Virtual Retina on Mac

Horacio Rostro-Gonzalez, Adrien Wohrer, Thierry Viéville, Pierre Kornbrobst
INRIA-Project Odyssee
E-mail : Horacio.Rostro_Gonzalez@sophia.inria.fr

This installation guide complement Adrien's work in order to install and compile Virtual Retina software on Mac.

I. Pre-installation of the Virtual Retina

– You need have installed X11 library in your computer

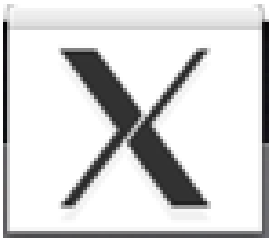

FIG. 1. X11 icon

If you do not have this library in your system you can do it as follows :

1) Panther on your Mac
   * X11 downloadable from http ://www.apple.com/support/downloads/x11formacosx.html
2) Tiger on your Mac
   * Installation from DVD of Tiger :

"the right package is X11User.pkg"

II. Installation and compilation of the Virtual Retina

A. *Creating local directories*

In order to install and compile all the necessary libraries we need create some **local** directories :

1) Open a Terminal
2) mkdir VirtualRetina
3) mkdir VirtualRetina/local
4) mkdir VirtualRetina/local/bin
5) mkdir VirtualRetina/local/lib
6) mkdir VirtualRetina/local/include
7) mkdir VirtualRetina/External_Libraries

B. *CMAKE*

1) For Mac Intel http ://www.cmake.org/files/v2.4/cmake-2.4.7-Darwin-universal.tar.gz
2) For Mac PPC http ://www.cmake.org/files/v2.4/cmake-2.4.7-AIX-powerpc.tar.gz
3) From the Terminal do :
   – tar -zfx "FILE (PPC or INTEL)"
   – rm "FILE*'
   – mv "FILE (unzipped)" External_Libraries/CMake

C. CImg

1) For Mac Intel http ://sourceforge.net/project/downloading.php ?group_id=96492&use_mirror=garr&filename=CImg-1.2.7.zip&29354337
2) For Mac PPC http ://cimg.sourceforge.net/download.shtml
3) From the Terminal do :
   - unzip CImg1.2.7.zip
   - rm CImg1.2.7.zip
   - mv CImg1.2.7 External_Libraries/CImg
   - cp External_Libraries/CImg/CImg.h /VirtualRetina/local/include

D. MvaSpike

1) Platform independent Intel http ://gforge.inria.fr/frs/download.php/2517/Mvaspike-1.0.16.tar.gz
2) From the Terminal do :
   - tar -zxf mvaspike-1.0.16.tar.gz
   - rm mvaspike-1.0.16.tar.gz
   - mv mvaspike-1.0.16 External_Libraries/MvaSpike
   - cd External_Libraries/Mvaspike
   - ./configure - -prefix=/VirtualRetina/local
   - make
   - make install
   - make clean

E. libxml++

To install libxml++ there are two options :

1) Using FINK (Package administrator for mac)
   If you do not have fink installed in your computer, you can download this from :

   - For Mac PPC http ://sourceforge.net/project/showfiles.php ?group_id=17203
   - For Mac Intel http ://sourceforge.net/project/showfiles.php ?group_id=17203

   When you have installed fink in your system, then you can install libxml++ as is described in the next step
   @ Open a Terminal
   @ In the first instance you can use, "**fink list libxml++**", in order to know if this library is available to use in your system. If you do not have any results, you must update your fink(**fink update-all**)
   @ Now you can install as **fink install libxml++ /VirtualRetina/External _Libraries**
   @ cd VirtualRetina/External_Libraries/libxml++/
   @ ./configure - -prefix=/VirtualRetina/local
   @ make
   @ make install
   @ make clean

2) If you prefer install the libxml++ library manually, you need install the XML++ dependencies. All the XML++ dependencies need be downloaded and copied in /VirtualRetina/External_Libraries/
   @ gettext → ftp ://mirrors.kernel.org/gnu/gettext
   @ glib → ftp ://ftp.gtk.org/pub/glib/2.12/
   @ sigc++ → http ://ftp.gnome.org/pub/GNOME/sources/libsigc++/2.1/
   @ glibmm → http ://ftp.gnome.org/pub/GNOME/sources/glibmm/2.15/
   @ libxml++ → http ://ftp.gnome.org/pub/GNOME/sources/libxml++/2.20/

   CONFIGURATION :

   - Open a Terminal
   - **gettext**
     cd VirtualRetina/External_Libraries/gettext
     ./configure - -prefix=/VirtualRetina/local

make
make install
make clean

– **glib**
cd VirtualRetina/External_Libraries/glib
./configure - -prefix=/VirtualRetina/local
make
make install
make clean

– **sigc++**
cd VirtualRetina/External_Libraries/sigc++
./configure - -prefix=/VirtualRetina/local
make
make install
make clean

– **glibmm**
cd VirtualRetina/External_Libraries/glibmm
./configure - -prefix=/VirtualRetina/local
make
make install
make clean

– **libxml++**
cd VirtualRetina/External_Libraries/libxml++
./configure - -prefix=/VirtualRetina/local
make
make install
make clean

Note : if you have some problems with dependencies, you can search for these in /usr/lib (- -prefix=/VirtualRetina/local :**/usr/lib**)

F. xmlParameters++ and VirtualRetina

1) Downloading and copying retina_package in /VirtualRetina/
   * http ://www-sop.inria.fr/odyssee/software/virtualretina/download.shtml
   * Open a Terminal
   * tar -zxf retina_package.tar.gz
   * rm retina_package.tar.gz
   * mv /retina_package/VirtualRetina /VirtualRetina/
   * mv /retina_package/External_Libraries/xmlParameters++ /VirtualRetina/External_Libraries/

2) Compiling and installing xmlParameters++
   Open a Terminal
   cd /VirtualRetina/External_Libraries/xmlParameters++
   cmake CMakeLists.txt -DLIBRARY_OUTPUT_PATH :PATH=/VirtualRetina/local/lib
   make
   Note : if you have some problems with dependencies, you can search for these in /usr/lib
   (-DLIBRARY_OUTPUT_PATH :PATH=/VirtualRetina/local/lib :**/usr/lib**)

3) Compiling and installing VirtualRetina
   Open a Terminal
   cd /VirtualRetina/VirtualRetina
   cmake CMakeLists.txt
   make

III. Using the built-in isight camera and OpenCv in order to capture an images sequence

Some Mac's have a built-in isight camera in order to have a video chat, but in this project we can use it for do an images capture, and use these as input of the Virtual Retina ; for do this we need to use the OpenCv library[1] as follows :

1) If you have not Xcode Tools[2] installed on your Mac you can install it from http ://developer.apple.com/tools/xcode/
2) To create a folder in /VirtualRetina/ named ImageCapture and download there the OpenCv library from http ://opencv-library.sourceforge.net/Mac_OS_X_OpenCV_Port
3) With Xcode tools and OpenCv library installed in your Mac, now you need create an project as follows :
   – Open Xcode located in /Developer/Applications.

FIG. 2. Xcode icon

   – To create a new project go to menu bar and select **File** → **New project**.
   – An assistant window will be displayed

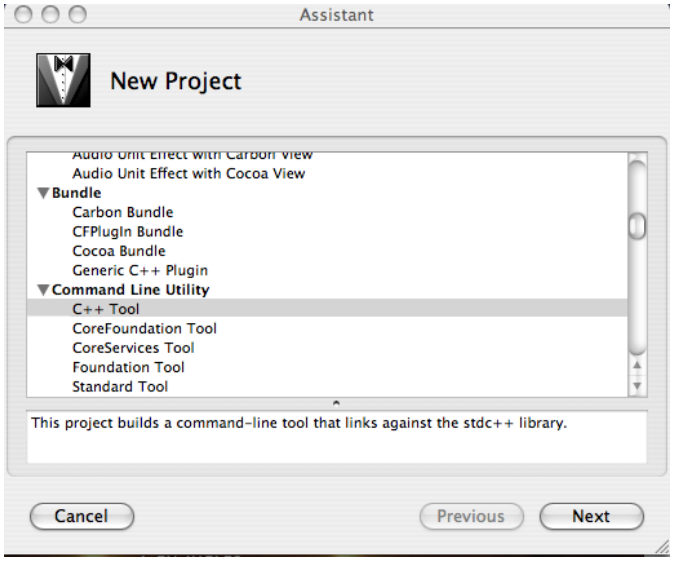

FIG. 3. Assistant Window

   – Select the option C++ Tool as Figure 3 shows and do click Next.
   – In the Next window you need put the name of the Project (the name could be anything) and the path (/VirtualRetina/ImageCapture), then do click Finish.
   – Now you have created your project, then a window with some files is displayed as Figure 4 shows.

[1] **OpenCV** is an open source computer vision library originally developed by Intel. It is free for commercial and research use under a BSD license. The library is cross-platform, and runs on Mac OS X, Windows and Linux. It focuses mainly on real-time image processing, as such, if it finds Intel's Integrated Performance Primitives (IPP) on the system, it will use these commercial optimized routines to accelerate itself.

[2] Xcode is Apple's premiere development environment for Mac OS X.

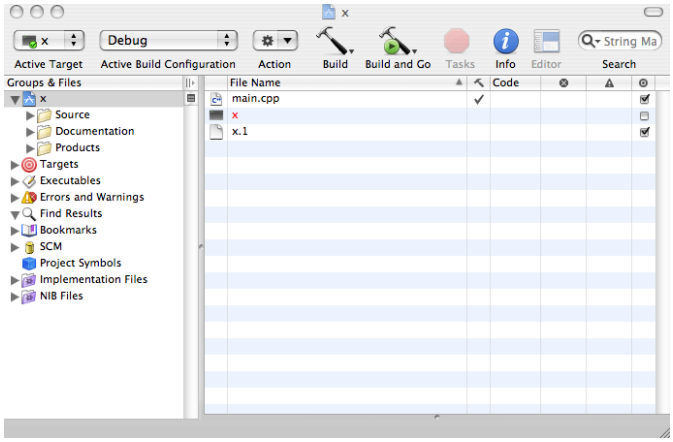

FIG. 4. Project Window

– Now you need add the OpenCv library, for do this go to Action Bottom in the project window and select **Add → Existing Frameworks** and locate the OpenCv library in /VirtualRetina/ImageCapture.
– Finally open the main.cpp file and replace all is contained in the file by the next code :

```
#include <OpenCV/OpenCV.h>
#include <cassert>
#include <iostream>

#define NF 50
#define CHECK(condition, message) { if (!(condition)) { fprintf(stderr, "Fatal error:
%s\n", message); exit(-1); } }
#define ALLOC(buffer, size, type) { buffer = (type *) calloc(size, sizeof(type));
CHECK(buffer != NULL, "Memory overflow"); }


const char  * WINDOW_NAME  = "Images capture";

using namespace std;

int main (int argc, char * const argv[])
{
int i=0,n;

char  *filename;
char  *path = "/VirtualRetina/ImageCapture/Sequence";
char  *name = "image";
char  *ext  = ".pgm";



cvNamedWindow (WINDOW_NAME, CV_WINDOW_AUTOSIZE);
    CvCapture * camera = cvCreateCameraCapture (CV_CAP_ANY);
IplImage *  current_frame = cvQueryFrame (camera);
    // you do own an iSight, don't you ?!?
    if (! camera)
        abort ();

n= strlen(path) + strlen(name) + strlen(ext);
ALLOC(filename,n+5,char);
```

```
while (current_frame = cvQueryFrame (camera)){

snprintf(filename,n+5,"%s%s%03d%s",path,name,i,ext);
cvSaveImage(filename,current_frame);
cvShowImage (WINDOW_NAME, current_frame);
i++;
int key = cvWaitKey (100);
        if (key == 'q' || key == 'Q')
            break;
}

return 0;

}
```

– Save the main.cpp file and do click on Build and Go bottom in the Project Window.

Now you have your images sequence in /VirtualRetina/ImageCapture/Sequence and then is possible execute the Virtual Retina with your own images sequence as follows :

– Open X11 terminal
– cd ∼ /VirtualRetina/local/bin
– ./Retina /VirtualRetina/ImageCapture/Sequenceimage*.pgm -ret /VirtualRetina/VirtualRetina/test/retina_files/EXAMPLE_primate_Pa
  -r 10 -outD /VirtualRetina/VirtualRetina/tmp/