iLab C++ Neuromorphic Vision Toolkit Overview

- Components:
  - Basic image processing and vision
  - Attention-related neural components
  - Object recognition-related neural components
  - Scene gist/layout-related neural components
  - Basic knowledge base / ontology
  - Hardware interfacing
  - Beowulf message passing
  - Applications

- Implementation:
  - C++, somewhat Linux-specific
  - Additional perl/matlab/shell scripts for batch processing
  - Uniprocessor as well as Beowulf

The basic architecture

- The diagram on the next slide is an overview of this computational neuroscience model.
- Suggested readings: see http://iLab.usc.edu/publications/

  - Start with Itti & Koch, Nature Reviews Neuroscience, 2001, for an overview.
  - Then see Itti, Koch and Niebur, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, for the core algorithm.
  - Then see Itti & Koch, Vision Research, 2000 and Itti & Koch, Journal of Electronic Imaging, 2001, for more advanced competition for salience.

  - See papers by Christian Siagian, Christopher Ackerman & Nitin Dhavale for more on robotics applications, gist, and localization.
  - See papers by Vidhya Navalpakkam, Rob Peters and Lior Elazary for more on scene understanding & top-down biasing.
  - See papers by Nathan Mundhenk for more on contour integration, surprise
  - Etc...

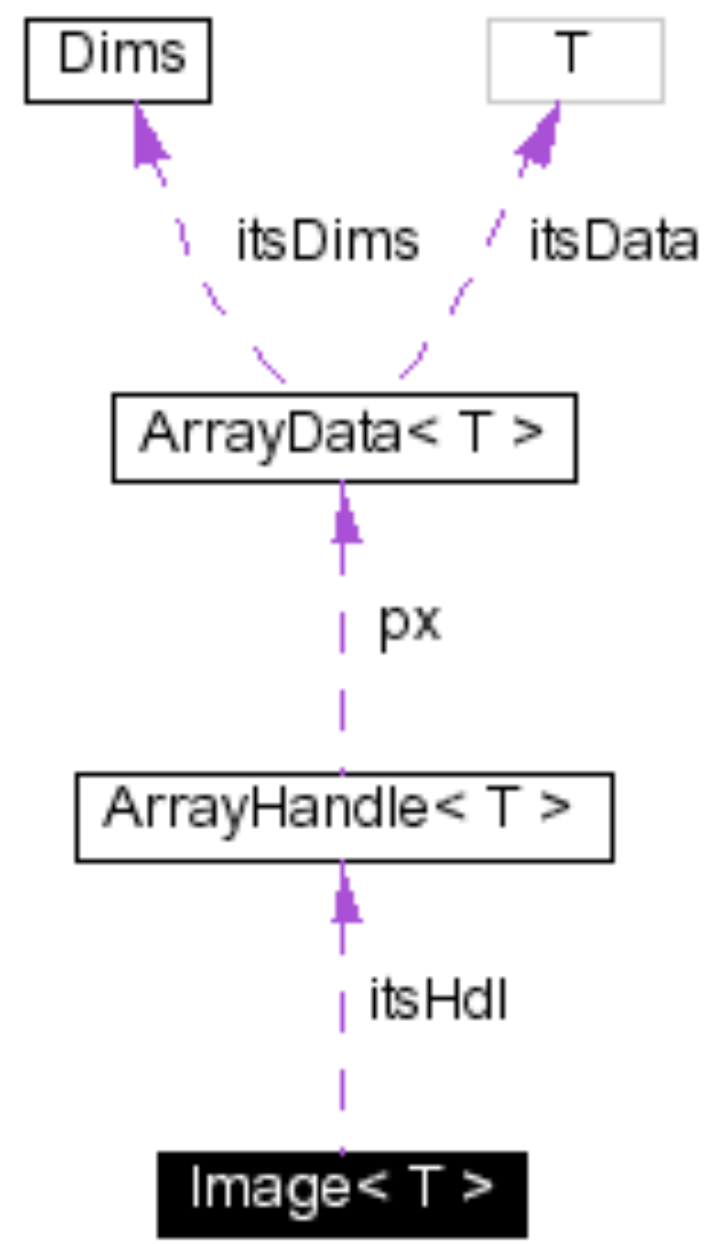
iLab C++ Neuromorphic Toolkit

Root: Image class

- Template class
  - e.g., Image<byte>, Image<PixRGB<float>>, Image<Neuron>

- Implemented using copy-on-write/ref-counting
  - Makes copying a light operation

- Many associated methods
  - Shape ops
  - Color ops
  - Mono only
  - Math ops
  - Matrix ops
  - I/O
  - Filter ops
  - Transforms



C++ Templates

- **The old way:** ByteImage, FloatImage, ColorImage, etc. yields lots of duplicated code that achieves essentially the same operations.

- **The C++ way:** write your algorithm only once, and make it operate on an unknown data type T. The compiler will then generate machine code corresponding to your algorithm and various data types for T, such as, T=byte, T=float, T=MyClass, etc

```
template <class T> class Image {
public:
    Image();
    T getPixelValue(const int x, const int y) const;
    void setPixelValue(const T& value, const int x, const int y);
private:
    T* data;
};

int main(const int argc, const char **argv) {
    Image<float> myImage; myImage.setPixelValue(1.23F, 10, 10);
    return 0;
}
```

See src/Image/Image.H

Operator overloads

- C++ allows you to define operators such as +, -, *, etc for your various classes.

- Example:

See Image/Pixels.H, Image.H

```
Image<byte> img1, img2;

img1 += 3;    // calls Image<T>::operator+=(const T& value)

img1 = img1*2 + img2/3;        // calls operator*(const T& value),
                               // operator/(const T& value),
                               // and operator+(const Image<T>& im)
```

Automatic type promotions

- Using type traits to determine at compile time whether the result of an arithmetic operation will fit in the same type as the operands.
- Extends the canonical C++ promotions to non-canonical types.
- Examples:

See Util/Promotions.H, Image/Pixels.H, Image/Image.H

```
Image<byte> im;

im + im        is an Image<int>
im * 2.0F      is an Image<float>
im * 2.0       is an Image<double>
```

Automatic type demotion with clamping

- Assignment from a strong type into a weak type will ensure that no overflow occurs.

- Example:

```
Image<byte> im1, im2;   Image<float> im3;

im1 = im3;        // will clamp values of im3 to 0..255 range and convert

im2 = im1 * 2.0;  // will create an Image<double> containing the
                  // result of im1 * 2.0, then clamp this image to
                  // 0..255 pixel range, then assign to im2.
```

Automatic type demotion with clamping

- Promotion rules (in Util/Promotions.H):

  Basically follow the C/C++ canonical promotions

  - byte, byte -> int;   Byte, int16 -> int; int16, int16 -> int; etc…
  - int, int -> int
  - byte, float -> float; int, float -> float; float, float -> float, etc…
  - byte, double -> double; int, double -> double; float, double -> double, etc…

Copy-on-write / ref counting

- The standard way:

Image object contains an array of pixels:

Image<T> object

Problem: copy is expensive, need to copy the whole data array (can be large, e.g., a 16MP RGB image uses 48MB of memory).

Copy-on-write / ref counting

In particular, this makes it very expensive to return Image objects from functions, hence essentially forbidding the natural syntax:

```
Image<float> source;
Image<float> result = filter(source);
```

With a function:

```
Image<float> filter(const Image<float>& source) {
    Image<float> res;
    // fill-up pixel values of res, processing values from source
    return res;
}
```

Indeed what happens here is:

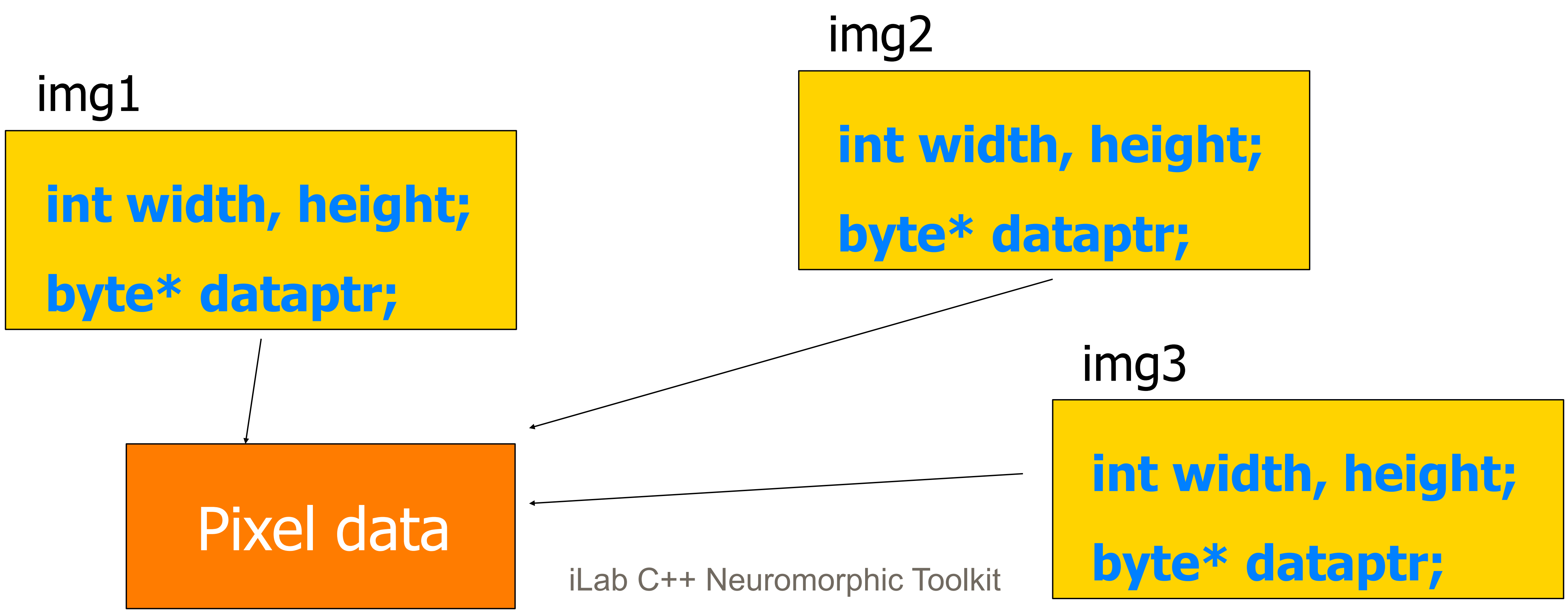
1) Inside filter(), allocate a new image res to hold the result
2) In the 'return' statement, copy that local image to some temporary
3) In the '=' statement, copy that temporary to Image 'result'

See Image/ArrayData.H, Image.H

Copy-on-write / ref counting

- The smart way: only keep a pointer to the actual pixel data in each Image object. When making copies of the Image object, keep track of how many are pointing to the same pixel data. When the last Image object is destroyed, free the pixel data. If the user attempts to modify the contents of one of the images that point to the same data, first make a copy of the data.

Image<byte> img1, img2, img3;   img2 = img1; img3 = img1;



Free functions rather than methods

- Given the copy-on-write mechanism, it is now very cheap to return Image objects. Thus, the more natural 'free function' syntax may be used for most image processing functions, instead of the 'class method' syntax.

- Example: let's say I want to pass an image through 3 successive filters, filter1(), filter2() and filter3():

Class method syntax: the filterX() are methods of class Image

```
const Image<float> source;
Image<float> result1, result2;
result1.filter1(source);
result2.filter2(result1);
result1.filter3(result2);
result2.freeMem();
```

See Image/*.H

Free function syntax: the filterX() are functions not attached to a class

```
const Image<float> source;
Image<float> result = filter3(filter2(filter1(source)));
```

Iterators

- Accessing data via pointers is error-prone, use iterators instead. Our classes that hold some data that can be iterated on provide iterator support very similar to that of the STL classes.

- Example:

See Image/Image.H

```
Image<byte> img;

Image<byte>::iterator itr = img.beginw(), stop = img.endw();
while (itr != stop) { *itr++ = 0; }
```

Shared pointers

- When objects communicate with lots of other objects, it is often difficult to know who will run out of scope first. When new memory is allocated for an object that will be passed around and used by several objects, we would like an automatic way of freeing the memory when everybody is done with it.
- Hence the class shared_ptr<T> which behaves like a pointer, except that when the last shared_ptr to an object runs out of scope, it will destroy/free the memory for that object.

- Example:

In obj1: SharedPtr<Message> mymsg(new Message());

In obj2: SharedPtr<Message> mymsg2(mymsg);

mymsg2->function();

See rutz/shared_ptr.h,
Also nub/ref.h

Message will be destroyed only when its SharedPtr's have run out of scope in both obj1 and obj2.

Elementary core classes

• Dims: for 2D (width, height) dimensions	Dims.H
• Point2D<T>: An (i, j) 2D point	Point2D.H
• PixRGB<T>: a (red, green, blue) triplet	Pixels.H
• Timer: to count time with arbitrary accuracy	Timer.H
• CpuTimer: to measure time and CPU load	CpuTimer.H
• Range: specifies a numeric range of values	Range.H
• LevelSpec: specifies scales for feature/saliency map	LevelSpec.H
• Rectangle: a rectangle	Rectangle.H
• Angle: an angle	Angle.H
• shared_ptr<T>: a shared pointer	shared_ptr.h
• VisualEvent	
• VisualObject	
• VisualFeature	
• ...	

Core definitions

- Promotions.H: the automatic type promotion rules
- atomic.H: atomic (one-CPU-instruction) operations
- Mathfunctions.H: basic math functions
- JobServer.H / WorkThreadServer.H: multithreading support
- Log.H: comprehensive logging facility
- StringConversions.H: convert various datatypes to/from string
- StringUtil.H: various string manipulation utilities (e.g., tokenize)
- sformat.H: like sprintf for std::string
- TypeTraits.H: compile-time information about types
- ...

Logs

- Provide a unified, convenient mechanism for text message output.
- 4 levels: LDEBUG, LINFO, LERROR, LFATAL
- printf()-like syntax
- Automatically adds class/function name, system error messages (use prefix ‘P’), a user id (use prefix ‘ID’), a line number (compile-time option)
- Can print to stderr or syslog

The hard way:

```
fprintf(stderr, “In myFunction(), could not open file ‘%s’ (error: %s)\n”, filename,
    strerror(errno));
```

>>>> In myFunction(), could not open file `test’ (error: file not found)

The easy way:

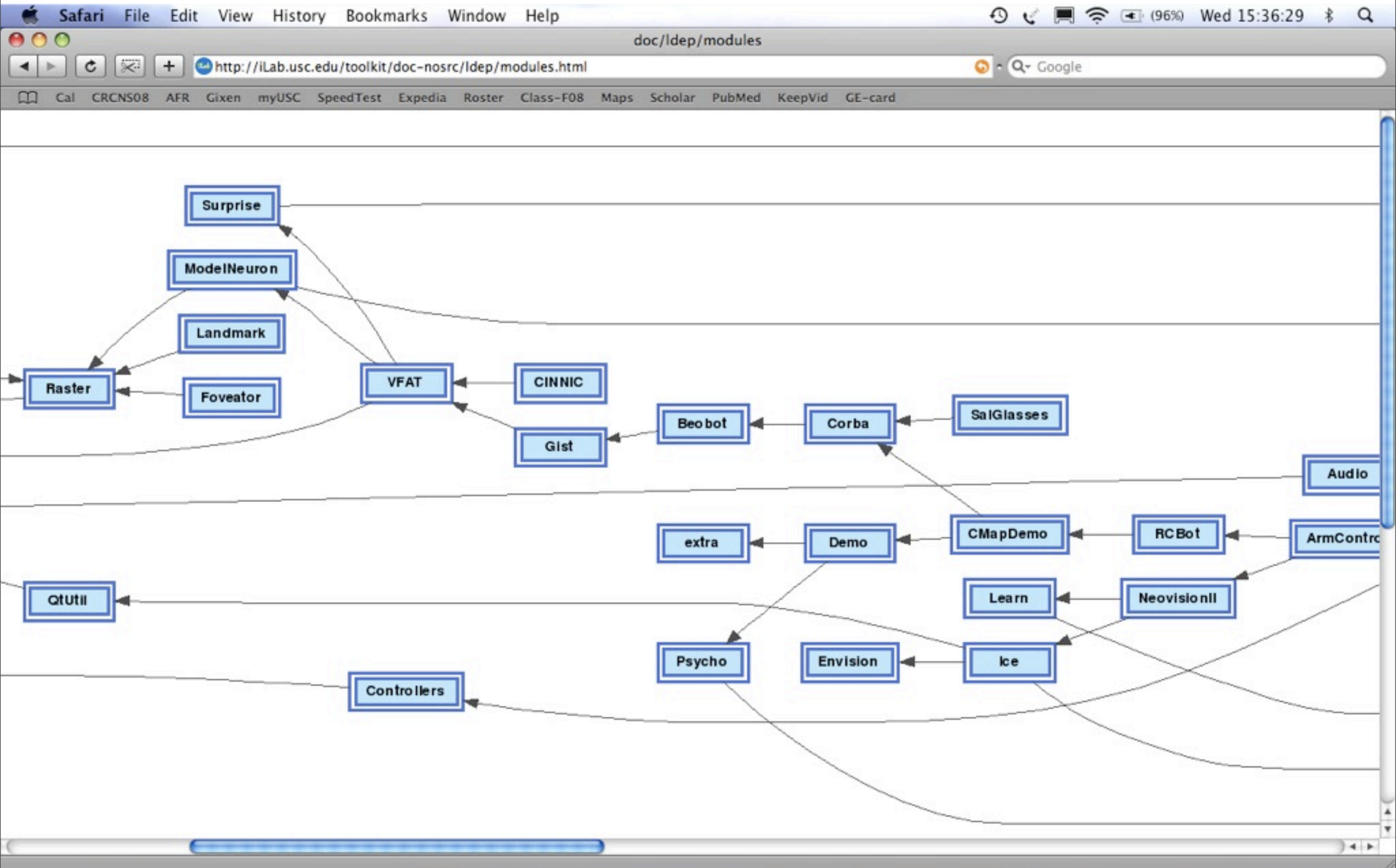
See Util/log.H

```
PLERROR(“Could not open file ‘%s’ ”, filename);
```

>>>> MyClass::myFunction: Could not open file ‘test’ (file not found)

Helper classes

- Raster: to read/write/display Images in various formats
- V4Lgrabber: to grab images from video source (PCI/USB)
- XWindow: to display image collections & interact
- FrameIstream, FrameOstream, FrameSeries: easily read images from image files, movies, cameras, etc
- Etc...



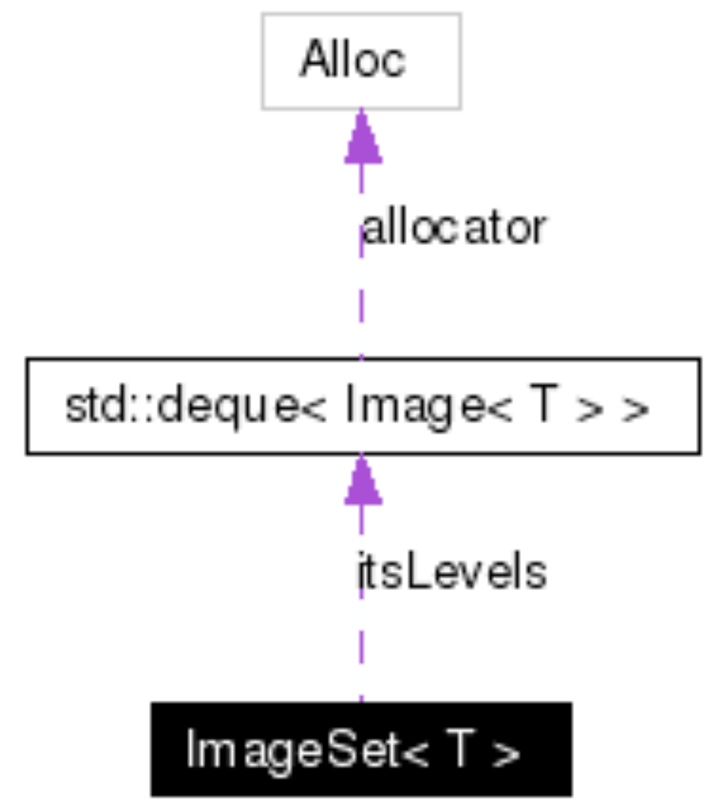
iLab C++ Neuromorphic Toolkit

src/ directories

AppDevices/	Channels/	GUI/	**nub**/	SceneUnderstanding/
AppEye/	CINNIC/	HMAX/	ObjRec/	Script/
AppGUI/	CMapDemo/	Ice/	Parallel/	SeaBee/
AppMedia/	CmuCam/	**Image**/	pbot/	SIFT/
AppNeuro/	Component/	inst@	plugins/	Simulation/
AppPsycho/	Controllers/	INVT/	PointCloud/	Surprise/
Apps/	Corba/	Landmark/	Psycho/	tcl/
ArmControl/	Demo/	Learn/	Qt/	TestSuite/
Audio/	Devices/	Matlab/	QtUtil/	TIGS/
Beobot/	Envision/	MBARI/	Raster/	Transport/
BeoSub/	extra/	Media/	RCBot/	**Util**/
Beowulf/	Foveator/	ModelNeuron/	Robots/	VFAT/
BO/	GA/	NeovisionII/	**rutz**/	Vgames/
BPnnet/	GameBoard/	Nerdcam/	SalGlasses/	Video/
Cell/	Gist/	Neuro/	Sbqa/	

ImageSets, a.k.a. Image Pyramids

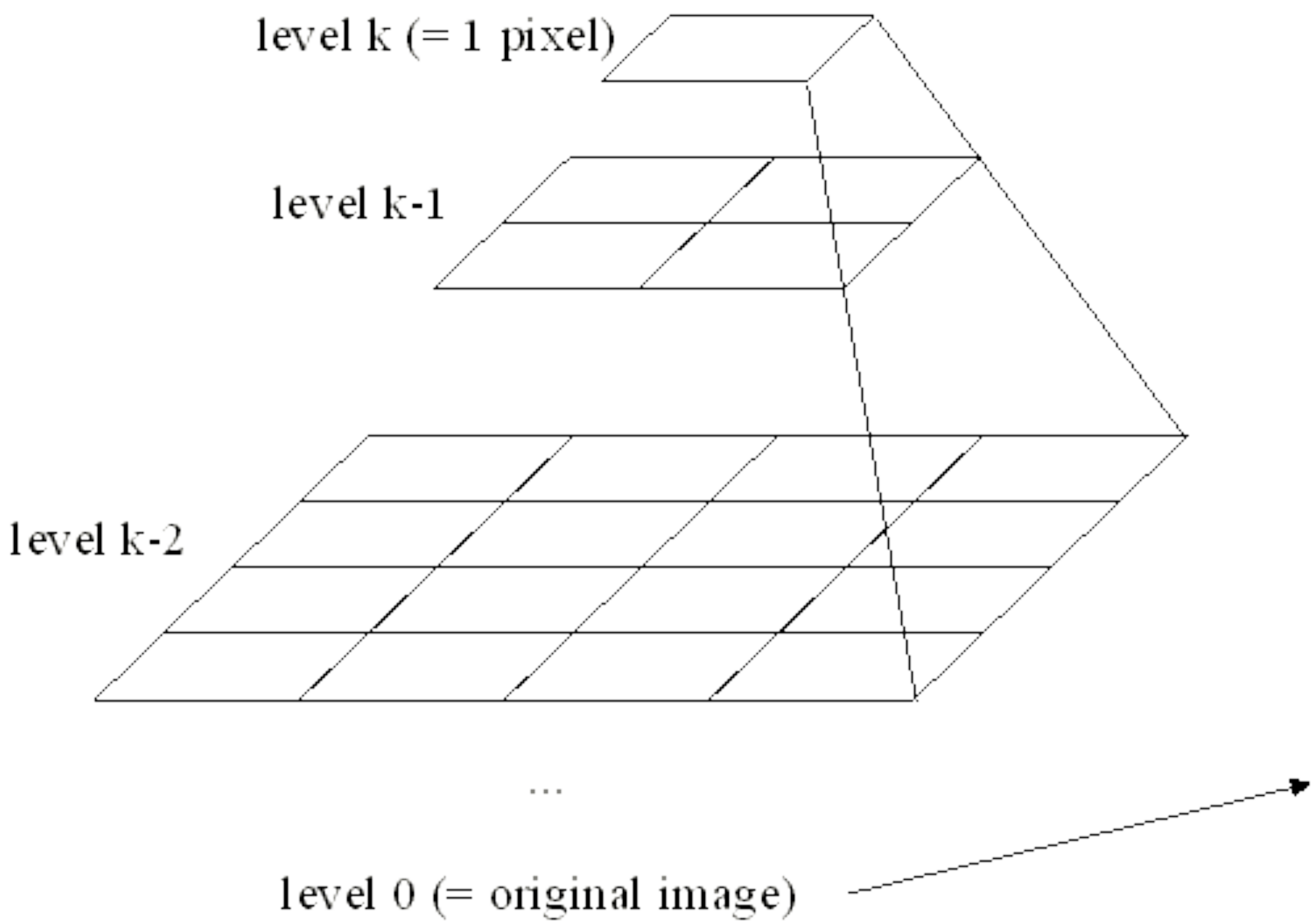
- Collection of images
- Dyadic image reduction from one level to next
- Various filters applied before reduction



See Image/ImageSet.H, ImageSetOps.H, PyramidOps.H, PyrBuilder.H, etc

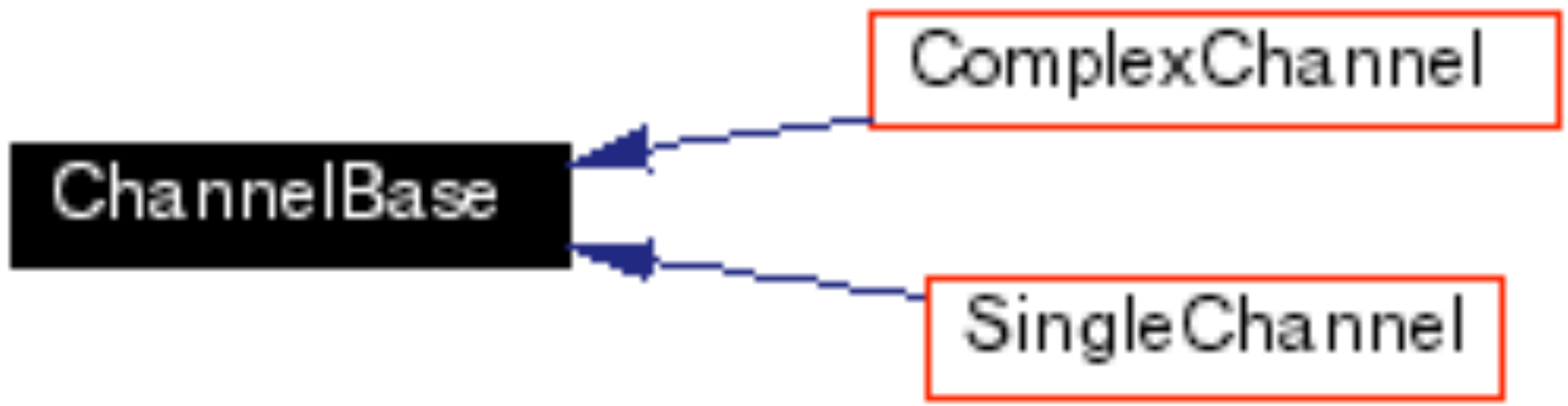
Gaussian Pyramid

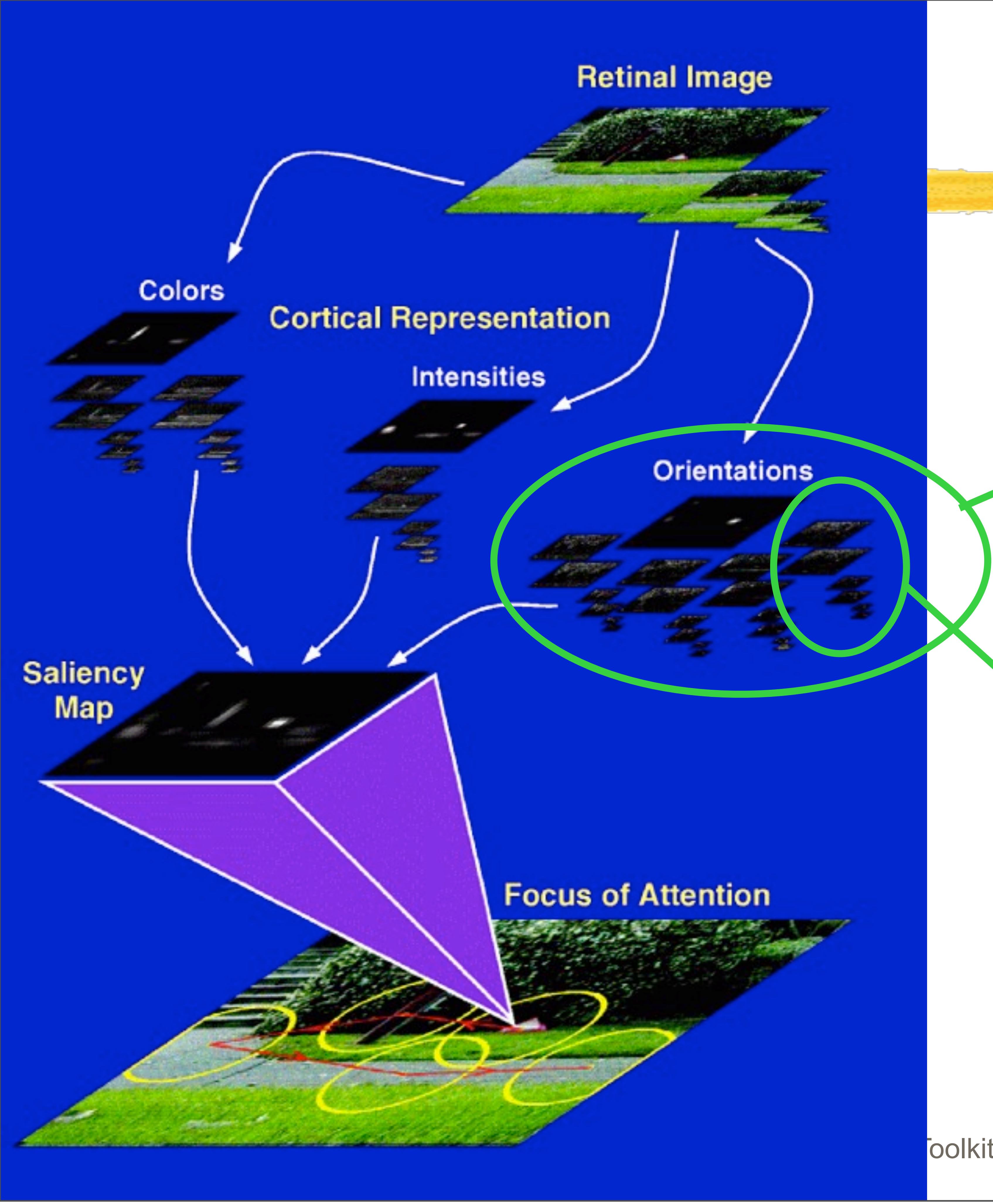
Idea: Represent NxN image as a "pyramid" of $1x1$, $2x2$, $4x4$,…, $2^k x 2^k$ images (assuming $N=2^k$)



Channels

- Implement a pyramid or collection of pyramids plus some I/O functions and additional processing
- Various derived instances can be identified by name
- SingleChannel: contains one pyramid
- ComplexChannel: contains a collection of SingleChannels





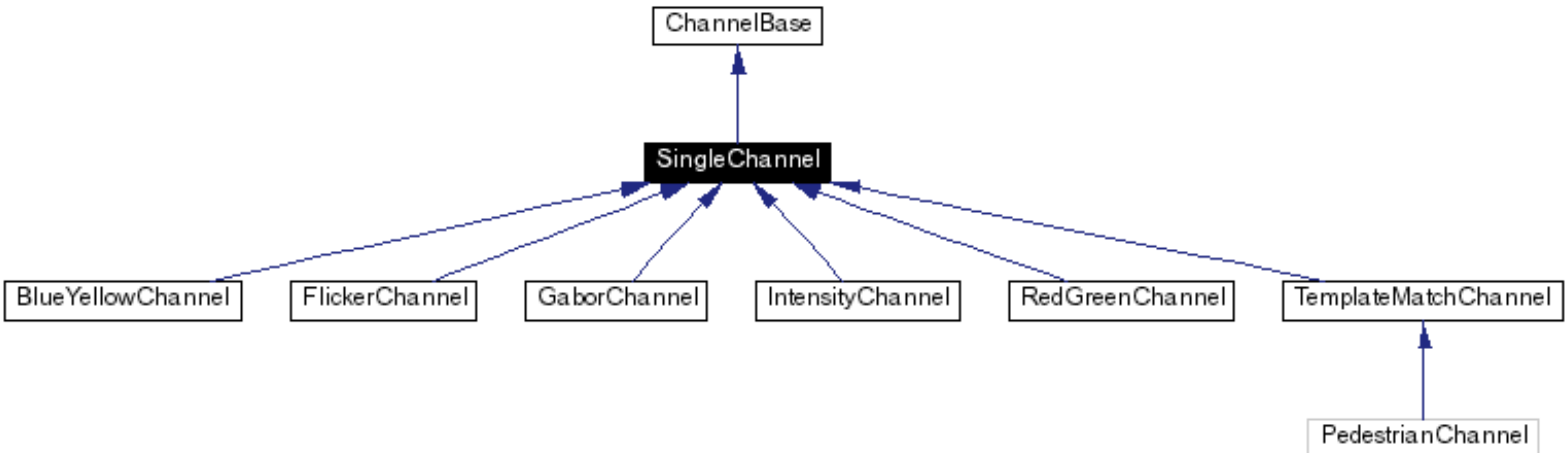
ComplexChannel

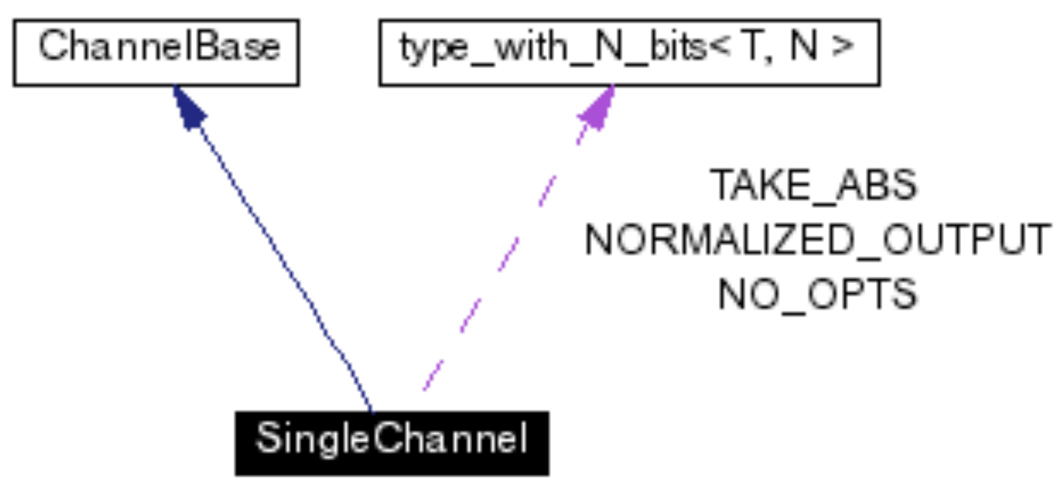
SingleChannel

Toolkit

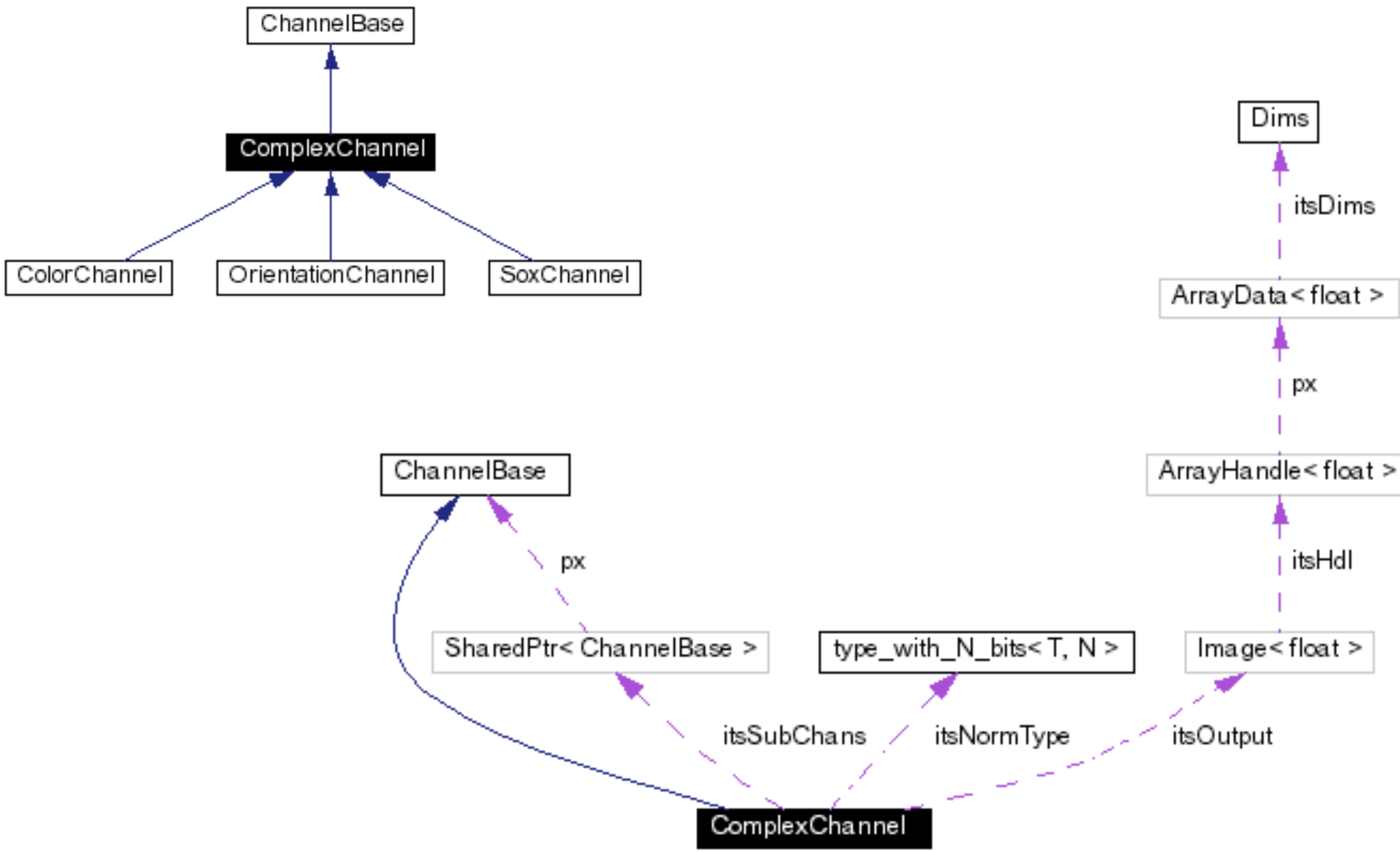
Itti & Koch,
Vision Research 2000

Single Channels



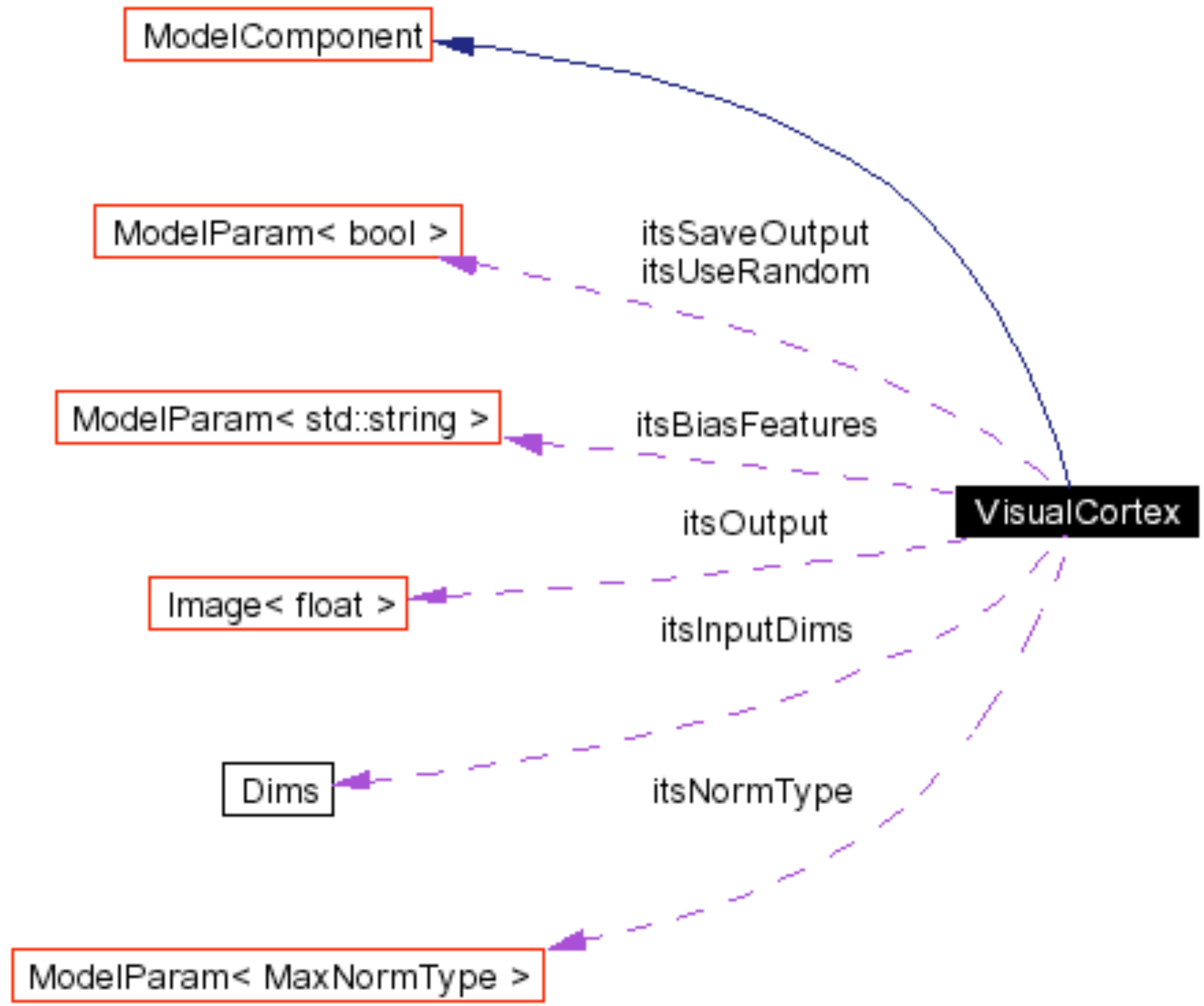


Complex channels



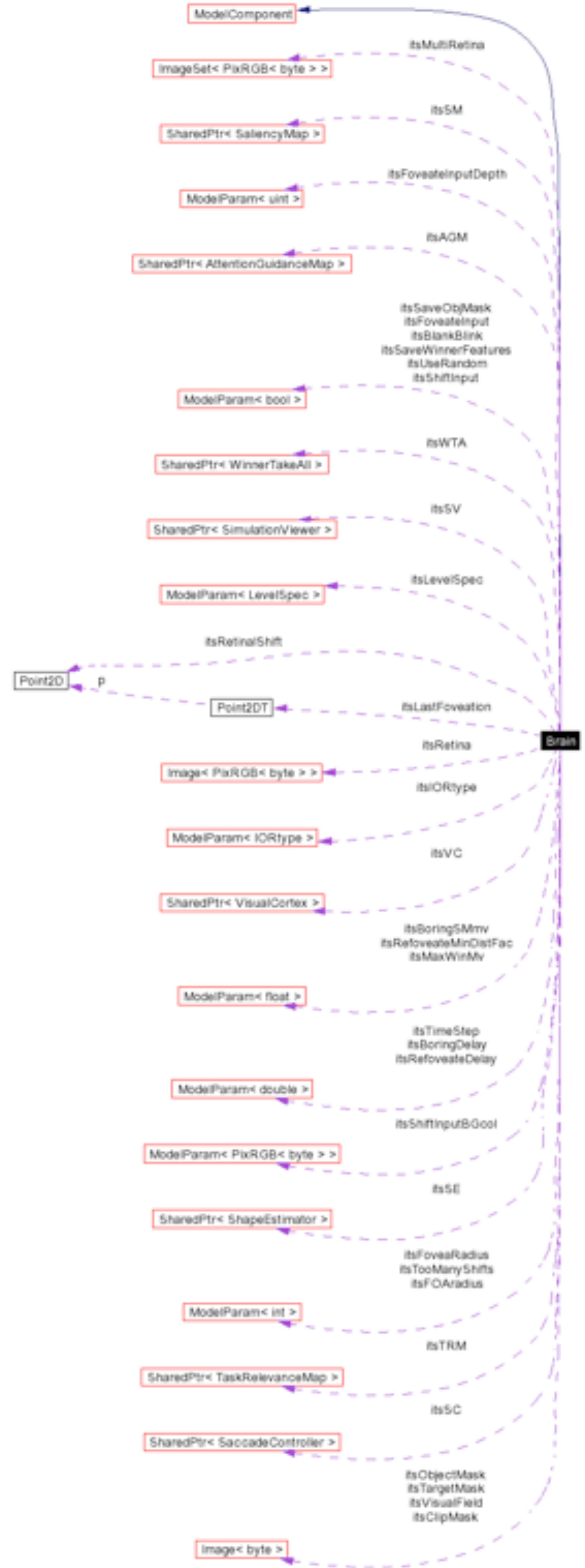
VisualCortex

- Run-time configurable collection of channels, plus additional I/O and access methods



Brain

VisualCortex plugged-in
at run-time

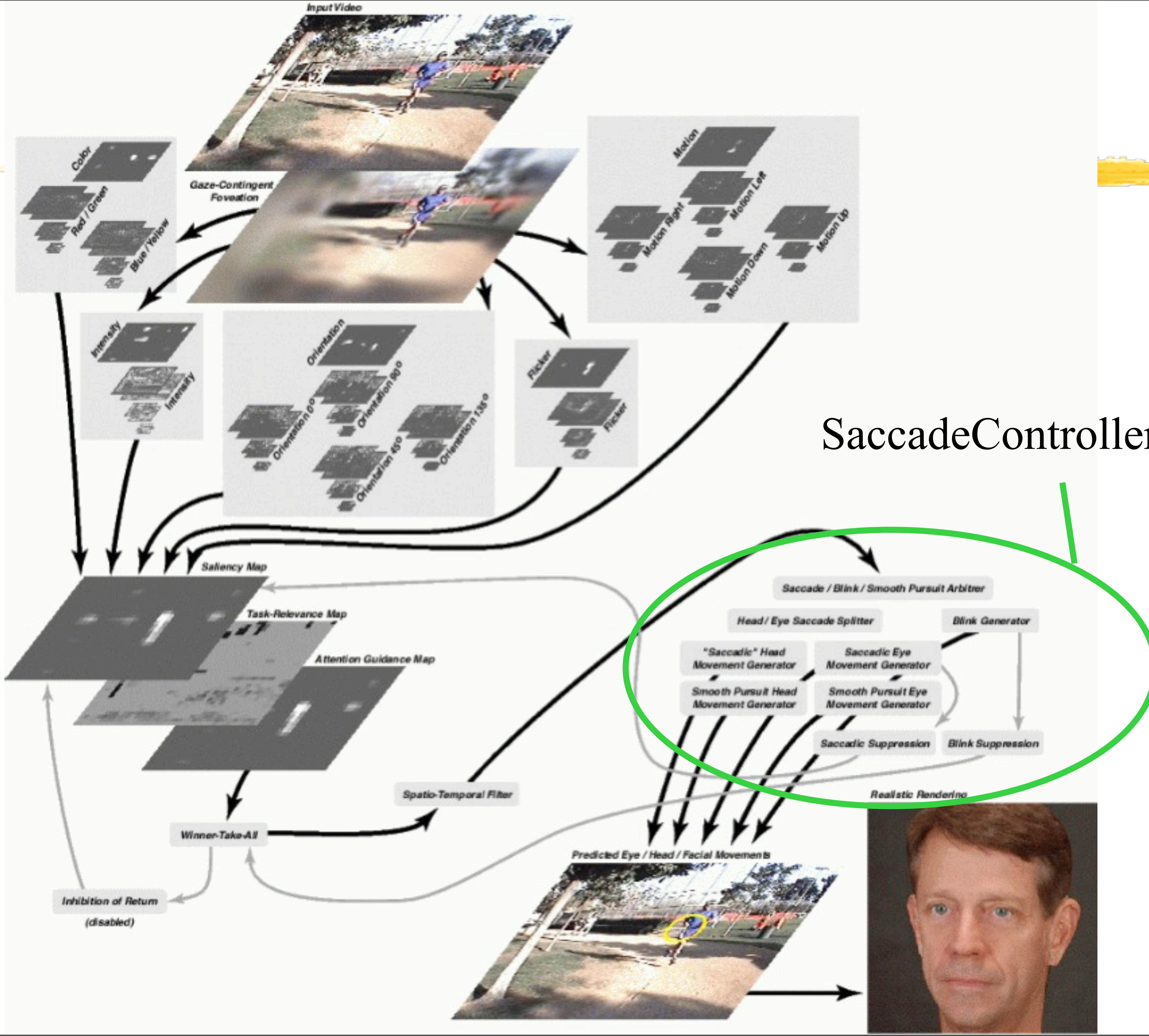


Brain: basic operation

Data flow is controlled by a **blackboard architecture**: Brain modules can post messages and can register callbacks which will be called when some messages are posted by other modules.

Processing flow is driven by reading new input images (stream oriented):

- Get an input image
- Process it through VisualCortex, get saliency map input
- Feed saliency map
- Let saliency map evolve
- Let task-relevance map evolve
- Combine saliency map and task-relevance map outputs to feed attention-guidance map
- Let attention-guidance map evolve
- Feed output of attention-guidance map to winner-take-all
- Get winner-take all output, if any
- Feed that to saccade controller
- Also feed it to shape estimator
- Activate inhibition of return
- ...


Input Video
Gaze-Contingent Foveation
Color
Red / Green
Blue / Yellow
Motion
Motion Right
Motion Left
Motion Down
Motion Up
Intensity
Intensity
Orientation
Orientation 0°
Orientation 90°
Orientation 45°
Orientation 135°
Flicker
Flicker
Saliency Map
Task-Relevance Map
Attention Guidance Map
Winner-Take-All
Inhibition of Return
(disabled)
Spatio-Temporal Filter
SaccadeController
Saccade / Blink / Smooth Pursuit Arbitrer
Head / Eye Saccade Splitter
Blink Generator
"Saccadic" Head Movement Generator
Saccadic Eye Movement Generator
Smooth Pursuit Head Movement Generator
Smooth Pursuit Eye Movement Generator
Saccadic Suppression
Blink Suppression
Realistic Rendering
Predicted Eye / Head / Facial Movements


Welcome to iLab! - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://ilab.usc.edu/
USC UNIVERSITY OF SOUTHERN CALIFORNIA
iLab
Welcome to iLab at the University of Southern California!
Research
Publications
People
Facilities
Classes
Opportunities
Events + Links
Search
http://ilab.usc.edu/publications/
Internet
Start
beobots
beobot030901
Welcome to iLab! - Microsof...
1:12 AM

Welcome to the Beobot Project - Microsoft Internet Explorer

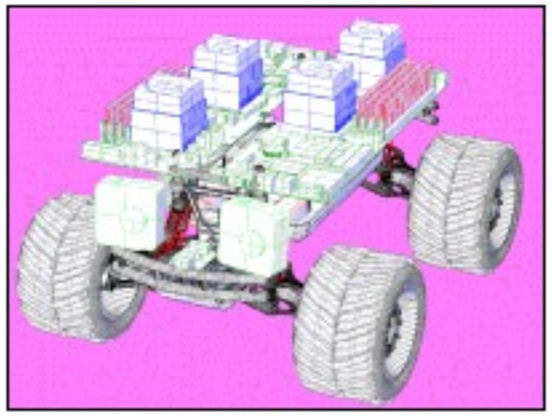
File Edit View Favorites Tools Help Address http://ilab.usc.edu/beobots/

iLab

beobot

Towards Visually-Guided Neuromorphic Robots

- Home
- Overview
- News
- Hardware
- Software
- Gallery
- Downloads
- Team
- Publications
- Sponsors
- Links



Welcome to the Beobot Project!

Beobots are autonomous robots whose brains are standard Linux clusters of computers which run real-time neuromorphic vision algorithms.

Just like **Beowulf Clusters** have revolutionized the world of high-performance computing, replacing costly and slowly-evolving custom supercomputer hardware by assemblies of inexpensive, mass-produced personal computers, we hope that Beobots (a **Beowulf** cluster on a mobile **robot**) will lead the way towards a new generation of robotics systems that are inexpensive, rapidly evolving, built from standard mass-produced components, and armed with sufficient computational power to run real-time neuromorphic vision algorithms.

- So **what exactly** is a Beobot?
- What **hardware** is it made of?
- What **software** does it run?
- Who are the **people** working on it?

Last CVS commit: Mon Sep 1 19:06:20 2003

Done | Internet

Start | beobots | beobot030901 | Welcome to the Beobot Pro... | 1:14 AM iLab C++ Neuromorphic Vision Toolkit Coding Home Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help Address http://ilab.usc.edu/toolkit/

Recent CVS / Forum Activity

Ordered by last CVS commit date/time.

User	Last CVS Commit			Last iLab Forum Post
walther	2003-09-01 at 19:06	saliency/src3/shapeEstimatorWebpage.C	1.5	Wed Aug 27 08:41:39 2003
zhanshi	2003-08-29 at 10:15	saliency/src3/dummySTL.H	1.3	Sun Aug 31 19:44:22 2003
itti	2003-08-27 at 11:04	saliency/src3/SimulationViewerEyeMvt.C	1.9	Fri Aug 29 14:04:17 2003
mundhenk	2003-08-19 at 20:40	saliency/src3/stats.conf	1.21	Sat Aug 23 15:57:03 2003
rjpeters	2003-08-02 at 11:06	saliency/src3/corrcoef.C	1.1	Wed Jul 23 18:03:45 2003
vidhya	2003-06-25 at 01:03	saliency/src3/VisualCortex.H	1.73	Sun Jan 12 11:43:12 2003
daesu	2003-05-13 at 14:40	saliency/src3/test-roadShape.C	1.1	---
dhavale	2003-05-09 at 18:58	saliency/src3/wrapping/test-Cam.C	1.2	Wed Aug 20 05:47:41 2003
beobot	2003-02-20 at 21:56	saliency/bin/bbsync	1.5	---
rhirata	2002-10-23 at 14:54	beobots/software/gyro/Gyro2.C	1.2	---
jsn	2001-12-10 at 13:16	beobots/software/lcd/lcd.C	1.2	---
juliet	---			Sun Jun 1 23:25:23 2003
aprilla	---			Mon Jun 3 01:53:33 2002

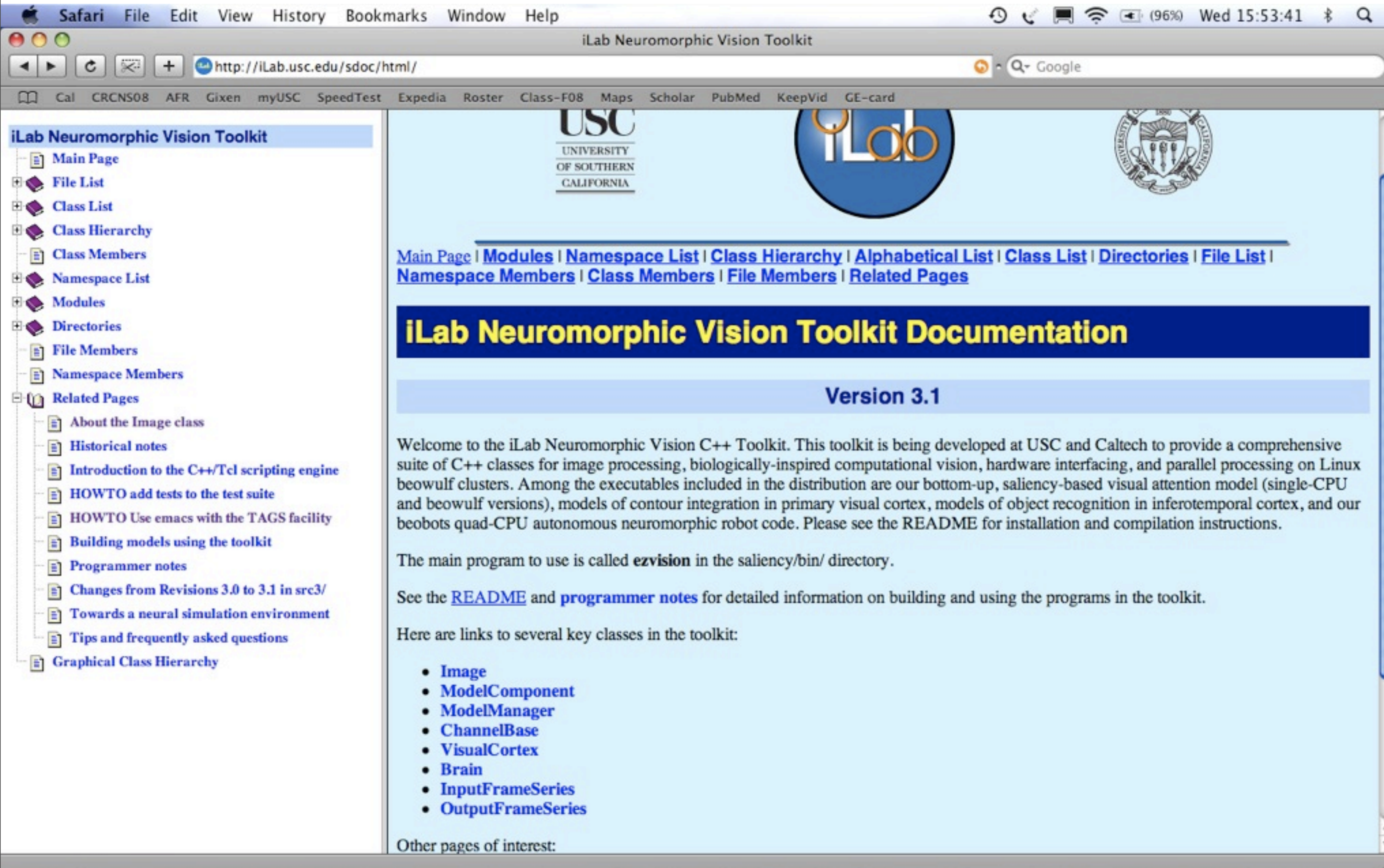
Latest CVS commits

Ordered by commit date/time.

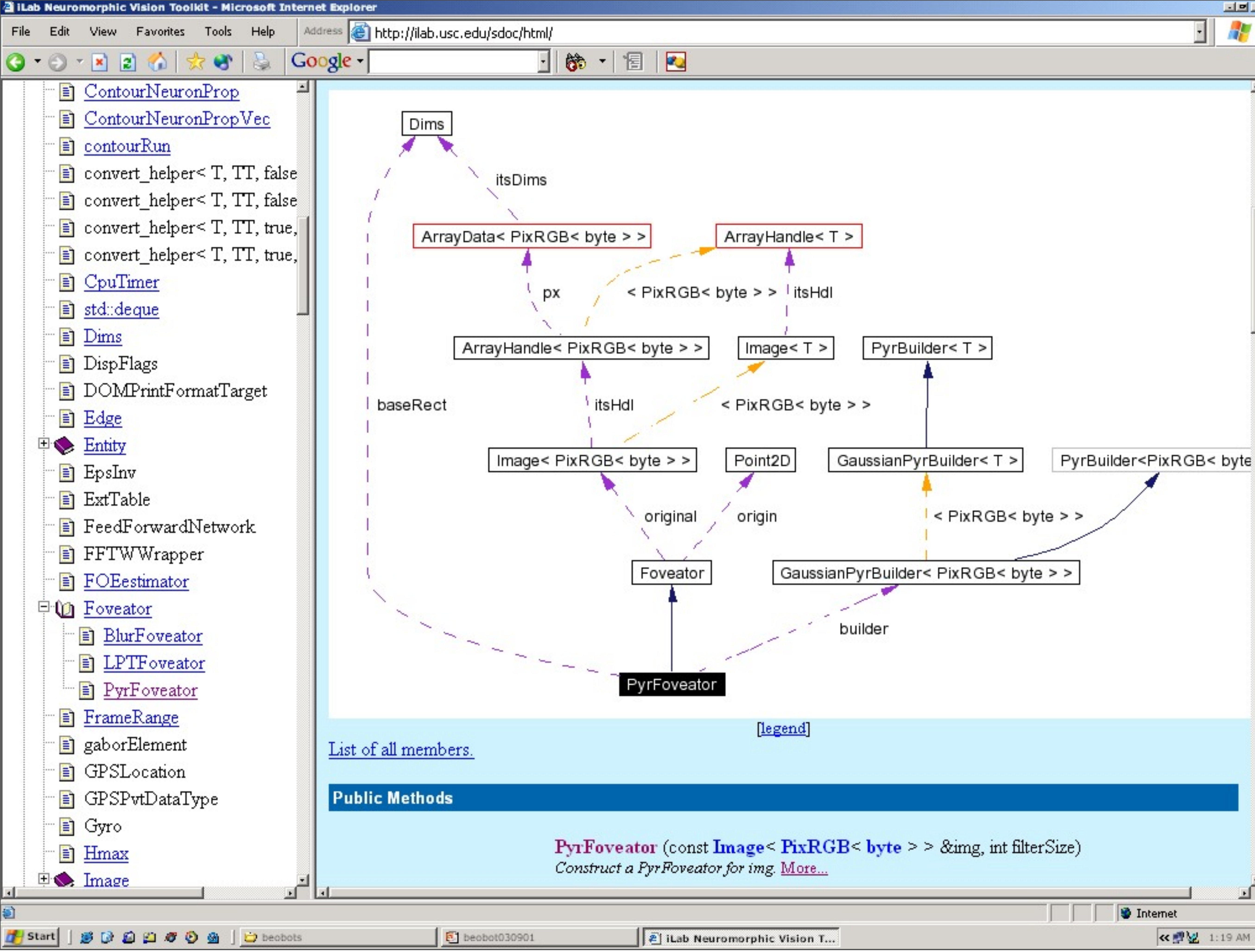
Date	Time (PST)	User	Version	File	Log Message

Done Internet

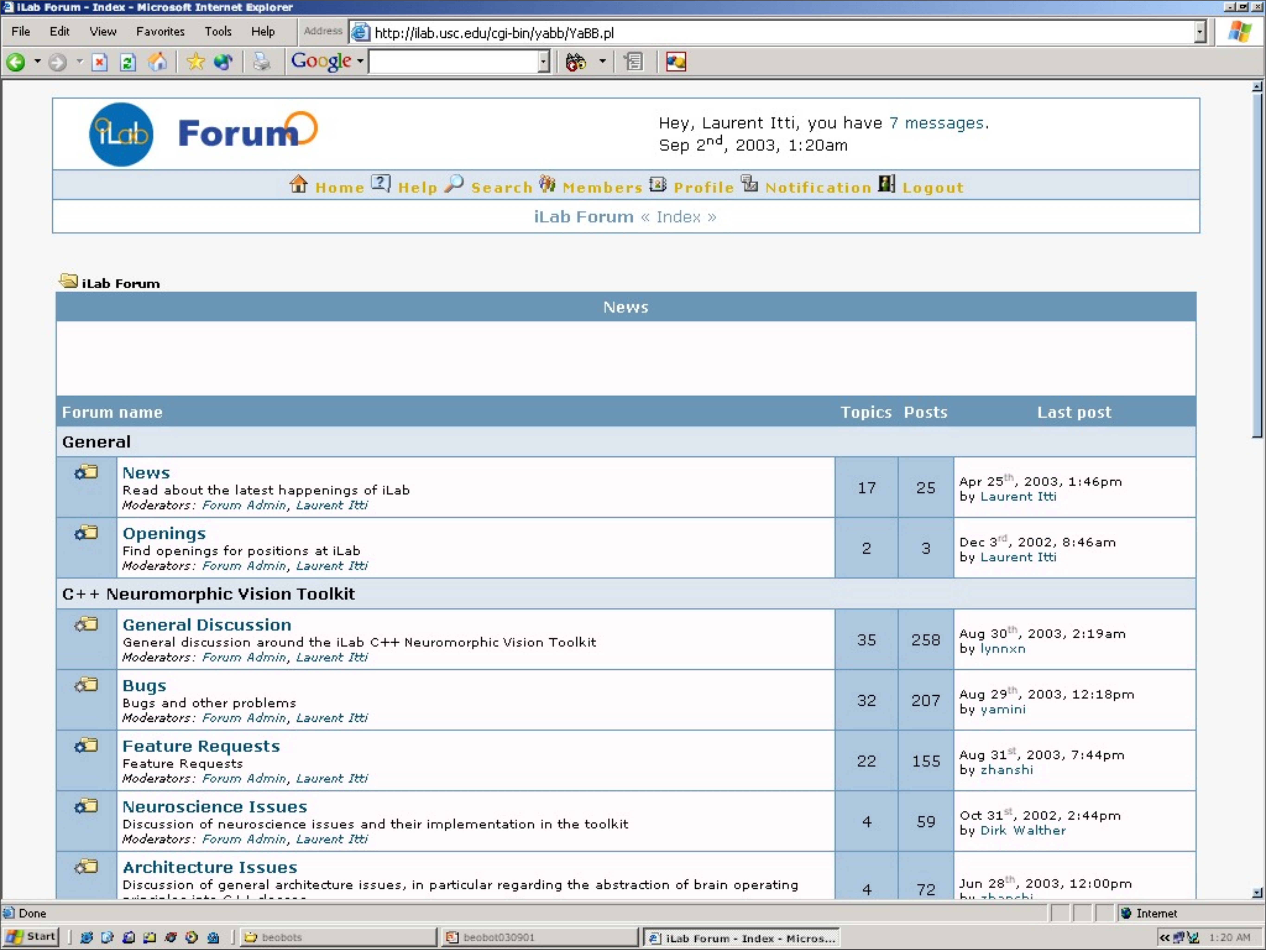
Start beobots beobot030901 iLab C++ Neuromorphic Vis... 1:16 AM



iLab C++ Neuromorphic Toolkit



Tuesday, July 20, 2010


iLab Forum - Index - Microsoft Internet Explorer
http://ilab.usc.edu/cgi-bin/yabb/YaBB.pl
Forum
Hey, Laurent Itti, you have 7 messages.
Sep 2nd, 2003, 1:20am
Home Help Search Members Profile Notification Logout
iLab Forum « Index »
iLab Forum
News
Forum name
Topics
Posts
Last post
General
News
Read about the latest happenings of iLab
Moderators: Forum Admin, Laurent Itti
17
25
Apr 25th, 2003, 1:46pm by Laurent Itti
Openings
Find openings for positions at iLab
Moderators: Forum Admin, Laurent Itti
2
3
Dec 3rd, 2002, 8:46am by Laurent Itti
C++ Neuromorphic Vision Toolkit
General Discussion
General discussion around the iLab C++ Neuromorphic Vision Toolkit
Moderators: Forum Admin, Laurent Itti
35
258
Aug 30th, 2003, 2:19am by lynnxn
Bugs
Bugs and other problems
Moderators: Forum Admin, Laurent Itti
32
207
Aug 29th, 2003, 12:18pm by yamini
Feature Requests
Feature Requests
Moderators: Forum Admin, Laurent Itti
22
155
Aug 31st, 2003, 7:44pm by zhanshi
Neuroscience Issues
Discussion of neuroscience issues and their implementation in the toolkit
Moderators: Forum Admin, Laurent Itti
4
59
Oct 31st, 2002, 2:44pm by Dirk Walther
Architecture Issues
Discussion of general architecture issues, in particular regarding the abstraction of brain operating
4
72
Jun 28th, 2003, 12:00pm
Done
Internet
Start
beobots
beobot030901
iLab Forum - Index - Micros...
1:20 AM

iLab Forum - Feature Requests - Microsoft Internet Explorer

File Edit View Favorites Tools Help Address http://ilab.usc.edu/cgi-bin/yabb/YaBB.pl?board=features

Forum

Hey, Laurent Itti, you have 7 messages.
Sep 2nd, 2003, 1:21am

Home Help Search Members Profile Notification Logout

iLab Forum « Feature Requests »

iLab Forum
C++ Neuromorphic Vision Toolkit
Feature Requests (Moderators: Forum Admin, Laurent Itti)

Feature Requests

Pages: 1 2 | Mark Topics as Read | Start new topic

Subject	Started by	Replies	Views	Last post
new X-Windows as command-line option	zhanshi	14	142	Aug 31st, 2003, 7:44pm by zhanshi
new IEEE1394 update	Laurent Itti	6	76	Aug 30th, 2003, 9:27pm by zhanshi
STL, doxygen, and graphviz	zhanshi	4	53	Aug 29th, 2003, 10:43am by Laurent Itti
new Added PNG write capability	Rob Peters	1	16	Jul 23rd, 2003, 6:34pm by Laurent Itti
new Methods for computing orientations	Dirk Walther	7	71	Apr 28th, 2003, 11:12am by Laurent Itti
new Dust off the Raster interface?	Rob Peters	11	88	Mar 21st, 2003, 5:49pm by Rob Peters
new Pyramids; LOGVERB+FULLTRACE « Pages 1 2 »	Rob Peters	22	185	Mar 8th, 2003, 4:53pm by Laurent Itti
new tests that take a long time	Laurent Itti	11	96	Mar 6th, 2003, 4:29pm by Laurent Itti
new threadsafe refcounting	Laurent Itti	4	54	Jan 25th, 2003, 11:20am by Laurent Itti
new Detection of targets by biasing features	vidhya	1	57	Jan 13th, 2003, 5:20pm by Laurent Itti
new itsLevels in PyramidBase	Dirk Walther	4	67	Dec 3rd, 2002, 11:44am

Done | Internet

Start | beobots | beobot030901 | iLab Forum - Feature Reque... | 1:21 AM iLab Publications - University of Southern California - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://ilab.usc.edu/publications/

University of Southern California
Hedco Neuroscience Building
Los Angeles, CA 90089-2520 - USA

Publications

Welcome to the iLab Publication Server!

115 publications, 73 with abstract, 55 available as PDF.

Publications by Year

in-press 2003 2002 2001 2000 1999 1998 1997 1996 1995 1994 1992 1991 1990 1989

Publications by Type and by Theme

- All Publications
- Journal Articles
- Publications in Press
- Book Chapters
- Proceedings from International Conferences
- Master Theses
- Patents and Copyrights
- Ph.D. Theses

- Beobots
- Model of Bottom-Up Saliency-Based Visual Attention
- Computer Vision
- Human Eye-Tracking Research
- Functional Neuroimaging
- Medical Research
- Medical Image Processing
- Computational Modeling
- Press Coverage
- Human Psychophysics
- Review Articles and Chapters

Done Internet

Start 0312_iNVTintro Microsoft PowerPoint - [iNVT-in... iLab Publications - Universi... 3:05 PM

Visual Attention Home Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help Address http://ilab.usc.edu/bu/

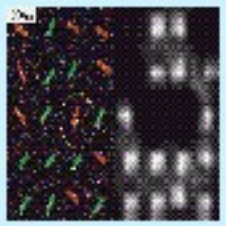
This project was started at Caltech with **Prof. Christof Koch**. It is actively being pursued both here and at Caltech (both jointly and in different directions).

The Theory

Details about the trainable model of bottom-up, task-independent visual attention under development in our laboratory.

The Images

A short overview of example images and the corresponding attentional trajectories. Test images, psychophysical stimuli, target detection images, natural scenes, artwork, etc.



The Movies

Several MPEG movies showing attentional trajectories and the temporal dynamics of the Saliency Map for test, psychophysical, artistic and natural images. Also shown are 3D warping of the original image onto the evolving saliency map.

The Interactive Demo

An interactive demonstration of the dynamic behavior of our attentional model, for a variety of complete image databases. Most recent Java™-aware Web browser required.

The Publications

Some pre-versions of our papers describing this research are available in HTML, Postscript and PDF format.

The Ongoing Projects

New! Previews of a few of our ongoing projects and preliminary screenshots. These include our **SaliencyVehicle** off-road muscle car, our real-time **SaliencyCam** which computes attentional deployment on live video feeds (15 frames/s), our **SaliencyAgent** which detects salient pedestrians in natural color scenes, and other exciting projects.

The C++ Source Code

The C++ source code and associated doxygen documentation are available through our CVS server. You will need the latest version of g++ (3.x) and several non-standard packages installed on your Linux distribution (e.g., IEEE1394

Done Internet

Start 0312_iNVTintro Microsoft PowerPoint - [iNVT-in... Visual Attention Home Page... 3:06 PM iLab - University of Southern California - Microsoft Internet Explorer

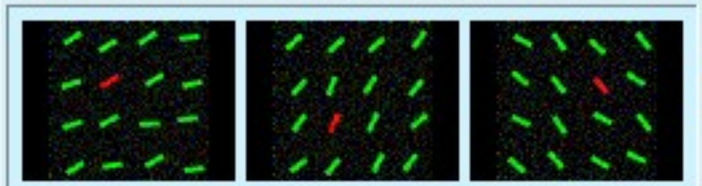
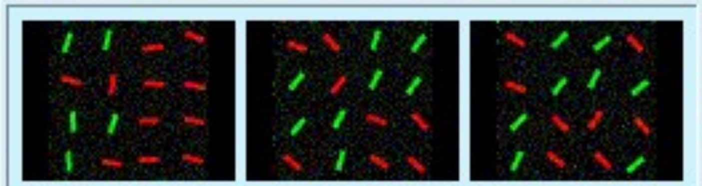
File Edit View Favorites Tools Help Address http://ilab.usc.edu/imgdbs/

iLab Image Databases

These image databases are provided for testing and evaluation only. Some of the images in the databases have been grabbed from the web, and may be subject to copyright. So, do not use these images in any commercial application!

All images are in *PPM* (24-bit color) or *PGM* (8-bit greyscale) format, compressed with *bzip2* and compiled in *tar* archives.

Note: We have put a lot of effort into making these databases available to you. By downloading any of the databases below, you agree to properly cite the associated master reference, which typically is the paper where we first described the database and used it with our model, and to provide a link to the present web page.

Samples	Database	# Images	Size	Description	Master Reference
	STIMart.tar	20	4.0 MB	Miscellaneous artwork, posters and portraits	Itti et al., IEEE PAMI, 1998
	STIMautobahn.tar	90 + 90	56 MB	Color images with German traffic signs + target masks	Itti & Koch, J. Elec. Imag., 2001
	STIMcoke.tar	104 + 104	53 MB	Color images with a red can + target masks	Itti & Koch, J. Elec. Imag., 2001
	STIMcolor.tar	180 + 180	2.1 MB	Color popout search arrays + target masks	Itti & Koch, Vis. Res., 2000
	STIMoricol.tar	180 + 180	2.1 MB	Orientation/color conjunctive search arrays + target masks	Itti & Koch, Vis. Res., 2000
		180 +	2.1	Orientation popout search arrays +	Itti & Koch, Vis. Res.

Internet

Start 0312_iNVTintro Microsoft PowerPoint - [iNVT-in... iLab - University of Souther... 3:12 PM