

# USBブータブルなAR/MRプログラム開発・ 検証環境(Casper Cartridge Project)

AR/MR Program Development and Verification Environment  
on USB Boot Linux

亀田能成

## 1. はじめに

拡張現実感 (AR: Augmented Reality) や複合現実感 (MR: Mixed Reality) は、未来を感じさせる新たなユーザインタフェースとして、現在多くの方々の注目を集めている。

しかしながら、研究成果は面白そうであるにもかかわらず、実際に AR/MR の研究を始めようとする、初学者は敷居の高さに少なからず面食らうことになる。

そこで、本稿では、筆者らが進めている Casper Cartridge Project<sup>(1)</sup>を紹介する。本プロジェクトでは、USB メモリを用いた Linux ブート環境を用意し、この分野について予備知識がない者でも、既存の PC 環境に手を加えることなく、AR/MR のデモンストレーションを実際に味わってみることができる。

本プロジェクトは元々 AR 研究成果の公平な比較検証を意図して開始されたが、初学者が単純にデモを楽しむ環境としても有用なので、少しでも興味がおありの方は是非一度試してみたい。

なお、本稿では、Casper Cartridge の試行に必要なもの、その上で試行できるプログラム群について述べ、実際の作業手順については誌面の都合上詳細を示さない。作業手順については、Project の Web ページ<sup>(注1)</sup>を参照されたい。

## 2. 準備

Casper Cartridge は Ubuntu Desktop Linux の ver. 12.4 (Long Time Support) をベースにしている。Cas-

per Cartridge は現在 ver. 5 である。本稿で紹介するのはそのうちの 64 bit OS をベースにした ver. 5.30 である。

新規に必要な USB メモリは、試行するだけであれば 8 GByte、ユーザの作業記録まで残したいのであれば 16 GByte が必要であり、Windows 上で作業をしてこの USB メモリ上に必要なファイル群を書き込む。

Casper を実行する PC として、Intel CPU 等が搭載されたいわゆる Windows パソコンが必要である。BIOS が USB ブートに対応していることが必要である<sup>(注2)</sup>。Casper は内蔵 HDD を使用しないので、Windows 環境が崩れることはない。

## 3. 試行可能なプログラム

### 3.1 ソフトウェアライブラリ

Casper Cartridge では以下のソフトウェアやライブラリは apt-get を用いずに直接ソースコードをコンパイルする形で用意している。

- OpenCV (Online 版)<sup>(注3)</sup>
- Point Cloud Library (PCL) (Online 版)<sup>(注4)</sup>
- OpenNI (ver. 1.5)

これらのライブラリでは、初めからサンプルプログラムが多数用意されている。特に、OpenCV と PCL ではそれぞれ 217 本<sup>(注5)</sup>と 148 本の C/C++ のサンプルプログラムが用意されていて、すぐに実行することができる。OpenCV ライブラリがほかのライブラリと連携す

亀田能成 正員：シニア会員 筑波大学大学院システム情報工学研究科知能機能システム専攻  
E-mail kameda@iit.tsukuba.ac.jp  
Yoshinari KAMEDA, Senior Member (Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba-shi, 305-8573 Japan).  
電子情報通信学会誌 Vol.96 No.7 pp.512-515 2013 年 7 月  
©電子情報通信学会 2013

(注 1) <http://www.kameda-lab.org/casper/index-j.html>

(注 2) ここ数年の WindowsPC であればおおよそ対応する。

(注 3) 2013/2 以降の版。

(注 4) 2013/2 以降の版。

(注 5) これと別に Python で OpenCV を実行するサンプルプログラムも 30 本以上用意されている。

るよう、コンパイルオプションを調整してある。例えば、図 1 は OpenCV から OpenNI を呼び出し、Xbox360 Kinect カメラで撮影している様子である。また、図 2 は USB カメラをつないで特徴点トラッキング

を実行した様子を示している。

PCL サンプルプログラムのうち 27 本では OpenNI と連携して、リアルタイムに Kinect からの Depth 画像を Point Clouds にして実時間処理を行う例となっている。最も単純な例を図 3 に示す。

もちろん、OpenNI から提供されているサンプルプログラムを実行することもできる。中間ライブラリ NITE と組み合わせて実行した例を図 4 に示す。

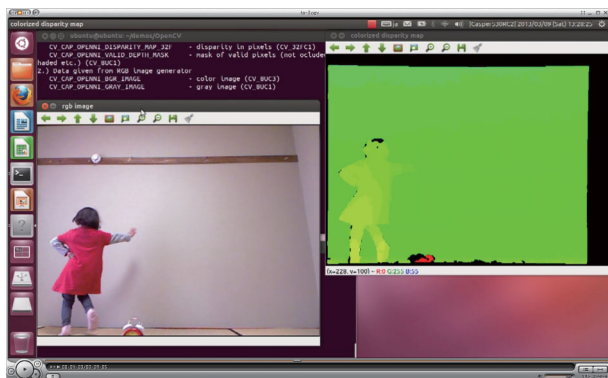


図 1 OpenCV サンプルプログラム “cpp-examples-openni\_capture”. 右側の Depth 画像の方はプログラム内で色変換している。Qt ライブラリとも連携しているので、プログラム本体に何も記述しなくとも画像の拡大や保存などの機能が UI から利用可能になっている（各ウィンドウ上部の白帯に並ぶアイコン）。

### 3.2 AR/MR アプリケーション

AR/MR アプリケーションとして用意してあるプログラムは、いずれもソースからコンパイルした様子を全て Casper Cartridge 上に保存してあるので、ソースをのぞいたり自分の手で変更してみることも簡単である。

図 5 は Kato らによる ARToolKit<sup>(2)</sup>の実行の様子である。こうした既知のマークを利用する方法の発展の一例として、Uchiyama らの Deformable random dot markers (DRDM)<sup>(3)</sup>の実行の様子を図 6 に示す。

一方で、マークを事前に用意せず、自然特徴をマークとしてカメラトラッキングを実現する方法として、

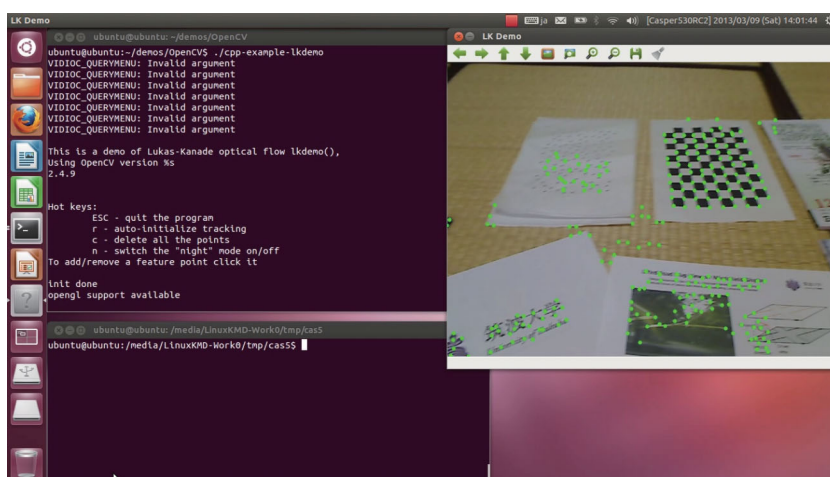


図 2 OpenCV サンプルプログラム “cpp-examples-lkdemo”. 緑色の点が追跡されている特徴点。

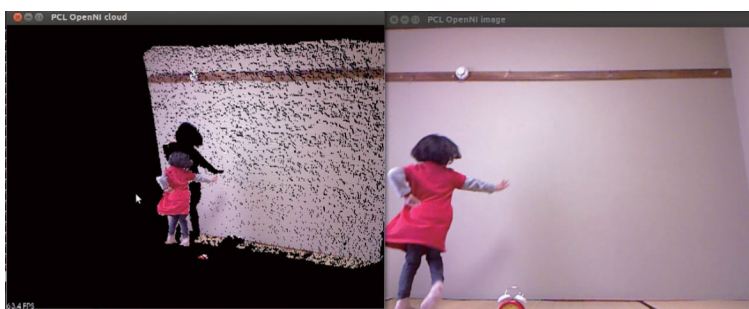


図 3 PCL サンプルプログラム “pcl\_openni\_viewer”. 左側は、Depth 画像を Point Clouds にしている様子を、少し視点を変えて見ている。右側は同時に取得されているカラー画像。

PTAM<sup>(4)</sup>という手法が提案されている。Casper ではこれを OpenCV ライブラリ経由で映像入力できるようにした改良版を利用できる。図7に実行の様子を示す。

更に、PTAM の改良版として、PTAM の複数地点版である PTAMM<sup>(5)</sup>や、エッジ情報を利用した PTAM<sup>(6)</sup>の実装<sup>(7)</sup>についても、ソースからコンパイルして作成した実行バイナリを試すことができる。

### 3.3 他のライブラリ<sup>(注6)</sup>や開発環境

Casper Cartridge ver. 5.30 では、並列化のため TBB など用意され、C/C++ 以外の言語として Java や python, ruby, R など初めから導入されている。

また、開発支援として、emacs のほかにも統合開発環境 eclipse (C/C++ 言語用) や meld が用意されている。

ちょっと変わったところでは、デスクトップ全体のリアルタイムキャプチャも用意してある。本稿のスナップショットも全て一旦デスクトップビデオを作成し、そこ

から適宜フレームを取り出して掲載している。

## 4. おわりに

Casper Cartridge は AR/MR 技術の今を実感できるよい機会を提供できると考えている。本プロジェクトにより、この分野に取り組む研究者や学生が一人でも増えることを期待する。なお、本稿の実行の様子は、プロ



図4 OpenNI サンプルプログラム “Sample\_NiUserTracker”. 2体の骨格の追跡が行われている。

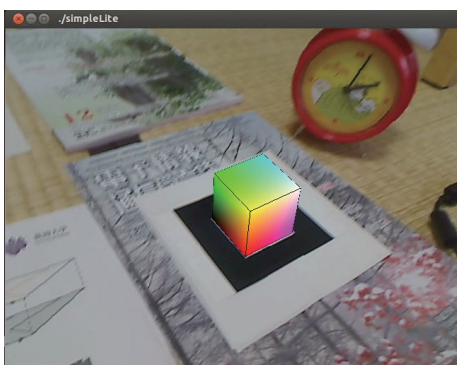
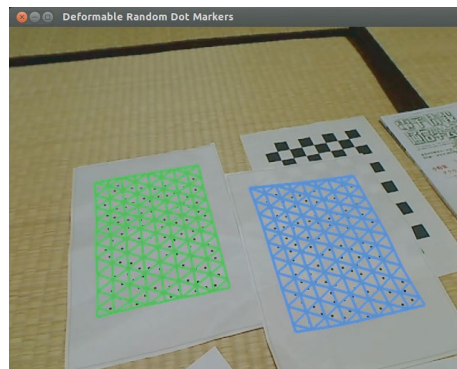
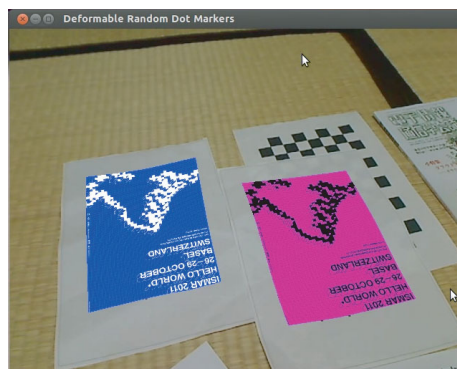


図5 ARToolKit プログラム “simpleLite”.



(a)



(b)

図6 DRDM プログラム (a)は追跡の様子。(b)はテクスチャを重ねた状況。



図7 PTAM プログラム カメラ視野中に広がって存在する自然特徴群を用いるので手がカメラ視野に入っても仮想物体は実世界に対して安定して表示され続ける。

(注6) 一定の手順を踏めば、Casper ver. 5.30 は Nvidia グラフィックカードを CUDA (ver. 5) で利用できる。詳細は Project Web ページ参照。

ジェクトページから映像で視聴可能にする予定である。

本プロジェクトの一部は、日本学術振興会基盤研究(B)「環境カメラ群映像の安心かつ効率的見える化の為の時空間解析と複合現実感的可視化」(課題番号23300064)の支援を受けて行われている。ここに謝意を表す。

なお、本稿中に現れる登録商標はそれぞれの所有者のものである。

## 文 献

- (1) Y. Kameda, I. Kitahara, and Y. Ohta, "Uniform software environment for AR performance evaluation based on usb boot linux," International Workshop on Benchmark Test Schemes for AR/MR Geometric Registration and Tracking Method (TrakMark) (in conjunction with ICPR2012), 2012.
- (2) H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," International Workshop on Augmented Reality, pp. 85-94, 1999.
- (3) H. Uchiyama and E. Marchand, "Deformable random dot markers,"

International Symposium on Mixed and Augmented Reality, pp. 237-238, 2011.

- (4) G. Klein and D. Murray, "Parallel tracking and mapping for small {AR} workspaces," International Symposium on Mixed and Augmented Reality, pp. 225-234, 2007.
- (5) R. Castle, G. Klein, and D. Murray, "Video-rate localization in multiple maps for wearable augmented Reality," International Symposium on Wearable Computers, pp. 15-22, 2008.
- (6) G. Klein and D. Murray, "Improving the agility of keyframe-based SLAM," European Conference on Computer Vision, pp. 802-815, 2008.
- (7) R. Bracha and E.B. Simhon.  
<http://www.youtube.com/watch?v=epm4ntinrH0&feature=share>

(平成 25 年 3 月 12 日受付)



かめだ よしなり  
亀田 能成 (正員：シニア会員)

平 3 京大・工・情報卒, 平 8 同大学院博士課程研究指導認定退学. 同年京大助手. 以来, 知的画像処理, マルチメディア処理解, 複合現実感等の研究に従事. 現在, 筑波大学院システム情報工学研究科准教授. 京大博士 (工学).