

Introducing OpenCV for Developers

*Eiichiro MOMMA (Nihon University), Takuya MINAGAWA (Vision & IT Lab)

1. Introduction

Open Source Computer Vision (OpenCV)[1] is a freely available library for real-time computer vision. It has interfaces for C, C++, Python, and will soon support Java running on Mac OS X, Linux, Windows, and Android operating systems. In this paper, we describe features of the library, including its design architecture, technologies currently available, and supporting communities. We also present a demonstration that is based on the technologies of the library.

2. Design Architecture

OpenCV has interfaces for C, C++, and Python. The library was originally developed by Intel for C, and in 2008, its development was transferred to Willow Garage[2]. In Ver. 2.0, its subsequent development resulted in its conversion into a C++ class library. In Ver. 2.2, which was released in 2010, its functionalities were subdivided and modularized. Fig. 1 shows an outline of design architecture and shows the relationship among image and video libraries, graphical user interface (GUI) libraries, computation and processing libraries, and OpenCV modules. The functionalities of the main modules are as follows.

- core:** Definitions of class and templates, and Core functionalities
- imgproc:** Image processing
- highgui:** I/O of images, videos and imaging devices, and GUI
- ml:** Machine Learning
- calib3d:** Camera calibration and 3D reconstruction
- features2d:** Feature extraction and descriptor extraction
- objdetect:** Object detection
- video:** video analysis
- flann:** Clustering and search multi-dimensional spaces
- gpu:** General-Purpose computation on GPU
- legacy:** Codes for compatibility

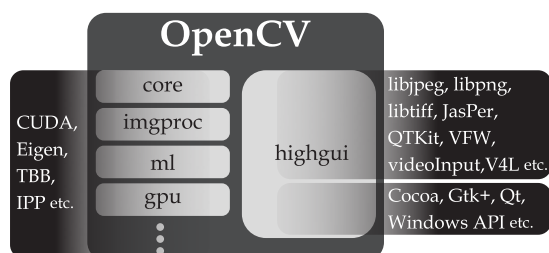


Fig. 1 Design Architecture

In the highgui module, various image formats (e.g. JPEG, PNG, TIFF, JPEG-2000), video codecs and imaging devices (e.g. QTKit, VFW, videoInput, V4L), and GUI frameworks (e.g. Cocoa, Gtk+, Windows API, Qt) were implemented using libraries such as other open source software. Interfaces for images, videos, imaging devices, and GUIs were integrated during the implementation. Therefore, developers can benefit from a combined I/O and GUI without having to consider image formats, video codecs, camera drivers, and different operating systems. For example, we can use `imread()` and `imshow()` to load and display an image, respectively. These operations usually incur the majority of the cost involved in image processing.

Furthermore, the performance improved significantly when the options used are Compute Unified Device Architecture (CUDA)[3], Eigen[4], Intel Threading Building Blocks (Intel TBB)[5], or Intel Integrated Performance Primitives (Intel IPP) [6].

3. Distribution

The distribution method for the library has a source tree that is managed in Subversion (SVN)[7] and a binary installer[8]. The advantage of the installer is that it is available immediately. However, based on the policy of Willow Garage, bugs are fixed only in the source tree. Therefore, stable functions are available to developers through use of the source tree. Fig.2 shows the structure of the source tree. The trunk is slightly unstable, because the latest technologies are always implemented in the trunk first. In contrast, branches are stable on keeping each version's specifications. When developers intend to use the implemented functionalities on a long term, the use of branches is recommended.



Fig. 2 Structure of Source Tree

4. Technologies immediately available

In OpenCV, classic image processing technologies and latest technologies are successfully implemented. In particular, technologies that are similar to robotics have been

implemented. Machine learning, which is indispensable to these technologies, has also been successfully implemented. As a result, `calib3d`, `features2d`, `objdetect`, `video`, and `ml` modules are updated frequently, and new functionalities are added to the trunk. Figs. 3 and 4 are examples of technologies that are immediately available in OpenCV. Fig. 3 is a result of face detection using haar-like features and AdaBoost[9], which increased the popularity of OpenCV. Fig. 4 is a result of image stitching implemented in Ver. 2.3. Features that are common in panned multiple images are extracted, and a panoramic image is created. An omnidirectional image can also be easily created using this functionality.

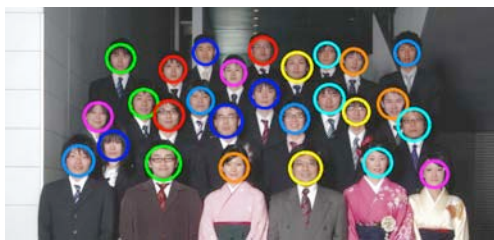


Fig. 3 Face Detection



Fig. 4 Image Stitching

5. Communities

There are active exchanges between OpenCV users. The greatest community of users is a mailing list in Yahoo! GROUPS[10], and topics range from installation to advanced image processing (in English). A community for developers (in English) also exists and is called `opencvlibrary-devel` ML [11]. There is currently no large-scale community in Japan, but some websites contain unified OpenCV information[12-16]. In addition, OpenCV topics are actively discussed on Twitter. Because it is a part of the overall field of computer vision, study sessions involving the CVIM tutorial series[17-19] of the

advanced communication media publication are held monthly throughout Japan[20-22].

6. Conclusion

In this paper, we described various features of OpenCV and the community that supports it. During the presentation of this paper, we plan to show demonstrations, such as "Hello, world" and technologies immediately available in OpenCV, and to share useful tips. We also plan to publicly release source codes and sample images.

REFERENCES

- [1] "OpenCV": <http://opencv.willowgarage.com/wiki/>
- [2] "Willow Garage": <http://www.willowgarage.com/>
- [3] "CUDA": http://www.nvidia.co.jp/object/cuda_home_new_jp.html
- [4] "Eigen": http://eigen.tuxfamily.org/index.php?title=Main_Page
- [5] "TBB Home": <http://threadingbuildingblocks.org/>
- [6] "Integrated Performance Primitives from Intel": <http://threadingbuildingblocks.org/>
- [7] "OpenCV SVN repository": <https://code.ros.org/svn/opencv/>
- [8] "Open Computer Vision Library": <http://sourceforge.net/projects/opencvlibrary/>
- [9] Paul Viola, Michael Jones: "Rapid Object Detection using a Boosted Cascade of Simple Features", *cvpr*, vol. 1, pp.511, 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01) - Volume 1, 2001
- [10] "Yahoo! GROUPS: OpenCV": <http://tech.groups.yahoo.com/group/OpenCV/>
- [11] "Opencvlibrary-devel Info Page": <https://lists.sourceforge.net/lists/listinfo/opencvlibrary-devel>
- [12] "OpenCV.jp": <http://opencv.jp/>
- [13] "ImagingSolution": <http://imaging-solution.net/>
- [14] "Point at infinity": <http://www.atinfinity.info/>
- [15] "Play with OpenCV": <http://playwithopencv.blogspot.com/>
- [16] "Momma's wiki": <http://www.eml.ele.cst.nihon-u.ac.jp/~momma/wiki/>
- [17] R. Kurazume, et al. : "Cutting-edge Guide to Computer Vision 1 -CVIM Tutrial Series-", Advanced Communication Media, 2008.
- [18] H. Fujiyoshi, et al. : "Cutting-edge Guide to Computer Vision 2 -CVIM Tutrial Series-", Advanced Communication Media, 2010.
- [19] T. Okatani, et al. : "Cutting-edge Guide to Computer Vision 3 -CVIM Tutrial Series-", Advanced Communication Media, 2010.
- [20] "Computer Vision Study Session in Kanto": <http://sites.google.com/site/cvsaisentan/>
- [21] "Kansai CV and PRML Study Session": <http://groups.google.com/group/cvprml>
- [22] "Nagoya CV and PRML Study Session": <http://sites.google.com/site/nagoyacv/>