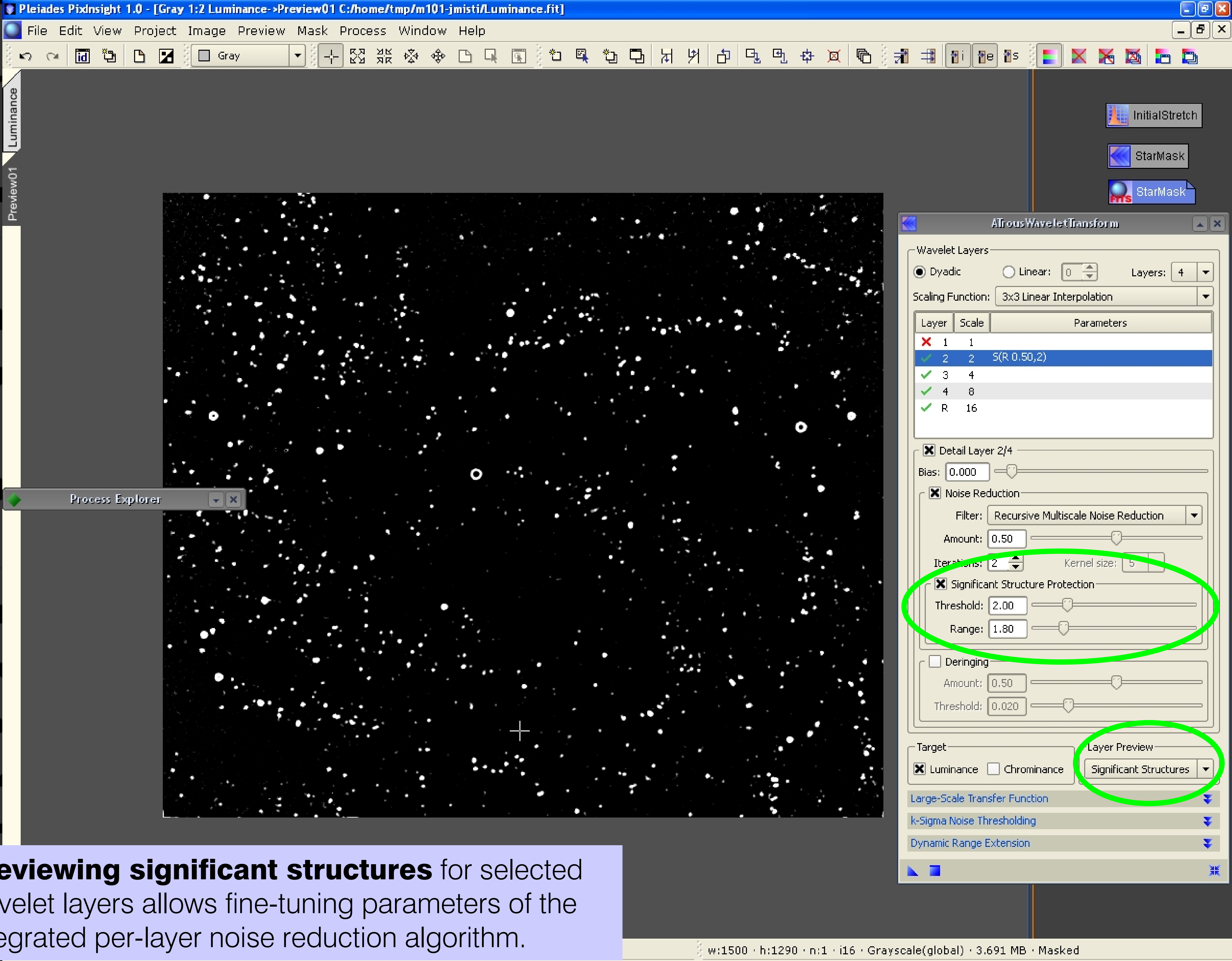
☒ Luminance □ Chrominance

Layer Preview

All Changes

i16 · Grayscale(global) · 3.691 MB · Masked

Initial multiscale analysis using the **wavelet layer preview** functionality of the ATrousWaveletTransform interface. By inspecting individual wavelet layers, significant structures can be identified and a wavelet processing strategy can be designed consistently.



Pleiades PixInsight 1.0 - [Gray 1:2 Luminance->Preview01 C:/home/tmp/m101-jmisti/Luminance.fit]

File Edit View Project Image Preview Mask Process Window Help

ATrousWaveletTransform

Wavelet Layers

● Dyadic ○ Linear: 0 Layers: 4

Scaling Function: 3x3 Linear Interpolation

Layer	Scale	Parameters
✗ 1	1	
✓ 2	2	S(R 0.50,2)
✓ 3	4	
✓ 4	8	
✓ R	16	

☒ Detail Layer 2/4

Bias: 0.000

☒ Noise Reduction

Filter: Recursive Multiscale Noise Reduction

Amount: 0.50

Iterations: 2 Kernel size: 5

☒ Significant Structure Protection

Threshold: 2.00

Range: 1.80

□ Deringing

Amount: 0.50

Threshold: 0.020

Target

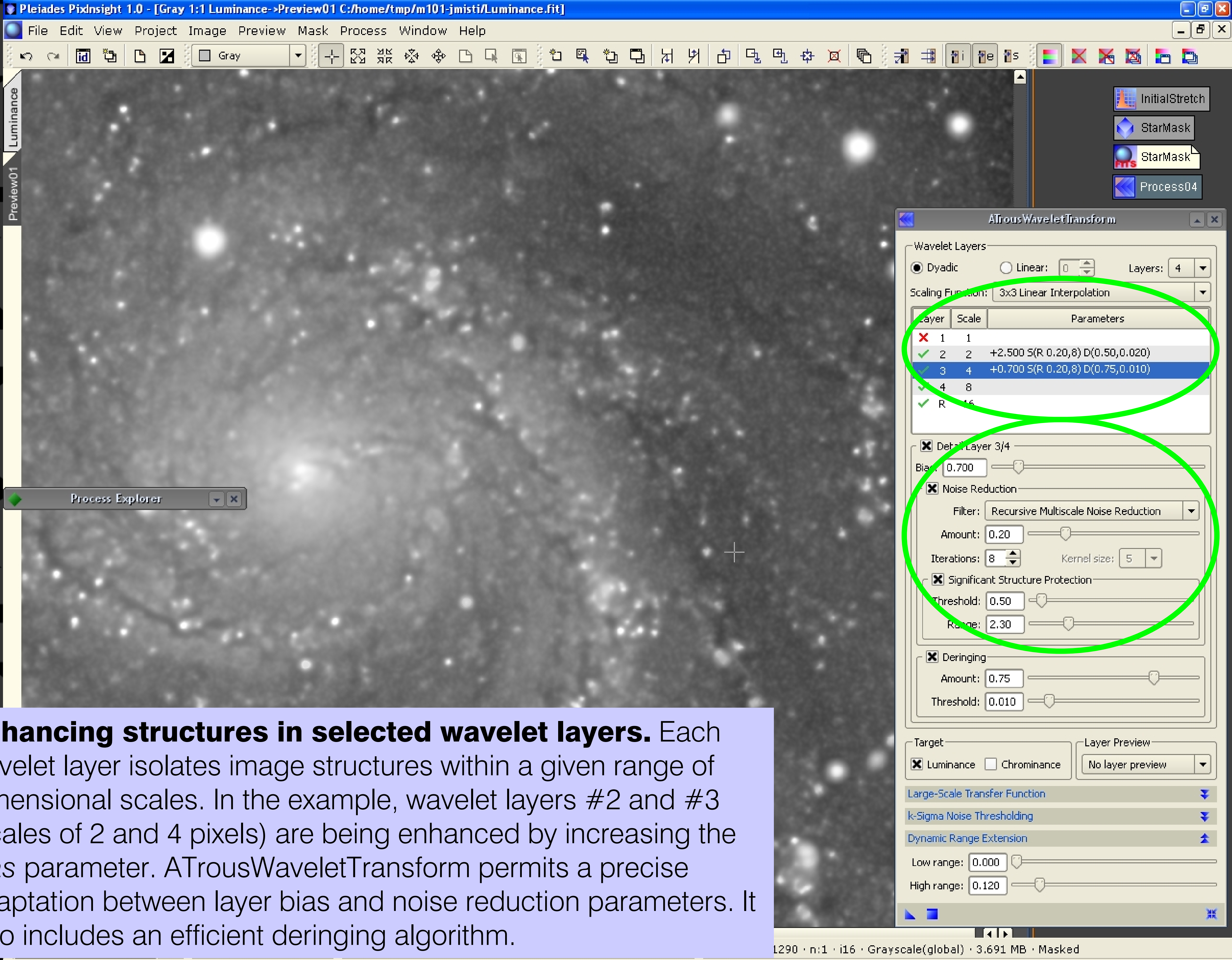
☒ Luminance □ Chrominance

Layer Preview

Significant Structures

w:1500 · h:1290 · n:1 · i16 · Grayscale(global) · 3.691 MB · Masked

Previewing significant structures for selected wavelet layers allows fine-tuning parameters of the integrated per-layer noise reduction algorithm.



Pleiades PixInsight 1.0 - [Gray 1:1 Luminance->Preview01 C:/home/tmp/m101-jmisti/Luminance.fit]

ATrousWaveletTransform

Wavelet Layers

● Dyadic ○ Linear: 0 Layers: 4

Scaling Function: 3x3 Linear Interpolation

	Layer	Scale	Parameters
✗	1	1	
✓	2	2	+2.500 S(R 0.20,8) D(0.50,0.020)
✓	3	4	+0.700 S(R 0.20,8) D(0.75,0.010)
✓	4	8	
✓	R	16	

☒ Detail Layer 3/4

Bias: 0.700

☒ Noise Reduction

Filter: Recursive Multiscale Noise Reduction

Amount: 0.20

Iterations: 8 Kernel size: 5

☒ Significant Structure Protection

Threshold: 0.50

Range: 2.30

☒ Deringing

Amount: 0.75

Threshold: 0.010

Target: ☒ Luminance □ Chrominance

Layer Preview: No layer preview

Large-Scale Transfer Function

k-Sigma Noise Thresholding

Dynamic Range Extension

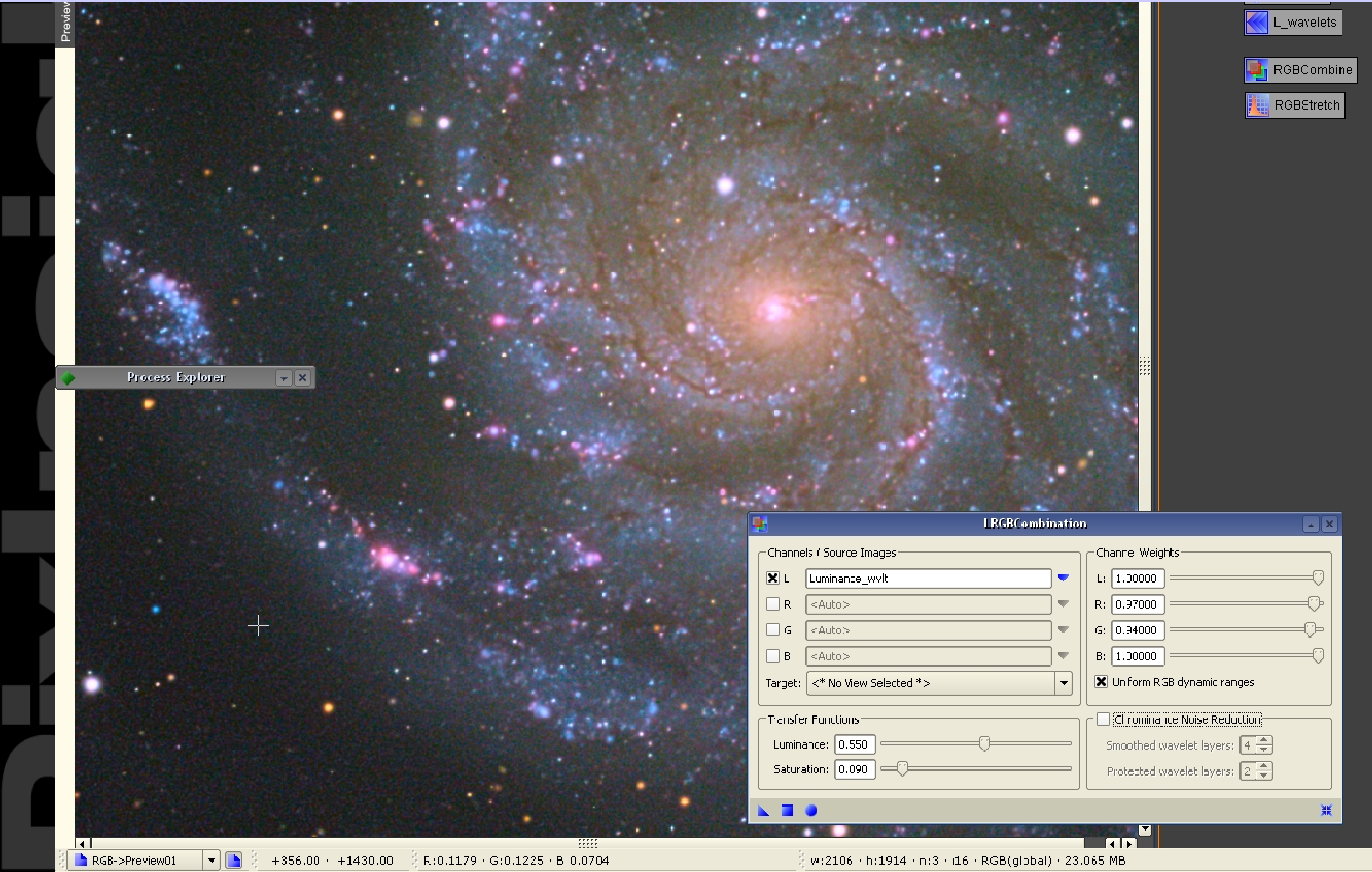
Low range: 0.000

High range: 0.120

1290 · n:1 · i16 · Grayscale(global) · 3.691 MB · Masked

Enhancing structures in selected wavelet layers. Each wavelet layer isolates image structures within a given range of dimensional scales. In the example, wavelet layers #2 and #3 (scales of 2 and 4 pixels) are being enhanced by increasing the *Bias* parameter. ATrousWaveletTransform permits a precise adaptation between layer bias and noise reduction parameters. It also includes an efficient deringing algorithm.

Combining an LRGB image with the LRGBCombination process. Different combine ratios (weights) can be specified for individual RGB channels. The luminance transfer function allows a precise adaptation between luminance and chrominance data. The saturation transfer function permits dramatic color saturation enhancements without affecting color balance (zero hue shifts).



LRGBCombination

Channels / Source Images

☒ L Luminance_wvlt
☐ R <Auto>
☐ G <Auto>
☐ B <Auto>
Target: <* No View Selected *>

Channel Weights

L: 1.00000
R: 0.97000
G: 0.94000
B: 1.00000
☒ Uniform RGB dynamic ranges

Transfer Functions

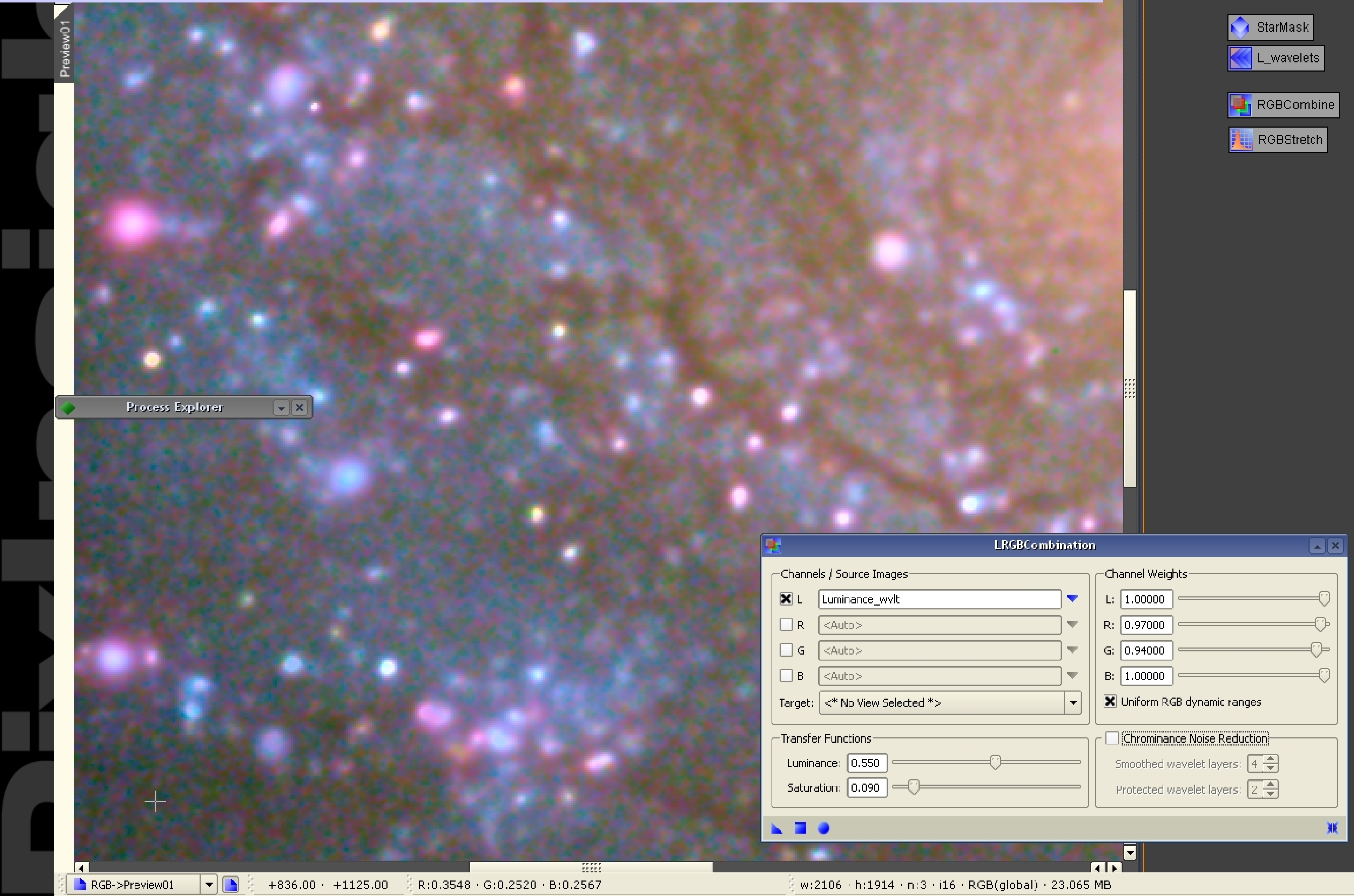
Luminance: 0.550
Saturation: 0.090

☐ Chrominance Noise Reduction

Smoothed wavelet layers: 4
Protected wavelet layers: 2

RGB->Preview01 · +356.00 · +1430.00 · R:0.1179 · G:0.1225 · B:0.0704 · w:2106 · h:1914 · n:3 · i16 · RGB(global) · 23.065 MB

Defining a strong color saturation increment with the LRGBCombination process. In this figure, **chrominance noise** is clearly visible. This problem has been perfectly fixed in our implementation, as the next slide demonstrates.



LRGBCombination

Channels / Source Images

☒ L Luminance_wvlt
□ R <Auto>
□ G <Auto>
□ B <Auto>
Target: <* No View Selected *>

Channel Weights

L: 1.00000
R: 0.97000
G: 0.94000
B: 1.00000
☒ Uniform RGB dynamic ranges

Transfer Functions

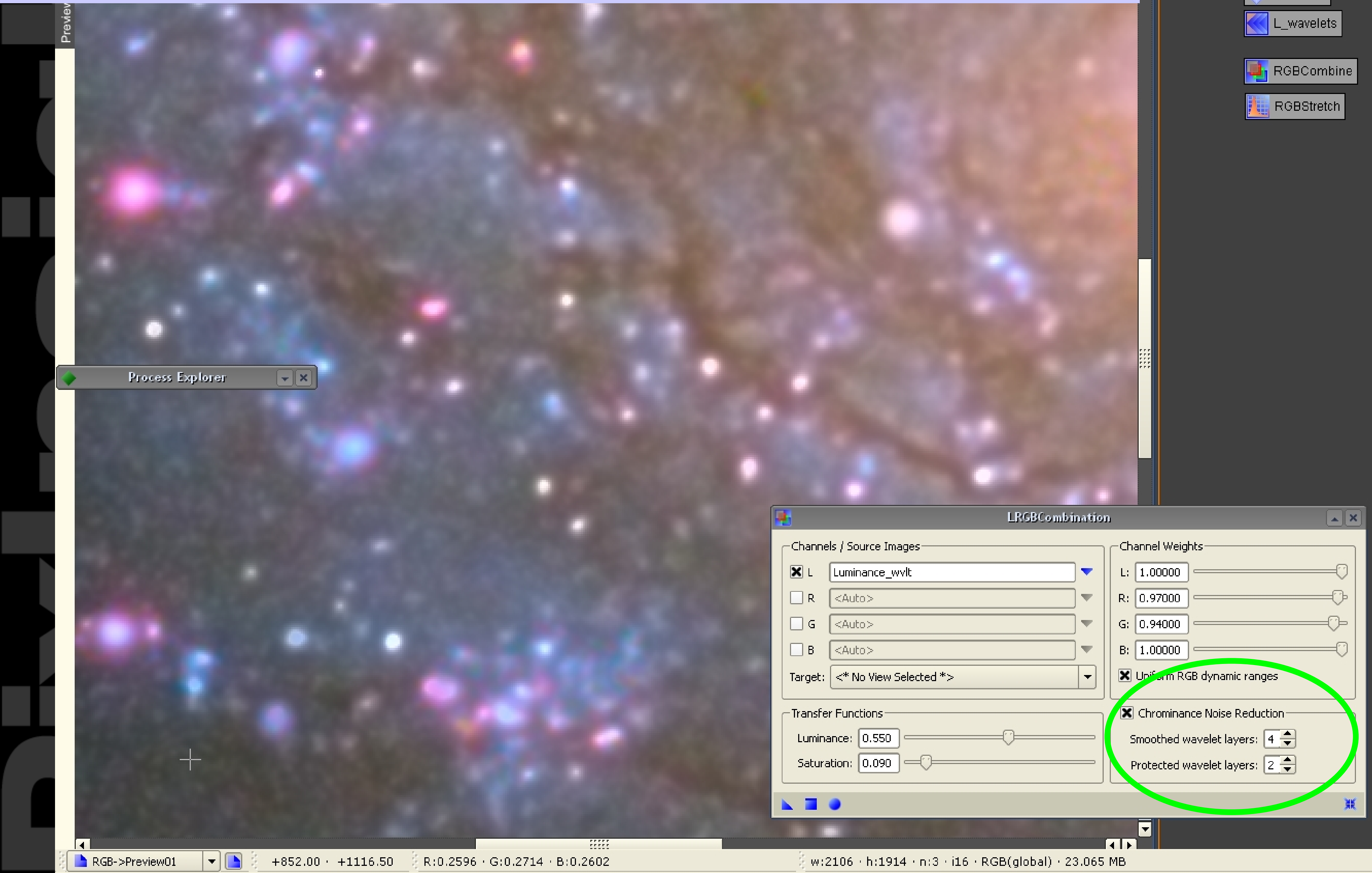
Luminance: 0.550
Saturation: 0.090

□ Chrominance Noise Reduction

Smoothed wavelet layers: 4
Protected wavelet layers: 2

RGB->Preview01 · +836.00 · +1125.00 · R:0.3548 · G:0.2520 · B:0.2567 · w:2106 · h:1914 · n:3 · i16 · RGB(global) · 23.065 MB

Applying chrominance noise reduction with the LRGBCombination process. Our implementation includes a chrominance-specific, multiscale (wavelets-based) noise reduction algorithm. This algorithm works in tandem with the saturation transfer function. Compare with the previous slide.



LRGBCombination

Channels / Source Images

☒ L Luminance_wvlt

☐ R <Auto>

☐ G <Auto>

☐ B <Auto>

Target: <* No View Selected *>

Channel Weights

L: 1.00000

R: 0.97000

G: 0.94000

B: 1.00000

☒ Uniform RGB dynamic ranges

Transfer Functions

Luminance: 0.550

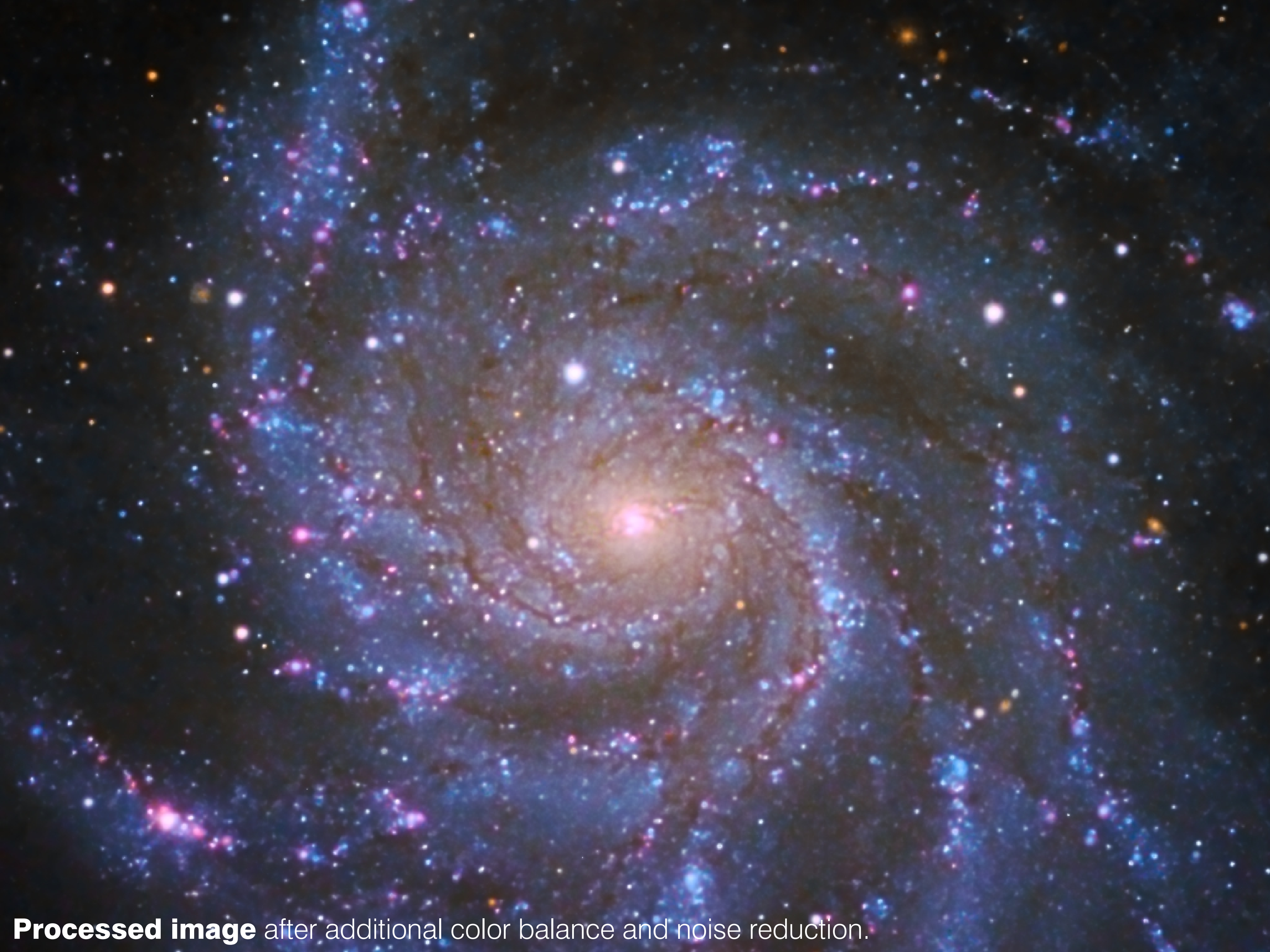
Saturation: 0.090

☒ Chrominance Noise Reduction

Smoothed wavelet layers: 4

Protected wavelet layers: 2

RGB->Preview01 +852.00 · +1116.50 R:0.2596 · G:0.2714 · B:0.2602 w:2106 · h:1914 · n:3 · i16 · RGB(global) · 23.065 MB



Processed image after additional color balance and noise reduction.

PixInsight

Image Authors
Luc Coiffier / Thomas W. Earle / Christopher Go /
José Luis Lamadrid / Carlos Milovic / Jim Misti / Vicent Peris