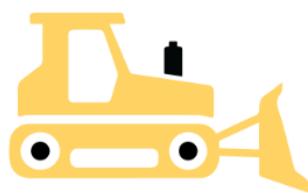


목 차

1. nerfstudio 사용가이드	1
2. setting velodyne LiDAR at ARIL	9
3. cloudsim 설치 및 오류해결	14
4. 개인공부 필기내역(cloudsim,LiDAR,NeRF)	32



nerfstudio 사용기

I) 준비

- nerfstudio에서는 NeRF모델을 구현하는데 필요한 Data Preprocess, Data Loader, Model Training, Visualizing, Rendering을 API형태로 제공함.
- Supported Components
 - [Cameras models](#)
 - [Sample representation](#)
 - [Ray samplers](#)
 - [Spatial distortions](#)
 - [Encoders](#)
- 지원되는 NeRF 모델들
 - [Instant-NGP](#)
 - [Splatfacto](#)
 - [Instruct-NeRF2NeRF](#)
 - [K-Planes](#)
 - [LERF](#)
 - [Mip-NeRF](#)
 - [NeRF](#)
 - [Nerfacto](#)
 - [Nerfbusters](#)
 - [NeRFPlayer](#)
 - [Tetra-NeRF](#)
 - [TensoRF](#)
 - [Generfacto](#)
 - [Instruct-GS2GS](#)
 - [PyNeRF](#)
 - [SeaThru-NeRF](#)
 - [Zip-NeRF](#)

● 준비단계

학습 데이터셋(Own data, 주어지는 데이터 모두 활용 가능), 서버작업시 포트포워딩 필수, CUDA 지원 그래픽카드, 사용 컴퓨터 환경 : Linux 18.04

※ 모든 활동은 conda 가상환경으로 하길 강력히 권장

II) 구성

- Web Viewer

NeRF 모델이 학습하며 모델을 구성하는 과정을 실시간으로 웹으로 확인할 수 있음, 이미지, 영상 뿐만 아니라 포인트클라우드, mesh 도 export 할 수 있으며 마우스와 WASD 를 통한 직관적인 조작 가능.
이를 위해선 서버를 통해 작업할 경우 포트포워딩이 필수.

서버에 배정받은 후 IP 주소와 포트 확인하여 자신의 컴퓨터에 포트포워딩 작업을 해준다.

조회 및 등록

```
sudo iptables -t nat -L --line-numbers  
sudo iptables -t nat -A PREROUTING -p tcp --dport [출발 포트번호] -j REDIRECT --to-port [도착 포트번호]
```

- 초기 세팅

Create Environment

```
conda create --name nerfstudio -y python=3.8  
conda activate nerfstudio  
python -m pip install --upgrade pip
```

Docker 설치

<https://docs.docker.com/get-docker/>

```
docker pull dromni/nerfstudio:0.1.18  
docker run --gpus all -v C:\d\nerfstudio\data:/workspace/ ^  
-v C:\d\nerfstudio\.cache:/home/user/.cahce/ ^  
-p 7007:7007 -p 3389:3389 ^  
--name nerfstudio -it dromni/nerfstudio:0.1.18
```

CUDA 설치

https://developer.nvidia.com/cuda-11-8-0-download-archive?target_os=Linux&target_arch=x86_64&Distribution=Ubuntu&target_version=18.04&target_type=runfile_local

※ ※ ※주의

서버 내의 가상 env 에 설치해야하며, 서버에 설치된 GPU에 맞는 CUDA 드라이버를 설치하여야 함.

다음 화면에서 맞는 CUDA 버전을 확인 후

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

Operating System	Linux	Windows							
Architecture	x86_64	ppc64le	arm64-sbsa	aarch64-jetson					
Distribution	CentOS	Debian	Fedora	KylinOS	OpenSUSE	RHEL	Rocky	SLES	Ubuntu
Version	18.04	20.04	22.04						
Installer Type	deb (local)	deb (network)	runfile (local)						

Download Installer for Linux Ubuntu 18.04 x86_64

wget

https://developer.download.nvidia.com/compute/cuda/11.8.0/local_installers/cuda_11.8.0_520.61.05_linux.run

주황색으로 표시된 문구를 바꿔줄 것

INSTALLING NERF STUDIO

pip install nerfstudio

III) Training

```
# Download some test data:  
ns-download-data nerfstudio --capture-name=poster  
# Train model  
ns-train nerfacto --data data/nerfstudio/poster
```

nerfacto 는 바닐라 nerf 모델 이용시 명령어 이며, 다른 nerf 모델 이용할 시 붉은 색 글씨 부분을 바꿔줘야함.

다른 nerf 모델 별 명령어 참고 :

<https://docs.nerf.studio/nerfology/methods/index.html>

바닐라 nerf 모델에 사용 되었던 nerf 학습 데이터셋임.

사용시 서버에 있는 4대의 GPU 중 0번부터 배정되어 시작,
다른 GPU 사용을 원할경우

```
# 사용가능한 GPU 조회  
nvidia-smi  
#GPU 배정  
CUDA_VISIBLE_DEVICES=2 작업명
```

해당 명령어 입력 시

```
user@aa0d86707cba4: /workspace  
[14:37:32] Auto image downscale factor of 1  
[14:37:33] Skipping 0 files in dataset split train.  
[14:37:33] Skipping 0 files in dataset split val.  
Setting up training dataset...  
Caching all 306 images.  
Setting up evaluation dataset...  
Caching all 34 images.  
No checkpoints to load, training from scratch  
[14:37:56] Printing max of 10 lines. Set flag --logging.local-writer.max-log-size=0 to disable line wrapping.  
Step (% Done) Train Iter (time) ETA (time)  
Step (% Done)  
9910 (99.70%) 40.468 ms 3 s, 642.142 ms 103.21 K  
9920 (99.73%) 42.695 ms 3 s, 415.579 ms 98.34 K  
9930 (99.77%) 43.932 ms 3 s, 75.238 ms 96.00 K  
9940 (99.80%) 41.387 ms 2 s, 483.248 ms 101.15 K  
9950 (99.83%) 42.549 ms 2 s, 127.447 ms 98.45 K  
9960 (99.87%) 42.750 ms 1 s, 710.018 ms 98.12 K  
9970 (99.90%) 40.796 ms 1 s, 223.895 ms 102.52 K  
9980 (99.93%) 40.914 ms 818.286 ms 102.23 K  
9990 (99.97%) 41.697 ms 416.971 ms 101.04 K  
9999 (100.00%) ---  
viewer at: https://viewer.nerf.studio/versions/23-02-8-0/?websocket_url=ws://localhost:7007  
◆◆◆ Training Finished ◆◆◆  
User starts to quit
```

viwer at : 주소를 입력 시 web viewer 를 통해 학습과정 및 학습 결과를 조회할 수 있음. (서버 작업시 포트포워딩 없이 열람 불가능)

기타 학습용, 바닐라 nerf 모델에 사용되었던 데이터셋

```
# Download all scenes from the Blender dataset, including the "classic"  
Lego model  
ns-download-data blender
```

```
# Download the subset of data used in the SIGGRAPH 2023 Nerfstudio  
paper  
ns-download-data nerfstudio --capture-name nerfstudio-dataset
```

```
# Download a few room-scale scenes from the EyefulTower dataset at  
different resolutions  
ns-download-data eyefultower --capture-name riverview seating_area  
apartment --resolution-name jpeg_1k jpeg_2k
```

```
# Download the full D-NeRF dataset of dynamic synthetic scenes  
ns-download-data dnerf
```

다운로드 후 위에 사용된 학습 명령어를 통해 추가 학습 가능

IV) Using own data

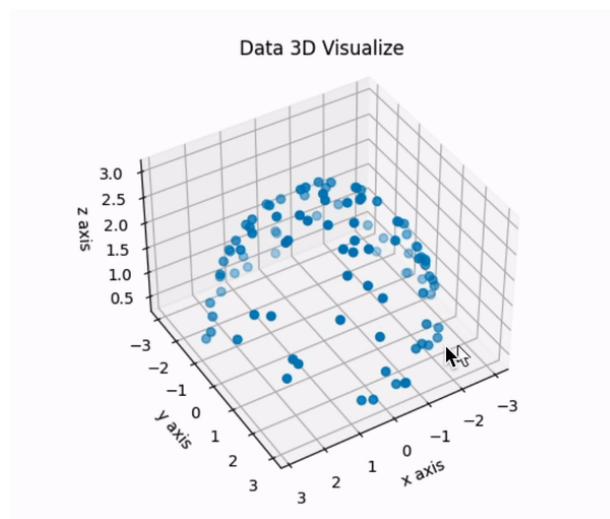
Supported Data Type

Data	Capture Device	Requirements	Speed
Images	Any	COLMAP	
Video	Any	COLMAP	
Polycam	IOS with LiDAR	Polycam App	
KIRI Engine	IOS or Android	KIRI Engine App	
Record3D	IOS with LiDAR	Record3D app	
Metashape	Any	Metashape	

이미지 활용

COLMAP 을 사용하여야 하기 때문에

정면을 기준으로 이글루모양으로 (총 3층), 빛 반사가 없는 사진을 활용하여야 함



(예시 camera pose)

<COLMAP 설치>

```
conda install -c conda-forge colmap  
git clone https://github.com/microsoft/vcpkg  
cd vcpkg  
./bootstrap-vcpkg.sh  
./vcpkg install colmap[cuda]:x64-linux
```

촬영된 이미지를 서버로 전송한 후, 한 폴더 내에 모아줌.

#training 실시

```
ns-process-data {images, video} --data {DATA_PATH} --output-dir  
{PROCESSED_DATA_DIR}
```

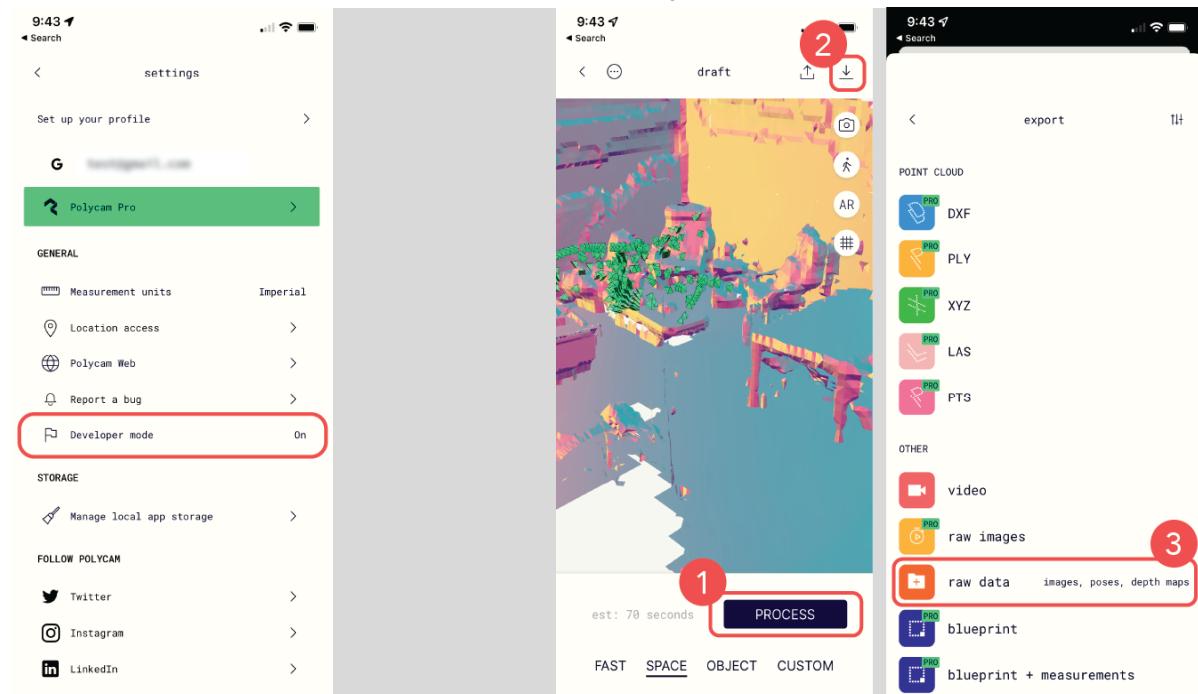
```
ns-train nerfacto --data {PROCESSED_DATA_DIR}
```

#완료 후 결과 view

```
ns-render --load-config outputs/-workspace-preprocessed_data/nerfa  
cto/2023-02-26_143732/config.yml --traj  
filename --camera-path-filename /workspace/preprocessed_data/cam  
era_paths/2023-02-26_143732.json --output-  
path renders/preprocessed_data/filename
```

LiDAR 데이터 사용

iphone 부착 라이다 사용하였으며, polycam 앱 다운로드 필요



polycam 다운로드 및 설치 후 (app store)

process - export(raw data 형식) 으로 파일 내보냄 - 데이터를
nerfstudio 인식 가능 데이터로 변환

#변환

```
ns-process-data polycam --data {OUTPUT_FILE.zip} --output-dir  
{output directory}
```

해당 데이터를 서버로 전송하면 라이다 포인트클라우드로 nerf 구현
가능

#라이다 데이터 트레이닝

```
ns-train nerfacto --data {output directory}
```

image



LiDAR



~Setting Velodyne VLP16 LiDAR at ARIL env~

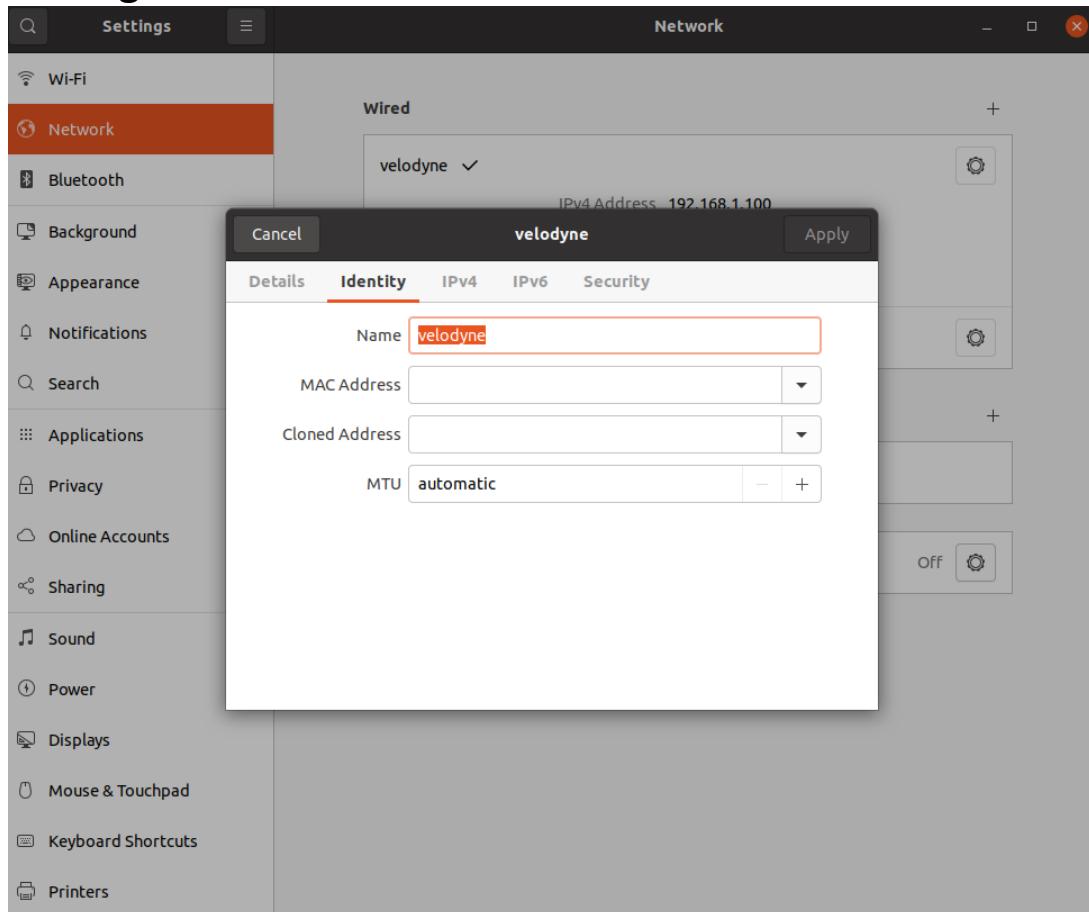
for who don't have network base

env :

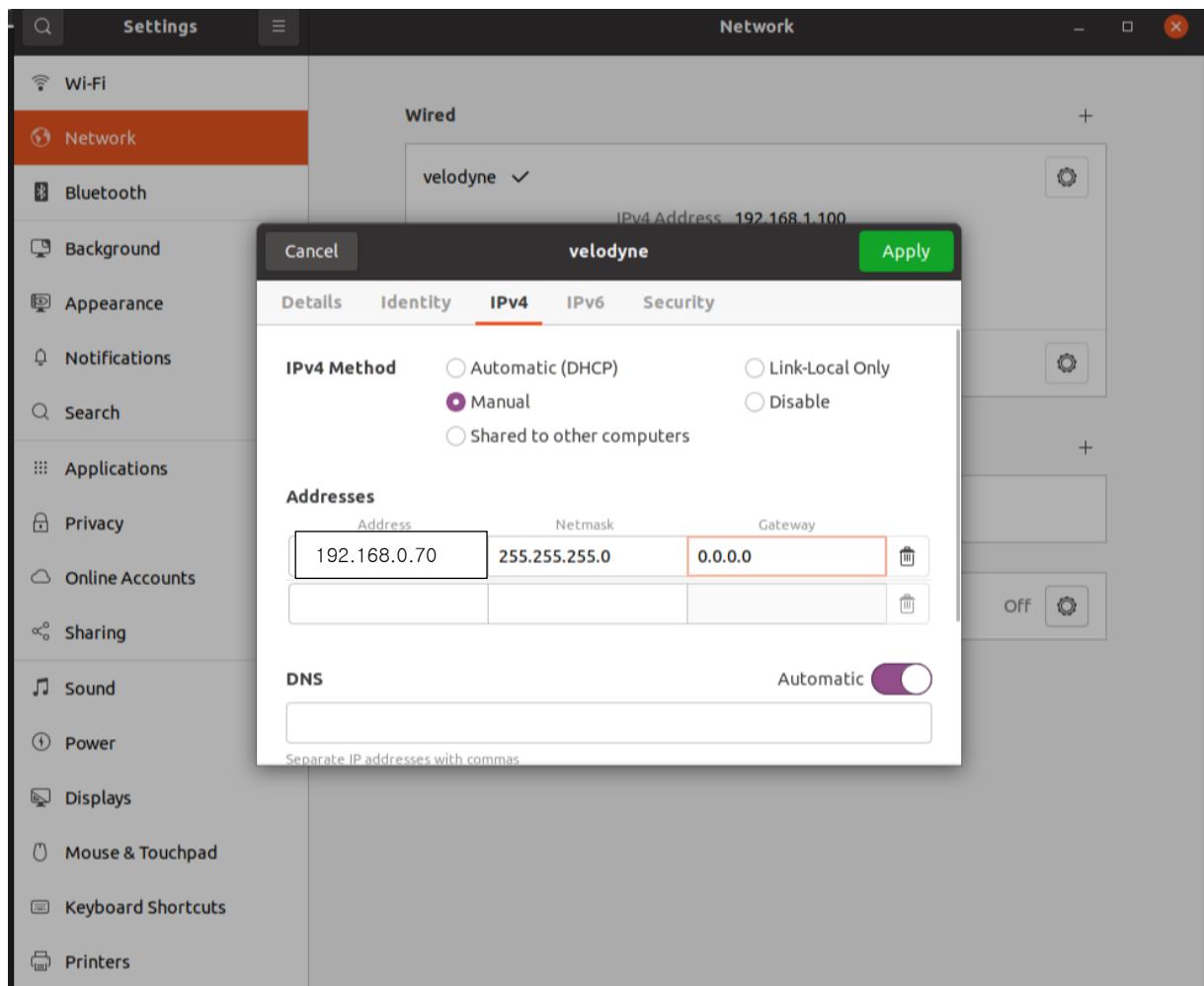
- Ubuntu 20.04 LTS
- Make sure LiDAR to be connected to computer
- Turn off Wi-Fi or unplug network cable until First step is done

I. Setting LiDAR

settings - network



to IPV4 tap



IPV4 Method 'Manual'

set

Addresses : 192.168.0.xx. (x 70~100)

Netmask : 255.255.255.0

Gateway :0.0.0.0

Apply - than run Terminal

Terminal 0||

```
$ sudo apt install net-tools
```

```
$ sudo ifconfig
```

sudo ifconfig 시 출력되는 '가장 위의' 네트워크 확인

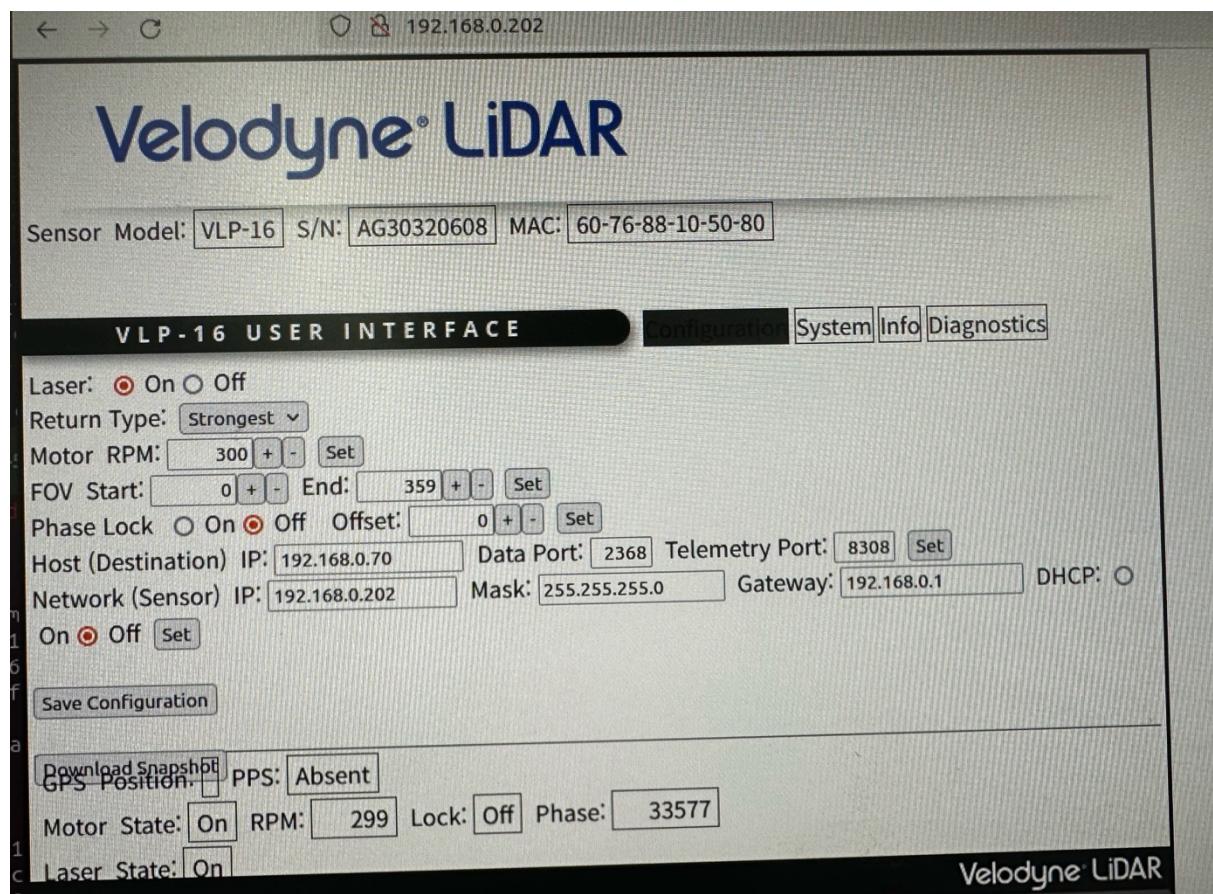
(manual 대로면 eth0이나 네트워크환경, 우분투 버전에 따라 다를 수 있음. 작성자의 경우 enp2s0 or enp2s1)

```
$sudo ifconfig enp2s0 192.168.0.70
```

```
$sudo route add 192.168.0.202
```

(연구실 LiDAR IP 주소)

Firefox 실행 후 192.168.0.202로 접속



해당 화면이 출력되면 성공

II. ROS dependencies 설치

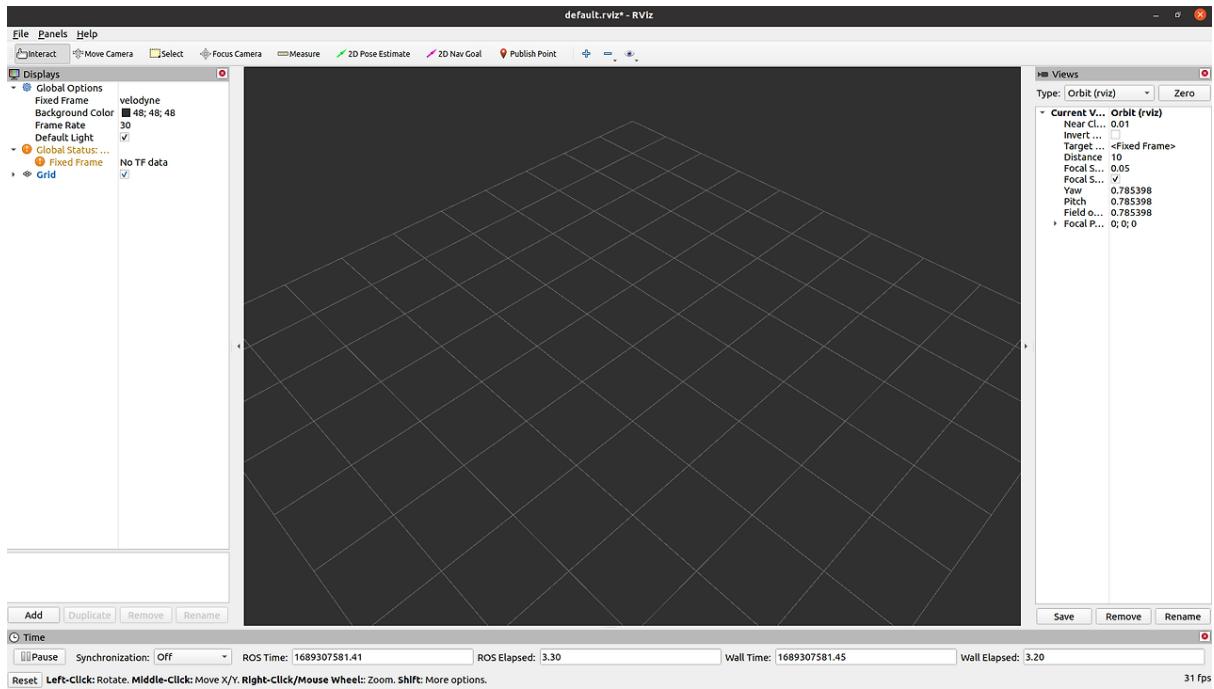
ubuntu 20.04 기준 호환 ROS 는 noetic 버전

터미널에

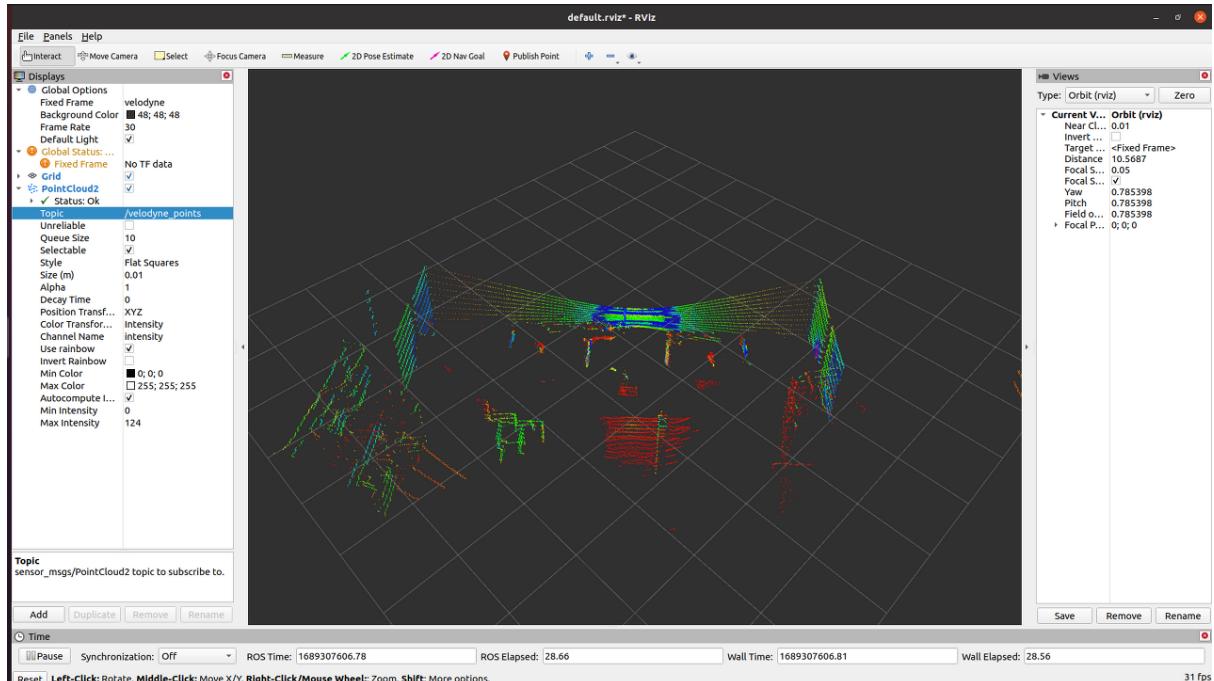
```
$sudo apt-get install ros-noetic-velodyne  
$cd ~/catkin_ws/src/ && git clone https://github.com/ros-drivers/velodyne.git  
$cd ..  
$rosdep install --from-paths src --ignore-src --rosdistro  
YOURDISTRO -y  
$cd ~/catkin_ws/ && catkin_make
```

III. Visualization

```
$roslaunch velodyne_pointcloud VLP16_points.launch  
$rosnode list  
$rostopic echo /velodyne_points  
$rosrun rviz rviz -f velodyne
```



좌측 하단의 Add 클릭



데이터 타입 PointCloud2 선택 :

Topic에서 /velodyne_points 선택

Done!

<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2023-12/R/eclipse-inst-jre-win64.exe>



Eclipse downloads - Select a mirror | Th…

The Eclipse Foundation - home to a global co…

www.eclipse.org

All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless otherwise specified.

 Download

Download from: Korea, Republic Of - Kakao Corp. (<https://www.eclipse.org>)

File: [eclipse-inst-jre-win64.exe](https://www.eclipse.org/eclipse-inst-jre-win64.exe) SHA-512

[>> Select Another Mirror](#)

eclipseinstaller

 by Oomph

type filter text



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration



Eclipse IDE for Enterprise Java and Web Developers

Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA and Data Tools, Maven and...



Eclipse IDE for C/C++ Developers

An IDE for C/C++ developers.



Eclipse IDE for Embedded C/C++ Developers

An IDE for Embedded C/C++ developers. It includes managed cross build plug-ins (Arm and RISC-V) and debug plug-ins (SEGGER J-Link, OpenOCD, pyocd, and QEMU), plus a number of templates to create ready...



Eclipse IDE for PHP Developers

The essential tools for any PHP developer, including PHP language support, Git client and editors for JavaScript, TypeScript, HTML, CSS and XML.

[Click here to raise an issue with Eclipse PHP Development Tools.](#)

Artifact download is progressing very slowly from <https://ftp.jaist.ac.jp>



Eclipse IDE for Java Developers

[details](#)

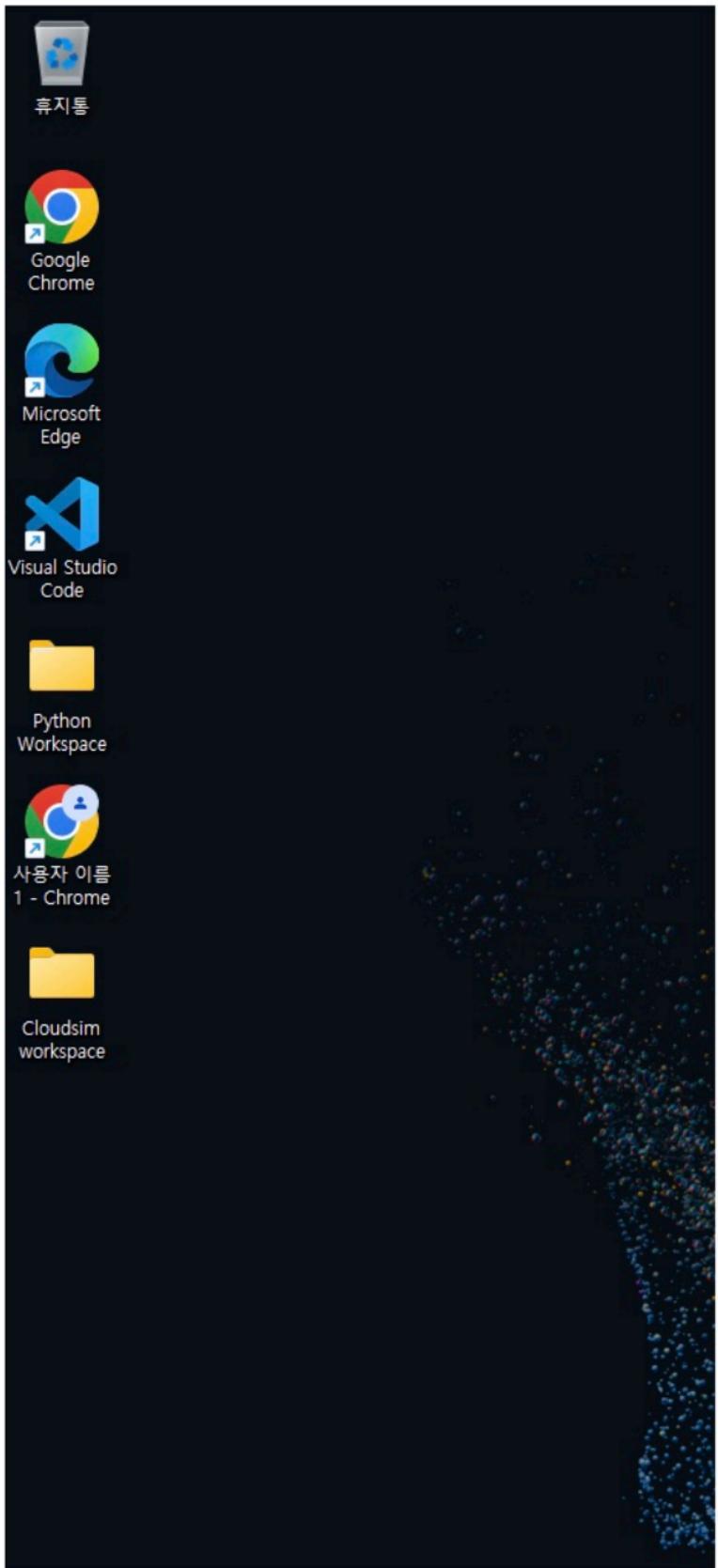
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration.

Java 17+ VMJRE 17.0.9 - <https://download.eclipse.org/justj/jres/17/updates/release/latest>**Installation Folder**

C:\Users\ARIL\eclipse\java-2023-12

 **create start menu entry** **create desktop shortcut****INSTALLING****Cancel Installation**

이제 CloudSim 이 필요합니다.



본인의 바탕화면에 아무 폴더 (저는 Cloudsim Workspace 로 하겠습니다) 를 생성해줍니다.

Cloudsim 3.0.3 버전을 사용할건데 (왜인지 모르지만 이것만 오류가 없음), 위 링크로 접속합니다

Project



cloudsim

Source

Issues

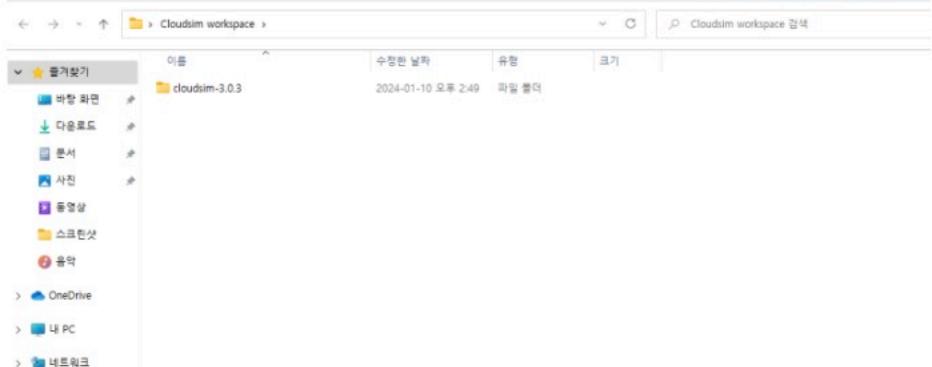
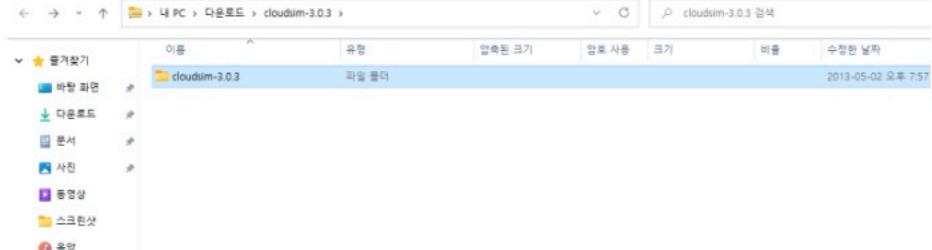
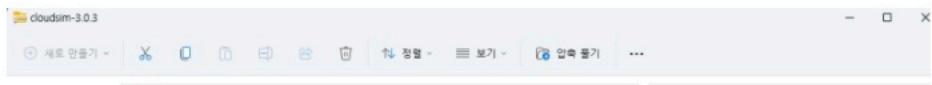
Wikis

Downloads

File	Summary + Labels	Uploaded	Size
cloudsim-3.0.3.tar.gz	CloudSim 3.0.3: bug fix release Featured Type-Package OpSys-All	May 3, 2013	9.9MB
cloudsim-3.0.3.zip	CloudSim 3.0.3: bug fix release Featured Type-Package OpSys-All	May 3, 2013	13.05MB
cloudsim-3.0.2.tar.gz	CloudSim 3.0.2: bug fix release Type-Package OpSys-All	Nov 7, 2012	9.9MB
cloudsim-3.0.2.zip	CloudSim 3.0.2: bug fix release Type-Package OpSys-All	Nov 7, 2012	13.05MB

cloudsim-3.0.3 zip 선택 후 다운로드, 압축을 풀어줍니다.

압축을 푼 폴더에서 cloudsim-3.0.3 폴더를



이름	유형	압축된 표기	압축 사용	크기	비율	수정한 날짜
docs	파일 폴더					2013-05-02 오후 7:57
examples	파일 폴더					2013-05-02 오후 7:57
jars	파일 폴더					2013-05-02 오후 7:57
sources	파일 폴더					2013-05-02 오후 7:57
build	XML 문서	1KB 아니요	3KB 62%	3KB	2013-05-02 오후 7:57	
changelog	텍스트 문서	5KB 아니요	12KB 64%	5KB	2013-05-02 오후 7:57	
examples	텍스트 문서	2KB 아니요	5KB 75%	2KB	2013-05-02 오후 7:57	
license	텍스트 문서	3KB 아니요	8KB 66%	3KB	2013-05-02 오후 7:57	
pom	XML 문서	2KB 아니요	4KB 64%	2KB	2013-05-02 오후 7:57	
readme	텍스트 문서	1KB 아니요	3KB 67%	1KB	2013-05-02 오후 7:57	
release_notes	텍스트 문서	2KB 아니요	3KB 53%	2KB	2013-05-02 오후 7:57	

(혹시나 헛갈리실가봐, 이 내용물이 있는 폴더를 넣어주는겁니다.)

Cloudsim 완료, 이제 JAR 파일이 필요합니다

<https://dlcdn.apache.org//commons/math/binaries/>

Index of /commons/math/binaries

Name	Last modified	Size	Description	Parent Directory	- common...
------	---------------	------	-------------	------------------	-------------

dlcdn.apache.org

위 링크에 접속하여 아래 화면에서 commons-math-3.0-bin.zip 을 클릭하여 다운받아줍니다.
(이게오류없었음)

새로 만들기



↑↓ 정렬 보기 ...

< > ⌂ ⌃ ⌄

Cloudsim workspace



Cloudsim workspace 검색

★ 즐겨찾기

바탕 화면

다음으로

문서

사진

Cloudsim workspace

동영상

스크린샷

음악

> OneDrive

> 내 PC

> 네트워크

2개 확장

이름

수정한 날짜

유형

크기

cloudsim-3.0.3

2024-01-10 오후 2:49

파일 폴더

commons-math3-3.6.1.jar

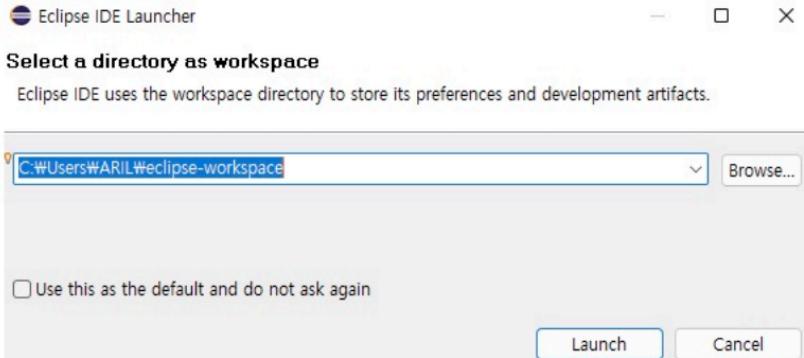
2016-03-17 오후 1:34

JAR 파일

2,162KB

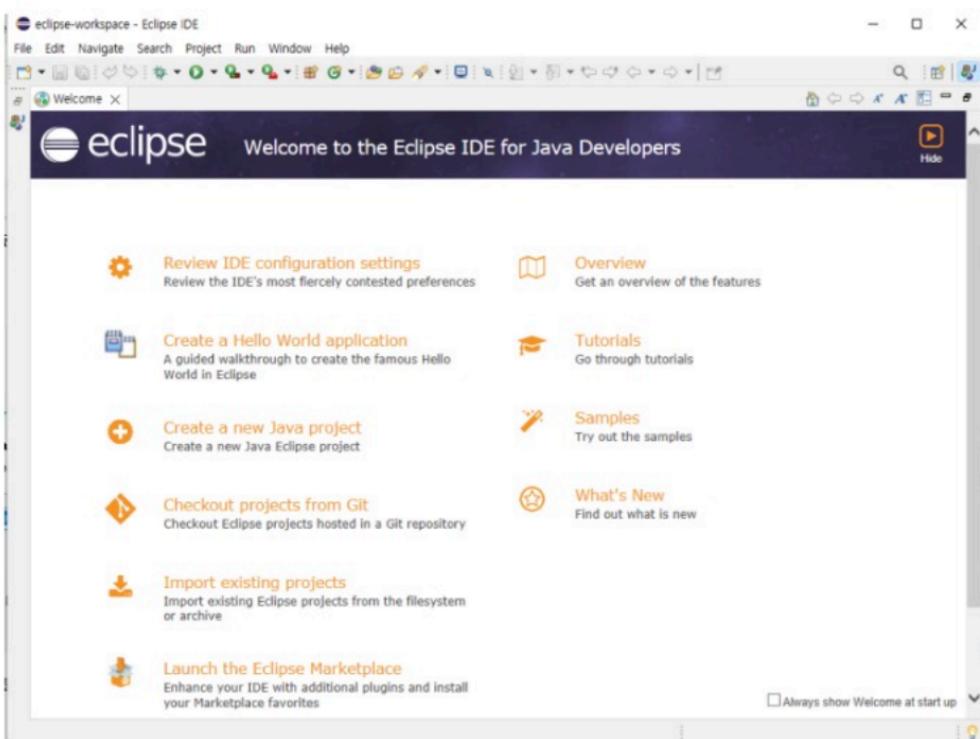
이렇게요.

이클립스 설치가 완료된 뒤 Launch 를 누르면 아래와 같은 화면이 나옵니다.



Launch 를 눌러 넘겨줍니다.

그럼 아래와 같이 뜹니다. Create a new Java project 를 눌러줍니다.



여기서 JRE 부분에서

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.



Project name: CloudsimARIL

Use default location

Location: C:\Users\ARIL\eclipse-workspace\CloudsimARIL

[Browse...](#)

JRE

Use an execution environment JRE:

JavaSE-1.7

Use a project specific JRE:

CDC-1.0/Foundation-1.0

Use default JRE 'jre' and workspace c

JRE-1.1

OSGI/Minimum-1.1

J2SE-1.2

OSGI/Minimum-1.2

J2SE-1.3

J2SE-1.4

J2SE-1.5

JavaSE-1.6

JavaSE-1.7

JavaSE-1.8

JavaSE-9

JavaSE-10

JavaSE-11

JavaSE-12

JavaSE-13

JavaSE-14

JavaSE-15

JavaSE-16

JavaSE-17

JavaSE-18

JavaSE-19

JavaSE-20

JavaSE-21

Project layout

Use project folder as root for sources

J2SE-1.3

Create separate folders for sources and

J2SE-1.4

J2SE-1.5

JavaSE-1.6

Working sets

Add project to working sets

JavaSE-1.7

Working sets:

JavaSE-1.8

JavaSE-9

JavaSE-10

JavaSE-11

JavaSE-12

JavaSE-13

JavaSE-14

JavaSE-15

JavaSE-16

JavaSE-17

JavaSE-18

JavaSE-19

JavaSE-20

JavaSE-21

① The default compiler compliance level for the current workspace is 17. The new project will use a project specific



< Back

Next >

Finish

Cancel

Use an excution environment JRE : 부분에서 JavaSE-1.7을 선택해줍니다.

Use default location

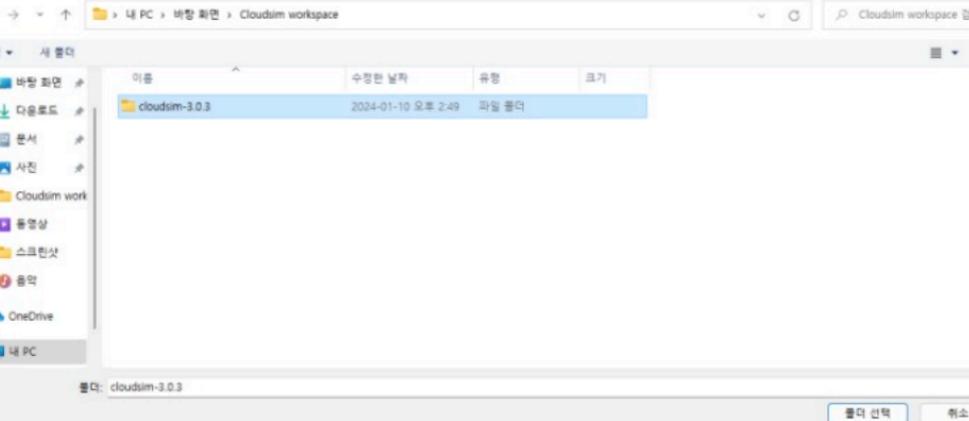
Location: C:\Users\ARIL\eclipse-workspace\CloudsimARIL

[Browse...](#)

JRE

그리고 위에 Use Default Loaction 체크박스를 해제한 뒤 Browse로

Project Folder Selection



아까 생성한 Cloudsim Workspace 폴더 내 cloudsim-3.0.3 폴더를 선택해줍니다.

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.



Project name:

Use default location

Location: C:\Users\ARIL\Desktop\Cloudsim workspace\cloudsim-3.0.3

[Browse...](#)

JRE

Use an execution environment JRE: JavaSE-1.7

Use a project specific JRE: jre

Use default JRE 'jre' and workspace compiler preferences

[Configure JREs...](#)

Project layout

Use project folder as root for sources and class files

Create separate folders for sources and class files

[Configure default...](#)

Working sets

Add project to working sets

[New...](#)

Working sets:

[Select...](#)

Module

Create module-info.java file

Module name:

Generate comments

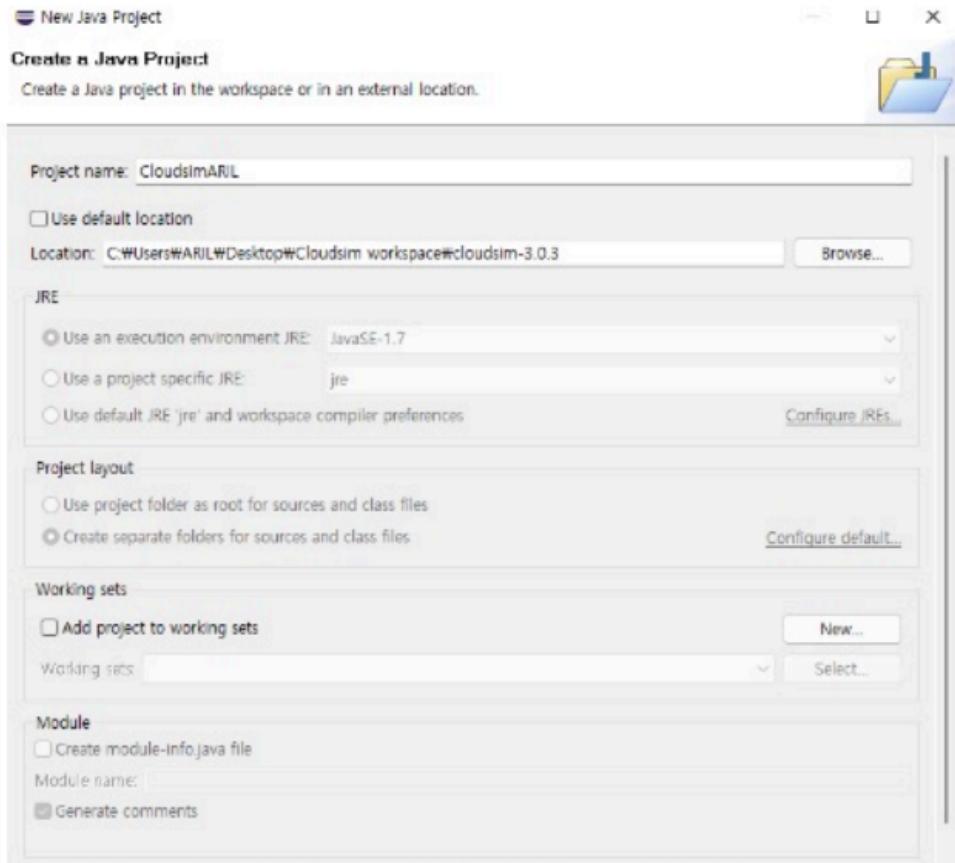
The wizard will automatically configure the JRE and the project layout based on the existing source. 24



그리고 위에 Use Default Location 체크박스를 해제한 뒤 Browse로



아까 생성한 Cloudsim Workspace 폴더 내 cloudsim-3.0.3 폴더를 선택해줍니다.



이런 화면이어야 합니다. Next 를 누릅니다. (Finish 누르면 안댐::)

New Java Project

Java Settings

Define the Java build settings.



Source Projects Libraries Order and Export Module Dependencies

CloudsimARIL

- examples
- sources
- jars
- build.xml
- changelog.txt
- examples.txt
- license.txt
- pom.xml
- readme.txt
- release_notes.txt

Details

[Create new source folder](#): use this if you want to add a new source folder to your project.

[Link additional source](#): use this if you have a folder in the file system that should be used as additional source folder.

[Add project 'CloudsimARIL' to build path](#): Add the project to the build path if the project is the root of packages and source files. Entries on the build path are visible to the compiler and used for building.

Allow output folders for source folders

Default output folder:

CloudsimARIL/bin

Browse...



< Back

Next >

Finish

Cancel

Java Settings

Define the Java build settings.



Source | Projects | Libraries | Order and Export | Module Dependencies

JARs and class folders on the build path:

Modulepath

Classpath

- > cloudsim-3.0.3.jar - CloudsimARIL/jars
- > cloudsim-3.0.3-sources.jar - CloudsimARIL/jars
- > cloudsim-examples-3.0.3.jar - CloudsimARIL/jars
- > cloudsim-examples-3.0.3-sources.jar - CloudsimARIL/jars
- > commons-math3-3.6.1.jar - C:\Users\ARIL\Desktop\Cloudsim workspace
- > JRE System Library [jre]

Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...

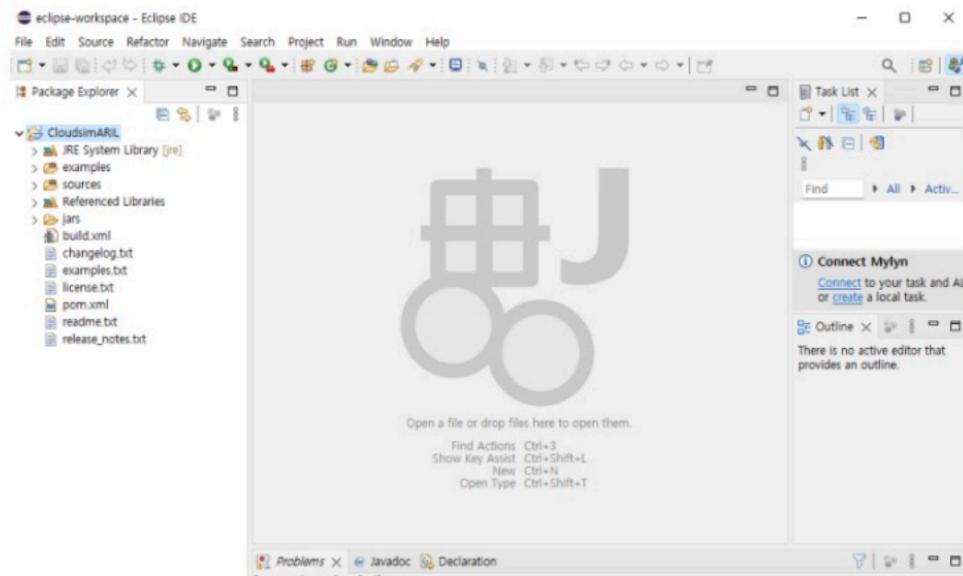


< Back

Next >

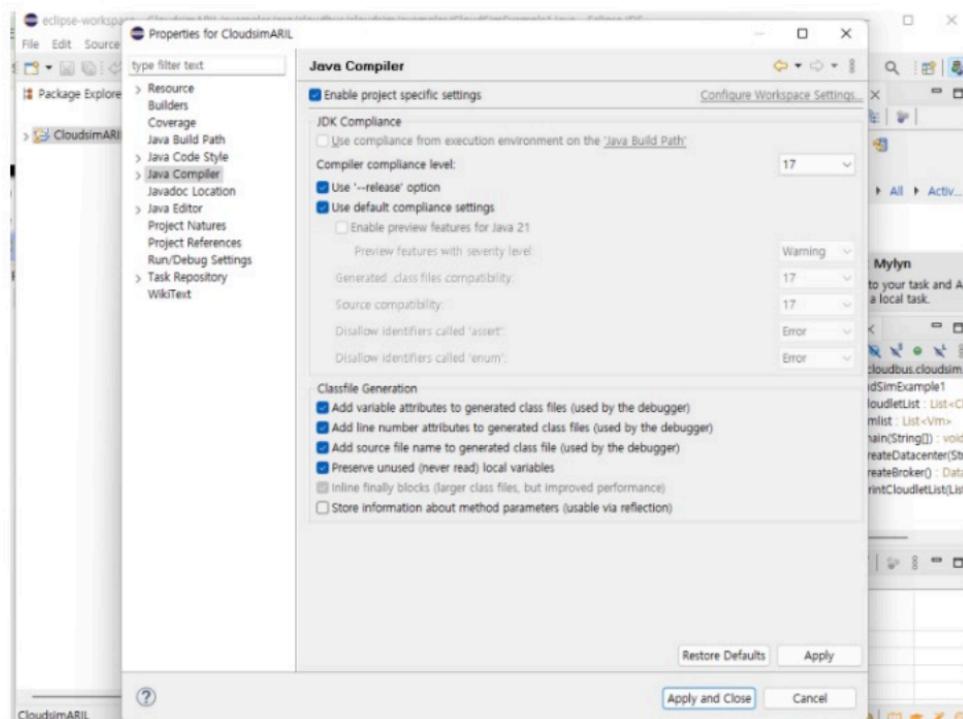
Finish

Cancel



아마 해당상태에선 오류가 뜬다고 이를립스가 호들갑을 떨 것입니다.

거의 다왔습니다. CloudsimARIL (제가 지은 임의의 이름입니다.) 즉 최상위 객체를 우클릭한 뒤 Properties로 가줍니다.



해당 화면의 Java Compiler 탭으로 이동해 Enable project specific settings에 체크를 해줍니다.

Important Notices

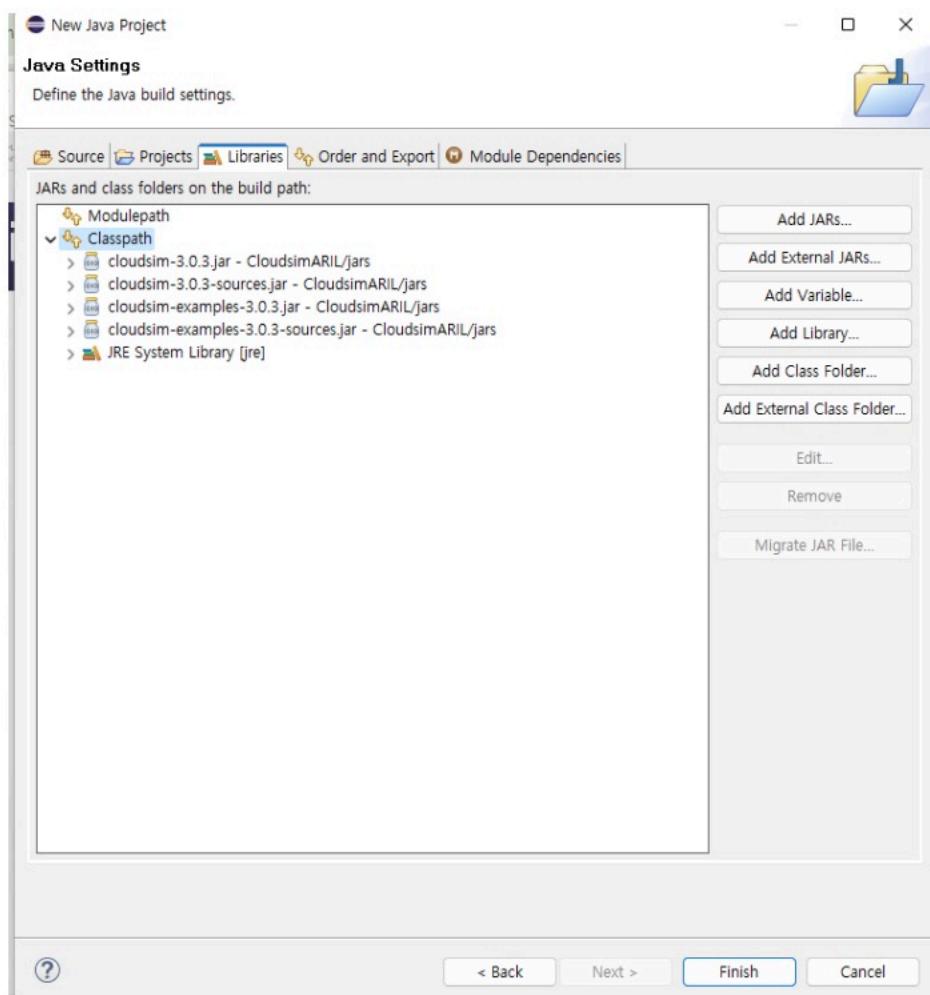
- Download from your nearest mirror site!
- PGP/GPG Signatures

Name	Last modified	Size	Description
Parent Directory	-	-	
commons-math-2.2.tar.gz	2013-03-03 21:07	10M	
commons-math-2.2.tar.gz.asc	2013-03-03 21:07	836	
commons-math-2.2.tar.gz.sha1	2013-06-14 23:39	62	
commons-math-2.2.tar.gz.sha256	2013-06-14 23:39	106	
commons-math-2.2.zip	2013-03-03 21:07	13M	
commons-math-2.2.zip.asc	2013-03-03 21:07	836	
commons-math-2.2.zip.sha1	2013-06-14 23:39	79	
commons-math-2.2.zip.sha256	2013-06-14 23:39	103	
commons-math3-3.6.1-bin.tar.gz	2016-03-21 12:48	18M	
commons-math3-3.6.1-bin.tar.gz.asc	2016-03-21 12:48	479	
commons-math3-3.6.1-bin.tar.gz.sha1	2016-06-14 23:39	89	
commons-math3-3.6.1-bin.tar.gz.sha256	2016-06-14 23:39	113	
commons-math3-3.6.1-bin.zip	2016-03-21 12:48	21M	
commons-math3-3.6.1-bin.zip.asc	2016-03-21 12:48	479	
commons-math3-3.6.1-bin.zip.sha1	2016-06-14 23:39	86	
commons-math3-3.6.1-bin.zip.sha256	2016-06-14 23:39	110	
commons-math4-4.0-beta1-bin.tar.gz	2022-12-19 13:19	13M	
commons-math4-4.0-beta1-bin.tar.gz.asc	2022-12-19 13:19	658	
commons-math4-4.0-beta1-bin.tar.gz.sha512	2022-12-19 13:19	129	
commons-math4-4.0-beta1-bin.zip	2022-12-19 13:19	14M	
commons-math4-4.0-beta1-bin.zip.asc	2022-12-19 13:19	658	
commons-math4-4.0-beta1-bin.zip.sha512	2022-12-19 13:19	129	

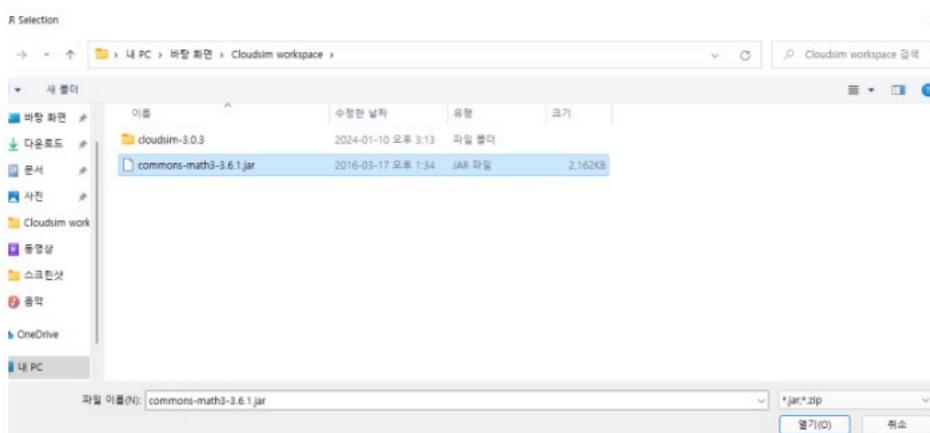
압축을 푼뒤, commons-math3-3.6.1jar 파일을 (이거 하나만)

아까 생성한 CloudsimWorkspace 폴더로 옮겨줍니다.

commons-math3-3.6.1							
새로 만들기		정렬		보기		압축 풀기	
▼	▶	이름	유형	압축된 크기	압축 사용	크기	비율
▼	▶	docs	파일 폴더				
		commons-math3-3.6.1.jar	JAR 파일	1,920KB	아니요	2,162KB	12%
		commons-math3-3.6.1-javadoc.jar	JAR 파일	6,241KB	아니요	6,680KB	7%
		commons-math3-3.6.1-sources.jar	JAR 파일	2,285KB	아니요	2,456KB	7%
		commons-math3-3.6.1-tests.jar	JAR 파일	2,642KB	아니요	2,906KB	10%
		commons-math3-3.6.1-test-sourc...	JAR 파일	1,783KB	아니요	1,914KB	7%
		commons-math3-3.6.1-tools.jar	JAR 파일	11KB	아니요	12KB	7%
		LICENSE	텍스트 문서	7KB	아니요	24KB	71%
		NOTICE	텍스트 문서	1KB	아니요	1KB	45%
		RELEASE-NOTES	텍스트 문서	1KB	아니요	2KB	48%



해당 탭에서 classpath 를 선택한 뒤 우측의 Add External JARs 를 클릭해줍니다.



바탕화면의 Cloudsim workspace 폴더로 이동한 뒤 commons-math3-3.6.1 jar 파일을 열어줍니다.

- type filter text
- > Resource Builders Coverage Java Build Path
 - > Java Code Style
 - > Java Compiler **Javadoc Location**
 - > Java Editor Project Natures Project References Run/Debug Settings
 - > Task Repository WikiText

Java Compiler

Enable project specific settings

[Configure Workspace Settings...](#)

JDK Compliance

Use compliance from execution environment on the 'Java Build Path'

Compiler compliance level: →

17
1.3
1.4
1.5
1.6
1.7
1.8
9
10
11
12
13
14
15
16
17
18
19
20
21

Use '--release' option

Use default compliance settings

Enable preview features for Java 21

Preview features with severity level:

Generated .class files compatibility:

Source compatibility:

Disallow identifiers called 'assert':

Disallow identifiers called 'enum':

Classfile Generation

Add variable attributes to generated class files (used by the debugger)

Add line number attributes to generated class files (used by the debugger)

Add source file name to generated class file (used by the debugger)

Preserve unused (never read) local variables

Inline finally blocks (larger class files, but improved performance)

Store information about method parameters (usable via reflection)

[Restore Defaults](#)

[Apply](#)



[Apply and Close](#)

[Cancel](#)

Complier compliance level :에서 1.7을 선택해줍니다.

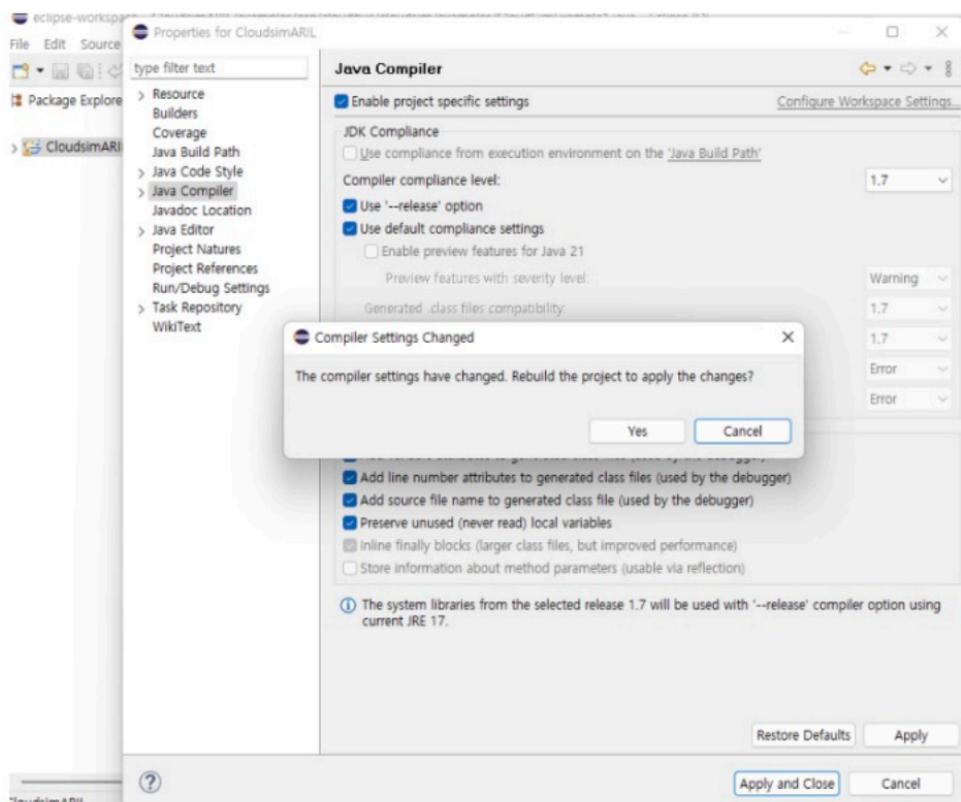
(1.7이 없다면 아래링크에서 Java 최신으로 설치할 것)

<https://www.oracle.com/java/technologies/javase-jdk21-doc-downloads.html>

Java Development Kit 21 Documentation

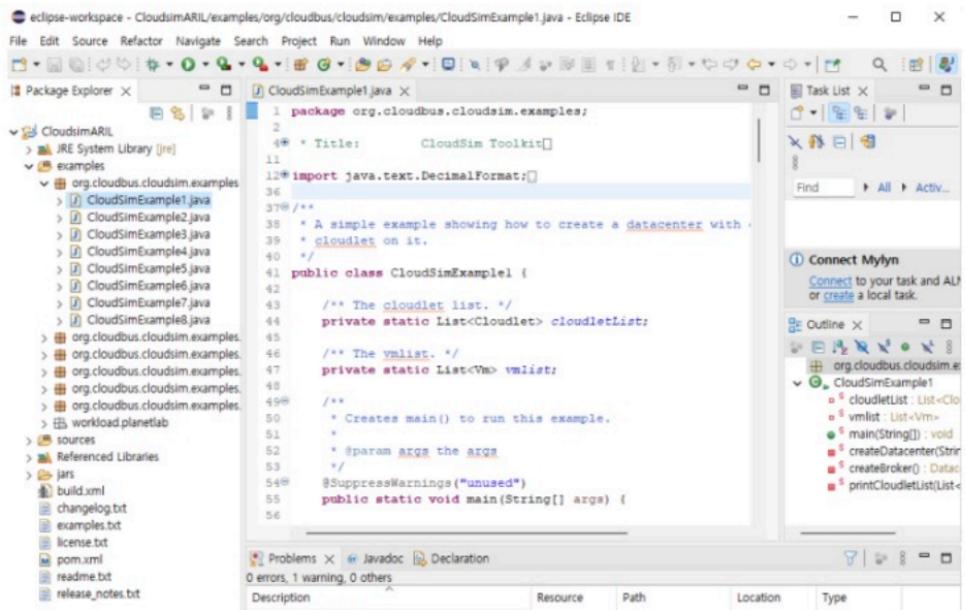
Java SE Development Kit 21 Documentation This software is lic...

www.oracle.com



Apply 를 누른 뒤 해당 창이 뜨면 Yes 를 눌러줍니다.

자동으로 리빌딩 되며 이클립스를 재시작합니다.



연 구 노 트
The Research Notes

과제명

TITLE Real time processing LIDAR Data on Cloud PROJECT NO.

과제번호

Continued from page :

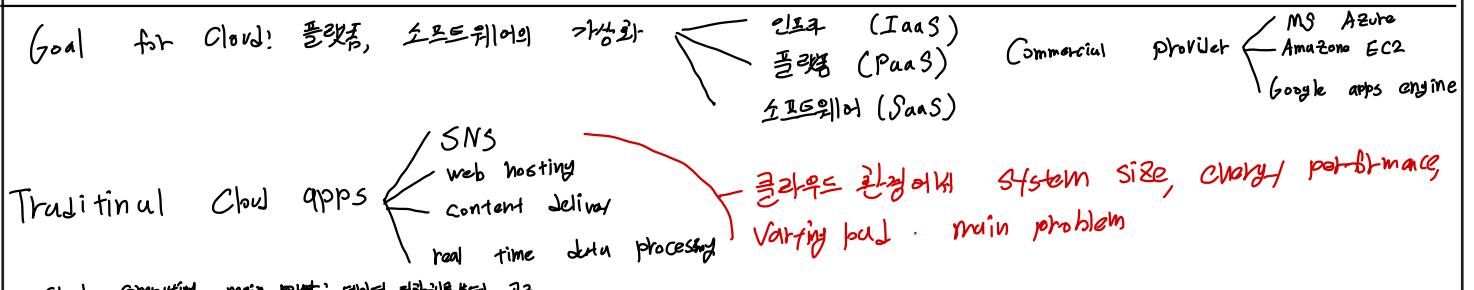
<CloudSim: Simulator for Cloud Computing infrastructure & modeling>

Cloud Computing: aims to deliver reliable, secure, fault-tolerant, sustainable, scalable infra for hosting Internet based services, / type of parallel distributed system consisting of a collection of interconnected, virtualized computers
→ 서비스를 위한 확장성, 안전성, 성능을 제공하는 시스템

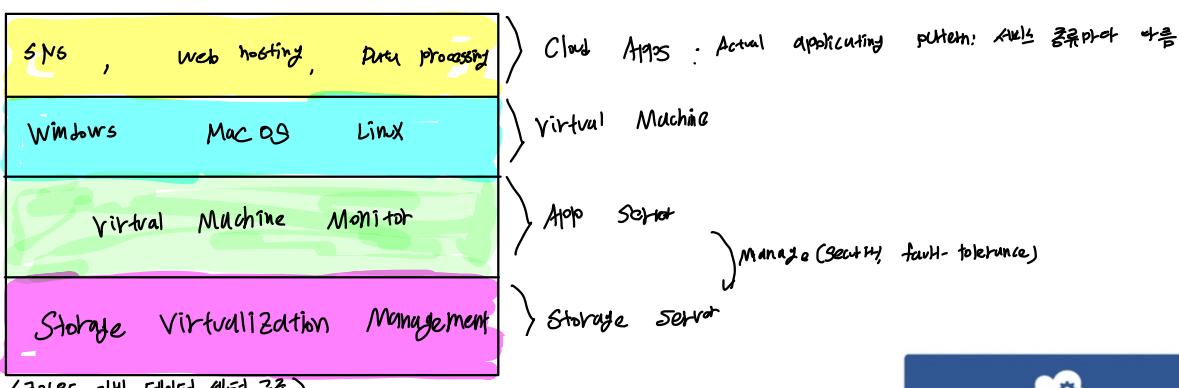
→ Modeling, Scheduling different apps, service for Cloud infra
→ Challenging task: requires different local, changing performance

→ Solved Using

CloudSim : a simulation framework provides simulation power to manage service modeling of cloud infra structure.



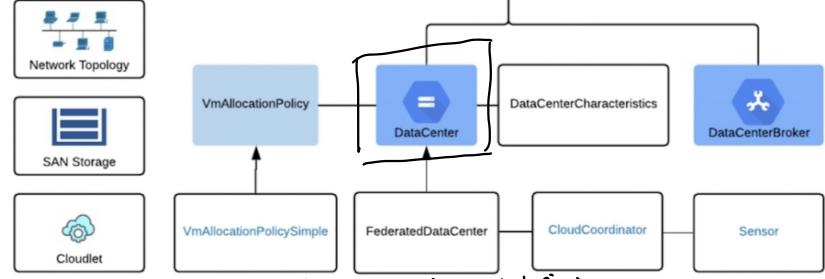
Cloud computing main point: 서비스 접근성을 높여 줌



CloudSim,

Extensible simulation toolkit that enables modeling and simulation of cloud computing system and applications provisioning environment

→ Save time, money, flexible environment



Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

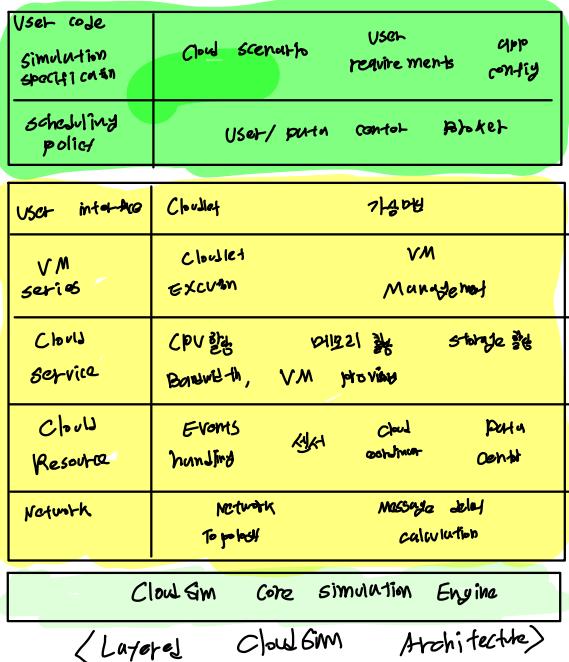
과제명

TITLE

과제번호

PROJECT NO.

Continued from page :



Cloud sim ?? : VM 서비스로 구현
Used to be developed by USInt
CloudSim simulation engine where the
services used to be developed by network
topologies /

VI : cloudlet / virtual machine

User code : simulation specification, schedule policy

→ (결과, 특정 클라우드 환경 에스토 하는 것)

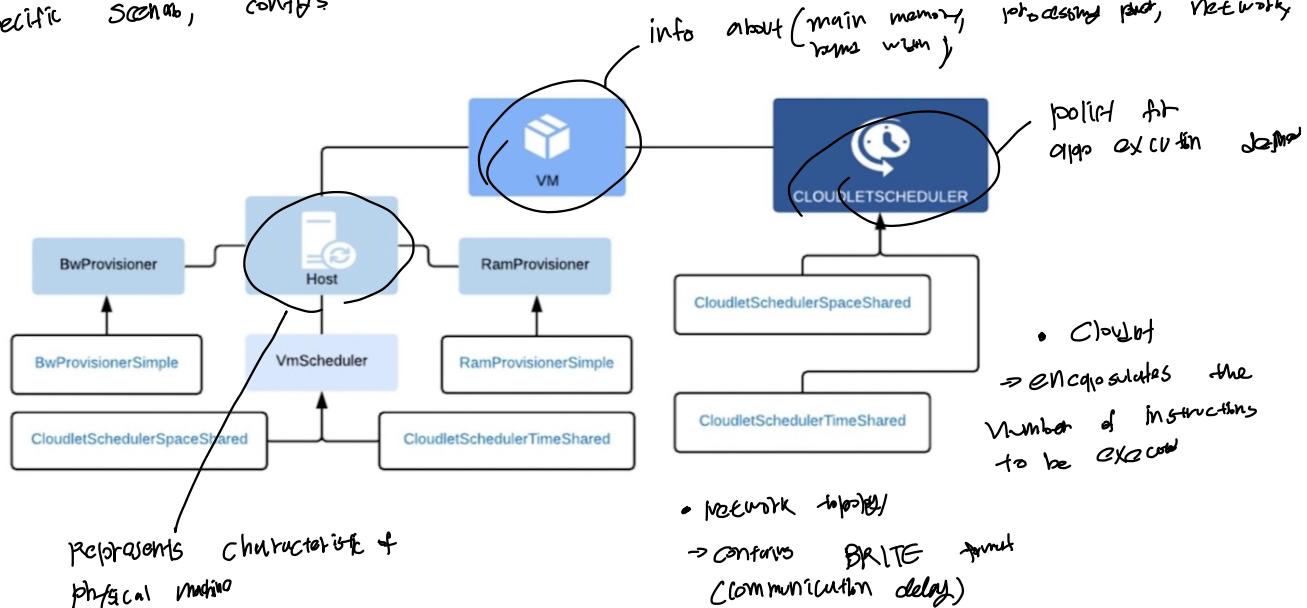
CloudSim : built-in a layered method that
reflects layered construction of cloud computing
environment themselves.

+ 클라우드 컴퓨팅 : 빠르게 발전中, 투, 배포 등 정해진
표준 부록 → 구체화 학습, 산업 활용

ex) defining core algorithms, policies, apps

- By extending basic functionalities already exposed by CloudSim, researchers would be able to perform tests based on specific scenarios, configs

JAVA 기반임 //



Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

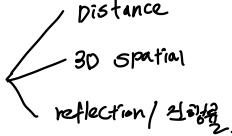
Big 3D spatial data processing Using cloud computing environment ~ 아기오와 대학~

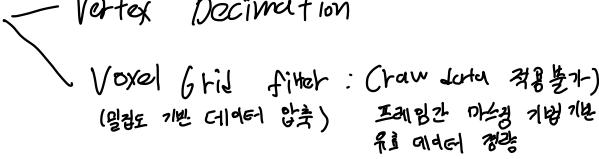
3

과제명 + 한국정보통신학회 Vol.25 No.12
TITLE

과제번호
PROJECT NO.

Continued from page :

LIDAR dataset  Highly Accurate. ↗ 전처리 시.
Enormous Size → 그거를 줄이는 알고리즘 개발

Size-reducing Algorithms 

Voxel Grid filter : (Raw data 적용불가)
(밀집도 기본 데이터 압축) 프레임간 마스킹 기법 기본
유 효 데이터 정량

→ 실시간 처리 가능 해야함 → 클라우드 이용. → Cloud Sim 이용해 최적조건 예측

pros & cons for CloudSim

pros

- Elasicity for configs
- Diverse Environment
- ↓ modeling for any environment
- 클라우드 기본 apo로 실행 by minimum time resource
- Ease use, customization

Cons

- GUI 지원 X
- restricted workload (basic network model)
- parallel experiments : not helpful
- data 읽기/쓰기 A동작이야
- A동작 결과는 여러번

각 구조마다
조건 설정

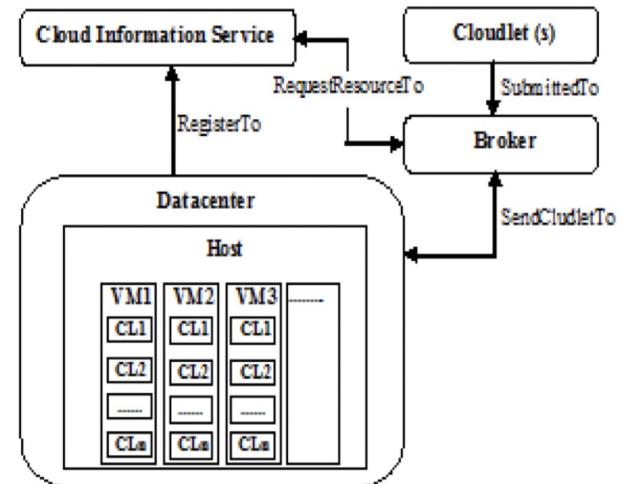


Fig. 1. Basic scenario of Simulation [8]:

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

과제명

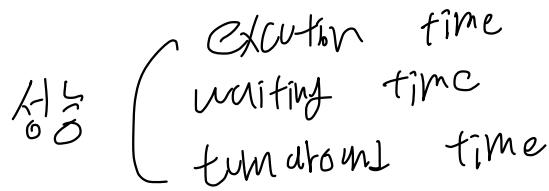
TITLE

과제번호

PROJECT NO.

Continued from page :

CloudSim 설정 manual (설정 가이드)



1) Turn around time

```

double totalTime = 0.0;
int NoCloudlets = cloudletList.size();
For int i=0; to NoCloudlets - 1;
totalTime += cloudletList.get(i).getFinishTime();
double avgTAT = totalTime / NoCloudlets;
End For
return avgTAT;
  
```

Examples 활용 결과값 예상

- Time Shared
- Space Shared cost
- Dynamic workload

2) Throughput time

```

calculateThroughput (List<? extends Cloudlet> cloudletList)
double maxFT=0.0;
int NoCloudlets = cloudletList.size();
For int i=0; to NoCloudlets-1;
  double currentFT = cloudletList.get(i).getFinishTime();
If "currentFT > maxFT" maxFT = currentFT; End If
  double throughput = NoCloudlets / maxFT;
End For
return throughput;
  
```

→ based on
VM allocation policy

3) Execution time

```

For int i=0; to NoCloudlets-1;
ET = list.get(i).getCloudletLength() / CapOfVM(p*q);
End For
  
```

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

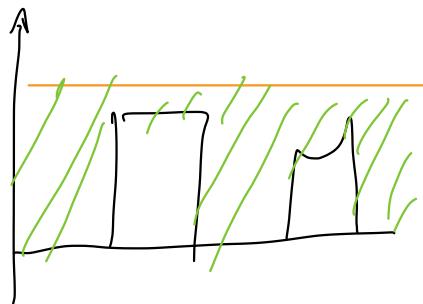
과제명

TITLE

과제번호

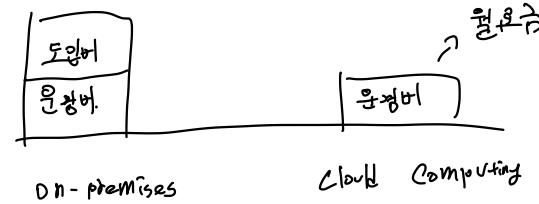
PROJECT NO.

Continued from page :

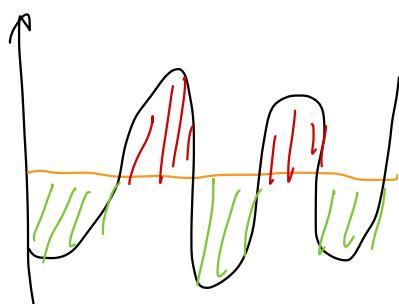
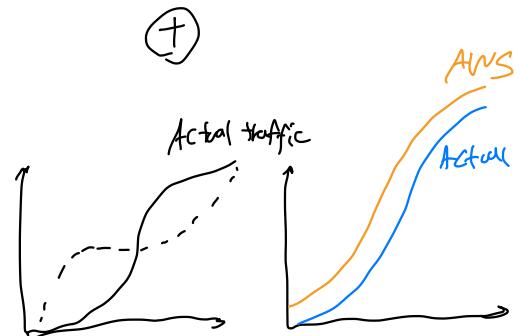


자·원·방·비
발·생

on & off



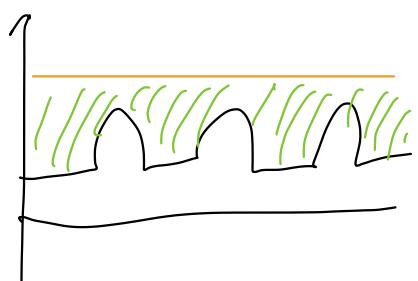
Fast Growth



Variable peaks

설계: 물리환경
경쟁X.
one-click
설정/구축

→ 외부, 전략적 / 비용, 인적



prediction peaks

AWS Deep reach
→ 물리 학습에 충분
있겠지만,,,

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

Calibration

$$w[x \ y \ z \ 1] = [x \ y \ z \ 1] P$$

↓
fix P

Scale parameter

외부 외부 : 3차원 월드 \rightarrow 3차- 카메라
내부 외부 : 3차원 월드 \rightarrow 2차원 화면.

체지보드 대상物体

$$P = \begin{bmatrix} R \\ e \end{bmatrix} K$$

↓
외부 파라미터

R : 3×3 회전 행렬 matrix

$e = [t_1, t_2, t_3]$: 1×3 평행 이동 vector

MHO 기법 장단점

$$K = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ C_x & C_y & 1 \end{bmatrix}$$

내부 파라미터 측정

precision \rightarrow Recall.

물체가 있는지
없는지?

있거나
없는지?

C_x, C_y : 영상 중심 좌표

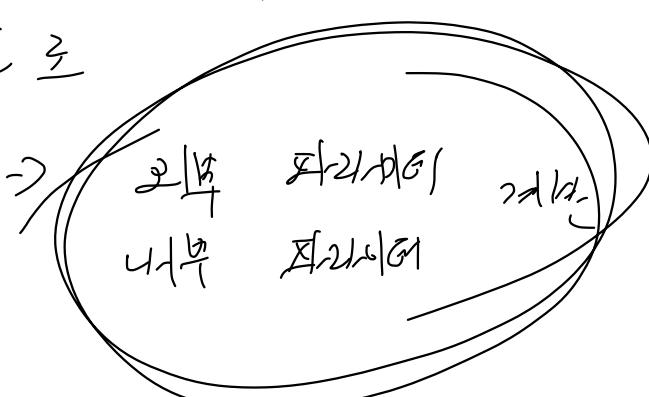
f_x, f_y : 카메라 단위 풀면 \rightarrow 실제 화면 거리 \rightarrow 초점 거리

$s = f_x \tan \alpha$: skew 계수

방향
증강학
체지보드
보정 필요
설정 해제?,

\rightarrow checker board

체지보드
코너점



1차 검출기.

RGB \rightarrow CNN \rightarrow 경로 예상 \rightarrow 경로 추적
특징량

ex) YOLOv3, Retina Net

2차 검출기

RGB \rightarrow CNN \rightarrow RPN \rightarrow proposal \rightarrow 경로 예상
(경로 예상)

↓
경로 예상

과제명

TITLE

과제번호

PROJECT NO.

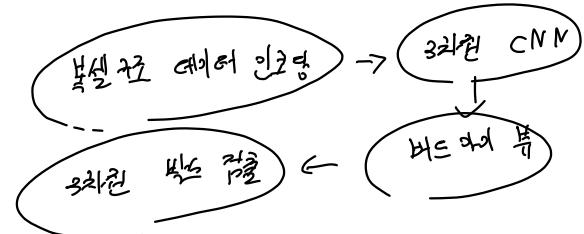
Continued from page :

Siamse network → 청가리한 살고리즘
 : 유사도 omega 대비 래시온 표현

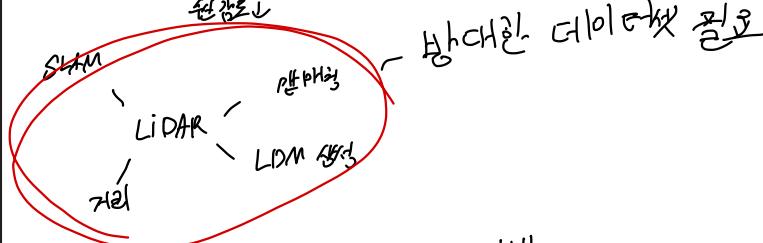
복설 기본
세컨드

LiDAR

기계식 : 전장부부
 solid state : optical phased array
 MemS : 센서



905nm → 1550nm
 고장 속도향향 원점으로



LiDAR DATA ←
 복설 기본
 (3차원간 복설 분화, 포인트 차이 인코딩)
 + 복설 포인트넷 분화
 포인트넷 기본
 (포인트넷 적용, 물체, 배경 포인트 분류)
 → 물체 해상도 보드 따로 처리

포인트넷
 포인트 RNN
 단계 : 대각적 움직
 단계 : 대각 추출

복설 포인트넷
흔한

PV-RCNN

2nd 차 데이터

종복한 표현 네트워크
제공

PCD : 순서대로 정렬되지 않음,
 → CNN 구조 사용하기 정렬되지 않음, 2차원 투영식 성능 저하.
 (증가화로 인한 정밀한)

물체 득점값 추출 필요

- 2차원 평면 투영 방법 (새로운 2차원 데이터 생성)
- 복설 기본표현 (작은 3차원 표현) : 각 복설에 있는 포인트들을
 대로 인코딩
 ↗ 포인트넷으로 인코딩한 벡터
- 포인트넷 사용 : 전체정렬
 대비/ 물체 정리
 ↗ 물체 추출,

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

LIDAR 데이터 물체추적

3차원 물체 추적 : 특징값 이용 유사도 측정

mm MOT 방법 : 포인트넷, 특징값 추출 : CNN 적용, 유사도 측정, 선형 프로그래밍

GNN BP MOT: LSTM 사용, 물체 순차적 움직임 분석, 그래서 구간 네트워크 사용

PCD의 특징

◦ 3차원 환경을 여러개 점群 표현 \rightarrow 다른 3차원 이미지

표현력 강점(별공한 표현 필요 없음)

◦ UNSTRUCTURED

unordered

Simple

permutation, invariant

CNN 적용 힘 \rightarrow (CNN은 주제와 관계)

◦ 3차원 표면드 표시, 색상, 노말 벡터로 구성

과제명

TITLE

과제번호

PROJECT NO.

Continued from page :

Velodyne VLP-16 LiDAR

- 16 카메라
- Accuracy $\pm 3\text{cm}$
- SW
- 탐지거리 0.5 ~ 100m

9103_ 3mm x 72.17mm

→ 실제 환경 데이터 처리 시스

Linux 16.04 LTS

ROS - noetic

on terminal

i) sudo ifconfig $\sim\sim$ 102.168.3.100

ii) sudo route add

102.168.0.202 (registered address)



LiDAR 설정 제작, 웹 접속

네트워크 설정

- WiFi-off

- IPv4

Address : 102.168.1.X

(X = 1 ~ 100 random)

Netmask : 255.255.255.0

Gateway : 0.0.0.0

Routes : use this

only for resources

on its network.

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

과제명

TITLE

과제번호

PROJECT NO.

Continued from page :

manual 엔 eth0 인터페이스

Mac UTM 가능 환경 : enp250

LG Linux 18.04 LTS : enp251

연결 케이블 : enp250 :

→ sudo route add 102.168.0.1 Factory address

→ 예기치 않음으로 연결에 실패함

102.168.0.202로 연결할 때 Connection timeout

: root user : 연결될 때까지 대기

본인은 30분 ~ 1시간 기다렸다 경험

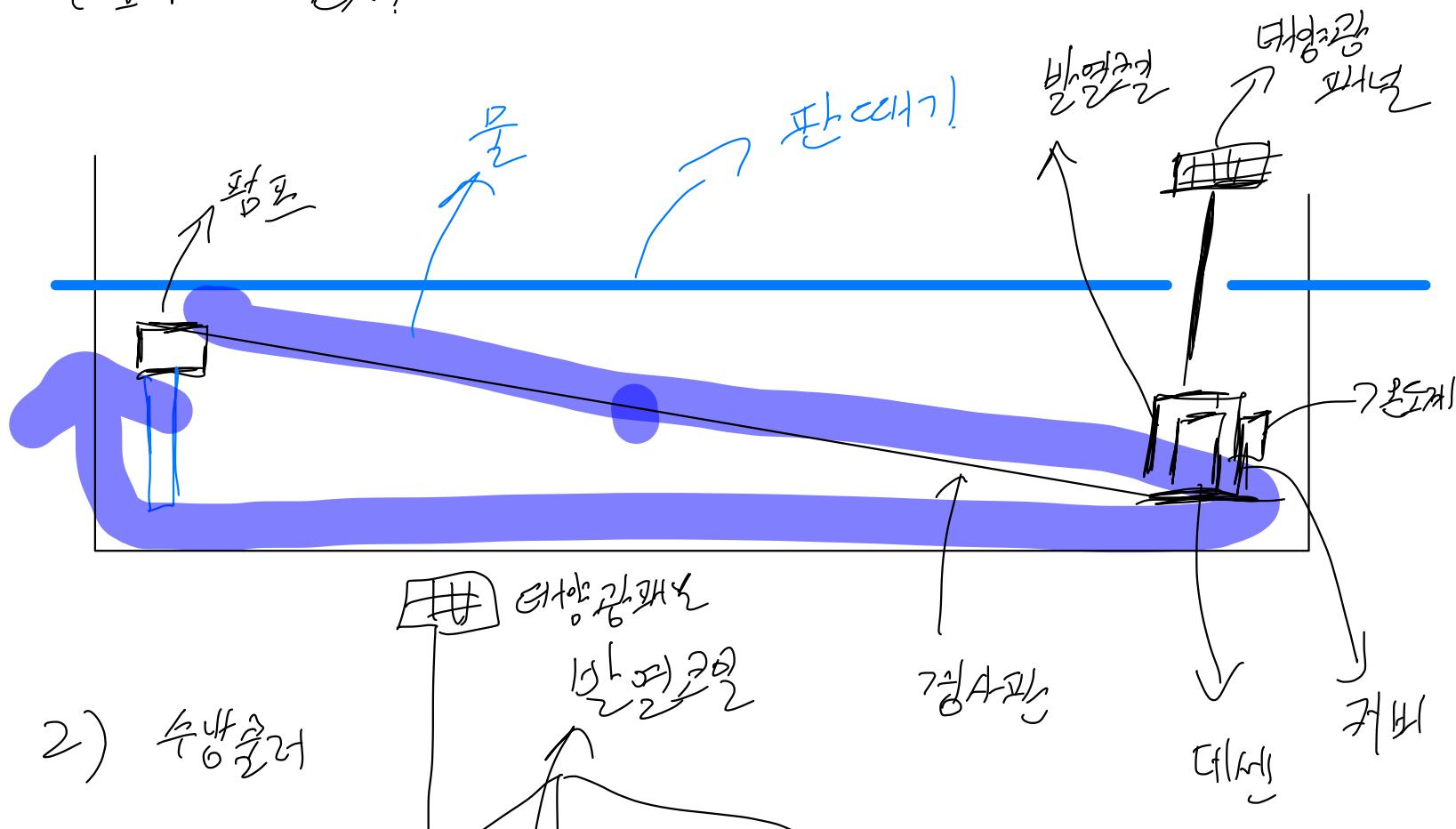
→ 아래와 같은 이유로 동일함

Continued from page :

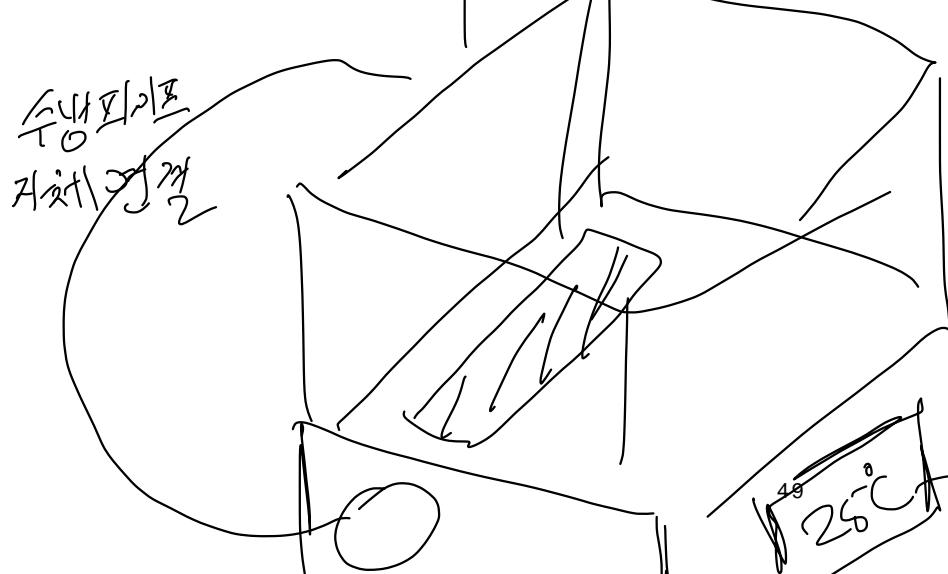
기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

1) 물 흐르기

(액체에서 물 것)



2) 수방출기



→ SC-320T 수방출기

외부는 대선처럼

구멍

온도 25°C

과제명

TITLE

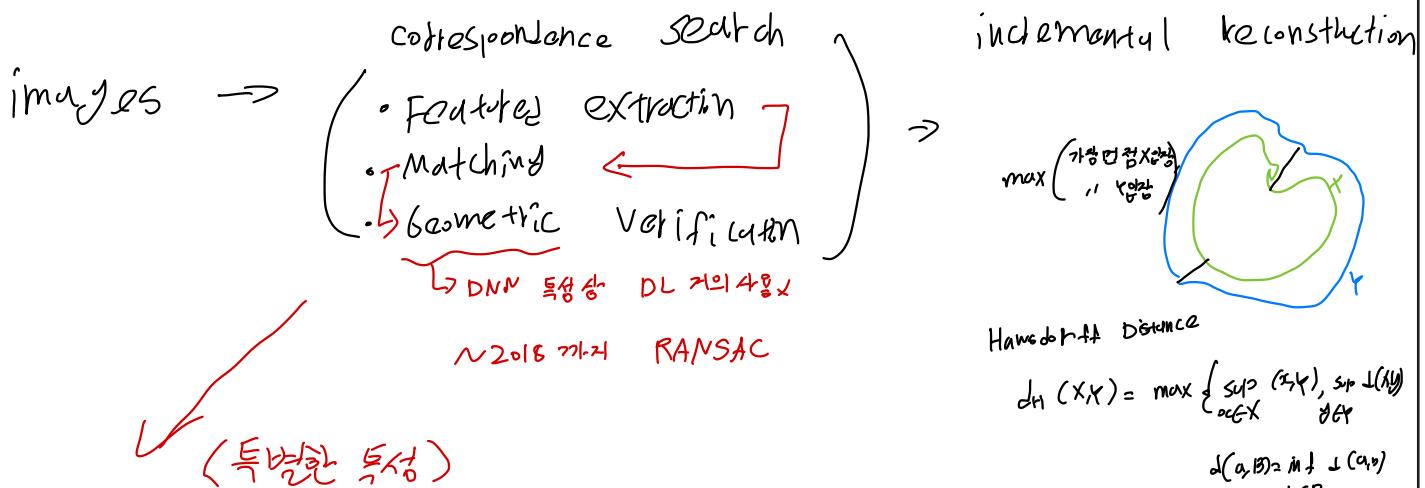
과제번호

PROJECT NO.

Continued from page :

LiDAR Point Cloud Processing ~ COLMAP

→ image 따로 처리해 특징 점을 빼고,
 모든 점가 비교 필요 없게 함.



후보 correspondence? 포인트 클라우드

→ 어느걸 먼저봐도 같은 결과가 나와야 함.

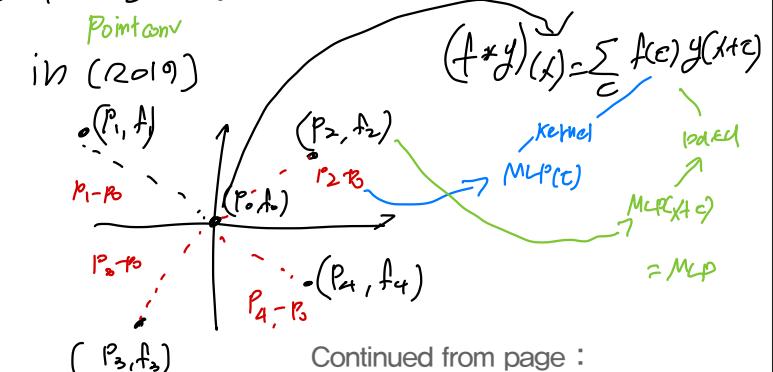
PointNet 활용 → 순서상관 없이 permutation invariant
 (2017) now pointnet++

최초 PC만 냄기기 (critical point sets)

↓ upnet bound-shapes : PC 개수증가 → 얼마나 물체인식을 못하나? → 학습 속도 향상 가능.

Bringing Actual Convolutions in (2019)

$$(f * g)(x) = \sum_c f(c) g(x+c)$$



기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

과제명

TITLE

과제번호

PROJECT NO.

Continued from page :

KPCorr(201n)

$$(f * g)(x) = \sum_{x_i \in N_x} f(x_i - x)g_i$$

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

과제명

TITLE

과제번호

PROJECT NO.

Continued from page :

우수한 연구원님 말씀으로 라이다 작동 성공함.. thank you..
 → Network Setting 및 IPV4에서 manual 대로 192.168.1.xx로 하지 말고
 LiDAR 주소가 192.168.0.202 까지 192.168.0.xx로 맞출것
 (~> ○ 으로 (1이 아니라))

기후 Raspiun/raspivid play 등 활용해 LiDAR 데이터 .bag 파일로 취득함

- play
 - . CSV → MIT 허브에 넣음
 - . PCD → rosrun pcl_ros

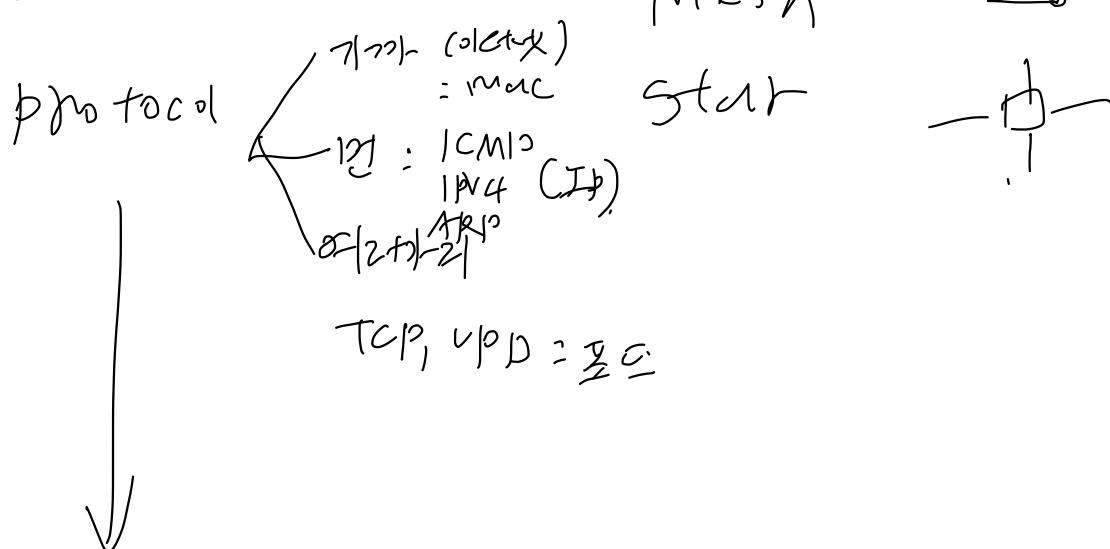
Pcd to image

- pcl-tools apt install
- Rosrun pcl_ros convert pcl_bags <파일명.pcd>

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

Network

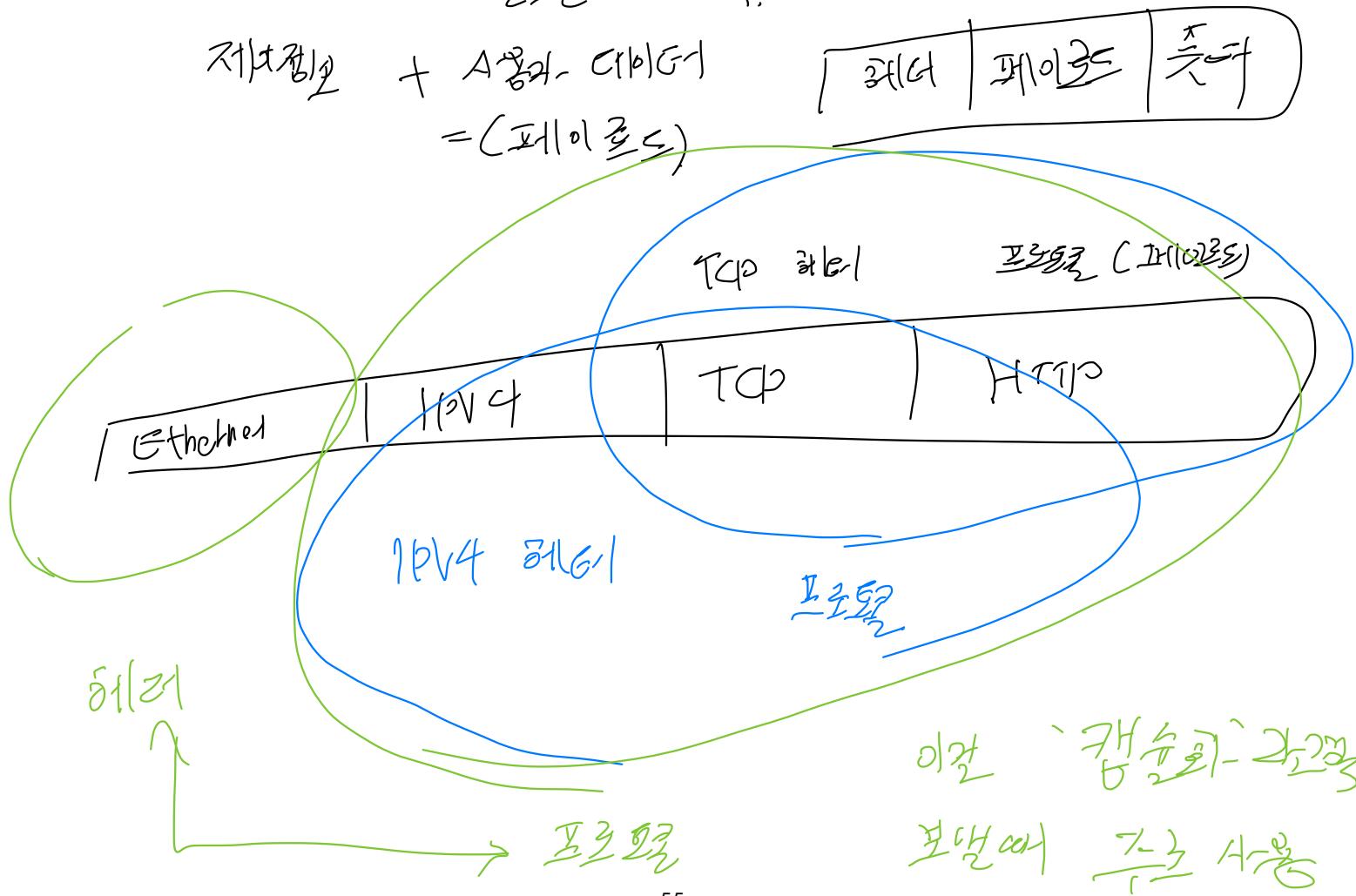
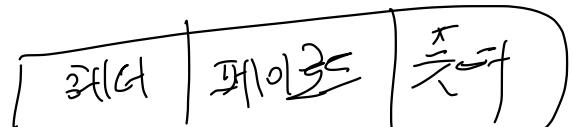


노드~노드 통신

마스터 데이터 주는 역할과 주는 역할

포함 : 네트워크 상 전송되는 데이터.

$$\text{제작점} + \text{A용자-cribge} \\ = (\text{페이지로드})$$



과제명

TITLE

과제번호

PROJECT NO.

Continued from page :

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

과제명

TITLE

과제번호

PROJECT NO.

Continued from page :

- CNN의 특징 : 2차원 이미지만 학습하기 좋다.
거널통과 계층적으로 지역적인 특징을 뽑는다.
- . 2차원 구조 : 정규적이지 못하니, point cloud : Unordered, permutation invariant.

→ PC를 PL로 적용하기 때문
 정규적 형태로 Voxel로 했을 때
 Viewpoint 상응하는 2차원 이미지 친합으로 빙글 끼쳤을 때
 핵심!!

→ 불필요한 empty space 제거 인식.

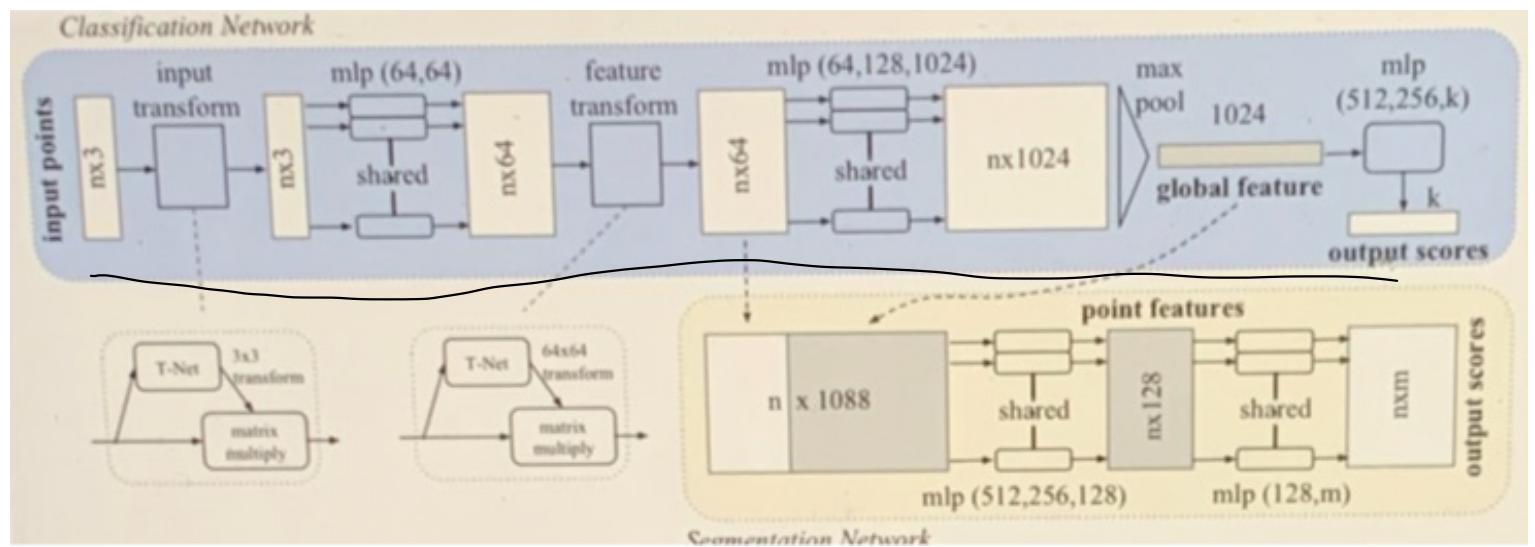
→ CNN, RNN → permutation invariant 하지 않음

PointNet

- ↑
PC : Unordered 한 입력을 전처리 없이 사용
(eg position, surface normal, Color)
- ↓
Classification
Segmentation

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date



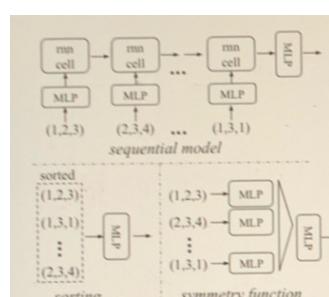
Point Set

- Unordered
- Interaction \rightarrow Local to Global
- Invariance

pointnet

- Max pooling
- Combination
- Alignment

부정적 인 오류율
정상 알맞은 예상



	accuracy
MLP (unsorted input)	24.2
MLP (sorted input)	45.0
LSTM	78.5
Attention sum	83.0
Average pooling	83.8
Max pooling	87.1

Unordered data processing

- 정렬 \rightarrow 고려할 수 있는 RNN \rightarrow 불가능한 X

- Symmetric function
- 2개

과제명

TITLE

과제번호

PROJECT NO.

Continued from page :

Joint Alignment Network

특성 추출에 앞서 input set 정렬

(교차원 이미지 정렬(각))

작은 크기 T-Net 통해 affine transformation Network 사용
 → 이를 입력해 적용

. 작은 이미지도 충적 변화 X

. 학습 포함 → 큰 영향 X

. 가로막 유실 등에도 큰 영향 X

T-Net의 적용

입력 포인트에 적용할 Transformation 추정

Feature 적용 Transformation 추정도 가능

정규화 적용 $L_{reg} = \|I - A A^T\|_F^2$

- ↳ 고차원 Transformation matrix A 를 orthogonal matrix로 변환,
- ↳ 정규화를 더 쉽게.

결론 : PointNet은 안정적으로 포인트 클라우드 Shape 정보를 확인할 수 있다.

Hierarchical point set feature learning
 Sampling layer → Grouping layer → point Net layer
 (Local region 학습)
 ↓
 Farthest point sampling 이용

PointNet ++

CNN처럼 전체 포인트 클라우드 영역 나누기,

(Neighborhood ball 구조)

- ↳ Center point 선정, 이를 기준으로 드는 ball 구조 주성

GCNN

(PointNet + GCN)

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

과제명

TITLE

과제번호

PROJECT NO.

Continued from page :

Point Cloud Matching.두 PC의 correspondence를 찾는 문제 \rightarrow (겹합 등 해결)

- PC 특징 추출

$$E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$

cross correlation function window func shifted intensity intensity

$$E(u, v, w) = \sum_{x, y, z} [I(x+u, y+v, z+w) - I(x, y, z)]^2$$

Open3D - ISS

3차원 shape keypoint 검출

$$\frac{\lambda_2(p)}{\lambda_1(p)} < r^1 \quad \frac{\lambda_3(p)}{\lambda_2(p)} < r^2$$

↓
point feature Histogram

Normal Aligned Radial Feature (NARF)

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

Iterative closest point (ICP)

$$\begin{cases} \text{point to point} & \min \sum \|y_n - \bar{x}_n\|^2 \\ \text{point to plane} & \min \sum (c_{y_n} - \bar{z}_n) \cdot n_d)^2 \end{cases}$$

Camera + LiDAR \rightarrow 0.21 ms

1단계 (가로망거)

YOLO RetinaNet RCNN

데이터 표현

카메라 : 카메라 좌표 (2차원)

Radar : 3차원 좌표

Sensor Fusion 유의사항

Calibration 필요

\rightarrow 차량이 달라서 one to one

방법 1
2번의 결과 \rightarrow 차체좌표
카메라 좌표 \rightarrow 3차원 좌표

같은 이유로 Calibration 필요

카메라 : 색상이

레이저 : $\begin{cases} \text{거리정보} & \text{제공} \\ (\text{색상} & \text{없음}) \end{cases} \rightarrow$ 색상으로 영향 받음.

레이저 흐름으로 차수 결정

레이저 흐름 \downarrow : Camera Focus

" " : LiDAR Focus

Camera + RADAR + LiDAR : 초점 합친 전략

\rightarrow 가로망 모드

63 알고리즘, HW 구조

과제명

TITLE

과제번호

PROJECT NO.

Continued from page :

+ 고성능 차전력 고속 충전

고성능 인지 요구사항 : 대용량 센서 데이터 처리 필요
 차량 주변 환경을 통한 차량 위치
 차량 데이터 속도와 처리 "

HW

전동적 차동학 : FCV 형태

↓
 차화 필요, MN 핵심 기능 가능
 (차량용 고속 네트워크)
 HW 성능 향상

센서 정보 VS LDM 비교

매칭

↑
 LPM을 통해 '정적 정보'와 같은 인식강도를 줄일 수 있음.

↓ 요구사항

주로 인도 도로 축출 경로
 → 최대 0.2M 이하
 → 차량의 위치를 매우 정확하게 알 수
 → 정지간 접두어로 정하고 → 그에 따른 출발지에서 완리

SW

- Android 같은 소프트웨어
- HW, DL 모델 정합화 보조

통신 플랫폼

V2X (Vehicle, Infra, Etc.)
 + DSlink (보통화)

MMS 이용

고정밀 GPS 미세정교

→ 측위)만

이거면 된다.

+ GPS

차량 위치 추적

+ RTK

높은 정밀도

Map Matching process (LPM 사용) 정확도 고증 How?
 Sensor Data Analysis

i) Odometry

→ 차량이 과거 위치로부터 얼마나 움직였는지?

고정밀 GPS에

ii) Map Matching

측정된 GPS와,

Map Matching Algorithm

이거만 가능.

MMS system

DGPS : 2차 이상 GPS 신호 이용

LibGPS

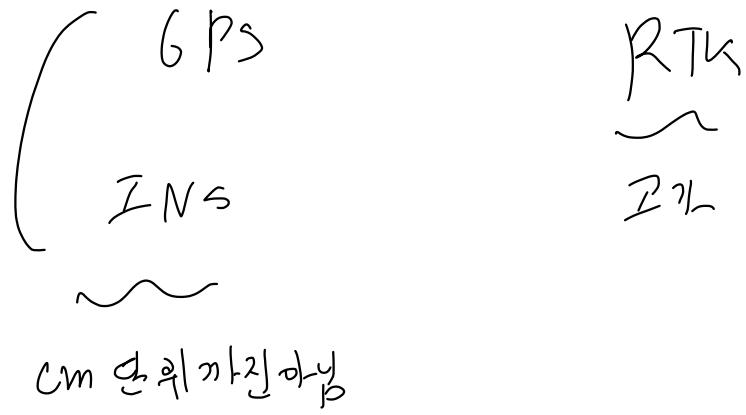
DMI : 차량 표전수

INS : 관성장치

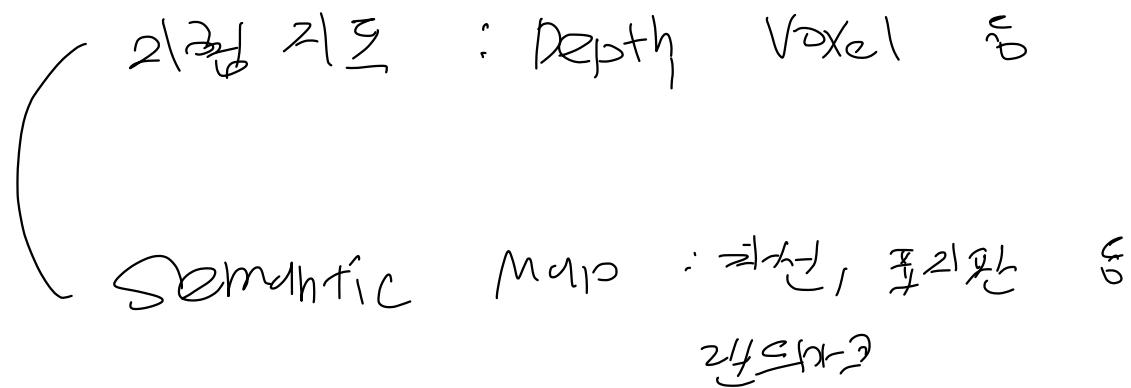
Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

측위 위한 장수)



측위 위한 213



Map Matching Algorithm \rightarrow AI 척도

과제명

TITLE

과제번호

PROJECT NO.

Continued from page :

Voxel 기반 방법 (세간드 포인트 필라)

i) 복선 구조 기반 포인트 대비하여 알고리즘 → 3차원 CNN 적용
↓

초감도 영역 특징 인식

(고전 : 3차원 PCN을 Bird Eye View로 두었)

ii) point RCNN

1단계) : 물체, 배경 point 구분

2단계) : 대상적 유의에 해당하는 조건으로 추출

iii) Voxel - point 혼합

PD - RCNN : Library 대비 더 끝부까지

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

과제명

TITLE

과제번호

PROJECT NO.

Continued from page :

카메라 — LiDAR Sensor Fusion

영상에 CNN 적용 → 물체에 대한 2차원 맵드 짐출

↓
프레스팅 : 2차원 빙스에 대응되는
3차원 광학

프레스팅 안에 존재하는 리아디 토이드 흐름

↓
포인트넷 등 인라닝 모델로 학습

↓
3차원 광학 정보 얻어냄

프레스팅 포인트넷 (CVPR 2018)

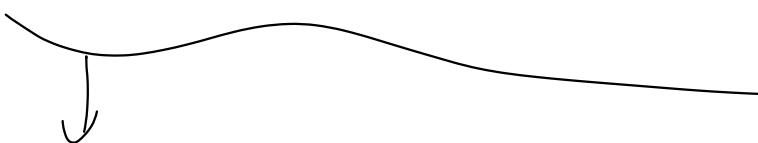
카메라 영상 → 2D CNN → 2차원 맵드 → 프레스팅

3차원 광학 분할 : 포인트넷 사용. 물체만 해석하는
파라미터 추정

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

CNN으로 학습
기미학 특징 many → 3차원 공간 특징
+
라이아 얼굴
특징값

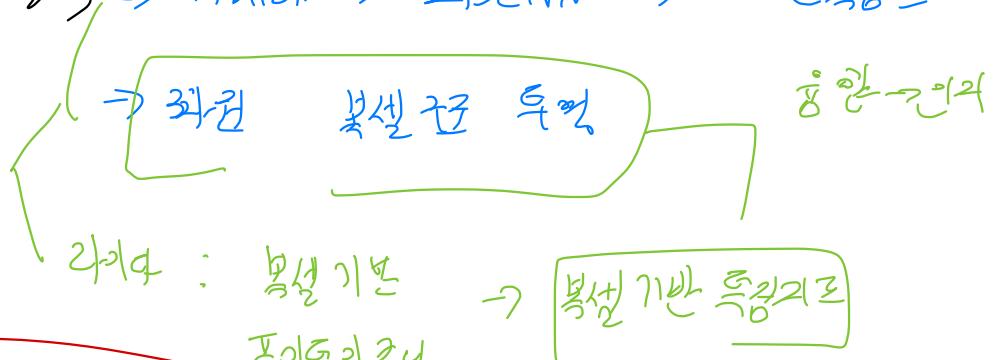


AV泊 (ICROS 16)

Cont Fuse (ECCV 16)

MMF (CVPR 10)

3D CV特征 (ECCV 20) → 기미학. → 2DCNN → 2차원 특징값



이전 알려지지 않은 현상

P2P CAN BUS

SOAOS

터널링 차의 자동차 통신

네트워크

공유화

과제명

TITLE

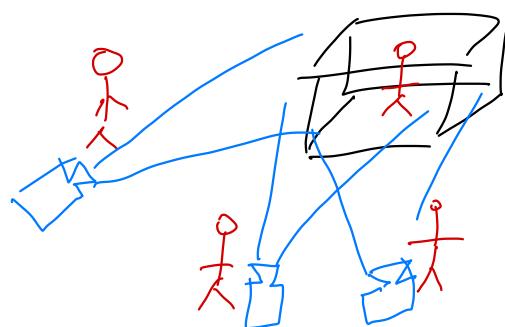
과제번호

PROJECT NO.

Continued from page :

3D vision

How?



모든 View의
Camera 시점으로
일부 View 만으로
구현, 표현 하는 것

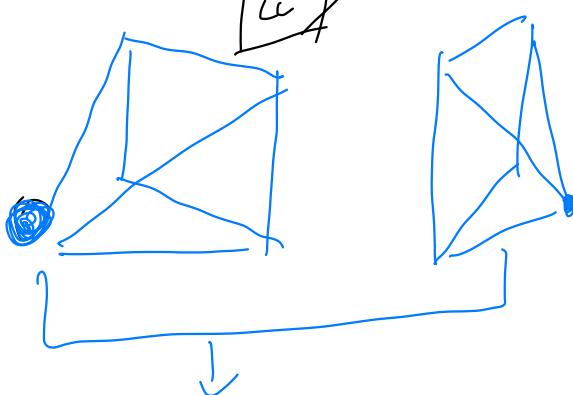
Black-Box Model

→ 아웃풋은 인간과 비슷,
인간은 그것을 보는 데



Building

2D First



Where are those Cameras?

Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

Homogeneous

coordinates

parametrizing a line

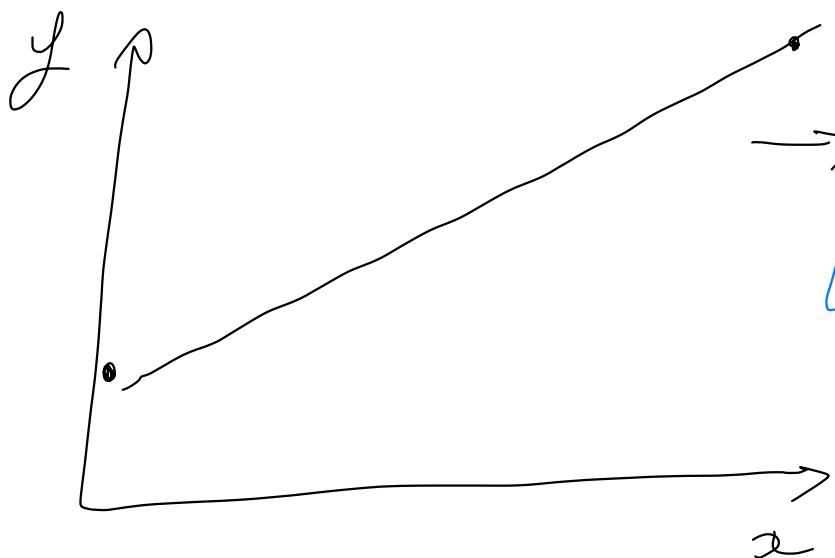
방정식 표현 가능

$$[a, b, c]^T [x, y, 1] = 0$$

$$[kx, ky, k]$$

▶ K점에

결과 변화 X.



Parameter

$$(a, b, c)$$

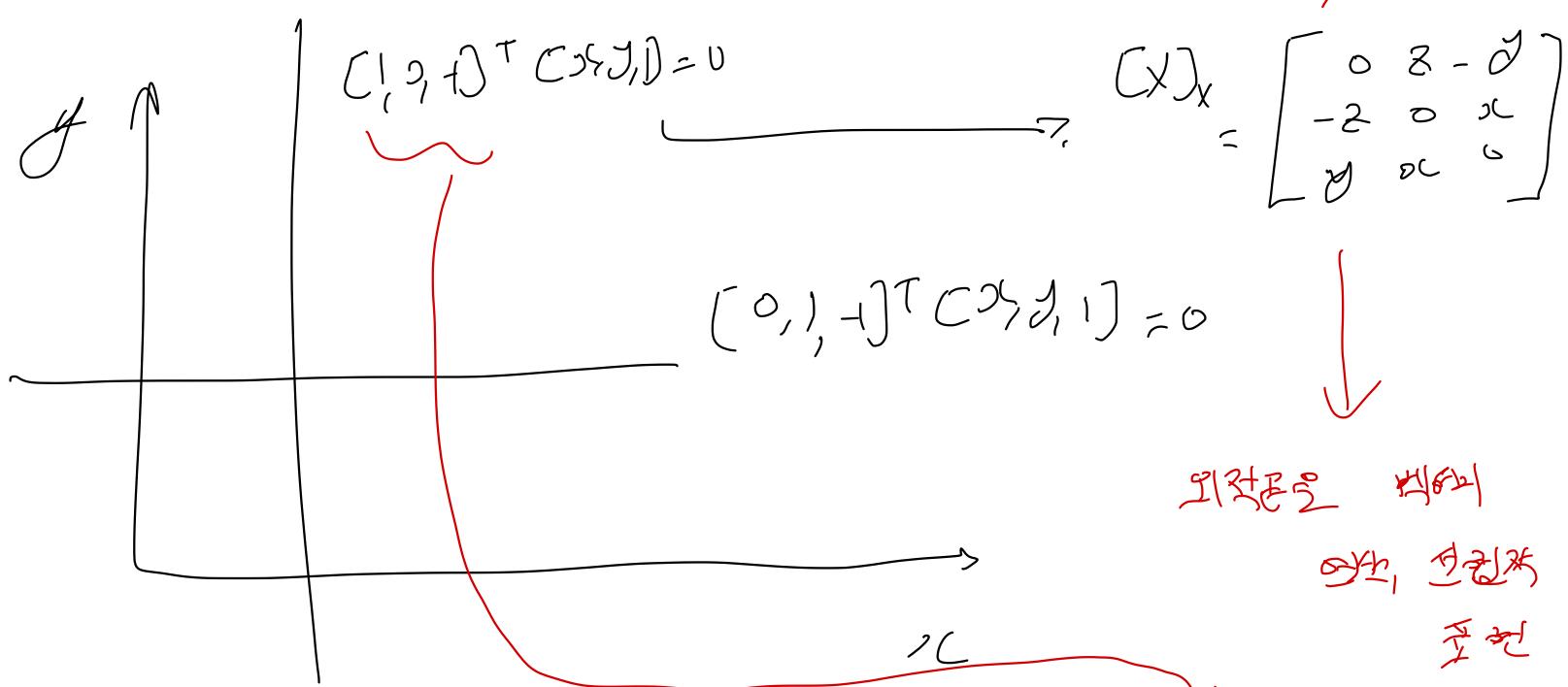
point

" $X = [x, y, 1]$ in homogeneous coordinates"

$$[a, b, c]^T [x, y, 1] = 0$$

- - - line "l = [a, b, c]"

스무스 미끄럼



외적으로 벡터

벡터, 선형대수

표현

다른 벡터

$$\begin{bmatrix} 0 & 1 & -1 \\ -2 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

: 외적을
표현하는 표현

과제명

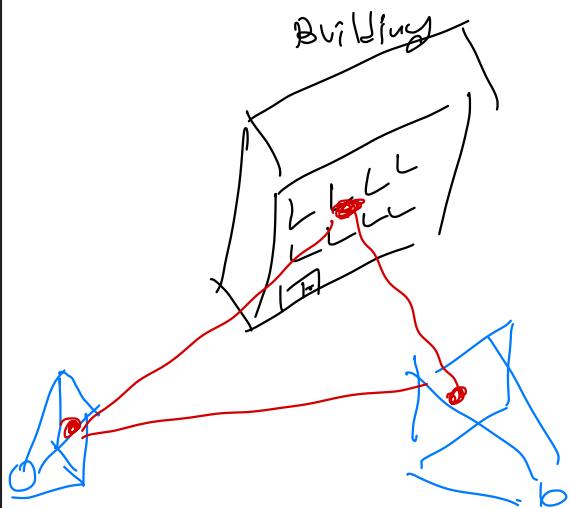
TITLE

과제번호

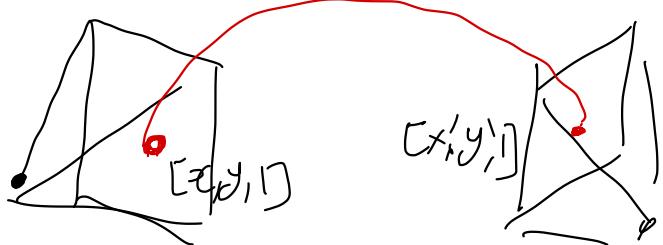
PROJECT NO.

Continued from page :

Homogeneous Coordinates



2D Homography

2D \rightarrow 2Dd₂ 2D \rightarrow 

3차원을 가지지 않고

2D로 표현 가능

$$x' = H x \quad \text{Proj transformation}$$

$$\begin{aligned} x'_1 &= x_1 + t_x \\ x'_2 &= x_2 + t_y \end{aligned}$$

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

4DoF) 2D로 2D로
1DoF) 2D로 2D로Affine transformation
회전, 평행 移動

Continued from page:

기록자 Invented by

점검자 Witnessed and Understood by

점검자 Witnessed and Understood by

일자 Date

일자 Date

일자 Date

Direct Linear Transformation (DLT)

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} h_{11} & \dots & \dots \\ \vdots & \ddots & h_{33} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \quad x' = Hx \quad \rightarrow \text{이제 } Hx = 0$$

$$h_1 = \begin{bmatrix} h_{11} \\ u_{21} \\ u_{31} \end{bmatrix}$$

$$\underbrace{x'}_{2\times 3} Hx = 0$$

$$\Leftrightarrow [x]_x Hx = 0$$

↓

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = (h_1 \ h_2 \ h_3) \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$\mathcal{L} = (Ah)^+ (Ah) + \lambda (I - h^T h)$$

Solving SVP with $Ah = 0 \quad s \in \|h\| \gamma$

$$\Delta_h \mathcal{L} = 2A^T Ah - 2\lambda h$$

Homography : 다른 위치로 촬영 필요,,
이미지 | 이미지

과제명

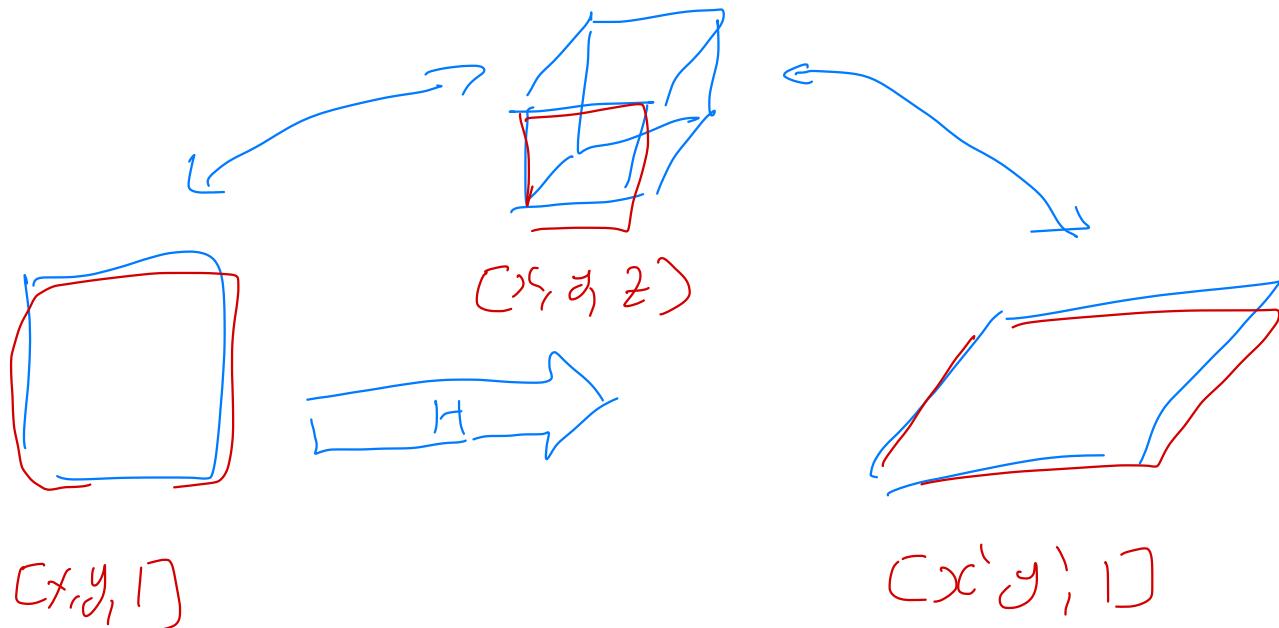
TITLE

과제번호

PROJECT NO.

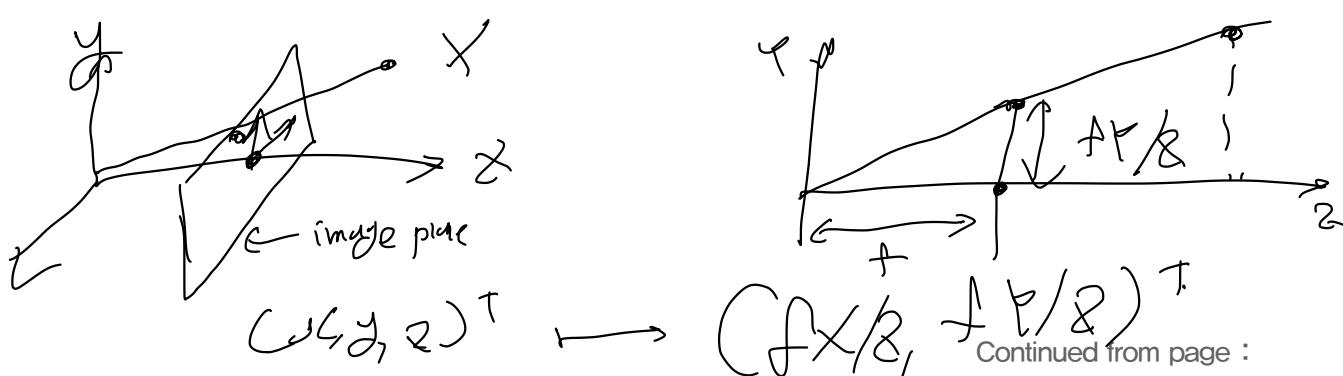
Continued from page :

From 2D to 3D



3D원 2D원 표현 해야 함...

Principle camera ~ (ISAS Year)



기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

$$(x, y, z)^+ = \underbrace{(Ax/2 + By, Ay/2 + Bz)}_{\frac{1}{2} \text{만 } x \leq 0}$$

3D T₀ 2P

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} Ax + 2By \\ Ay + 2Bz \\ z \end{pmatrix} = \begin{pmatrix} f & 2B & 0 \\ f & 2B & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$X = k[I|0] \underbrace{X_{\text{curr}}}_{\sim 2D \text{ 2D}} \quad \sim 3D \text{ 3D}$$

Radial Distortion

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = \mathcal{L}(r) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}$$

$$p_x = p_y \rightarrow \alpha y \\ \rightarrow \text{hologram}$$

\rightarrow 2D 2D

$(x, y, 1)$

(x, r, z)

$$X = k[I|0] X_{\text{curr}}$$

과제명

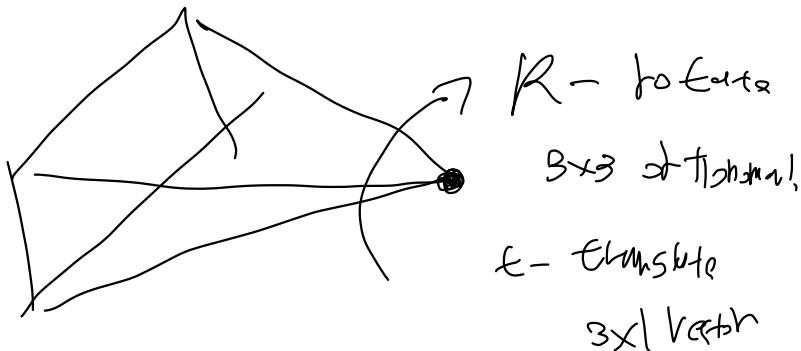
TITLE

과제번호

PROJECT NO.

Continued from page :

$$X_{cam} = [R|t] X_{world} \rightarrow \text{Simultaneous Localization and Mapping (SLAM)}$$



$$X_{world} = [R_{C2W} | t_{C2W}] X_{cam} : 6 \text{ dof}$$

$$R_{w2C} R_{C2W} = I$$

$$\boxed{H = K[r_1, r_2 | t]}$$

$$R_{w2C} t_{C2W} + t_{C2W} = 0$$

→ Corners
Calibration

Calibration

$$R X = K[r_1, r_2, r_3 | t] X$$

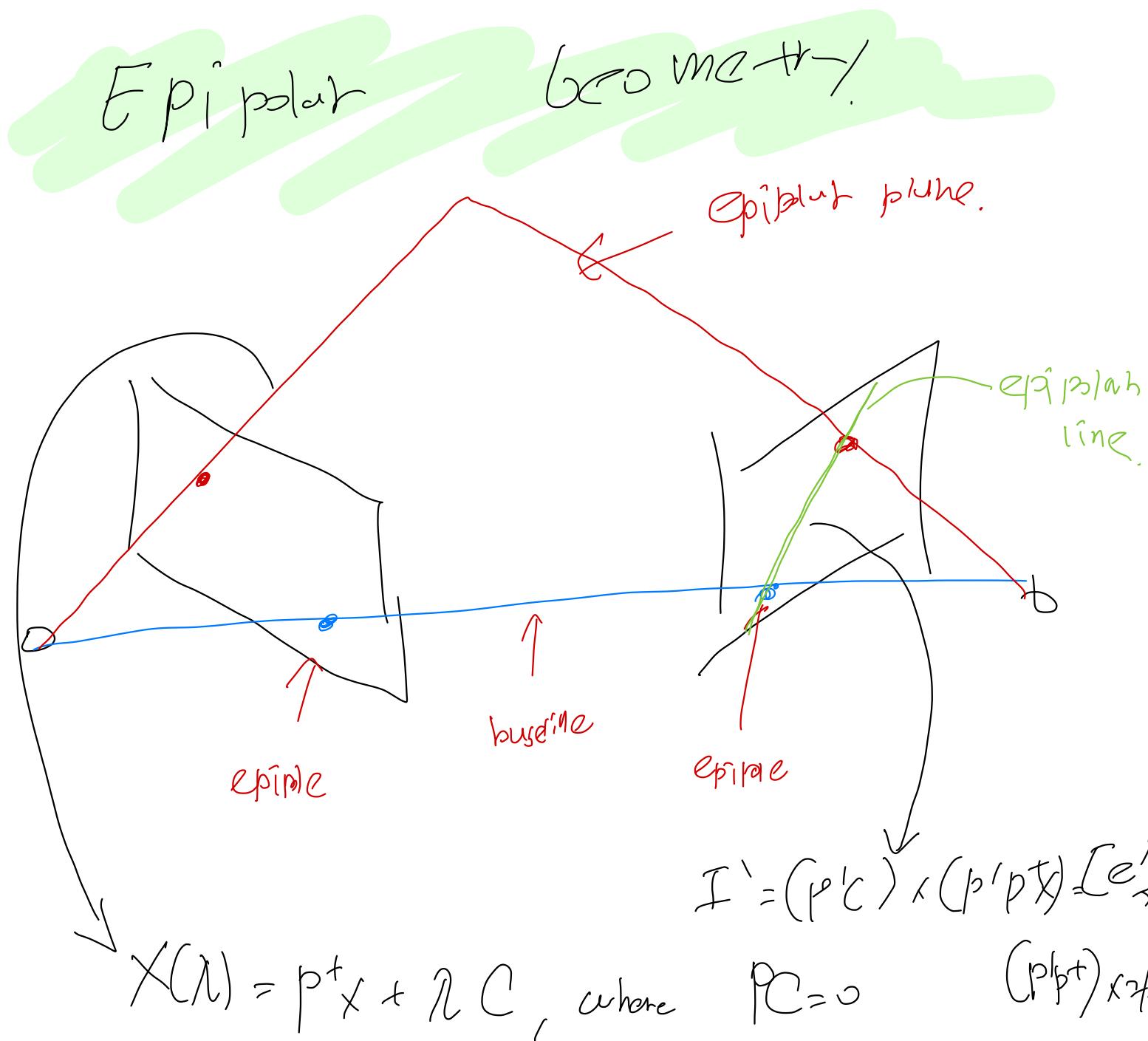
Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

Leverberg Mat d'and f. $\rightarrow \hat{x}_1 \hat{x}_2 \hat{y} \in$ Calibration

$$h_1^T K^{-T} K^{-1} h_2 = 0$$

만족하는 K 찾기 \rightarrow open CV $\approx 1\%$

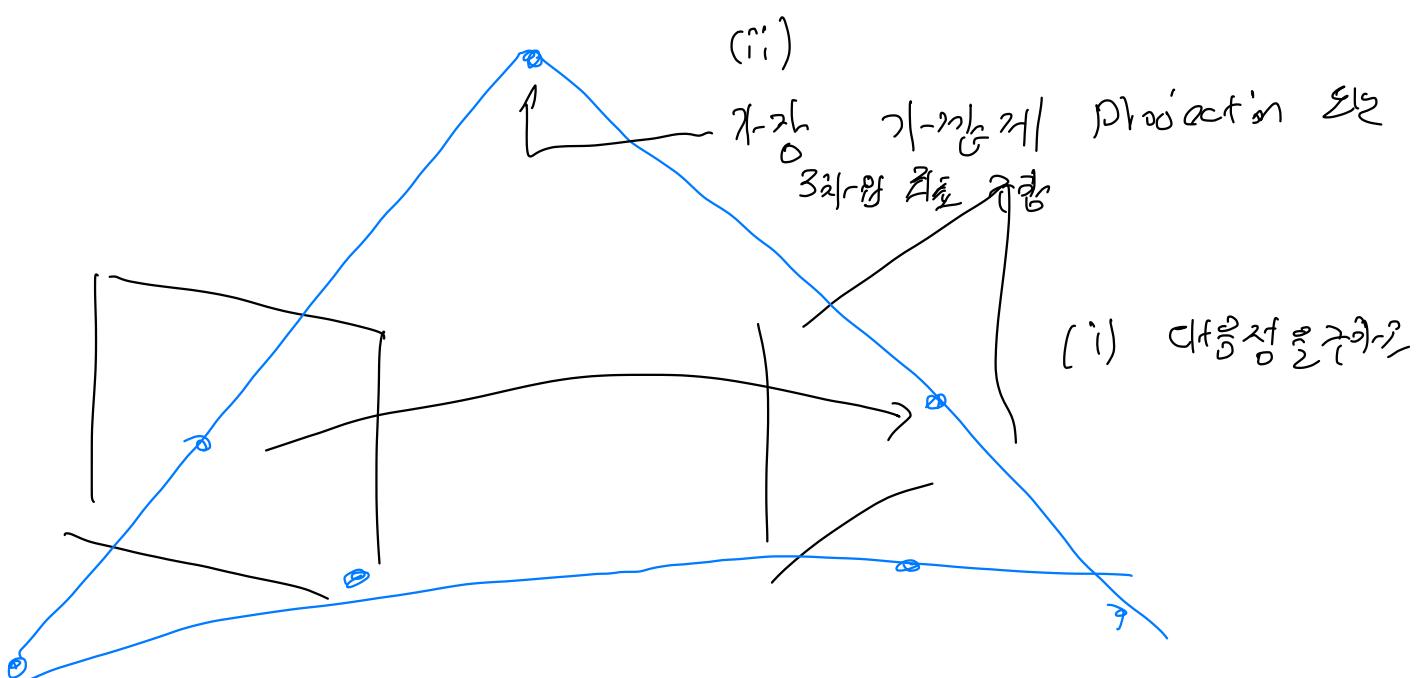


과제명
TITLE

과제번호
PROJECT NO.

Continued from page :

Diagram illustrating matrix multiplication $T = T^T F^T +$ by setting up a 2×2 matrix. The matrix is shown as a rectangle with vertices labeled with coordinates: top-left $(0, 405)$, top-right $(154, 405)$, bottom-left (10) , and bottom-right $(154, 0)$. An arrow points from the matrix to the equation $T = T^T F^T +$.



Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

COLMAP 2017 'ChongJin' 편집

ChongJin point cloud type
↳ 2017 sketch point net

point net part segmentation : object sketch?

과제명

TITLE

과제번호

PROJECT NO.

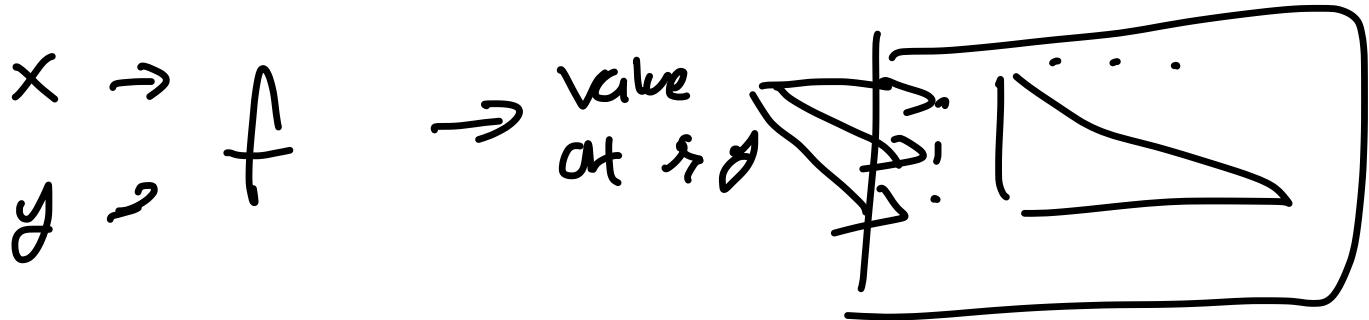
Continued from page :

Why NeRF?

Camera : 3D to 2D $(x, y, z) \rightarrow (x', y')$

NeRF : 2D to 3D (Novel View Synthesis)

2007: Compositional Ray-tracing rendering methods



2019: occupancy networks

 \rightarrow implicit (어디에 있는지가- 차라- 알수 / 알수)

$$f_f : \mathbb{R}^3 \times \mathcal{X} \rightarrow [0, 1]$$

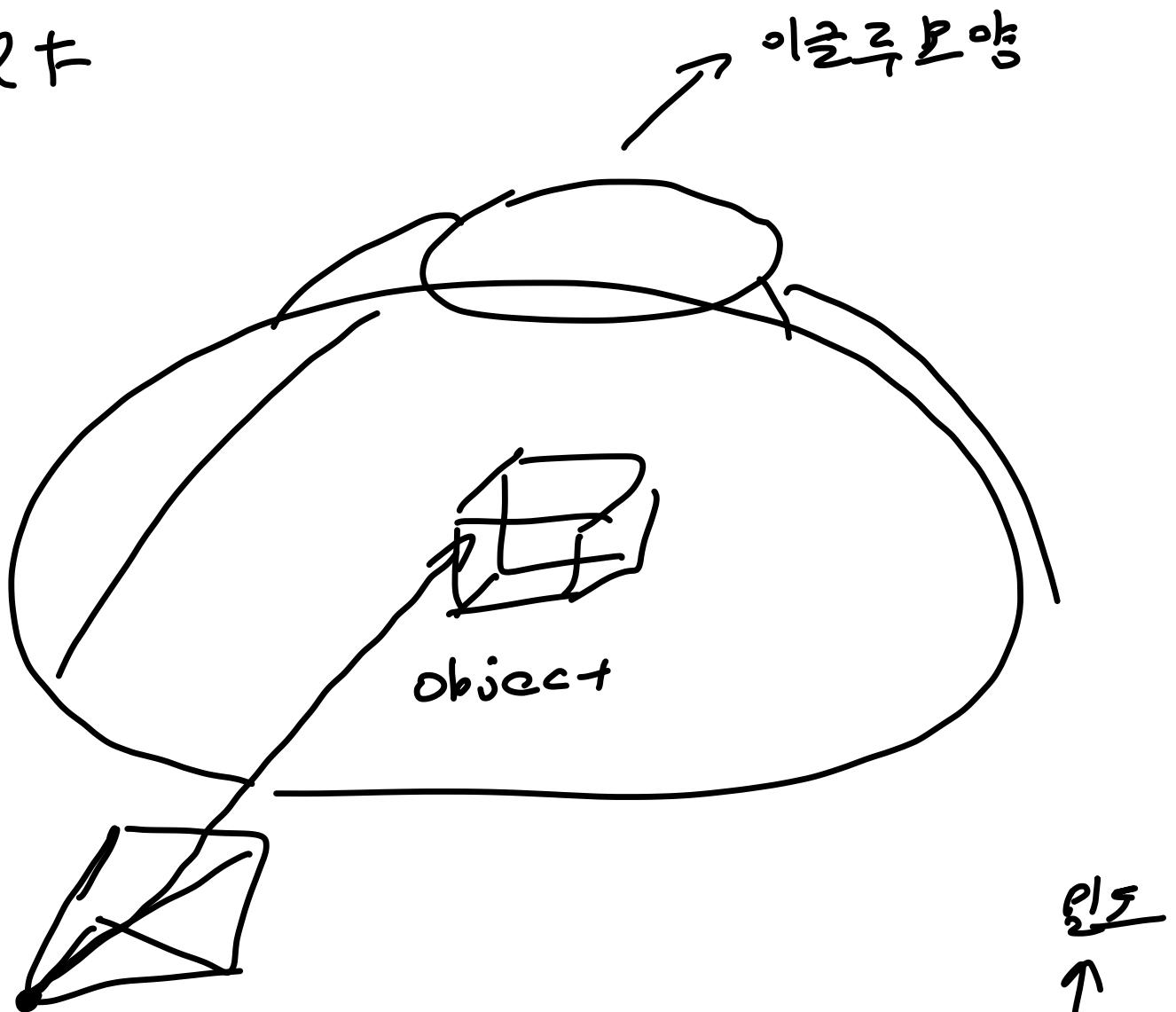
↗ ^{DN}
 f_f : \mathbb{R}^3 $\times \mathcal{X}$ $\rightarrow [0, 1]$
 ↘
 obsrv out in
 3D条件

Continued from page :

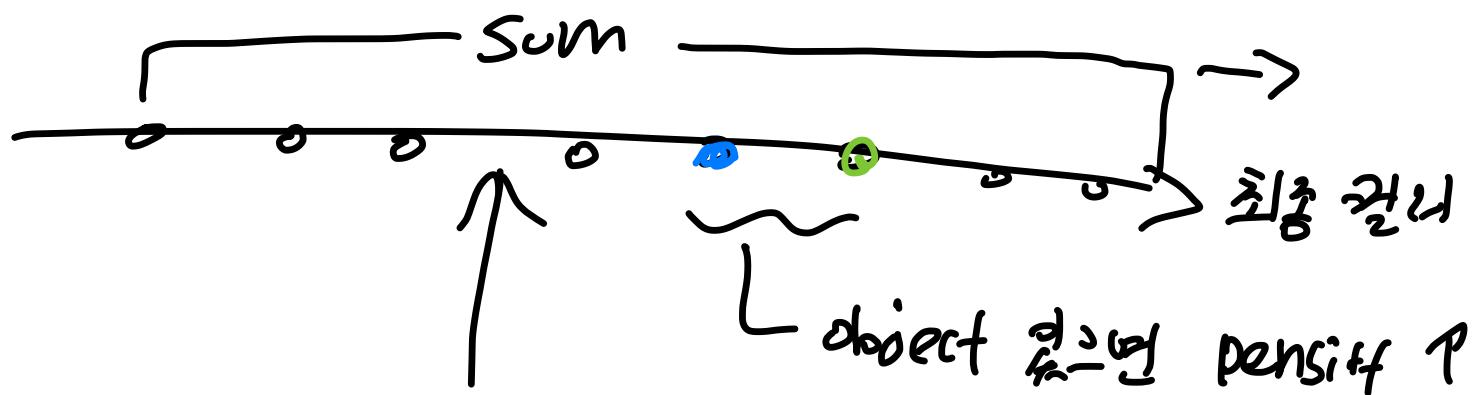
기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

NeRF (Volume Rendering) VS SRN (NN only)

NeRF



$$(x, y, z, \theta, \phi) \rightarrow f \rightarrow (R, G, B, \sigma)$$



Raf (7-2)

과제명
TITLE

과제번호
PROJECT NO.

Continued from page :

$$r(t) = o + \epsilon \zeta$$

$$C(r) = \int_{\epsilon n}^{t_A} T(t) \delta(r(t)) C(k(t), \zeta) dt$$

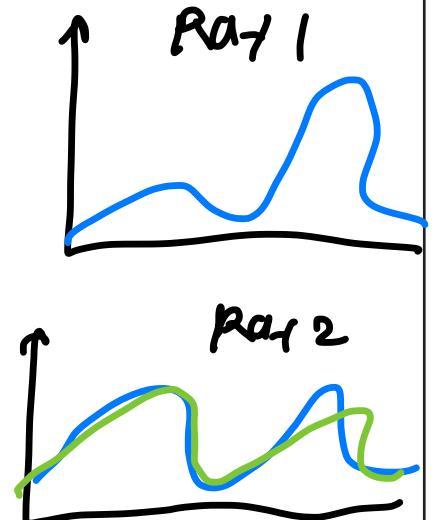
Density rate car

$$= e^{- \int_{\epsilon n}^{\epsilon} \delta(s) ds}$$

volume rendering

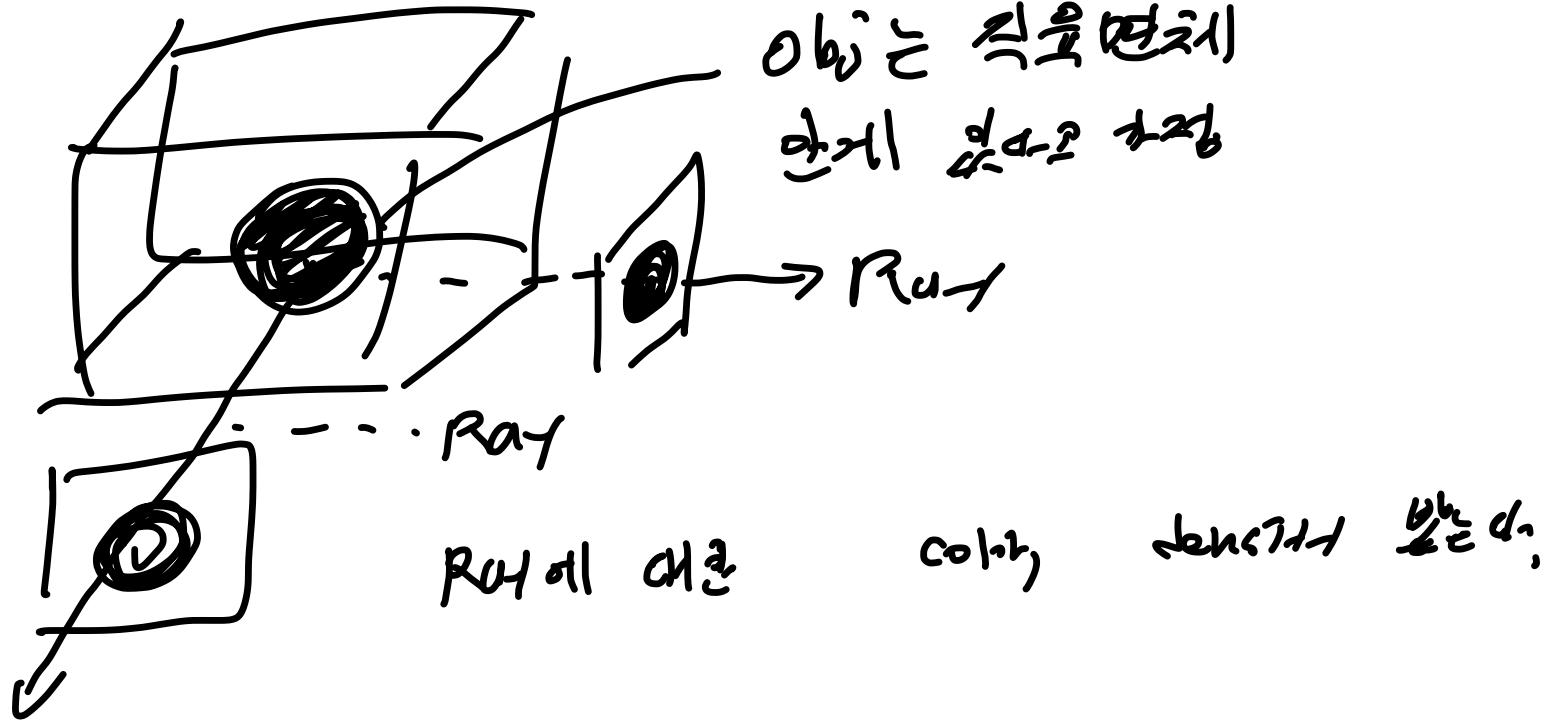
SD input

$$(x_0, y_0, z_0, \theta_0, \phi_0) \rightarrow (R, G, B)$$



Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date



$$r(\epsilon) = o + \epsilon d$$

$\hat{C}(r) = \int_{\epsilon_0}^{\epsilon_f} T(\epsilon) \delta(C(\epsilon)) C(\epsilon) d\epsilon$

원점 o

$$F_\theta : (X, d) \xrightarrow{\text{MLP}} (\mathcal{C}, \sigma)$$

Color Density

Volume rendering

$$r(\epsilon) : r(\epsilon) = o + \epsilon d$$

$$C \approx \sum_{i=1}^N T_i \alpha_i C_i$$

weights

Color

pixel value

$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$

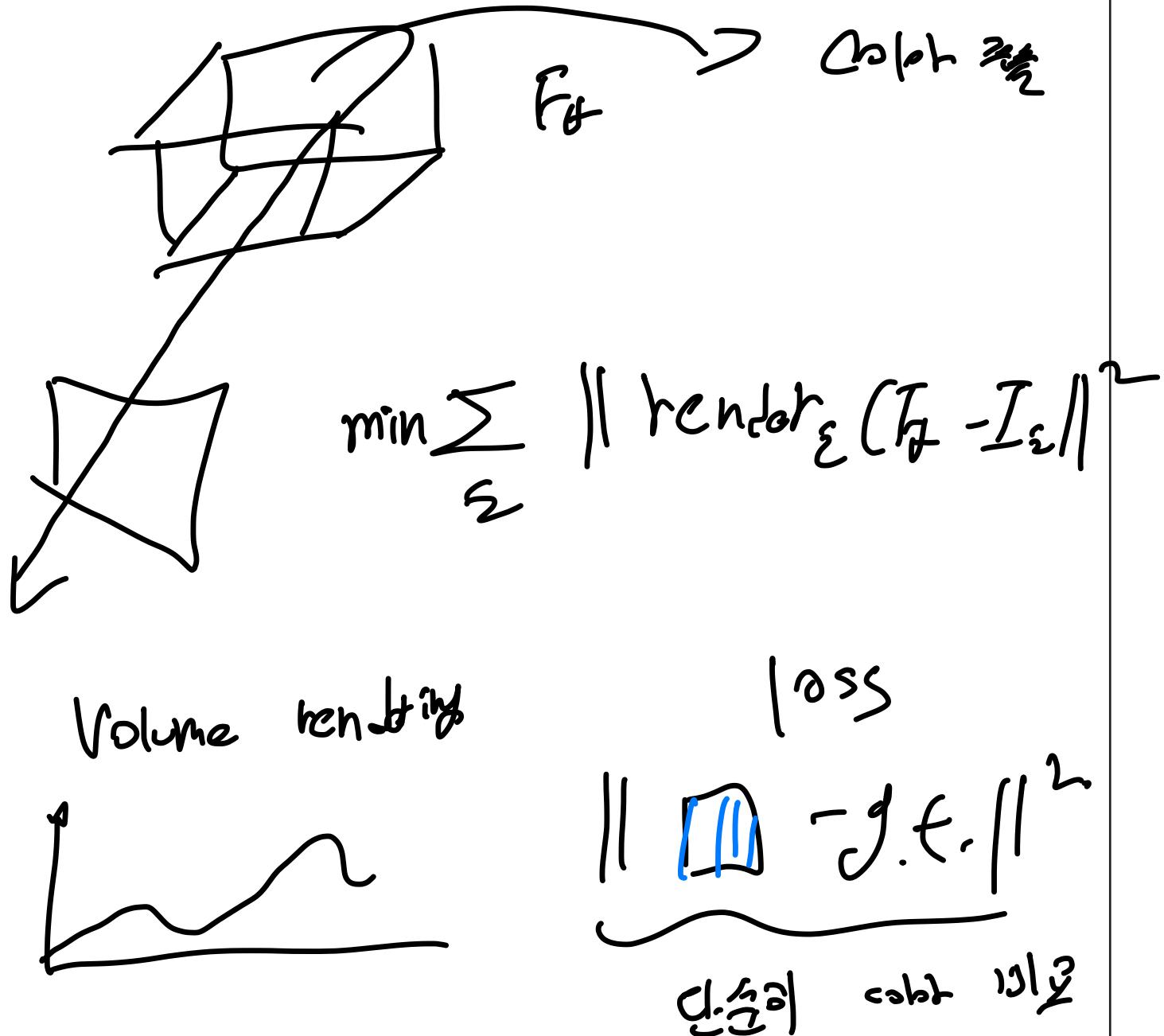
과제명

TITLE

과제번호

PROJECT NO.

Continued from page :



Continued from page :

기록자 Invented by	점검자 Witnessed and Understood by	점검자 Witnessed and Understood by
일자 Date	일자 Date	일자 Date

NeRF 개선 연구 (속도) -(Mip - NeRF)

기존 NeRF : 3x-4x Pixel, 빠른 ROI.

: 각각 다른 사건 학습시 렌더링 오류,
(해상도 등) aliasing 발생

→ Super sampling은 해결, but
Not too effective cuz NeRF uses MP

NeRF : Sample 때 camera 관찰 영역
거리가 일정해야 출 차중

Mip - NeRF : 거리에 따라 광센크기 조절
(거리마다 1번 : canonical from 각각 위치)
,, 가까이 : > " 각각