

Comment obtenir le
parfum des fleurs de
prunier sans un rhume
glacial ?

Si vous n'avez pas assez de temps pour mener la vie que vous souhaitez,
vous serez éventuellement obligé de passer beaucoup de temps à vivre votre
vie dont vous ne voulez pas.

[Blogue Jardin](#) [première page](#) [Nouvel essai](#) [connecter](#) [abonnement](#) [gérer](#)

Sous WSL-Ubuntu18.04, le processus d'installation d'Ope nREALM et les problèmes rencontrés et solutions

Collection-Multi-UAV SLAM(4)

- 1. Sous WSL-Ubuntu18.04, le processus d'installation d' Aerial_ma... 2023-10-31
- 2. Sous WSL-Ubuntu20.04, le processus d'installation d'OpenCV-3,... 2023-11-08
- 3. Sous WSL-Ubuntu18.04, le processus d'installation d'OpenR... 2023-11-19
- 4. Introduction à quelques configurations de paramètres et méthodes d'ut... 01-30

[fermer](#)

Installez CMake-3.18.4 à partir du code source

OpenREALM nécessite que la version de CMake soit supérieure à 3.15.

Installez CMake-3.18.4 :

- **Commande 1 :** `sudo apt install build-essential libssl-dev`
- **Commande 2 :**
`wget https://github.com/Kitware/CMake/releases/download/v3.18.4/cmake-3.18.4.tar.gz`
- **Commande 3 :** `tar -zxvf cmake-3.18.4.tar.gz`
- **Commande 4 :** `cd cmake-3.18.4`
- **Commande 5 :** `./bootstrap`
- **Commande 6 :** `make`
- **Commande 7 :** `sudo make install`

Installer OpenCV-3.3.1

1. Installez CUDA et les environnements dépendants

- **Commande 1 :** `sudo apt-get install nvidia-cuda-toolkit`
- **Commande 2 :** `sudo apt-get install build-essential`

annonce

Surnom : LesPlumes
Âge : 1 an et 5 mois
Fans : 2Abonnés
: 0
[+Ajouter un suivi](#)

< décembre 2024 >						
jour	un	deux	troi	Quatr	cin	six
			s	e	q	
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	vingt-et-un
vingt-de	vingt-tr	vingt-qua	25	26	27	28
ux	ois	tre				
29	30	31	1	2	3	4
5	6	7	8	9	10	11

recherche

Cherchez-

mes tags

WSL (3)

ROS (2)

OpenREALM (2)

géoserveur (2)

ouvrir cv (1)

- **Commande 3 :**

```
sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev
```
- **Commande 4 :**

```
sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libdc1394-22-dev
```
- **Commande 5 :**

```
sudo add-apt-repository "deb http://security.ubuntu.com/ubuntu xenial-security main"
```
- **Commande 6 :** `sudo apt update`
- **Commande 7 :** `sudo apt install libjasper1 libjasper-dev`

2. Installez gcc-6 et g++-6 et définissez-les comme options actuelles.

Téléchargez gcc : `sudo apt-get install gcc-6 g++-6`

Une fois terminé, vous pouvez saisir `ls /usr/bin/gcc*` et `ls /usr/bin/g++*` pour afficher les versions de gcc et g++ installées sur le système.

Après avoir installé gcc-6, spécifiez gcc-6 pour la compilation :

```
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-6 100
```

Mettez à jour la configuration par défaut de la version gcc :

```
sudo update-alternatives --config gcc
```

Compilez en spécifiant g++-6 :

```
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-6 100
```

Mettez à jour la configuration par défaut de la version G++ :

```
sudo update-alternatives --config g++
```

3. Installez OpenCV-3.3.1 à partir du code source (**remarque !! Doit avoir opencv_contrib**):

gcc-6 et g++-6 sont utilisés ici car CUDA-9 est utilisé pour la compilation ultérieure, donc **les versions de GCC et G++** peuvent être conservées à 6.

Pour le processus d'installation d'OpenCV-3.3.1, veuillez vous référer à ce blog : [Installez OpenCV-3.3.1 avec opencv_contrib \(cliquez sur ce lien pour ouvrir\)](#).

Installer OpenVSLAM

Environnement dépendant : CUDA-9, OpenCV-3.3.1, Pangolin-0.5

Remarque : Cette version est une version modifiée par l'auteur d'OpenREALM. L'installation d'autres versions d'OpenVSLAM peut empêcher OpenREALM de fonctionner correctement.

1. Installez les bibliothèques dépendantes

- **Commande 1 :** `sudo apt-get install libglew-dev`
- **Commande 2 :** `sudo apt-get install libboost-dev libboost-thread-dev libboost-filesystem-dev`
- **Commande 3 :**

```
sudo apt-get install libx11-dev libxmu-dev libglu1-mesa-dev libgl2ps-dev libxi-dev g++ libzip-dev libpng12-dev libcurl4-gnutls-dev libfontconfig1-dev libsqlite3-dev libglew*-dev libssl-dev
```
- **Commande 4 :** `sudo apt-get install glew-utils`
- **Commande 5 :** `sudo apt install libglew-dev`

aérienne_mapper (1)

Collection

SLAM multi-UAV(4)

Développement de stations au sol(1)

Archives des articles

avril 2024(2)

Janvier 2024(2)

Novembre 2023(3)

Octobre 2023(1)

Derniers commentaires

1. Re : Processus d'installation d'OpenRE
ALM et problèmes rencontrés et solutions
sous WSL-Ubuntu18.04

@LesPlumes Je l'ai réalisé en utilisant la
machine virtuelle ubuntu18.04, en utilisant
opencv-contrib, et opencv sont tous deux
3.3.1...

--szjjjj

2. Re : Processus d'installation d'OpenRE
ALM, problèmes rencontrés et solutions so
us WSL-Ubuntu18.04

@szjjjj Pouvez-vous fournir des informatio
ns de configuration plus détaillées ? Quelle
s sont les versions de chaque bibliothèque
? Y a-t-il une utilisation d'opencv-contrib ?
...

--LesPlumes

3. Re : Processus d'installation d'OpenRE
ALM et problèmes rencontrés et solutions
sous WSL-Ubuntu18.04

@LesPlumes J'ai réinstallé opencv mais ç
a ne semble pas fonctionner...

--szjjjj

- Erreur 1 : Les packages suivants ont des dépendances non satisfaites :
libglewmx-dev : Conflits : libglew-dev mais 2.0.0-5 doit être installé E : Impossible de corriger les problèmes, vous avez conservé des packages cassés.
- Raison : Il se peut que la version existante du système ne corresponde pas à la version à installer.
- Solution:

```
sudo apt update
```

```
sudo apt-get install libglewmx-dev
```

2.Installez Pangolin-0.5

Téléchargez le code source de Pangolin-0.5 sur GitHub (Lien : [Pangolin-0.5](#))

- **Commande 1 :** `cd Pangolin`
- **Commande 2 :** `mkdir build && cd build`
- **Commande 3 :** `cmake -DCPP11_NO_BOOST=1 ..`
- **Commande 4 :** `make -j6`
- **Commande 5 :** `sudo make install`

3.Installez OpenVSLAM

Téléchargez le code source d'OpenVSLAM sur GitHub (lien : [openvslam](#))

Après avoir téléchargé OpenVSLAM, ouvrez d'abord `lopenvslam.gitlconfig` et modifiez

<https://github.com/OpenVSLAM-Community/FBoW.git> en : <https://github.com/stella-cv/FBoW.git>

Exécutez ensuite la commande suivante.

- **Commande 1 :** `cd ~ && cd app && cd openvslam`
- **Commande 2 :** `git clone git@github.com:laxnpander/openvslam.git`
- **Commande 3 :** `cd openvslam`
- **Commande 4 :** `git submodule init && git submodule update`
- **Commande 5 :** `mkdir build && cd build`
- **Commande 6 :**

```
cmake \

-DUSE_PANGOLIN_VIEWER=ON \

-DINSTALL_PANGOLIN_VIEWER=ON \

-DUSE_SOCKET_PUBLISHER=OFF \

-DUSE_STACK_TRACE_LOGGER=ON \

-DBUILD_TESTS=ON \

-DBUILD_EXAMPLES=ON \

..
```

- Erreur 1 : Impossible de télécharger la version-1.10.0.tar.gz
- Raison : bloqué
- Solution : URL <https://github.com/google/googletest/tags> , accédez au navigateur pour télécharger **googletest-release-1.10.0.tar.gz** , puis remplacez le fichier dans le chemin suivant par le package compressé que vous venez de téléchargé : **openvslam\build\googletest-download\googletest-download-prefix\src\release-1.10.0.tar.gz**

Notez que le package compressé doit être renommé **release-1.10.0.tar.gz**

Ensuite, exécutez à nouveau cmake.

- **Commande 7 :** `make -j6`
- **Commande 8 :** `sudo make install`

InstallerOpenREALM

(Tutoriel d'installation officiel : <https://github.com/laxnpander/OpenREALM>)

4. Re : Processus d'installation d'OpenREALM et problèmes rencontrés et solutions sous WSL-Ubuntu18.04

@szjjj Il semble que ce soit un problème de pncv. Essayez peut-être de réinstaller opencv. Il y a une erreur de liaison, donc le programme ne s'exécute pas, donc il n'y a pas de sortie de données. rviz n'est qu'un outil d'affichage et n'a rien à voir avec le programme openrealml lui-même. ...

--LesPlumes

5. Re : Processus d'installation d'OpenREALM, problèmes rencontrés et solutions sous WSL-Ubuntu18.04

Bonjour, après l'avoir exécuté, la fenêtre RViz est également vide et une erreur se produit : [ERREUR] [1721553822.250432764] : PluginlibFactory : Le plugin pour la classe '...

--szjjj

1. Téléchargez le package d'installation

- **Commande 1 :** `git clone https://github.com/laxnpander/OpenREALM.git`
- **Commande 2 :** `git submodule init`
- **Commande 3 :** `git submodule update`

2. Installez les packages de dépendances

Ouvrez `OpenREALM/tools/install_deps.sh` et commentez la partie `#OpenVSLAM`.

```
156 # OpenVSLAM
157 cd ~ && cd app && cd opencvslam
158 git clone git@github.com:laxnpander/opencvslam.git
159 cd opencvslam
160 git submodule init && git submodule update
161 mkdir build && cd build
162 cmake \
163     -DUSE_PANGOLIN_VIEWER=ON \
164     -DINSTALL_PANGOLIN_VIEWER=ON \
165     -DUSE_SOCKET_PUBLISHER=OFF \
166     -DUSE_STACK_TRACE_LOGGER=ON \
167     -DBUILD_TESTS=ON \
168     -DBUILD_EXAMPLES=ON \
169     ..
170 make -j4
171 sudo make install
172
```

Si le téléchargement de `g2o` est lent, vous pouvez modifier le lien sur la ligne 138 en :

`git@github.com:RainerKuemmerle/g2o.git`

```
136 # G2O
137 cd ~ && cd app && mkdir g2o && cd g2o
138 git clone git@github.com:RainerKuemmerle/g2o.git
139 cd g2o
140 git checkout 9b41a4ea5ade8e1250b9c1b279f3a9c098811b5a
141 mkdir build && cd build
142 cmake \
```

- **Commande 1 :** `cd OpenREALM/tools`
- **Commande 2 :** `chmod u+x install_deps.sh`
- **Commande 3 :** `./install_deps.sh`

3. Compilez et installez OpenREALM

- **Commande 1 :** `cd ..`
- **Commande 2 :** `mkdir build && cd build`
- **Commande 3 :** `cmake ..`
 - Erreur 1 : `realm_densifier/realm_densifier_impl/psl` ne contient pas de fichier `CMakeLists.txt`.
 - Raison : PSL non téléchargé
 - Solution : accédez à GitHub pour télécharger PSL, puis re **-cmake**

```
cd OpenREALM/modules/realm_densifier/realm_densifier_impl/psl
```

```
git clone git@github.com:laxnpander/PSL.git
```

◦

Erreur 2 : Erreur CMake dans `modules/realm_stages/CMakeLists.txt:105 (add_library)` : la cible "realm_stages" est liée à la cible "fbow::fbow" mais la cible n'a pas été trouvée. Peut-être qu'un appel `find_package()` est manquant pour un IMPORTED cible, ou une cible ALIAS est manquante ?

```
CMake Error at modules/realm_stages/CMakeLists.txt:105 (add_library):
  Target "realm_stages" links to target "fbow::fbow" but the target was not
  found. Perhaps a find_package() call is missing for an IMPORTED target, or
  an ALIAS target is missing?

CMake Error at modules/realm_vslam/realm_vslam_base/CMakeLists.txt:113 (add_library):
  Target "realm_vslam_base" links to target "fbow::fbow" but the target was
  not found. Perhaps a find_package() call is missing for an IMPORTED
  target, or an ALIAS target is missing?

CMake Error at modules/realm_stages/CMakeLists.txt:105 (add_library):
  Target "realm_stages" links to target "fbow::fbow" but the target was not
  found. Perhaps a find_package() call is missing for an IMPORTED target, or
  an ALIAS target is missing?

CMake Error at modules/realm_vslam/realm_vslam_base/CMakeLists.txt:113 (add_library):
  Target "realm_vslam_base" links to target "fbow::fbow" but the target was
  not found. Perhaps a find_package() call is missing for an IMPORTED
  target, or an ALIAS target is missing?

-- Generating done
CMake Generate step failed. Build files cannot be regenerated correctly.
```

- Raison : le package de dépendances `FboW` est introuvable
- Solution : ouvrez `OpenREALM/CMakeLists.txt` et ajoutez les phrases suivantes :

```
set(FBOW_INCLUDE_DIRS "/usr/local/include")

set(FBOW_LIBS "/usr/local/lib/libfbow.so")

find_package(fbow REQUIRED)
```

Puis redémarrez cmake.

- Commande 4 : `make -j6`
- Commande 5 : `sudo make install`

Installer l' ensemble de données de test OpenREALM

1.Créez un espace de travail ROS pour la compilation (exécuté sous le chemin OpenREALM)

- Commande 1 : `mkdir catkin_ws && cd catkin_ws`
- Commande 2 : `mkdir src && cd src`

2.Obtenez le package ROS et les dépendances supplémentaires

- Commande 1 : `git clone https://github.com/laxnpander/OpenREALM_ROS1_Bridge.git`
(Mode SSH : `git clone git@github.com:laxnpander/OpenREALM_ROS1_Bridge.git`)

Dépendances supplémentaires du package ROS

- Commande 2 : `sudo apt install -y -q python-catkin-tools build-essential git wget`
- Commande 3 : `sudo apt install -y -q ros-$ROS_DISTRO-geographic-msgs`
- Commande 4 : `sudo apt install -y -q ros-$ROS_DISTRO-geodesy`
- Commande 5 : `sudo apt install -y -q ros-$ROS_DISTRO-cv-bridge`
- Commande 6 : `sudo apt install -y -q ros-$ROS_DISTRO-rviz`
- Commande 7 : `sudo apt install -y -q ros-$ROS_DISTRO-pcl-ros`

3.Assurez-vous que vous êtes dans `catkin_ws` , pas dans `src`, puis construisez-le.

- Commande 1 : `cd ..`
- Commande 2 : `catkin_make -DCMAKE_BUILD_TYPE=Release`
 - Erreur 1 : erreur CMake sur OpenREALM_ROS1_Bridge/realm_ros/CMakeLists.txt:143 (add_executable) :

Ciblez les liens "realm_exiv2_grabber" vers la cible "Boost::filesystem" mais le

la cible n'a pas été trouvée. Peut-être qu'un appel find_package() est manquant pour un

Cible IMPORTÉE ou une cible ALIAS est manquante ?

```
CMake Error at OpenREALM_ROS1_Bridge/realm_ros/CMakeLists.txt:143 (add_executable):
  Target "realm_exiv2_grabber" links to target "Boost::filesystem" but the
  target was not found. Perhaps a find_package() call is missing for an
  IMPORTED target, or an ALIAS target is missing?

CMake Error at OpenREALM_ROS1_Bridge/realm_ros/CMakeLists.txt:143 (add_executable):
  Target "realm_exiv2_grabber" links to target "FLANN::FLANN" but the target
  was not found. Perhaps a find_package() call is missing for an IMPORTED
  target, or an ALIAS target is missing?

CMake Error at OpenREALM_ROS1_Bridge/realm_ros/CMakeLists.txt:143 (add_executable):
  Target "realm_exiv2_grabber" links to target "fbow::fbow" but the target
  was not found. Perhaps a find_package() call is missing for an IMPORTED
  target, or an ALIAS target is missing?

CMake Error at OpenREALM_ROS1_Bridge/realm_ros/CMakeLists.txt:139 (add_executable):
  Target "realm_ros_grabber" links to target "Boost::filesystem" but the
  target was not found. Perhaps a find_package() call is missing for an
  IMPORTED target, or an ALIAS target is missing?

CMake Error at OpenREALM_ROS1_Bridge/realm_ros/CMakeLists.txt:139 (add_executable):
  Target "realm_ros_grabber" links to target "FLANN::FLANN" but the target
  was not found. Perhaps a find_package() call is missing for an IMPORTED
  target, or an ALIAS target is missing?

CMake Error at OpenREALM_ROS1_Bridge/realm_ros/CMakeLists.txt:139 (add_executable):
  Target "realm_ros_grabber" links to target "fbow::fbow" but the target was
  not found. Perhaps a find_package() call is missing for an IMPORTED
  target, or an ALIAS target is missing?
```

- Raison : Le fichier correspondant est introuvable
- Solution : (Référence : <https://github.com/laxnpander/OpenREALM/issues/75>)

Ouvrez le fichier `OpenREALM\catkin_ws\src\OpenREALM_ROS1_Bridge\realm_ros\CMakeList.txt`,

À la ligne 53, ajoutez :

```
# boost

find_package(Boost COMPONENTS thread filesystem date_time system REQUIRED)
```

À la ligne 74, ajoutez :

```
find_package(fbow REQUIRED)

find_package(CGAL REQUIRED)
```

À la ligne 115, ajoutez :

```
set(CMAKE_MODULE_PATH "/usr/lib/x86_64-linux-gnu/cmake")

add_library(FLANN::FLANN INTERFACE IMPORTED)
```

À la ligne 125, ajoutez à `target_link_libraries(...)` :

```
${Boost_FILESYSTEM_LIBRARIES}

${Boost_SYSTEM_LIBRARIES}
```

```
123 target_link_libraries(${PROJECT_NAME} PUBLIC
124     ${catkin_LIBRARIES}
125     ${cmake_modules_LIBRARIES}
126     ${OpenCV_LIBRARIES}
127     ${Boost_FILESYSTEM_LIBRARIES}
128     ${Boost_SYSTEM_LIBRARIES}
129     realm_stages
130 )
131
```

Exécutez l'ensemble de données de test OpenREALM

1. Téléchargez l'ensemble de données (il est recommandé de le télécharger puis de le faire glisser dans Ubuntu) :

<https://drive.google.com/open?id=1-2h0tasl4wzxZKLBbOz3XbJ7f5xIxIMe>

2. Saisissez le chemin où se trouvent les données, puis décompressez :

- Commande 1 : `tar -xvzf open_realm_edm_dataset.tar.gz`

3. Confirmez le chemin de l'ensemble de données dans le fichier de lancement :

Le chemin du dossier des profils est : `OpenREALM\catkin_ws\src\OpenREALM_ROS1_Bridge\realm_ros`

Besoin d'ouvrir avant de courir `OpenREALM\catkin_ws\src\OpenREALM_ROS1_Bridge\realm_ros\launch` ,

Vous devez modifier la ligne 13 du fichier de lancement et définir la valeur sur l'adresse absolue de votre propre ensemble de données .

```

Ubuntu-18.04 > home > yill > app > OpenREALM > catkin_ws > src > OpenREALM_ROS1_Bridge > realm_ros > launch > alexa_noreco.launch
1  <?xml version="1.0"?>
2  <launch>
3      <arg name="camera_id" default="alexa"/>
4      <arg name="topic_adapter" default="/realm/${arg camera_id}/input"/>
5      <arg name="topic_pose_est" default="/realm/${arg camera_id}/pose_estimation/frame"/>
6      <arg name="topic_dense" default="/realm/${arg camera_id}/densification/frame"/>
7      <arg name="topic_surf" default="/realm/${arg camera_id}/surface_generation/frame"/>
8      <arg name="topic_rect" default="/realm/${arg camera_id}/ortho_rectification/frame"/>
9      <arg name="topic_mosaic" default="/realm/${arg camera_id}/mosaicing/frame"/>
10
11     <node pkg="realm_ros" type="realm_exiv2_grabber" name="realm_exiv2_grabber" output="screen">
12         <param name="config/id" type="string" value="${arg camera_id}"/>
13         <param name="config/input" type="string" value="/home/yill/download/edm_big_overlap_50p"/>
14         <param name="config/rate" type="double" value="10.0"/>
15         <param name="config/profile" type="string" value="alexa_noreco"/>
16         <!--param name="config/opt/poses" type="string" value=""-->
17         <!--param name="config/opt/surface_pts" type="string" value=""-->
18         <!--param name="config/opt/set_all_keyframes" type="bool" value="True"-->
19         <!--param name="config/opt/working_directory" type="string" value="/home/alex/Documents/OpenREALM"/-->
20     </node>

```

4. Installez le package de fonctions AerialMapDisplay :

```
sudo apt-get install ros-$ROS_DISTRO-rviz-satellite
```

5. Exécutez le fichier de lancement :

Entrez d'abord **OpenREALM/catkin_ws/src/OpenREALM_ROS1_Bridge/realm_ros/launch** le chemin, puis entrez la commande de test suivante.

- Cartographie GNSS uniquement :

Commande: `roslaunch realm_ros alexa_gnss.launch`

- Cartographie 2D avec SLAM visuel :

Commande: `roslaunch realm_ros alexa_noreco.launch`

- Cartographie 5D avec SLAM visuel et reconstruction de surface :

Commande: `roslaunch realm_ros alexa_reco.launch`

Erreurs que vous pouvez rencontrer lors de l'exécution :

- Erreur 1 : `RLEException: [alexa_gnss.launch] n'est ni un fichier de lancement dans le package [realm_ros] ni [realm_ros] un nom de fichier de lancement.`

Le traçage de l'exception a été écrit dans le fichier journal

```

yill@DESKTOP-MRW992C:~/app/OpenREALM/catkin_ws/src/OpenREALM_ROS1_Bridge/realm_ros/launch$
roslaunch realm_ros alexa_gnss.launch
RLEException: [alexa_gnss.launch] is neither a launch file in package [realm_ros] nor is [r
ealm_ros] a launch file name
The traceback for the exception was written to the log file

```

- Raison : L'environnement realm_ros n'a pas été ajouté au chemin ROS, il n'a donc pas pu être trouvé.
- Solution:

Ouvrez le fichier bashrc :

```
sudo vim ~/.bashrc
```

Saisissez ensuite dans la ligne du bas (modifiez le chemin en fonction du chemin de votre ordinateur) :

```
source /home/your_name/app/OpenREALM/catkin_ws/devel/setup.sh
```

```
export ROS_PACKAGE_PATH=${ROS_PACKAGE_PATH}:/home/your_name/app/OpenREALM/catkin_ws/
```

Enregistrez et quittez, puis redémarrez bashrc :

```
source ~/.bashrc
```

Vérifiez si le chemin est ajouté :

```
echo $ROS_PACKAGE_PATH
```

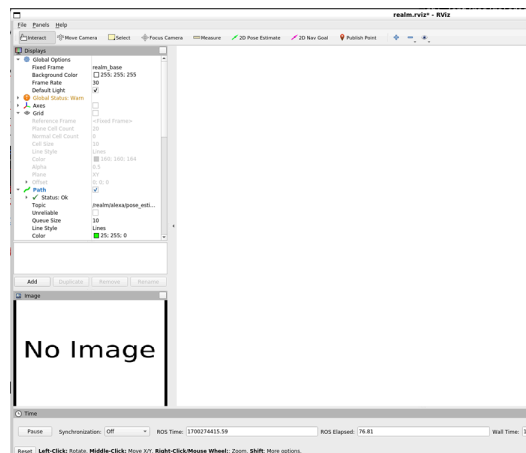
```
# >>> fishros initialize >>>|
source /opt/ros/noetic/setup.bash
# <<< fishros initialize <<<
source /home/yill/app/OpenREALM/catkin_ws/devel/setup.bash
export ROS_PACKAGE_PATH=${ROS_PACKAGE_PATH}:/home/yill/app/OpenREALM/catkin_ws/
```

- Erreur 2 : Une fenêtre RViz vide apparaîtra après l'exécution.

En même temps, le terminal demandera également :

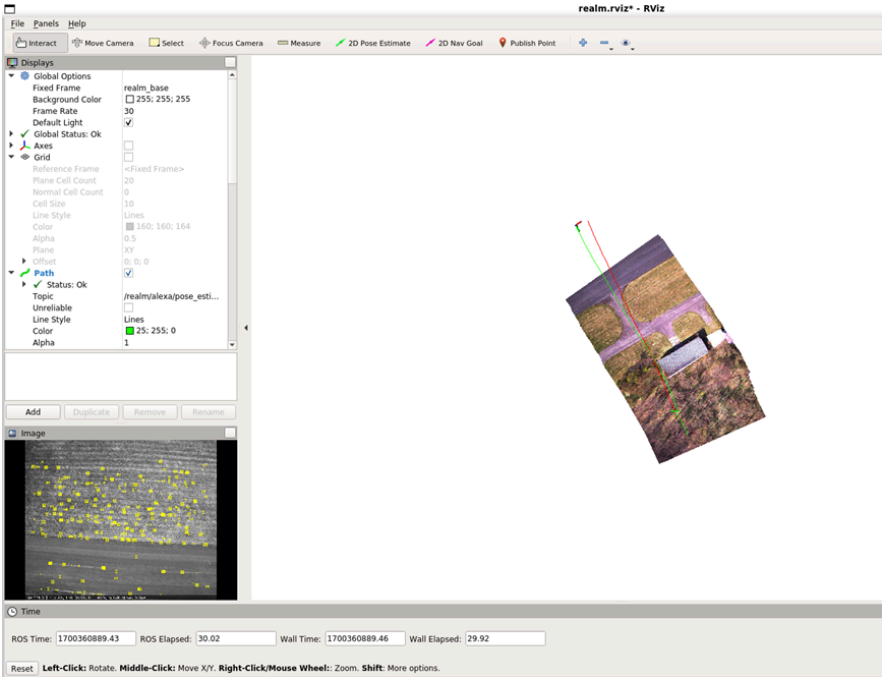
[ERREUR] [1700274476.866316760] : PluginlibFactory : Le plugin pour la classe

'rviz_plugins/AerialMapDisplay' n'a pas pu être chargé. Erreur : Selon les descriptions du plugin chargé, la classe rviz_plugins/AerialMapDisplay avec le type de classe de base rviz::Display n'existe pas.



- Raison 1 : package de fonctions AerialMapDisplay manquant
- Solution: `sudo apt-get install ros-$ROS_DISTRO-rviz-satellite`
- Raison 2 : OpenCV n'est pas fourni avec **le package de dépendances OpenCV_contrib** , ce qui entraîne l'échec de l'exécution d'OpenVSLam. Par conséquent, les points caractéristiques de l'image ne peuvent pas être suivis et l'épissage échoue, donc l'interface Rviz est vide.
- Solution : réinstallez OpenCV-3.3.1 avec le package de dépendances OpenCV_contrib ; veuillez vous référer à mon processus d'installation ci-dessus ;

Diagramme de résultat d'une opération réussie :



Collection : SLAM multi-UAV

Étiquettes : ROS , WSL , OpenREALM

Bon article à soutenir

suis-moi

Récupérez cet article

Partage WeChat



LesPlumes
Fans - 2 Abonnés - 0

+Ajouter un suivi

0 0

Surclasser pour devenir membre

publié @ 2023-11-19 12:19 LesPlumesLire (1571 19) Commentaires () Modifier CollectionReport

Actualisez la page et revenez en haut

Vous devez vous connecter pour afficher ou publier des commentaires. [Connectez-vous](#) maintenant ou [visitez](#) la page d'accueil du blog.

- [Recommandé] 100% open source ! Le code source C++ des logiciels multiplateformes industriels à grande échelle est fourni, modélisé et configuré !
- [Recommandé] Replay vidéo de la conférence FFA 2024 : Le passé, le présent et le futur d'Apache Flink
- [Recommandé] L'assistant IA de Douyin Doubao, votre encyclopédie intelligente, gratuite et illimitée
- [Recommandé] Outil SSH léger et performant IShell : la bénédiction de l'IA, une étape plus rapide



Recommandation de l'éditeur :

- Une brève discussion des principes sous-jacents du .NET Core asynchrone (Async)
- Qu'est-ce que la chaîne fortement typée de .NET ?
- Rappelez-vous l'analyse bloquée d'un certain système d'armoire intelligente .NET hdp.
- De quelle quantité de mémoire chaque langage de programmation aura-t-il besoin pour exécuter 1 million de tâches simultanées en 2024 ?
- Comment écrire du code : dix expériences mènent au parcours d'une programmation efficace



Classement de lecture :

- Enfin résolu le problème du système de service client en ligne .net toujours faussement signalé par 360 (signature n umérique du logiciel)
- Le site Web a été touché par une attaque DDoS juste après sa mise en ligne !
- Un outil open source pour optimiser les CV "GitHub Hot Spots"
- Basé sur .NetCore pour développer le projet de blog StarBlog - (32) Le premier numéro est terminé
- (Série 13) Vue3+Echarts construit un super beau panneau système