Appendix B

The SaliencyToolbox

B.1 Introduction

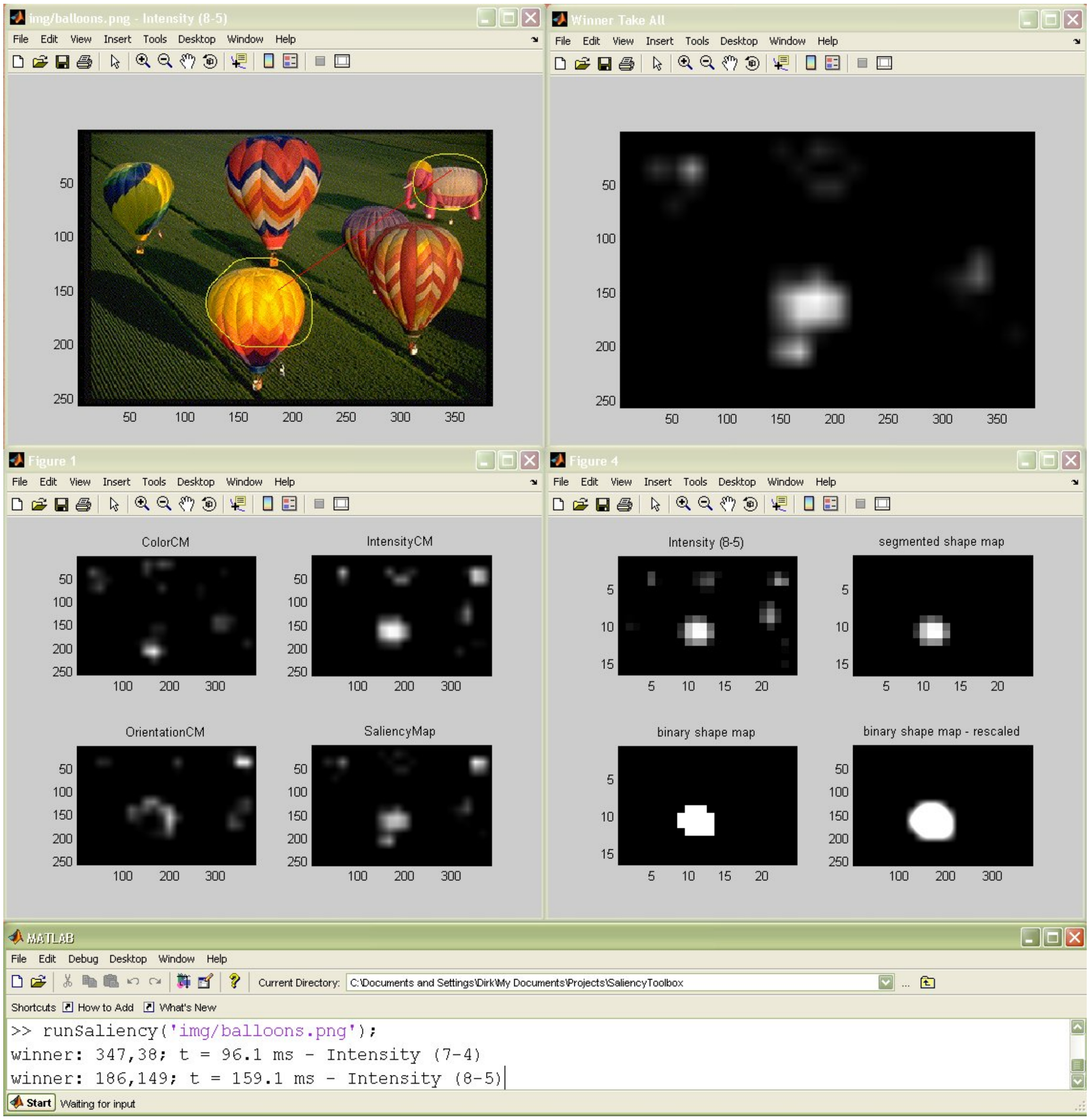
The SaliencyToolbox is a collection of Matlab functions and scripts for computing the saliency map for an image, for determining the extent of a proto-object as described in chapter 2, and for serially scanning the image with the focus of attention. Being mostly written in Matlab, the code is easily accessible, easy to experiment with, and platform independent.

Major parts of the code are reimplemented from the *iLab Neuromorphic Vision C++ Toolkit* (iNVT) at Laurent Itti's lab at the University of Southern California (`http://ilab.usc.edu/toolkit`). The iNVT code base contains much functionality beyond the actual saliency computations. Unfortunately, its feature richness (contained in 360,000 lines of code) makes the code increasingly difficult to understand for novices. The SaliencyToolbox complements the iNVT code in that it is more compact (about 5,000 lines of code) and easier to understand and experiment with. The SaliencyToolbox only contains the core functionality for attending to salient image regions. Although most time critical processing steps are coded in C++ mex files, processing an image with the SaliencyToolbox in Matlab takes longer than with the iNVT code. Whenever processing speed or feature richness are paramount, the iNVT code should be preferred. For computing the saliency map or attending to salient proto-objects in an image in a transparent and platform independent way, the SaliencyToolbox is a good choice.

The SaliencyToolbox requires a Matlab Release 13 or 14[1] and the Matlab Image Processing Toolbox. Most time critical parts of the code are coded in C++ mex files. Pre-compiled binaries of the mex files are included for Microsoft Windows, Mac OS X, Linux 32 bit Intel/AMD, and Linux 64 bit AMD Opteron. The source code can be compiled on any system with the GNU C compiler gcc (see section B.4 for details).

The SaliencyToolbox is licensed under the GNU General Public License (`http://www.gnu.org/copyleft/gpl.html`). Some portions of the code are covered under two U.S. patent applications.

[1] Other versions might work but were not tested.



Current Directory: C:\Documents and Settings\Dirk\My Documents\Projects\SaliencyToolbox

```
>> runSaliency('img/balloons.png');
winner: 347,38; t = 96.1 ms - Intensity (7-4)
winner: 186,149; t = 159.1 ms - Intensity (8-5)
```

Figure B.1: Screen shot of a typical display while running the SaliencyToolbox.

(Koch and Itti 2001; Rutishauser et al. 2004b). Since the code and the documentation are evolving, please refer to the web site (`http://www.saliencytoolbox.net`) for the most recent version.

B.2 Installation

The code can be downloaded free of charge from `http://www.saliencytoolbox.net`. Unpacking the code creates a directory structure under SaliencyToolbox. Simply add the SaliencyToolbox directory with its subdirectories to your Matlab path:

```
addpath(genpath('<your SaliencyToolbox path>'));
```

If access to the toolbox is desired every time Matlab starts, the above command should be places into your startup.m file, which is located in `~/matlab/startup.m` for Linux, Mac OS X and other Unix flavors, and in the "Start in" directory for your Matlab desktop shortcut under Microsoft Windows.

B.3 Quick Start

In Matlab, change to the SaliencyToolbox directory, then type:

```
runSaliency('img/balloons.png');
```

After a few moments, you should see four figure windows with various intermediate results, and in the Matlab window you will see details about the most salient location, which is also marked in the image (figure B.1). You can now hit "Return" to go to the next most salient location and so forth. To quit, simply enter "q" and press "Return".

If you receive the following fatal error: "MEX file xxx could not be found," then you need to compile the mex files for your system (section B.4).

The `SaliencyToolbox/img` directory contains a few example images. The `runSaliency` function also takes an optional second argument, a `saliencyParams` structure. Start exploring the documentations for `runSaliency`, `defaultSaliencyParams`, and `dataStructures` to get an idea of what is happening. It may be a good idea to make a copy of `runSaliency.m` and start dissecting and adapting it for your purpose.

For more feedback on what is happening at each time step, try the following:

```
declareGlobal;
DEBUG_FID = 1;
runSaliency('img/balloons.png');
```

See documentation for `initializeGlobal`, `declareGlobal`, and `debugMsg` for what is happening here.

B.4 Compilation

Most users do not need to compile the mex files. Binaries for the most common architectures are included with the toolbox. Compilation may become necessary if the binaries for your operating system and CPU combination are not in the `SaliencyToolbox/bin` directory yet. In this case, please send me a note once you have successfully compiled your mex files, and I may include the binaries for your system in the next release. Compilation is also necessary, of course, if you modify the C++ code of the mex files.

B.4.1 Linux, Mac OS X, and other Unix flavors

This requires the GNU gcc compiler, version 3.2.3, if possible. The compiler should be named "gcc-3" and its C++ equivalent "g++-3". Create an alias or link if your compiler is called differently. Check by typing:

```
gcc-3 --version
```

The 3.3 variants in Fedora Core and OS X work as well. Gcc 3.4.x definitely does NOT work for mex files; they will not link correctly at Matlab runtime. Gcc versions before 3.0 will not work because they do not fully implement the standard template library. Gcc 4.x does not appear to work, either.

The `mex` shell script that comes with Matlab needs to be in the executable path. The `mex` scripts of Matlab releases 13 and earlier do not understand the "-cxx" option to signal C++ code; release 14 requires it. Hence, if your Matlab version is R13 or older, you need to edit two lines around line 25 of your `SaliencyToolbox/mex/Makefile` to this:

```
#MEXFLAGS := -cxx
MEXFLAGS :=
```

Now, change into directory `SaliencyToolbox/mex` and type:

```
make
```

This will create the binaries in `SaliencyToolbox/bin`. If you get a message saying that nothing needs to be done, but you are sure that you need to recompile the code, then use this command:

```
make clean all
```

B.4.2 Microsoft Windows

To compile the code under Windows, MinGW (minimalistic GNU for Windows) and MSYS (Minimalistic SYStem) need to be installed. This is free software in the public domain. Follow these steps to install a working environment for compiling mex files (all downloads are available at `http://www.mingw.org/download.shtml`):

(i) Download mingw-runtime-3.8.tar.gz and unpack into a new folder `c:\mingw`.

(ii) Download binutils-2.15.91-20040904-1.tar.gz and unpack into `c:\mingw`.

(iii) Download gcc-3.2.3-20030504-1.tar.gz and unpack into `c:\mingw`. If you are asked if you wish to overwrite some existing files, click on “yes to all”.

(iv) Download w32api-3.3.tar.gz and unpack into `c:\mingw`.

(v) Download MSYS-1.0.10.exe and run the installer, install into `c:\msys`.

(vi) During the postinstall, answer “y” on the question if you have mingw, then enter `/c/mingw` as its installation path.

Start your newly installed MSYS environment, which will give you a Unix-like shell, and change the directory to `SaliencyToolbox/mex`. Make sure that the global environment variable `MATLABROOT` is set to the Matlab installation directory, e.g., by typing

```
export MATLABROOT=/c/MATLAB7
```

Now you can compile the code by typing

```
make
```

which creates the Windows `.dll` files in `SaliencyToolbox/bin`. If you get a message saying that nothing needs to be done, but you are sure that you need to recompile the code, then use this command:

```
make clean all
```

B.5 Generating the Documentation

In order to re-create the documentation for the C++ source code for the mex files, you need to have doxygen and graphviz (for the `dot` command) installed and in your executable path. Then change to the `SaliencyToolbox/mex` directory and type:

```
make doc
```

This will create the documentation in `SaliencyToolbox/doc/mexdoc`.

The documentation for the m-files is generated from within Matlab using the m2html toolbox (`http://www.artifact.tk/software/matlab/m2html/`). Make sure that m2html is in your Matlab path, then change to the SaliencyToolbox base directory and type:

```
STBgenerateDoc;
```

This creates the documentation for the m-files in `SaliencyToolbox/doc/mdoc`.