Acquisition d'un flux d'images RGBD et Détection des informations 3D des points clé du squelette humain

Curtis Martelet

Tuteur : Wanchen Li

SOMMAIRE

Objectifs

- Migrer la Node enregistrement de IAI_Kinect2 de ROS1 vers ROS2
- Se familiariser avec Openpose
- Comprendre le fonctionnement de Openpose_ROS2
- Développer des environnements cloisonnés pour les différentes tâches

Bimanual Agile Zany Anthropomorphic Robot - BAZAR

Caméra Kinect v2

Récapitulation du Projet S1

1

CALIBRATION

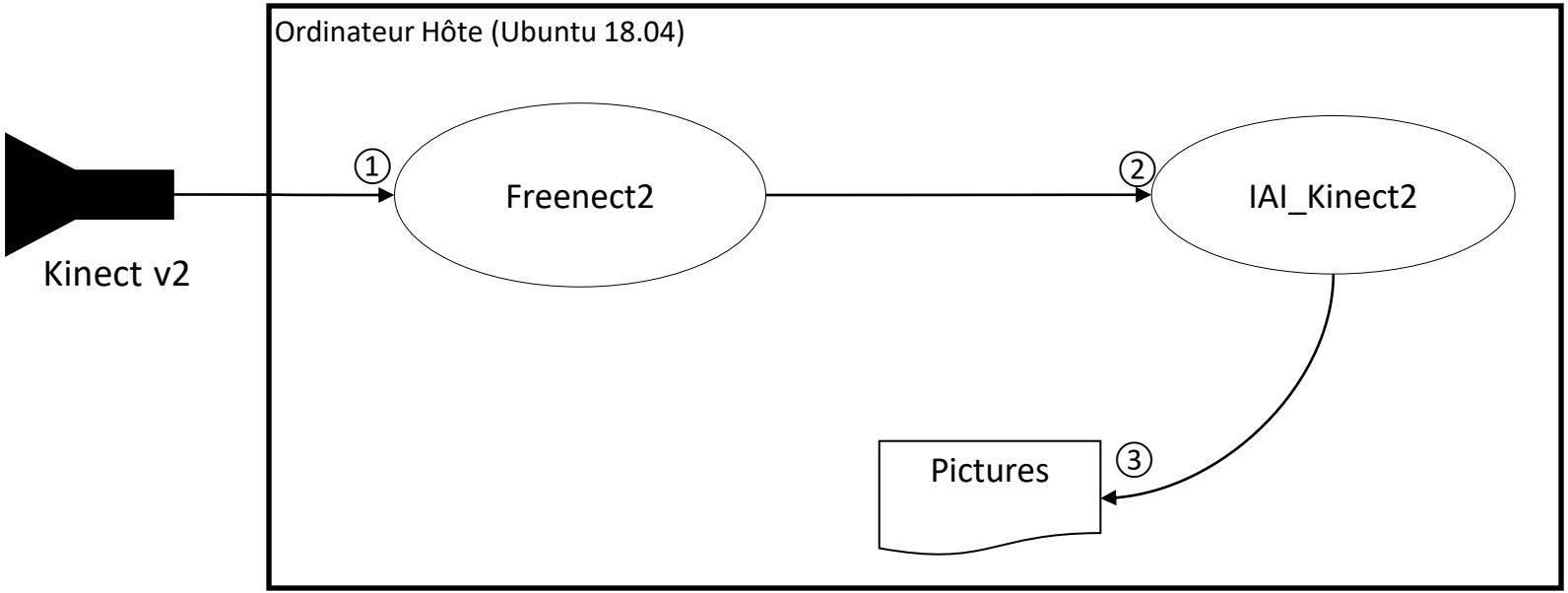
Calibrer la caméra.

2

CAPTURE & ENREGISTREMENT

Enregistrer le flux d'images puis le sauvegarder sur le disque dur.

Schéma de fonctionnement du Projet S1



Projet S2

1

CAPTURER

Capturer
l'environnement.

2

TRAITER

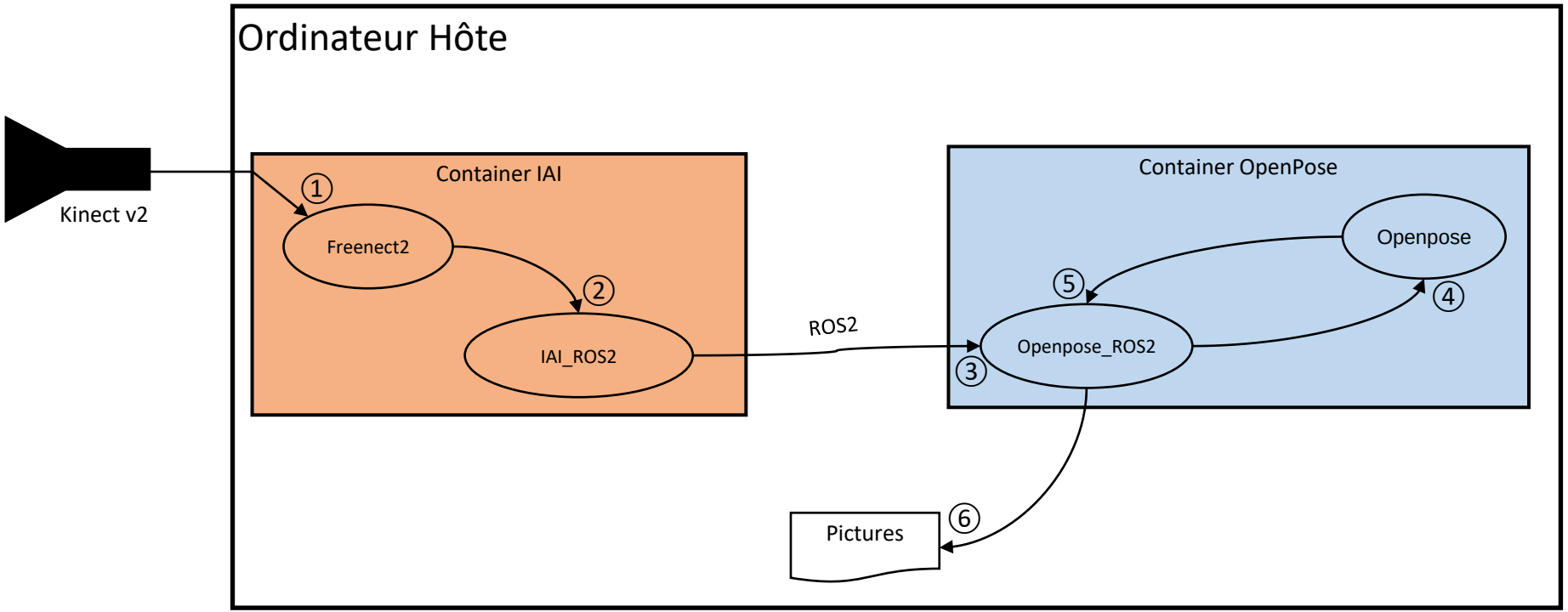
Extraire la position
des joints.

3

ENREGISTRER

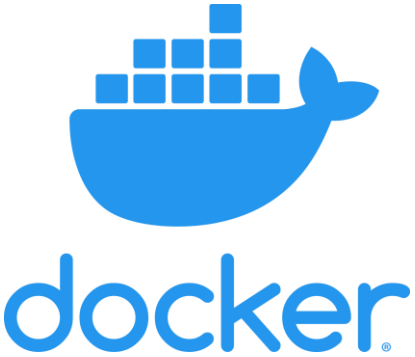
Enregistrer les positions
sur le disque dur.

Schéma de fonctionnement du Projet S2



Langages

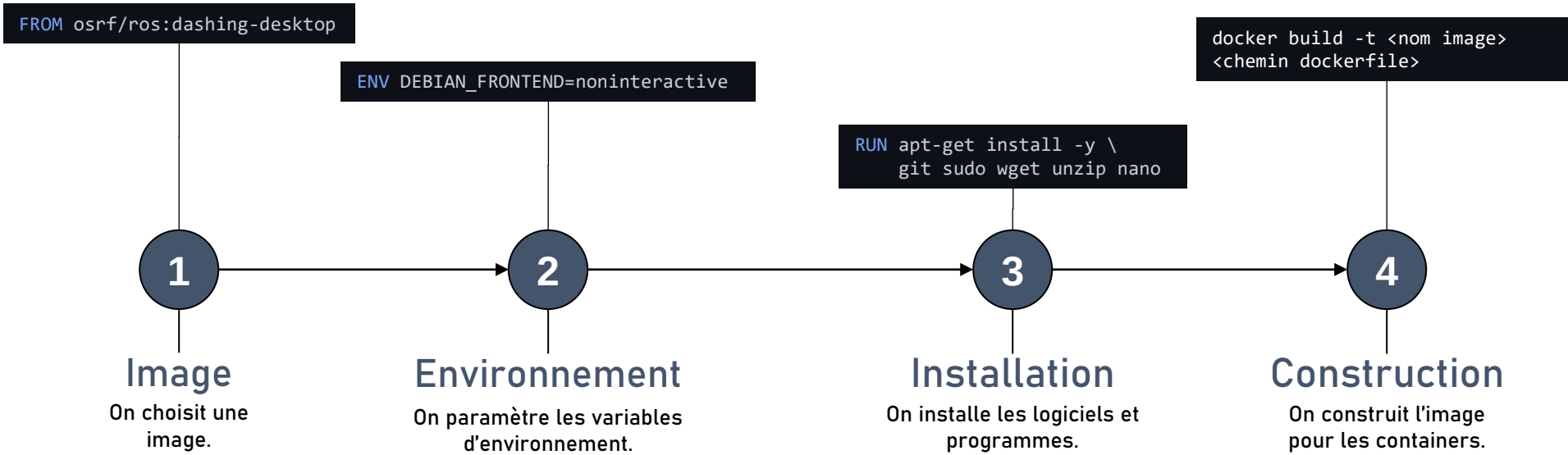
- Performant.
- Populaire.



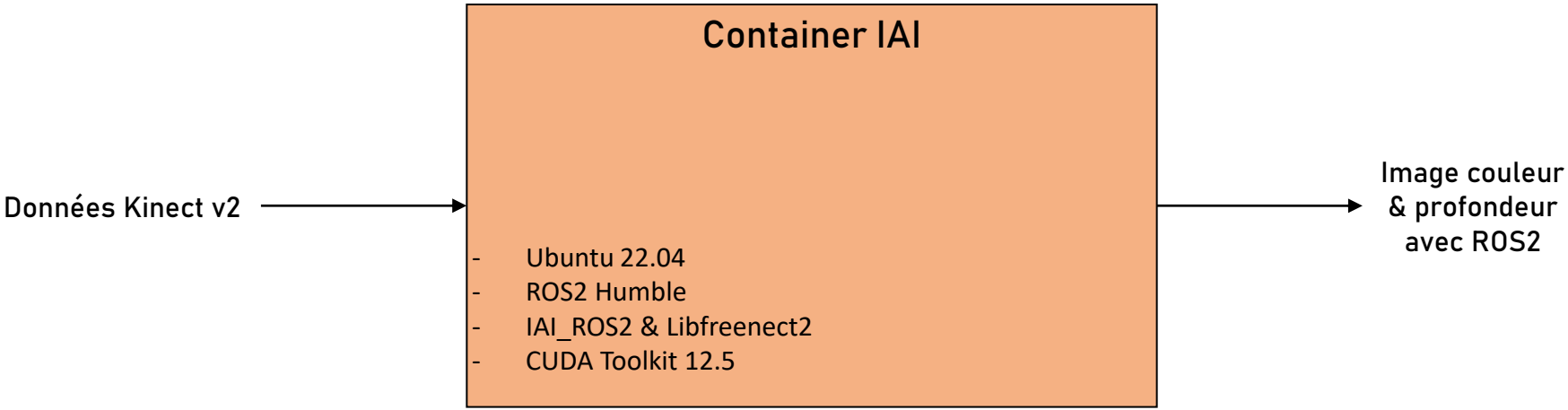
- Portable.
- Simple.
- Populaire
- Isolé

Simulation

Dockerfile


FROM osrf/ros:dashing-desktop
ENV DEBIAN_FRONTEND=noninteractive
RUN apt-get install -y \
git sudo wget unzip nano
docker build -t <nom image>
<chemin dockerfile>
1
2
3
4
Image
On choisit une image.
Environnement
On paramètre les variables d'environnement.
Installation
On installe les logiciels et programmes.
Construction
On construit l'image pour les containers.

IAI_Docker


Container IAI
Données Kinect v2
- Ubuntu 22.04
- ROS2 Humble
- IAI_ROS2 & Libfreenect2
- CUDA Toolkit 12.5
Image couleur & profondeur avec ROS2

IAI_Docker

1. Ubuntu 22.04 + CUDA Toolkit 12.5
nvidia/cuda:12.5.0-devel-ubuntu22.04

2. Installe ROS2 Humble

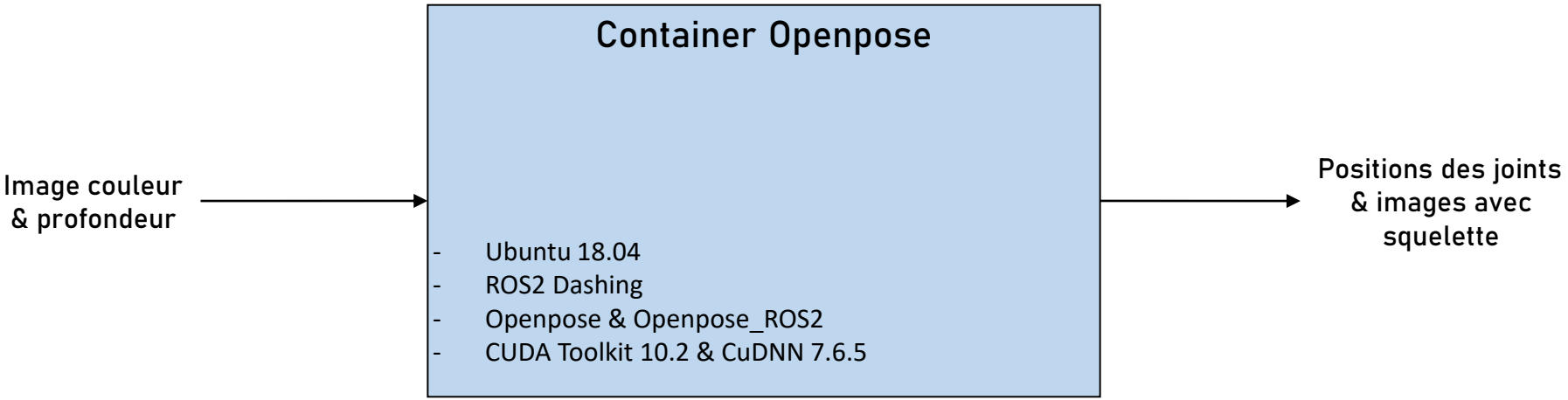
3. Installation de cuda_sample

4. Installation de Libfreenect2

```
[  2%] Building CXX object CMakeFiles/generate_resources_tool.dir/tools/generate_resources.cpp.o
[  5%] Linking CXX executable bin/generate_resources_tool
[  5%] Built target generate_resources_tool
[  7%] Generating resources.inc.h
[ 10%] Building NVCC (Device) object CMakeFiles/cuda_compile_1.dir/src/cuda_compile_1_generated_cuda_depth_packet_processor.cu.o
/root/libfreenect2/src/cuda_depth_packet_processor.cu:31:10: fatal error: helper_math.h: No such file or directory
   31 | #include <helper_math.h>
      |          ^~~~~~~~~~~~~~~
compilation terminated.
CMake Error at cuda_compile_1_generated_cuda_depth_packet_processor.cu.o.RelWithDebInfo.cmake:220 (message):
  Error generating
  /root/libfreenect2/build/CMakeFiles/cuda_compile_1.dir/src/./cuda_compile_1_generated_cuda_depth_packet_processor.cu.o


make[2]: *** [CMakeFiles/freenect2.dir/build.make:87:
CMakeFiles/cuda_compile_1.dir/src/cuda_compile_1_generated_cuda_depth_packet_processor.cu.o] Error 1
make[1]: *** [CMakeFiles/Makefile2:148: CMakeFiles/freenect2.dir/all] Error 2
make: *** [Makefile:136: all] Error 2
```

Openpose_Docker



Openpose _Docker

1. Ubuntu 18.04 + ROS2 Dashing

`osrf/ros:dashing-desktop`

2. Variables d'environnement et ports exposés

`EXPOSE 9090 11311 8080`

Défini le ROS_DOMAIN_ID, ROS_LOCALHOST_ONLY, ROS_MASTER_URI etc…

3. Installation de CUDA Toolkit 10.2 & CuDNN 7.6.5 :

Clé de CUDA : https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-keyring_1.0-1_all.deb

4. Installer Cmake

5. Installer Openpose

6. Installer Openpose_ROS2

Acquisition

IAI_ROS2 & Libfreenect2

IAI_ROS2 :

Collection d'outils et librairies pour intégrer la Kinect2 à ROS2.

Fonctionnalité intégrée :

- Pont entre Libfreenect2 et ROS2.
- Un outil de correction de perspective.

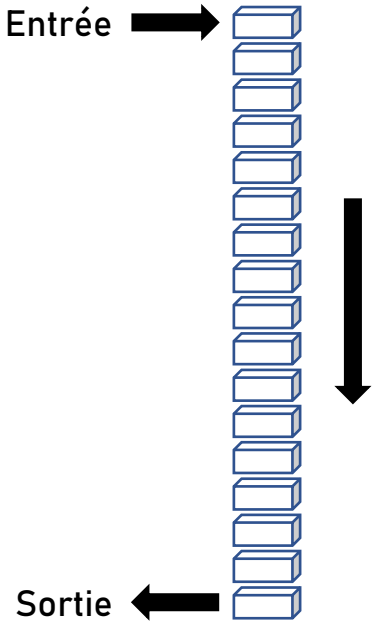
Fonction ajoutée :

- Enregistrement.

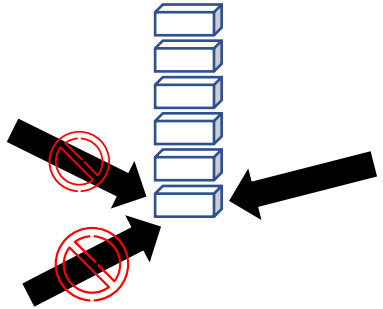
Libfreenect2 :

Un driver open source pour la Kinect v2.

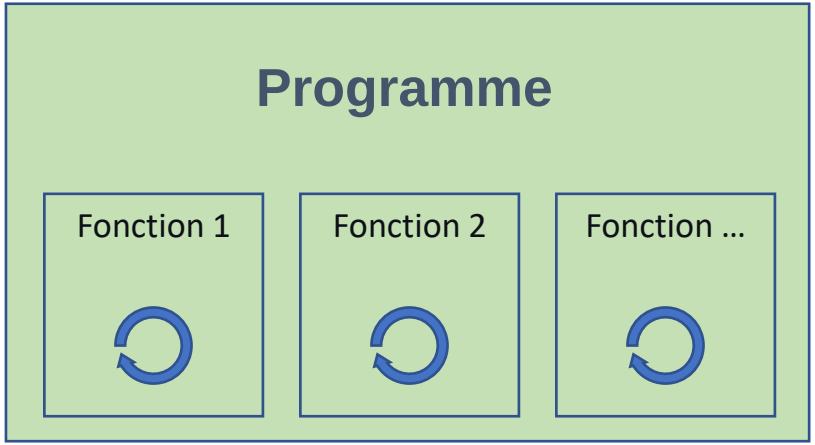
Pile & Threads



PILE

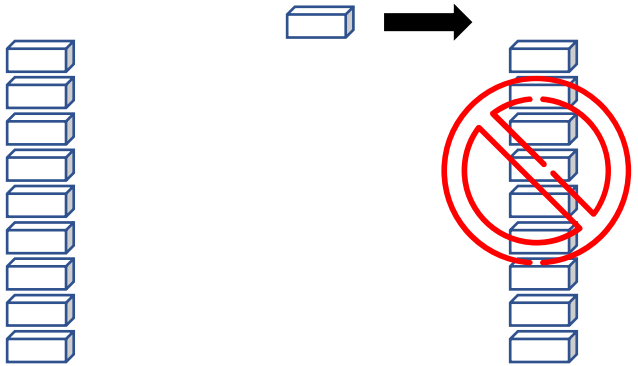


MUTEX

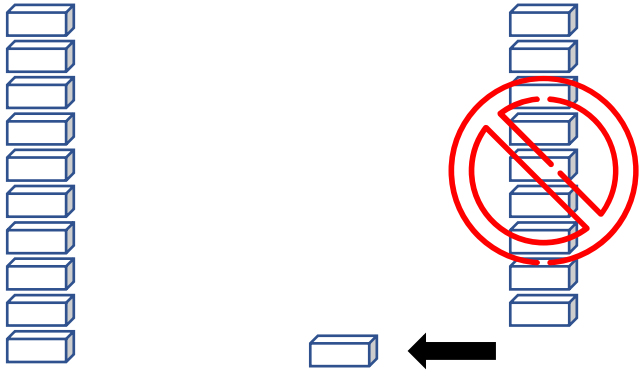


THREAD

Kinect2_record 1



Ajout d'une paire d'images



Retrait d'une paire d'images

Kinect2_record 2

```
// Structure images
struct ImagePair
{
    cv::Mat color_image;
    cv::Mat depth_image;
};

// Threadpool implementation
queue<ImagePair> image_queue_;
mutex queue_mutex_;
condition_variable queue_cond_var_;
size_t num_thread_;
vector<thread> consumer_threads_;
```

Déclaration variables

Record = True

```
for (size_t i = 0; i < num_thread_; ++i)
    consumer_threads_.emplace_back(&Kinect2Record::consumerThread, this);
```

Création des threads

```
cv::Mat color_image =
cv_bridge::toCvCopy(color_msg,sensor_msgs::image_encodings::RGB8)->image;
cv::Mat depth_image =
cv_bridge::toCvCopy(depth_msg,sensor_msgs::image_encodings::TYPE_16UC1)->image;
{
   lock_guard<mutex> lock(queue_mutex_);
   image_queue_.emplace(ImagePair{color_image, depth_image});
}
queue_cond_var_.notify_one();
```

Callback des images

Kinect2_record 3

```
ImagePair image_pair;
{
  unique_lock<mutex> lock(queue_mutex_);
  queue_cond_var_.wait(lock, [this] {
    return !recording_||!image_queue_.empty();
  });

  if (!recording_ && image_queue_.empty())
    return;
  if (!image_queue_.empty()) {
    image_pair = image_queue_.front();
    image_queue_.pop();
  }
  else
    return;
}
queue_cond_var_.notify_one();
saveImagesToDisk(image_pair.color_image, image_pair.depth_image);
```

Consumer Thread

```
{
  lock_guard<mutex> lock(queue_mutex_);
  recording_ = false;
}
queue_cond_var_.notify_all();

for (auto& thread : consumer_threads_)
{
  if (thread.joinable())
    thread.join();
}
consumer_threads_.clear();
```

Suppression des Threads

Traitement

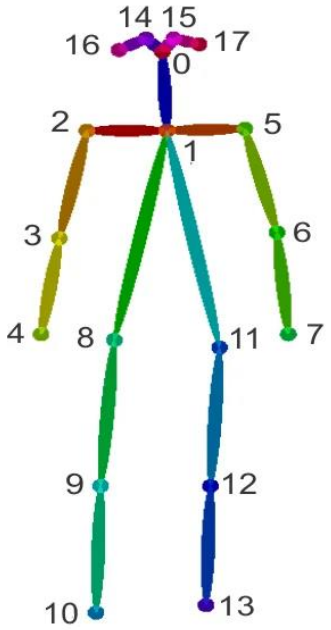
Openpose

Openpose :

Une bibliothèque de vision par ordinateur en temps réel.

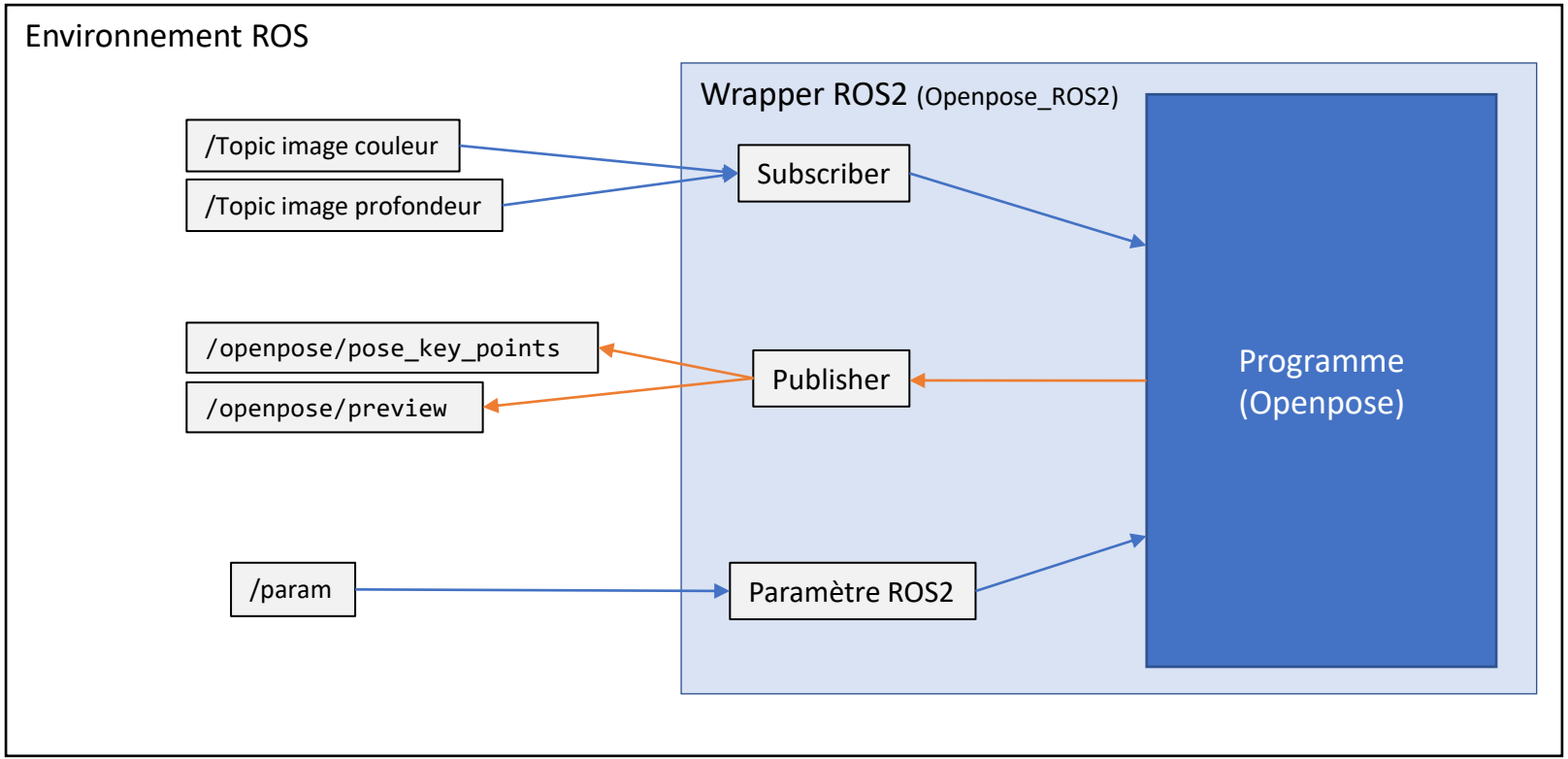
Estime les poses du corps, des mains, du visage et des pieds.

Utilise des images ou des vidéos comme source.



0. Nose
1. Neck
2. Right Shoulder
3. Right Elbow
4. Right Wrist
5. Left Shoulder
6. Left Elbow
7. Left Wrist
8. Right Hip
9. Right Knee
10. Right Ankle
11. Left Hip
12. Left Knee
13. Left Ankle
14. Right eye
15. Left Eye
16. Right Ear
17. Left Ear

Openpose_ROS2



Résultats

- ✗ Migration de la Node enregistrement Vers ROS2
- ✓ Se familiariser avec Openpose
- ✗ Comprendre le fonctionnement de Openpose_ROS2
- ✓ Développer des environnements cloisonnés pour les différentes tâches

Conclusion

Merci de votre attention