

Digital Imaging with Free Software

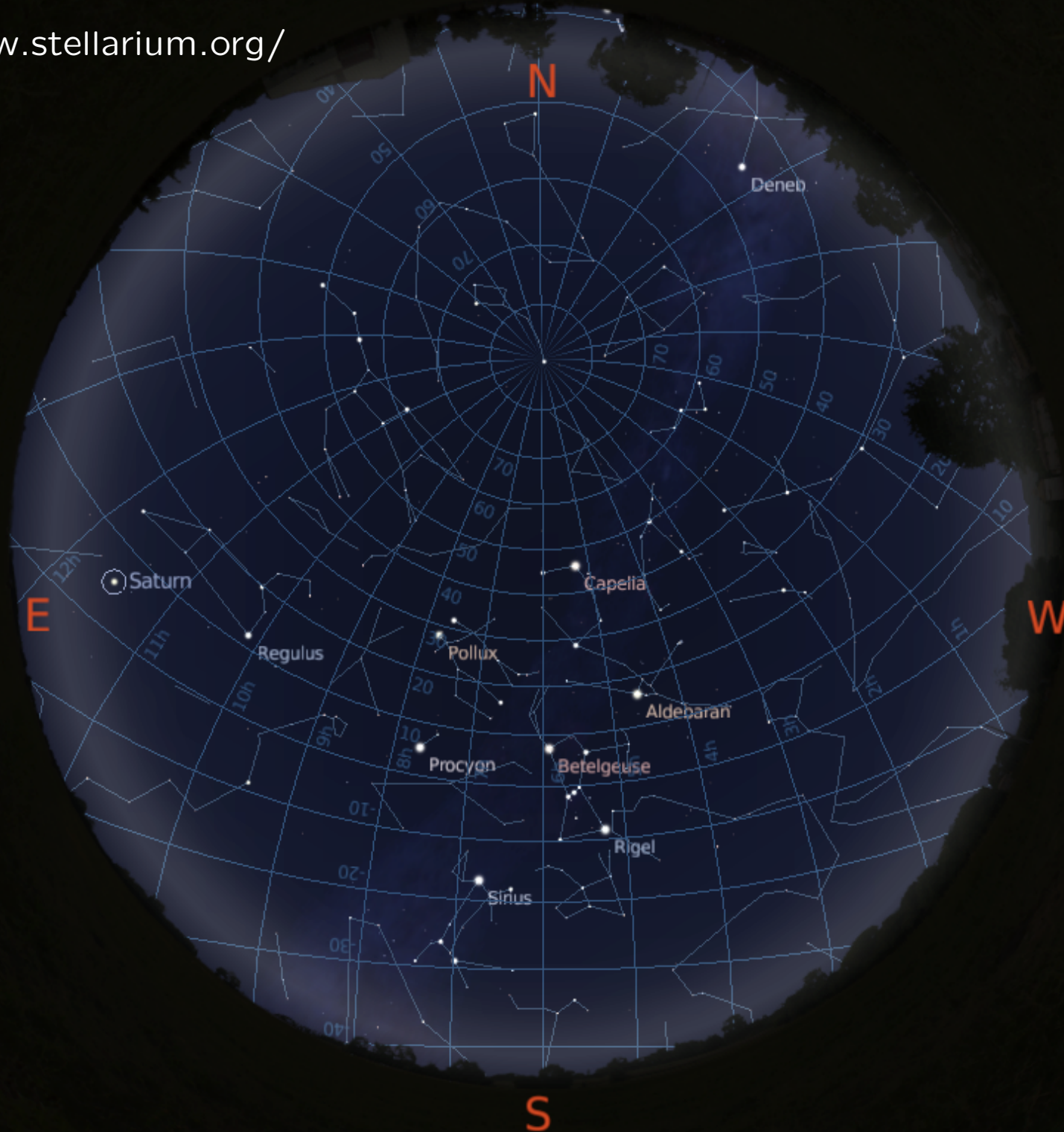
Sheffield Astronomical Society
Beginner's Meeting

Digital Imaging with Free Software

Jan Wedekind

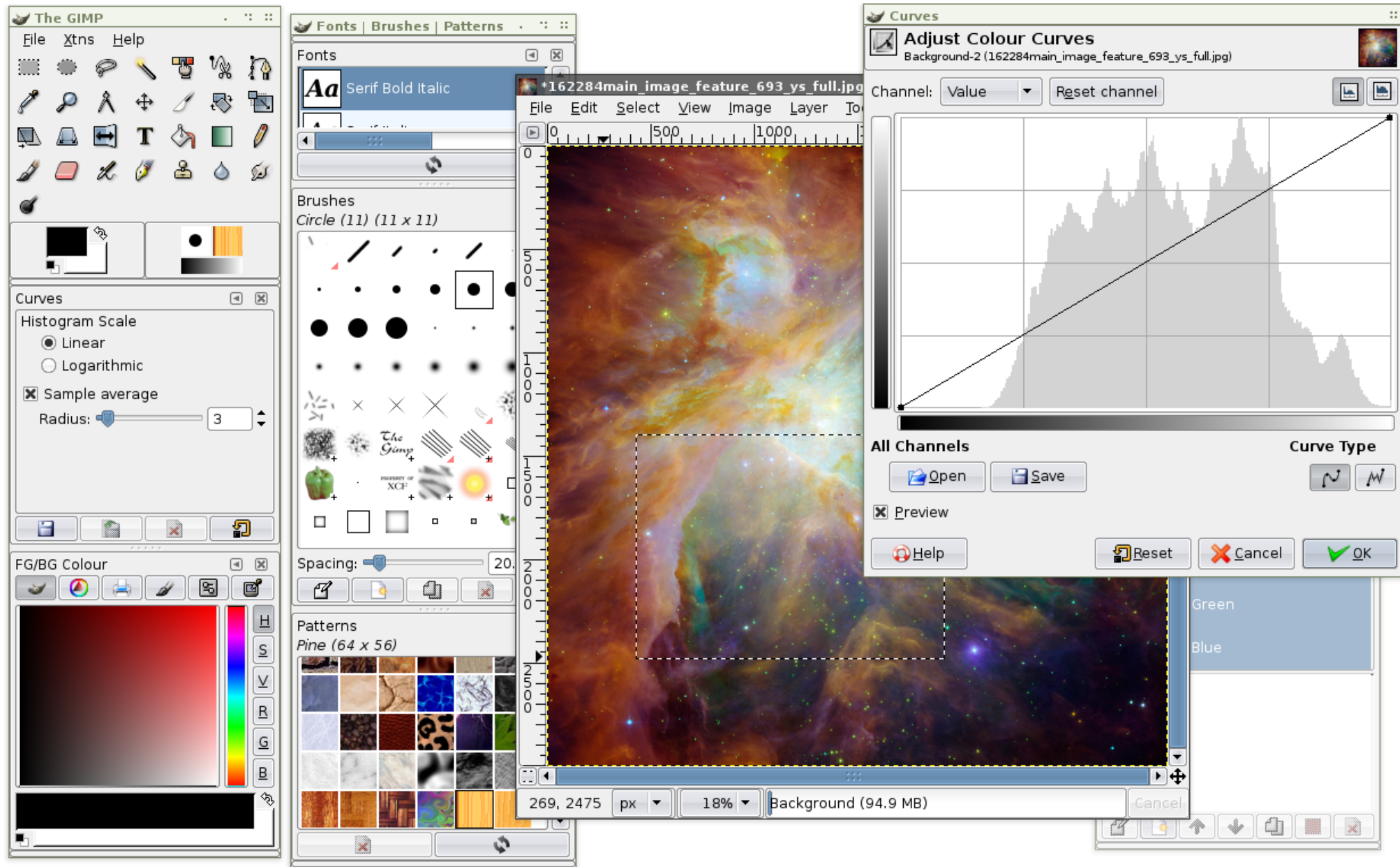
Monday, January 19th 2009

<http://www.stellarium.org/>



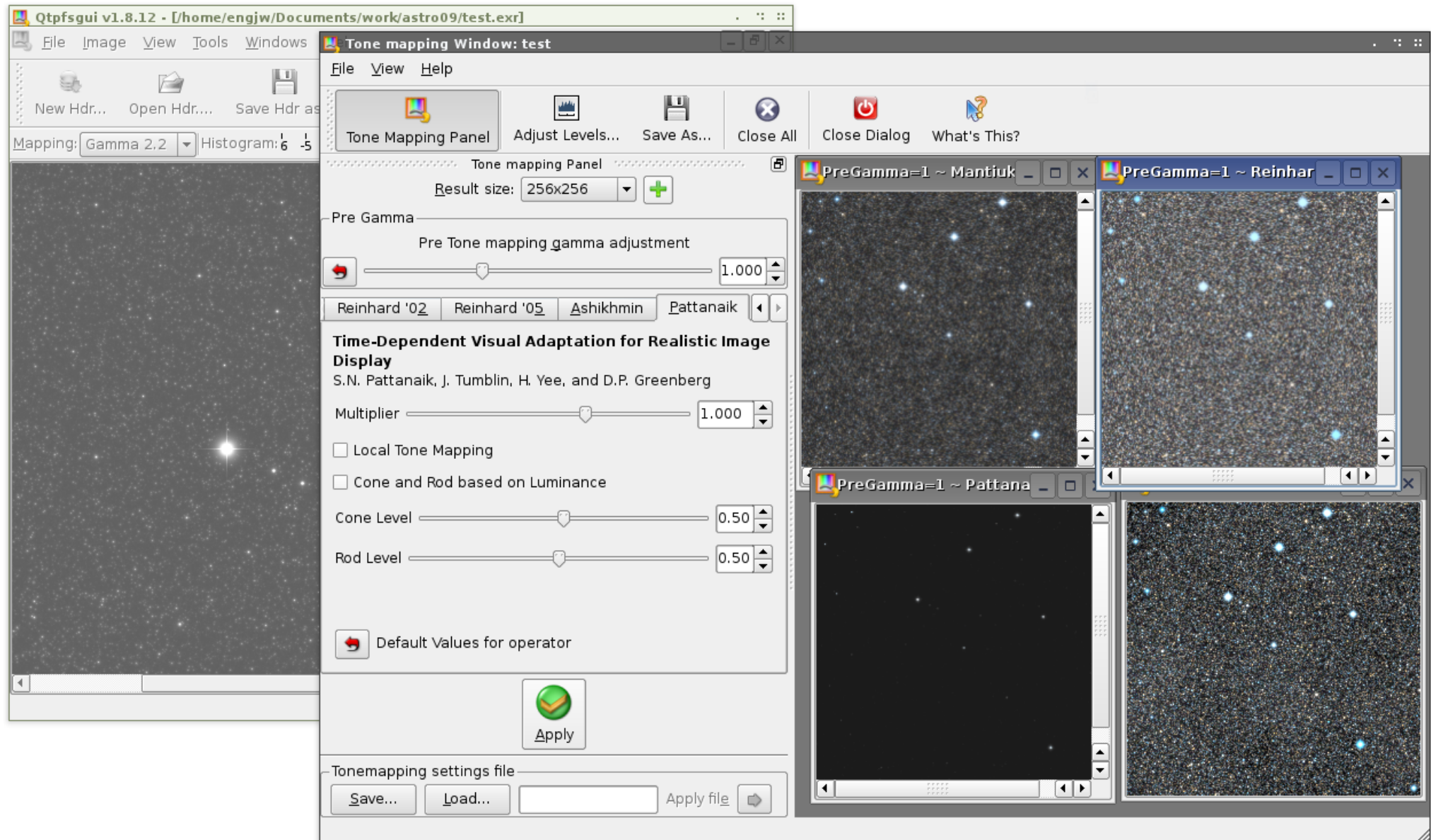
The GIMP (The GNU Image Manipulation Program)

<http://www.gimp.org/>



QtPfsGui (High Dynamic Range Imaging)

<http://qtpfsgui.sourceforge.net/>



Kubuntu 8.04 (Distribution of GNU/Linux)

<http://www.kubuntu.org/>



Try Kubuntu without any change to your computer

Install Kubuntu

Check CD for defects

Test memory

Boot from first hard disk

Press F4 to select alternative start-up and installation modes.

F1 Help F2 Language F3 Keymap F4 Modes F5 Accessibility F6 Other Options

Free Software

Free Software Definition

freedom 0: run the program, for any purpose

freedom 1: study how the program works, and adapt it to your needs

freedom 2: redistribute copies so you can help your neighbor

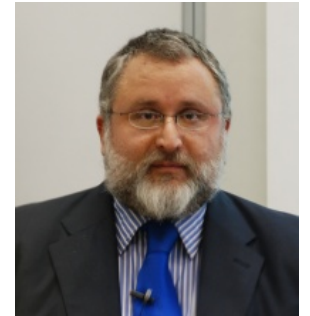
freedom 3: improve the program, and release your improvements to the public, so that the whole community benefits



Richard Stallman - Free Software Foundation

“Imagine if you will for a moment a society in which mathematics has become property, and it’s owned by people. Now every time you want to do anything useful ... your first stop is at the mathematics store to buy enough mathematics to complete the task which lies before you. ... You can predict that every other activity in society, whether undertaken for economic benefit or for the common good, will pay taxes in the form of mathematics payments.”

Eben Moglen - Keynote at Plone Conference 2006



HornetsEye

<http://rubyforge.org/projects/hornetseye/>



Log Out Support My Account

This project's trackers Search Advanced search

Home My Page Project Tree Code Snippets Project Openings **HornetsEye - Ruby Computer Vision Lib**

Summary Admin Forums Tracker Surveys News Files

Video processing and computer vision library for GNU/Linux offering interfaces to do image- and video-I/O with ImageMagick/Magick++, Xine, firewire digital camera (DC1394), and video for linux (V4L). Port to MS Windows is underway.

- Development Status: [3 - Alpha](#), [4 - Beta](#), [5 - Production/Stable](#)
- Environment: [Console \(Text Based\)](#), [X11 Applications](#)
- Intended Audience: [Developers](#)
- License: [GNU General Public License \(GPL\) version 3](#)
- Natural Language: [English](#)
- Operating System: [Windows](#), [Linux](#)
- Programming Language: [C++](#), [Ruby](#)
- Topic: [Capture](#), [Artificial Intelligence](#), [Mathematics](#)



Registered: 2006-12-04 15:10
Activity Percentile: 59.47%
View project activity [statistics](#).

Developer Info

Project Admins:
[Jan Wedekind](#)

Developers:
1 [\[View Members\]](#)

Latest File Releases

Package	Version	Date	Notes / Monitor	Download
hornetseye	0.31	January 11, 2009	 - 	Download

[\[View All Project Files\]](#)

Public Areas

 [Project Home Page](#)

Latest News

[HornetsEye-0.31 released](#)

Using Existing Free Software

Libraries Integrated



C++ Boost



libdc1394



DotGNU



FFTW



MPlayer



OpenEXR



Qt4-QtRuby



RMagick



NArray



libxine



Xorg, Mesa3D

Tools in Use



Bazaar



GCC



make, automake, autoconf



NaturalDocs



NSIS



Ruby

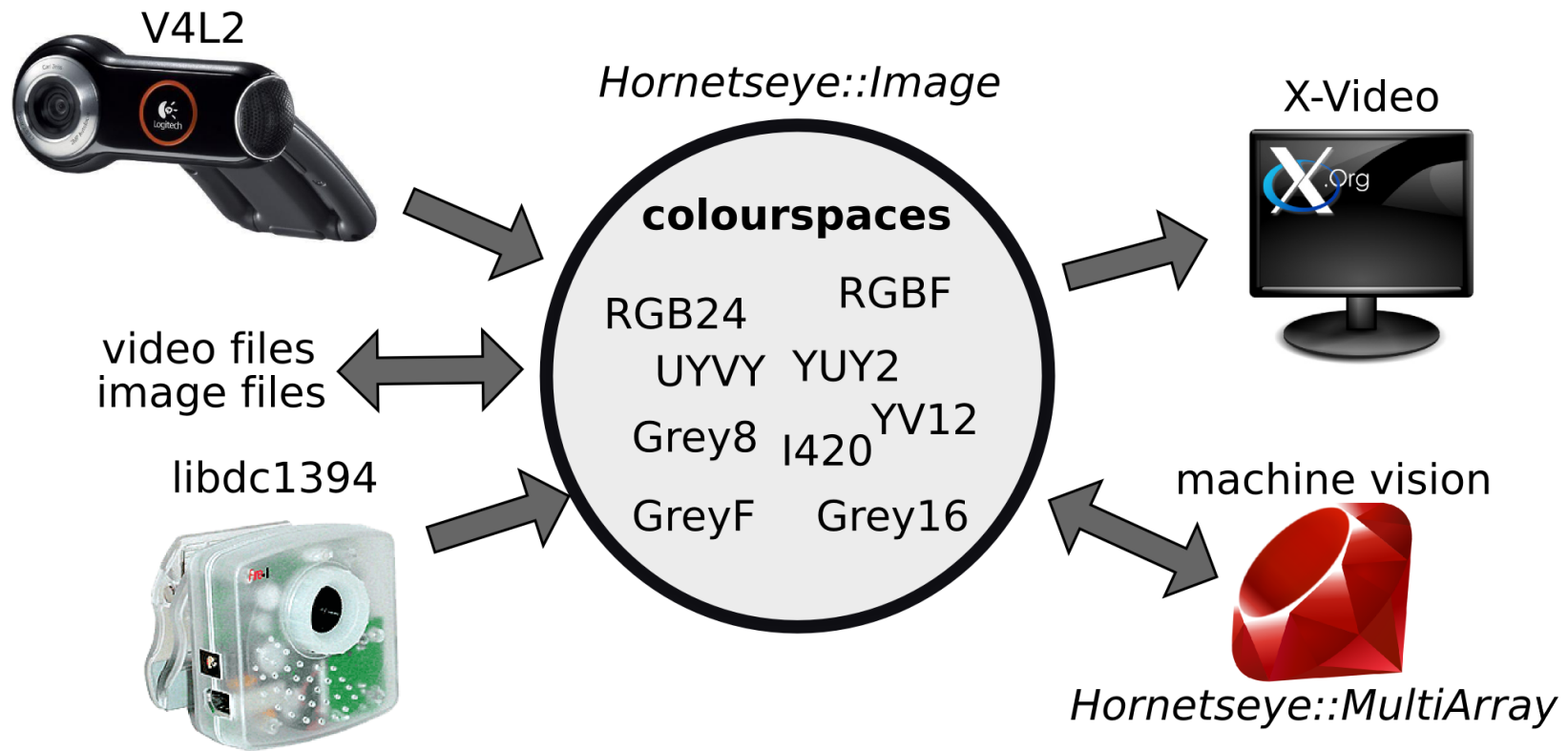
GNU/Linux ⇔ Microsoft Windows

Cross-Platform Support



V4LInput	VFWInput
V4L2Input	DShowInput
DC1394Input	—
XineInput	—
MPlayerInput	MPlayerInput
MEncoderOutput	MEncoderOutput
X11Display	W32Display
X11Window	W32Window
XImageOutput	GDIOutput
OpenGLOutput	—
XVideoOutput	—

Input/Output and Colourspace Conversions



$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.500 \\ 0.500 & -0.418688 & -0.081312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

also see: <http://fourcc.org/>

Opening Webcam/Framegrabber

```
# Open V4L2-compatible Video Device  
require 'hornetseye'  
include Hornetseye  
input = V4L2Input.new  
img = input.read  
img.display
```

Capture Image

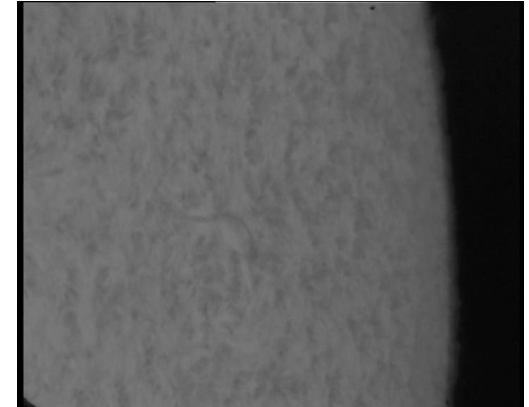
```
# Capture Image
require 'hornetseye'
include Hornetseye
input = V4L2Input.new
img = nil
X11Display.loop { img = input.read }
img.save_rgb24 'test.png'
```

Capture Video

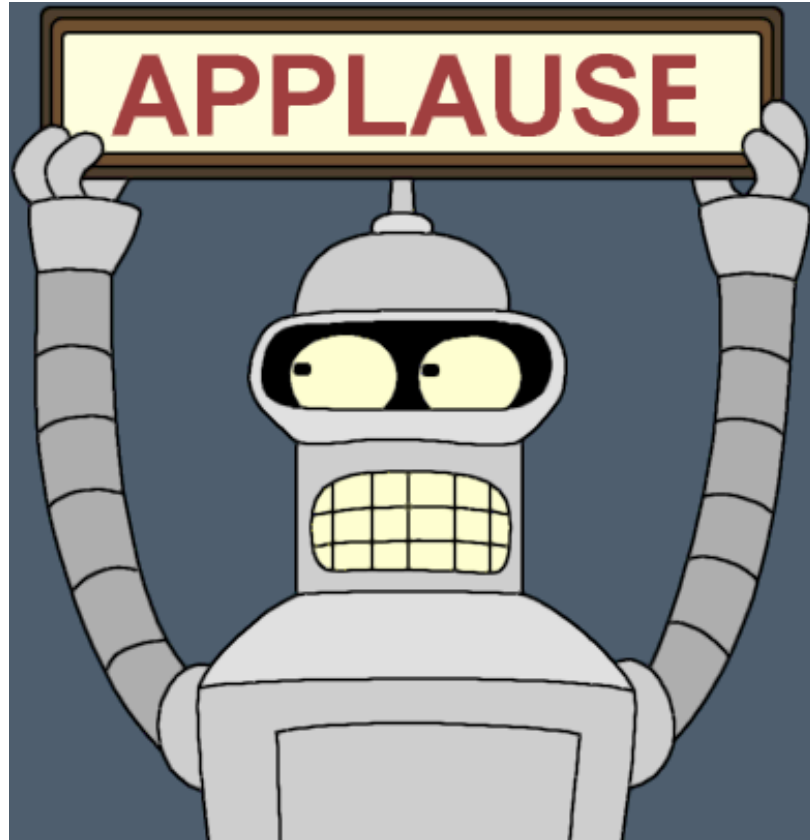
```
# Capture Video
require 'hornetseye'
include Hornetseye
output = MEncoderOutput.new( 'test.avi', 10,
    '-ovc lavc -lavcopts vcodec=msmpeg4:vhq:vbitrate=4000' )
X11Display.loop do
    img = input.read
    output.write( img )
    img
end
output.close
```


Averaging Frames

```
# Averaging Frames
require 'hornetseye'
include Hornetseye
input = XineInput.new 'solar.avi'
arr = []
X11Display.loop do
  img = input.read_grey8
  arr.push( img )
  arr = arr[ 1..-1 ] if arr.size > 25
  result = arr.inject { |a,b| a.to_lint + b }
  result.normalise
end
```



Questions?



<http://vision.eng.shu.ac.uk/jan/astro09.pdf>

<http://tinyurl.com/9876fr>