

facebook

Surround 360 Build

24.04.16

(https://docs.google.com/presentation/d/18VUOHLNWw_buBJLpbrPLrP22hy3eJLY4DxQ_6CnOGyU/edit?usp=drive_link)

Prerequisites

[우분투 OS 설치]

Surround360 오픈소스는 Ubuntu 16.04 64-bit 를 공식 지원하므로 Ubuntu 16.04 OS 를 설치해줘야 한다(surround360_camera_ctl_ui\README.md 참고).

- Ubuntu 16.04 64-bit 설치
<https://mirror.kakao.com/ubuntu-releases/xenial/> 에서 “64-bit PC (AMD64) desktop image” 링크를 다운로드 (ubuntu-16.04.7-desktop-amd64.iso) 후 설치 (AMD64 는 x86의 64비트 확장 아키텍처 (x86-64, x64, EM64T, Intel64 로도 불림))
- 설치된 우분투 버전 확인
`$ lsb_release -a | grep Desc`
Description: Ubuntu 16.04.7 LTS

[네트워크 설정]

우분투 설치시 기본적으로 유선랜이 잡혀야 하지만, RTX8000 달린 서버의 경우, 메인보드의 유선랜이 잡히질 않고 있음 (ifconfig 명령어시 lo 만 표시됨)

이 경우, 별도의 랜포트 어댑터를 연결할 경우, 유선랜이 인식됨 (e.g. ITime 또는 델노트북 어댑터)

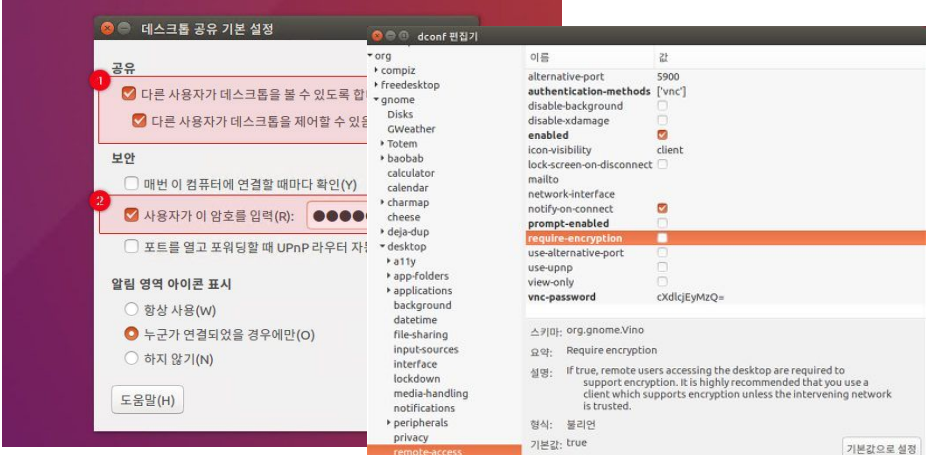
[원격 접속 설정]

윈도우에서 접속을 위해 <https://www.realvnc.com/en/connect/download/combined/> 에서 RealVNC® Viewer for Desktop 를 다운로드 (Server 버전은 유료임) (https://drive.google.com/file/d/1JAB5raVX58zmhgdyepsE1b9aSTHbrNel8/view?usp=drive_link)

우분투 설치하자 마자 <https://baekjakka.tistory.com/17> 그대로 따라하면 원격접속 성공함! (enabled 체크, prompt-enabled / require-encryption 체크 해제하고 테스트했음)
우분투에서 로그인한 상태여야 원격 접속이 성공함! 편의를 위해 설정에서 자동 로그인을 ON --> 재부팅후 ip 주소가 바뀌어서 원격 접속이 안되는 경우가 있으니 ip 주소 확인이 필요!
\$ netstat -tlnp 로 확인시 5900 포트 listening 중이어야 함
Desktop 화면에서 데스크톱 공유 앱을 오픈해 (des 입력), 공유 항목에 체크, 지시와 다르게 데스크톱 앱에서 “사용자가 이 암호를 입력” 항목을 체크 해제함 (상관없는 듯)

추가로 버그로 인해 VNC 암호화 연결 기능 비활성화 설정이 필요함!!!
`sudo apt install dconf-editor` 설치 후, Desktop 화면에서 dconf 앱을 검색 & 실행 후, org > gnome > desktop > remote-access 화면에서 require-encryption 항목을 체크 해제 해주어야 RealVNC Viewer 앱에서 접속에 성공하고 있음(require-encryption 이 켜져 있으면 타임아웃 에러)

*RDP 설치가 필요함. <https://c-norgy.be/blog/?p=15978>
wget <https://www.c-norgy.be/download/xrdp-installer-1.2.2.zip>
unzip xrdp-installer-1.2.2.zip && chmod +x xrdp-installer-1.2.2.sh && ./xrdp-installer-1.2.2.sh
삭제시는 -f 옵션을 추가, needs reboot!!!



Prerequisites

[NVIDIA 드라이버 설치]

// RTX8000 서버 (172.16.151.134)
공식 www.nvidia.co.kr/download 에서 Quadro RTX 8000 을 검색한 결과인
Linux X64 (AMD64/EM64T) Display Driver 버전: 550.67 페이지에서 다운로드 버튼을 클릭해
NVIDIA-Linux-x86_64-550.67.run 를 다운로드 후, chmod +x ~ 로 실행 권한 부여

Ctrl + Alt + F1 로(X server 탈출) virtual terminal (tty) 에 진입 후,
\$ sudo service lightdm stop 를 해주고
\$ sudo ./NVIDIA-Linux-x86_64-550.67.run 로 드라이버 설치

The distribution-provided pre-install script failed! Are you sure you want to continue?
--> 모니터 케이블을 NVIDIA Display Port 에 연결하지 않고 메인보드의 HDMI 포트에 연결시 이
에러가 발생함!!! (HDMI 에 연결한 경우, 화면이 계속 깜박이는 현상이 있음)

[git clone]

\$ sudo apt-get install git // git 설치
\$ mkdir ~/cubox && cd ~/cubox
\$ git clone [git@github.com](https://github.com/CUBOX-Co-Ltd/Surround360.git):CUBOX-Co-Ltd/Surround360.git

--> Permission denied (publickey) 에러 발생시
ssh-keygen -t ecdsa -b 521 -C "skc0833@cubox.ai" 로 생성 후,
cat ~/.ssh/id_ecdsa.pub 내용을 github 우상단 아이콘 / Settings 클릭, 좌측 SSH and GPG Keys
클릭 후, 좌상단에 New SSH Key 버튼을 클릭해 등록

ssh 키 생성시 ssh-keygen 입력 후 그냥 엔터만 쳐서 ~/.ssh/id_rsa, id_rsa.pub 를 생성 후,
id_rsa.pub 를 SSH Key 에 등록하면 git push -u origin main 시도시
ERROR: You're using an RSA key with SHA-1, which is no longer allowed. 에러가 발생하게됨
따라서, 위처럼 ssh-keygen -t ecdsa -b 521 -C "skc0833@cubox.ai" 로 SSH Key 를 생성해줘야
함

[Optional] vscode 설치
<https://code.visualstudio.com/> 에서 deb 파일 다운로드 & 설치
sudo dpkg -i code_1.88.0-1712152114_amd64.deb

[Optional] chrome browser 설치
sudo dpkg -i google-chrome-stable_current_amd64.deb
1) dpkg: dependency problems prevent configuration of google-chrome-stable:
google-chrome-stable 패키지는 다음 패키지에 의존: libnss3 (>= 2:3.35): 하지만:
시스템에 있는 libnss3:amd64의 버전은 2:3.28.4-0ubuntu0.16.04.12입니다.
--> wget
http://security.ubuntu.com/ubuntu/pool/main/n/nss/libnss3_3.35-2ubuntu2.16_amd64.deb
다운로드 후, sudo dpkg -i libnss3_3.35-2ubuntu2.16_amd64.deb
from
<https://support.google.com/chrome/thread/210446256/latest-version-of-chrome-requires-a-dependency-that-is-not-available?hl=en>

2) Retry sudo dpkg -i google-chrome-stable_current_amd64.deb
dpkg: dependency problems prevent configuration of google-chrome-stable:
google-chrome-stable 패키지는 다음 패키지에 의존: libu2f-udev: 하지만:
libu2f-udev 패키지는 설치하지 않았습니다.
--> wget http://archive.ubuntu.com/ubuntu/pool/main/libu/libu2f-host/libu2f-udev_1.1.4-1_all.deb
sudo dpkg -i libu2f-udev_1.1.4-1_all.deb

3) Retry sudo dpkg -i google-chrome-stable_current_amd64.deb
google-chrome-stable 패키지는 다음 패키지에 의존: libvulkan1: 하지만:
libvulkan1 패키지는 설치하지 않았습니다.
--> sudo apt-get -f install
&& retry OK!!!

[Optional] SSH Server 설치
sudo service ssh status // Loaded: not-found 이면 미설치 상태임
dpkg -l | grep openssh
sudo apt-get install openssh-server
service --status-all | grep + // 현재 실행중인 전체 서비스 목록
sudo apt-get install ssh // 클라이언트 설치(불필요)
sudo gedit /etc/ssh/sshd_config // config 파일 수정(불필요)
service ssh restart or start // config 수정시 재시작

Build 환경 구축

기본적으로 Surround360 소스 내의 README.md 를 따라서 빌드환경을 구축했으며 , 이를 기반으로 필요한 빌드 절차 및 추가 내용을 아래 정리함

github.com/facebookarchive/Surround360 의 commit 로그 확인시 Surround360 개발환경 구성 시점은 2017년중순~2018년초 정도로 보이므로 , 기타 오픈소스 빌드시에 해당 시점을 소스를 사용했음 (e.g, Install Halide 시점은 Mar 15 2017)

```
$ gcc --version      // g++ --version 도 동일한 버전임
gcc (Ubuntu 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609

$ python --version   Python 2.7.12
$ which python       /usr/bin/python
/usr/bin/python -> python2.7*           // python3.5 도 존재함
```

[README.md]

카메라 빌드없이 rendering software 를 아래 샘플 데이터셋으로 테스트가 가능하며 , 이 샘플들은 소스 빌드 완료후 테스트 예정
"Palace of Fine Arts Take 1" - 2 frames - (337.4MB)
"Facebook Building 20" - 190 frames - (21.15GB)

[surround360_camera_ctl_ui\README.md]

아래처럼 surround360_render 를 먼저 빌드하라고 가이드하고 있으므로 , surround360_render 빌드를 먼저 진행함

Compile the Surround 360 Rendering Software first. The Camera Control Software uses its ISP. Follow the steps in surround360_render/README.md (compile to use accelerated ISP)

[surround360_render\README.md]

surround360_camera_ctl_ui 에서 accelerated ISP 사용을 위해 Surround 360 Rendering Software 를(surround360_render) 먼저 빌드해줘야 함

Installing Dependencies of the Surround 360 Rendering Software

- Install CMake (method 2 - Linux only)
sudo apt-get install cmake // 아래 명령어 대신 이렇게만 해도 3.5.1 버전이 설치됨
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:george-edison55/cmake-3.x
sudo apt-get update
sudo apt-get install cmake && sudo apt-get upgrade cmake
- Install Git (method 1 - Linux only) // 이미 설치했음
git --version git version 2.7.4 // default 버전
- Install Subversion (method 1 - Linux only) // 불필요함
- Install Python (method 1 - Linux only):
sudo apt-get install python python-dev
python --version Python 2.7.12 // default 버전과 동일
- Install gflags (method 1 - Linux only):
sudo apt-get install libgflags2v5 libgflags-dev
- Install glog (method 1 - Linux only):
sudo apt-get install libgoogle-glog-dev

Build 환경 구축

- Install folly (method 1):

see <https://github.com/facebook/folly>

1) mkdir ~/cubox && cd ~/cubox 후에
git clone <https://github.com/facebook/folly>

2) folly main branch 를 README 가이드대로 빌드시 print(f"I don't ~") 에서 아래
에러가 발생하므로
cd folly && sudo ./build/fbcode_builder/getdeps.py install-system-deps --recursive
File "./build/fbcode_builder/getdeps.py", line 425
f"I don't know how to install any packages on this system {host_tuple}"

3) v2017.12.15.00 버전으로 시도
git checkout v2017.12.25.00
folly/README 에 따라서 googletest-release-1.8.0 를 다운로드 및 복사
(BOOST 는 없는 상태이므로 export LIBRARY_PATH=\$BOOST_ROOT/ 는 수행안함)
Ubuntu 13.10 의 필요 패키지들 추후 필요하므로 설치(e.g, automake, autoconf, ...)

libiberty-dev, double-conversion.git 는 불필요해 보이므로 skip 함
Download and install folly with the parameters listed below: 에 따라서 folly 를 빌드 &
설치

- Install Ceres (method 1 - Linux only):
see <http://ceres-solver.org/installation.html>

```
sudo apt-get install libatlas-base-dev && \  
sudo apt-get install libeigen3-dev && \  
cd ~/cubox && \  
git clone https://ceres-solver.googlecode.com/ceres-solver && \  
cd ceres-solver && \  
git checkout 85c6b5c && \  
mkdir ceres-bin && \  
cd ceres-bin && \  
cmake .. && \  
make -j32 && \  
sudo make install && \  
sudo ln -s /usr/include/eigen3/Eigen /usr/local/include/Eigen
```

- Install numpy (method 1 - Linux only):
sudo apt-get install python-numpy

- Install OpenCV (method 1 - Linux only):

```
cd ~/cubox && \  
git clone https://github.com/Itseez/opencv.git && \  
cd opencv && \  
git checkout f109c01 && \  
cmake -DWITH_IPP=OFF && \  
make && \  
sudo make install
```

Build 환경 구축

- Install COLMAP
see <https://colmap.github.io/install.html>

<https://repology.org/project/colmap/versions> 에서 16.04 바이너리는 누락됐으므로
직접 빌드를 진행해야함
<https://colmap.github.io/install.html#linux> 대로 빌드 진행

```
sudo apt-get install \  
  git \  
  cmake \  
  ninja-build \  
  build-essential \  
  libboost-program-options-dev \  
  libboost-filesystem-dev \  
  libboost-graph-dev \  
  libboost-system-dev \  
  libeigen3-dev \  
  libflann-dev \  
  libfreeimage-dev \  
  libmetis-dev \  
  libgoogle-glog-dev \  
  libgtest-dev \  
  libsqlite3-dev \  
  libglew-dev \  
  qtbase5-dev \  
  libqt5opengl5-dev \  
  libcgald-dev \  
  libceres-dev
```

- main branch 는 cmake .. -GNinja 시에 아래 cmake version 에러 발생
CMake 3.10 or higher is required. You are running version 3.5.1
git checkout 3.0 후 빌드를 진행함

```
git clone https://github.com/colmap/colmap.git  
cd colmap  
mkdir build  
cd build  
cmake .. -GNinja  
ninja  
sudo ninja install
```

- COLMAP 빌드시 cmake .. -GNinja 시에 GLUT 를 못찾는 에러 발생
Could NOT find GLUT (missing: GLUT_glut_LIBRARY GLUT_INCLUDE_DIR)
sudo apt install freeglut3-dev 후 시도하면 해당 에러 사라짐.
그러나 CMakeLists.txt:77 의 find_package(Qt5 REQUIRED) 에서 에러 발생

에러의 원인은 /usr/lib/x86_64-linux-gnu/libEGL.so 의 링크파일인
/usr/lib/x86_64-linux-gnu/mesa-egl/libEGL.so -> libEGL.so.1.0.0 파일이 없어서 에러
/usr/lib/x86_64-linux-gnu/libEGL.so.1 -> libEGL.so.1.1.0 는 존재하므로 아래처럼 이를
대체하면 성공함
sudo rm /usr/lib/x86_64-linux-gnu/libEGL.so
sudo ln /usr/lib/x86_64-linux-gnu/libEGL.so.1 /usr/lib/x86_64-linux-gnu/libEGL.so

libGL.so 도 마찬가지로 처리해줘야함
sudo rm /usr/lib/x86_64-linux-gnu/libGL.so && sudo ln -s /usr/lib/libGL.so.1
/usr/lib/x86_64-linux-gnu/libGL.so

--> /usr/lib/x86_64-linux-gnu/cmake/Qt5Gui/Qt5GuiConfigExtras.cmake:50 디버깅시
message(INFO "@@skc \${Qt5Gui_\${CMAKE_LIB_NAME}_LIBRARY}")
INFO@@skc /usr/lib/x86_64-linux-gnu/libEGL.so 와 libGL.so 로 에러원인 확인했었음

이후 다시 cmake 시 GLUT_Xi_LIBRARY, GLUT_Xmu_LIBRARY NOTFOUND 에러
sudo apt-get install libxmu-dev libxi-dev 후 재시도시 cmake, ninja install 까지 성공!

- Install ffmpeg (method 1):
see <https://trac.ffmpeg.org/wiki/CompilationGuide>

```
$ sudo add-apt-repository ppa:jonathonf/ffmpeg-4 // 불필요  
$ sudo apt-get install ffmpeg  
$ ffmpeg --version  ffmpeg version 2.8.17-0ubuntu0.1
```

Build 환경 구축

- Install Gooley (method 2 - Linux only):

```
아래 명령어는 print 관련 에러가 발생중이니 python 3.8 설치부터 진행을 추천함
sudo apt-get install python-pip && \
sudo pip install --upgrade pip && \
sudo pip install Gooley && \
sudo apt-get install python-wxgtk2.8
```

Command "python setup.py egg_info" failed with error code 1 in
/tmp/pip-build-Qs3PB1/pip/ 에러가 발생중임 (python version 문제 (2.7.11)??)

- python 3.8 설치
디폴트는 2.7, 3.5 이지만, 위 `print(f"asdf")` 에러 해결을 위해 적당한 버전(3.8)을 설치
(<https://dlehqgm10480.tistory.com/8> 참고)

```
wget https://www.python.org/ftp/python/3.8.19/Python-3.8.19.tgz
sudo apt-get install build-essential checkinstall
sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev \
    libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev libffi-dev zlib1g-dev
cd Python-3.8.1
sudo ./configure --enable-optimizations
sudo make altinstall
```

```
$ python3.8 -V      Python 3.8.19
$ sudo update-alternatives --config python
update-alternatives: error: no alternatives for python
```

```
$ sudo update-alternatives --install /usr/bin/python python /usr/local/bin/python3.8 1
$ python -V        Python 3.8.19
이제서야 sudo pip install --upgrade pip 와 sudo pip install Gooley 가 성공!!!
sudo apt-get install python-wxgtk2.8 는 아직도 E: Package 'python-wxgtk2.8' has no
installation candidate 에러 발생 중이며, 이후의 If python-wxgtk2.8 not available (e.g.
Ubuntu 16.04): 항목에 해결책이 제시되고 있음
```

- If python-wxgtk2.8 not available (e.g. Ubuntu 16.04):

```
echo "deb http://archive.ubuntu.com/ubuntu wily main universe" | sudo tee
/etc/apt/sources.list.d/wily-copies.list && \
sudo apt update && \
sudo apt install python-wxgtk2.8 && \
sudo rm /etc/apt/sources.list.d/wily-copies.list && \
sudo apt update
```

sudo apt update 실행시에 아래 에러가 발생하고 있다....

```
Err:9 http://archive.ubuntu.com/ubuntu wily Release
404 Not Found [IP: 185.125.190.36 80]
```

16.04 는 공식 지원이 끊겨서 발생한 에러이며, 다음처럼 강제로 예전 주소로 변경하여 성공
(The wily version is in EOL (End of Life): no longer supported.)

// 강제로 sudo vi /etc/apt/sources.list.d/wily-copies.list 해서 내용을 지우고 아래 내용을 넣고
저장

```
deb http://old-releases.ubuntu.com/ubuntu/ wily main universe restricted multiverse deb-src
http://old-releases.ubuntu.com/ubuntu/ wily main universe restricted multiverse

deb http://old-releases.ubuntu.com/ubuntu/ wily-security main universe restricted multiverse
deb-src http://old-releases.ubuntu.com/ubuntu/ wily-security main universe restricted multiverse

deb http://old-releases.ubuntu.com/ubuntu/ wily-updates main universe restricted multiverse
deb-src http://old-releases.ubuntu.com/ubuntu/ wily-updates main universe restricted multiverse
```

이후 sudo apt update 부터 python-wxgtk2.8 등등 성공

Build 환경 구축

- Install PIL (method 1 - Linux only)
sudo apt-get install python-pil
- Install extra dependencies (Linux only):
sudo apt-get install libtinfo-dev libjpeg-dev
- (to use accelerated ISP) Install LLVM
현재 github 상의 llvm 소스는 예전 방식으로 빌드가 어려워보이니, 새로 소스를
다운받아 빌드를 시도함
<https://releases.llvm.org/> 에서 아래 파일을 Download 후, 압축 해제
LLVM source code (.sig) --> llvm-3.7.0.src.tar.xz
Clang source code (.sig) --> cfe-3.7.0.src.tar.xz

cd ~/cubox/ && mkdir llvm3.7/tools/clang/ 후에
tar -xf llvm-3.7.0.src.tar.xz && mv llvm-3.7.0.src llvm3.7
tar -xf cfe-3.7.0.src.tar.xz && mv cfe-3.7.0.src/* llvm3.7/tools/clang/

cd ~/cubox/llvm3.7 && mkdir build && cd build
cmake -DLLVM_ENABLE_TERMINFO=OFF
-DLLVM_TARGETS_TO_BUILD="X86;ARM;NVPTX;AArch64;Mips;PowerPC"
-DLLVM_ENABLE_ASSERTIONS=ON -DCMAKE_BUILD_TYPE=Release ..
make -j32
<-- 빌드 성공함

export LLVM_CONFIG=\$HOME/cubox/llvm3.7/build/bin/llvm-config && export
CLANG=\$HOME/cubox/llvm3.7/build/bin/clang

Build 환경 구축

- * (to use accelerated ISP) Install Halide (last known version to work from Mar 15 2017)

```
cd ~/cubox
git clone https://github.com/halide/Halide.git 는
--> 아래 git checkout 970f749 시에 error: pathspec '970f749' did not match any file(s)
known to git. 에러가 발생하므로
git clone https://github.com/jingpu/Halide-HLS.git 에서 받도록 함!
```

```
mv Halide-HLS Halide && cd Halide
git checkout 970f749
```

```
mkdir cmake_build
cd cmake_build
cmake -DLLVM_DIR=$HOME/cubox/llvm3.7/build/share/llvm/cmake
-DCMAKE_BUILD_TYPE=Release -DLLVM_VERSION=37
-DWARNINGS_AS_ERRORS=OFF ..
make -j32 // 기존 README 에는 make 항목이 누락됐었음
```

이렇게 빌드를 해주지 않으면 cmake_build/include/Halide.h 가 생성되지 않아
surround360_render 에서 cmake -DCMAKE_BUILD_TYPE=Release
-DHALIDE_DIR=\$HOME/cubox/Halide/cmake_build 후 make 시에 Halide.h 헤더를
못찾는 아래 에러가 발생하게 된다.
/home/cubox/cubox/Surround360/surround360_render/source/camera_isp/Cameralisp
Gen.cpp:10:20: fatal error: Halide.h: No such file or directory

Compiling the Surround 360 Rendering Software

- After installing all of the dependencies as described above, run:
cd ~/cubox/Surround360/surround360_render
cmake -DCMAKE_BUILD_TYPE=Release
make -j32
- (to use accelerated ISP):
cd ~/cubox/Surround360/surround360_render
cmake -DCMAKE_BUILD_TYPE=Release
-DHALIDE_DIR=\$HOME/cubox/Halide/cmake_build
make -j32
- To test that compilation is successful, run:
./bin/TestRenderStereoPanorama --help

Build 환경 구축

[surround360_camera_ctl-ui\README.md]

카메라를 제어하고 로컬 스토리지에 저장하는 Surround 360 Camera Control Software (UI)

Installing Dependencies of the Surround 360 Camera Control Software

- Install dependencies:
sudo apt-get install libgtkmm-3.0-1v5 libgtkmm-3.0-dev libglibmm-2.4-dev libglibmm-2.4-1v5 libgtk-3.0 libglew-dev libglew1.13 zlib1g zlib1g-dev && \\\nsudo apt-get install libgladmm-2.4-1v5 libgtkmm-2.4-1v5 && \\\nsudo apt-get install libgtkmm-2.4-dev libgladmm-2.4-dev libgtkglextmm-x11-1.2-dev && \\\nsudo apt-get install libtinfo-dev
--> 위 명령어가 실패하며 sudo apt-get -f install 를 하라고 표시되면, 이를 수행 후, 다시 시도한다.
- Download Point Grey FlyCapture SDK and Grasshopper 3 USB 3.0 (GS3-U3-41C6C-C) Firmware:
FlyCapture SDK - Linux Ubuntu (64-bit) 를 다운로드
<https://www.flir.com/products/flycapture-sdk/?vertical=machine%20vision&segment=iis> 에서 flycapture2-2.13.3.31-amd64-pkg_Ubuntu16.04.tgz 를 다운로드
로그인은 skc0833@cubox.aero / s12~!

OR wget

https://github.com/ErnestDeiven/flycapture2/blob/master/flycapture2-2.13.3.31-amd64-pkg_Ubuntu16.04.tgz

- Install FlyCapture SDK:
tar -zxvf flycapture2-2.13.3.31-amd64-pkg_Ubuntu16.04.tgz
cd flycapture2-2.13.3.31-amd64
chmod +x install_flycapture.sh
sudo sh install_flycapture.sh
- FlyCapture SDK:설치 후, 재부팅이 반드시 필요하며, RTX8000 서버의 경우, 재부팅시 (후) HDMI 대신 DP 포트를 연결해야 화면 해상도가 제대로 표시됨

- To install extra dependencies when prompted:
sudo apt-get -f install
- /etc/default/grub 파일을 오픈 후,
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash" 를
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash usbcore.usbfs_memory_mb=8000 transparent_hugepage=never" 로 교체
- Then update grub:
sudo update-grub
- Install Grasshopper 3 USB 3.0 (GS3-U3-41C6C-C) Firmware (optional):
이 부분은 필수가 아니므로 skip 함!
- Set up Nvidia proprietary drivers:
이 부분도 이미 설치를 했음 (RTX8000 -> NVIDIA-Linux-x86_64-550.67.run)
- Install FreeGLUT:
sudo apt-get install freeglut3 freeglut3-dev

Build 환경 구축

Compiling the Surround 360 Camera Control Software

- After installing all of the dependencies as described above, run:

```
cd ~/cubox/Surround360/surround360_camera_ctl_ui
cmake -DCMAKE_BUILD_TYPE=Release
-DHALIDE_DIR=$HOME/cubox/Halide/cmake_build
make -j32
```
- To use the user interface, run:

```
./bin/CameraControlUI
```

[참고 사항]

본 가이드에서 설치했던 대부분의 파일들은 아래 구글 드라이브에도 저장돼 있음
(AI-2TEAM / 기타 / Surround360)

<https://drive.google.com/drive/u/0/folders/13WReX9kQx-iZ1KkC0GYT51SUmo3Rem4I>

본 가이드 문서의 구글 닥스 링크

https://docs.google.com/presentation/d/18VUOHLNWw_buBJLpbrPLrP22hy3eJLY4DxQ_6CnOGyU/edit?usp=sharing

[주의 사항]

모든 생성되는 폴더를 ~/cubox/ 하위에 생성했으므로
공식 가이드의 \$HOME 경로 대신에 \$HOME/cubox/ 혹은 ~/cubox/ 로 교체해줘야 함

TODO: 추후에는 공식 가이드처럼 \$HOME 하위에 Surround360 등등 필요한 폴더를 바로 만들자!

[문제 사항]

16.04 는 공식 지원이 끊긴 상태라서

1) 소스 repository 가 사라져 버린 경우가 존재함

(e.g, python-wxgtk2.8 는 apt repository 가 사라져 에러가 발생하므로 강제로 repository 주소를
예전 주소로 변경해줘야 성공함

2) 소스 repository 가 이동된 경우가 존재함

LLVM 빌드시 svn repository 를 github 로 옮기면서 기존 빌드 명령이 실패함 (루트
CMakeLists.txt 가 사라짐). 예전 비슷한 시점의 소스 (llvm-3.7.0.src.tar.xz) 를 찾아 빌드해서
넘어감

3) python 버전 불일치로 파이썬 소스 수정이 필요함

python 소스에서 print "xxx" 로 된 부분을 전부 print("xxx") 로 수정해줘야 함
개발환경 설정에서 Gooye 에서는 print("xxx") 형태를 사용중이므로
python 2.x 버전이 아니, 3.x 대 버전으로 설정해줘야 하므로

4) vscode 도 예전 버전을 사용해야 Open folder 및 디버깅이 가능함
(code_1.64.2-1644445741_amd64.deb)