Uniform Software Environment for AR Performance Evaluation based on USB boot Linux

Yoshinari KAMEDA, Itaru KITAHARA, and Yuichi OHTA
Graduate School of Systems and Information Engineering, University of Tsukuba
kameda@iit.tsukuba.ac.jp

Abstract

This paper proposes a new software environment on which performance of different AR approaches can be uniformly evaluated. The proposed environment is provided in a bootable USB of Linux system. It is rather easy to install and/or develop AR software and libraries on the environment since the usability of the system is same as normal Linux system. Once prepared, whole software set-up of the system can be re-distributed with simple procedure. Therefore, the evaluation could be done uniformly at any places if only they can prepare the same hardware.

We call this project "casper cartridge" and it has been released since 2010. We announce the latest version that includes major software tools and libraries, and demostrate the camera calibration evaluation of PTAMM with the TRAKMARK dataset.

1. Introduction

In augmented reality literature, system performance is one of the key elements because AR technologies should be evaluated by users in general. As system performance is a result of integration of computer hardware and software, it could be realized and reproduced if adopted hardware is listed and software (driver, OS, libraries, and application program) are shown. Unfortunaetely, it is not easy to make it successful these days and comparison can not be made fairly.

The hardware problem could be solved if the detailed list of the adopted products is revealed.

The software problem, however, becomes serious because of sophistcated on-line update system. In case of Linux system, for example, one typical linux system just after installation already has 1,420 software packages and 704 run-time shared libraries. Then by updating packages and installing necessary softwares, number of the installed packages reaches almost 2,000 and installed shared libraries will be double. A detailed list of all the kernel/OS patches, installed packages, and libraries and the order of procedure could help, but some packages and/or softwares may not be accesible later on on-line update system. The ultimate solution is to freeze the whole software environment, including not merely newly installed softwares, but packages, drivers, and operating system itself too.

One close solution for the software problem is to make a distribution by reseachers themselves. Once a distribution is made, anyone can have same software environment. However, it is not a easy task and it needs some preparation.

In this paper, we propose to provide a new software environment on which researchers can easily setup and demonstrate their program. Then the whole software environment is to be instantly frozen so that the frozen environment is ready to be re-distributed to others.

We call this software environment "casper cartridge," which has been firstly announced in Sep. 2009. It is based on the Ubuntu linux system of long term support (LTS) version. This papar introduces the latest version of casper cartridge and describes its advantage. We take up some advanced AR applications and show they could run on the casper cartridge. We also demonstrate an evaluation of 3D camera calibration based on the TRAKMARK dataset.

2. Features of Casper Cartridge

Casper cartrige is actually a linux system booted from USB stick and it adopts Union file system (UnionFS)[6]. It hence inherits the features of bootable USB stick. In this section these features are shown to be good advantages on AR research and development.

(1) Direct control to hardware
Unlike other virtual PC, the kernel directly runs over the hardware. There is no overhead to control the hardware. The only disadvantage is the access speed to the

file system which resides on USB stick. Since AR application may not need big access to disk area, it will not be a serious problem.

(2) Complete software set
As casper cartridge provides all the software matters (same as CDROM boot), it can guarantee the software part is completely same at its first boot. This means that a reproduced system will show exactly the same performance if only they prepare the same hardware as the original system. In other words, if there is performance difference, it should come from the hardware difference.

(3) Read-write access to file system
Different from live CD-ROM boot, users have a writable root file system. Any files in the root file system can be created, changed, and deleted without any limitation. These changes are maintained on the next boot too.

(4) Snapshot for back-up and/or re-distribution
While the system runs, all the file operations to the root file system are recorded in a single file (named "casper-rw") on USB stick. After the system is shut down, this casper-rw becomes a snapshot at this moment. By distributing casper-rw to others, they can receive the same software setup. Copying the casper-rw actually means making a back-up. The copied casper-rw could be redistributed easily because it is just a single file.

3. System Overview

The linux operating system is selected for casper cartridge project because the project will re-distribute the OS itself. Among linux distributions, Ubuntu 12.04 long term support version is targeted because it is announced to be supported until Apr. 2017.

3.1. Bootable USB stick

A USB stick is prepared on Windows. Though a special software[1] is needed, files on the USB stick are accessible from Windows.

It has three important components from user view of casper cartridge.

1. boot loader
2. **casper-FS**: read-only file system
3. **casper-rw**: read-write file system

These three components should be consistent. For example, if the boot loader is installed to call Ubuntu 12.04.1/64bit, casper-FS and casper-rw should be the ones for that distribution. Casper-FS corresponds to a set of a few files when the stick is mounted on Windows, and it will correspond to one read-only file system once the linux starts. Casper-rw is a single file on Windows, and it is invisible on linux because it is unified with casper-FS.

On linux, both casper-FS and casper-rw form the root file system by union file system technology [6]. Users can create/modify/delete files arbitrarily. All the changes are saved in casper-rw, and casper-FS is static. It means that once casper-rw is removed (or renamed) before booting, it always falls back to the original setting. This is useful to check the difference between the original behavior and the modified one.

3.2. casper-FS

When the USB stick is made to be bootable, it has default read-only file system that comes from original Ubuntu distribution.

Version 5 of casper cartridge basically provides casper-FS and it will replace the original read-only file system. The corresponding files on Windows are found at "\casper \filesystem.squashfs" (and three associated files of which the file name start with "filesystem").

The size of the filesystem.squashfs varies from about 700MB to 4GB (upper limit of Windows file system on USB stick).

3.3. casper-rw

When the USB stick is made to be bootable, "casper-rw" should be placed at "\casper-rw" on Windows. When the kernel finds it on booting, the unionFS is activated to record file manipulations. Otherwise, all the file manipulations will be abandoned on shutting down the system.

At this moment, major cartridges of version 5 do not provide casper-rw. The original (clean) casper-rw can be prepared and it can be used fully by users. The size of casper-rw depends on the free area of the USB stick. If the USB stick is 16GB or more[2], casper-rw could be set to its maximum size of 4GB.

4. Content of Casper Cartridge

Version 5 is the latest casper cartridges[3]. Each cartridge is named by three digits, and the first letter is 5.

Cartridges are designed to support some different software setup, but basically any cartridge comes with suffient software environment for AR/MR development.

[1] Universal USB installer, http://goo.gl/zxujo

[2] boot loader section usually needs about 100MB

[3] http://www.kameda-lab.org/casper

We take up Ver.514 here to introduce the content.

4.1. Libraries and Softwares

The libraries marked by [c] indicates that it is compiled from source tree on this casper cartridge. This means users can have the libraries together with sample sources and its binaries in the casper cartridge environment. The number after slash in the list indicates the version.

- Based on Ubuntu / 12.04.1 LTS (64bit)
- Point Cloud Library / 1.6.0 [c]
- OpenNI / 1.5.4.0 [c]
- driver for Microsoft XBox360 Kinect
- driver for PrimeSense/ASUS Xtion PRO LIVE
- NITE for OpenNI / 1.5.2.21 (Figure. 1)
- OpenCV / 2.4.2 [c]
  (with FFMPEG, Gstreamer, OpenNI, V4L, QT, QT/OpenGL, TBB, Eigen, Python support) (Figure. 2, Figure 3)

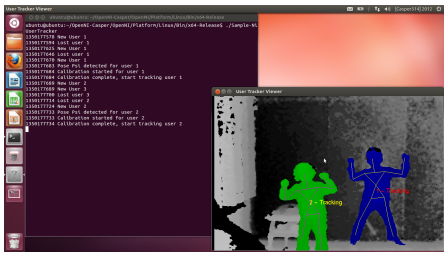

Figure 1. OpenNI: Sample-NiUserTracker

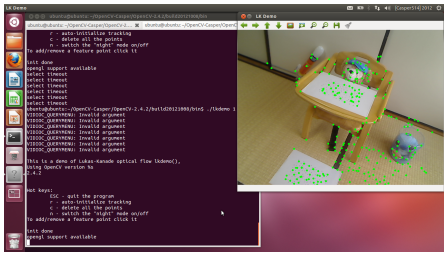

Figure 2. OpenCV: lkdemo

Some integrated development environments are also prepared to help programming and analysis.

- eclipse for C developper / JUNO
- processing / 2.0-beta-3
- libsvm / 3.12, liblinear / 1.91, R / 2.14.1, gnuplot 4.4 patchlevel 3

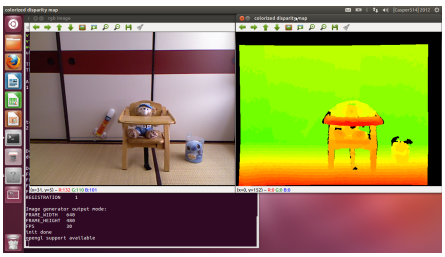

Figure 3. OpenCV: openni_capture

4.2. AR Application

The casper cartridge has some major pre-compiled AR application to show its applicability.

- ARToolKitPlus / 2.3.0 [c]
- Deformable Random Dot Markers (UCHIYAMARKERS2.0) [c] [5] (Figure.4)
- Parallel Tracking and Multiple Mapping (PTAMM) [c] [1] (Figure.5)

ARToolKitPlus, which is an alternative implementation of original ARToolKit[2], has been installed in the cartridge.

As for an example of advanced planner marker tracker, deformable random dot marker (DRDM)[5] has been installed. Original source of the DRDM was provided in windows environment, but it is rather straightfoward porting to the casper cartridge.

So-called PTAM[3] and PTAMM[1] is thought to be one of the most practical on-line 3D camera tracking method up to date. The source code of PTAMM[4] is open for evaluation, but the original code does not support wide variety of camera input. Hence it is ported to show the casper cartridge is a good environment for AR/MR use.

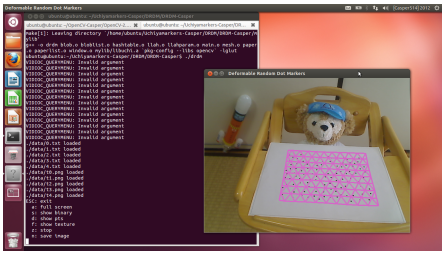

Figure 4. DRDM

[4]PTAM and PTAMM shares the same code for camera tracking part.

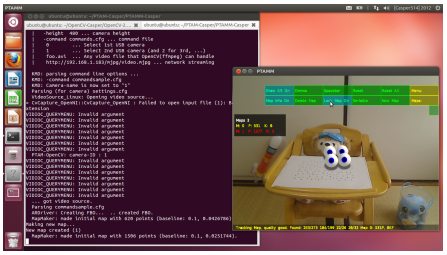

Figure 5. PTAMM with OpenCV

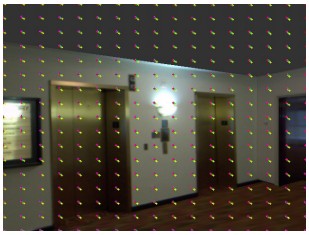

Figure 7. Screen Snapshot of 3D Camera Calibration Evaluation

4.3. Camera Tracking Evaluation on TRAKMARK dataset

A video data set for Evaluation of 3D camera calibration methods is provided as TRAKMARK[4]. As it provides internal and external camera parameters, researchers can evaluate how good a camera tracking method is.

One of our collaborators is working on defining new evaluation criteria for AR use, and the software development is being done by adding codes to PTAMM on the casper cartridge (Figure 6 and Figure 7).

Though the original TRAKMARK data set is provided by Blue-Ray Disc, we have prepared on-line storage of the TRAKMARK dataset and the program directly reads image sequence by http connection.

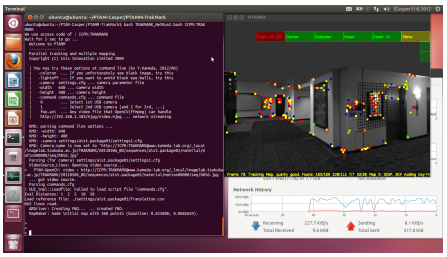

Figure 6. 3D Camera Calibration Evaluation of PTAMM on TRAKMARK Data set

5. Conclusion

We have proposed casper cartridge project; a new software environment for AR performance evaluation on bootable USB stick of Ubuntu Linux.

By providing complete software environment (from drivers and OS to application), any researchers can reproduce the AR system that has same performance of the original one if only they prepare same hardware.

In addition, the casper cartridge is appropriate for preliminary AR software development. It will be also good for education because some recent AR applications are ready to run from the beginning.

Note that the casper cartridge is designated to see the evaluation and demonstration only. As the softwares installed on the casper cartridge have their own copyright notice, users should read and follow them before using them.

Acknowledgment

Part of this research is supported by grant-in-aid for scientific research (B)(23300064) of Japanese government.

References

[1] R. O. Castle, G. Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *International Symposium on Wearable Computers (ISWC)*, pages 15–22, September 2008.

[2] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. *International Workshop on Augmented Reality (IWAR)*, pages 85–94, October 1999.

[3] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234, November 2007.

[4] H. Tamura, H. Kato, and T. W. Group. Proposal of international voluntary activities on establishing benchmark test schemes for ar/mr geometric registration and tracking methods. *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 233–236, 2009.

[5] H. Uchiyama and E. Marchand. Deformable random dot markers. *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 237–238, October 2011.

[6] C. P. Wright, J. Dave, P. Gupta, H. Krishnan, E. Zodok, and M. N. Zubair. Versatility and Unix Semantics in a Fan-Out Unification File System. Technical Report FSL-04-01b, Computer Science Department, Stony Brook University, October 2004.