

# 室内外场景定位

## Installation

### 基础环境

ubuntu 16.04, cuda8.0(cuda9.0), MATLABR2016b

### TensorFlow安装(ContextDesc环境配置)

安装虚拟环境py2\_tf112流程如下(python==2.7, tensorflow==1.12.0)

[tensorflow](#)

```
pip install --upgrade pip
pip install https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-1.12.0-cp27-none-linux_x86_64.whl
pip install numpy
pip install opencv_contrib (to enable SIFT)
```

其余包请在运行 ContextDesc 时自行安装! (匹配 python2 即可)

首次运行程序, 模型将从预设路径自动下载, 如有问题请及时联系.

### OPENMVG 安装及配置

[openmvg](#)

#### 1. 工具准备

CMake

Git

C/C++ compiler (GCC, Visual Studio or Clang)

#### 2. 获取源码

```
git clone --recursive https://github.com/openMVG/openMVG.git
```

#### 3. 安装依赖项

```
sudo apt-get install \
    libpng-dev libjpeg-dev \
    libtiff-dev libxxf86vm1 \
    libxxf86vm-dev \
    libxi-dev \
    libxrandr-dev
```

可选: 需要可视化, 可安装Graphviz

```
sudo apt-get install graphviz
```

#### 4. 配置和编译

```
mkdir openMVG_Build && cd openMVG_Build
cmake -DCMAKE_BUILD_TYPE=RELEASE ../openMVG/src/
cmake --build . --target install
make test
```

## 5. 测试安装

```
make test
```

# COLMAP 安装及配置

[colmap](#)

## 1. 获取源码

```
git clone https://github.com/colmap/colmap
```

## 2. 安装依赖项

```
sudo apt-get install \
    git \
    cmake \
    build-essential \
    libboost-program-options-dev \
    libboost-filesystem-dev \
    libboost-graph-dev \
    libboost-regex-dev \
    libboost-system-dev \
    libboost-test-dev \
    libeigen3-dev \
    libsuitesparse-dev \
    libfreeimage-dev \
    libgoogle-glog-dev \
    libgflags-dev \
    libglew-dev \
    qtbase5-dev \
    libqt5opengl5-dev \
    libcgall-dev
```

PS:在Ubuntu16.04下, CGAL的cmake配置是损坏的, 必须安装CGAL的qt5包

```
sudo apt-get install libcgall-qt5-dev
```

## 3. 编译安装Ceres-Solver

[Ceres-Solver](#)

```
sudo apt-get install libatlas-base-dev libsuitesparse-dev
git clone https://ceres-solver.googlecode.com/ceres-solver
cd ceres-solver
git checkout $(git describe --tags) # Checkout the latest release
mkdir build
cd build
cmake .. -DBUILD_TESTING=OFF -DBUILD_EXAMPLES=OFF
make
sudo make install
```

## 4. 配置和编译colmap

```
git clone https://github.com/colmap/colmap.git
cd colmap
git checkout dev
mkdir build
cd build
cmake ..
make
sudo make install
```

## 5. 运行colmap

```
colmap -h
colmap gui
```

# Structure

## 工程文件夹及主要程序文件说明

1. dataset - 存放数据及数据预处理脚本, 其中officialData存放官方原始数据
2. scripts - 存放算法主要源代码
3. software - 存放算法所使用的软件框架
4. upload - 存放中间运行结果及结果处理脚本
5. workspace - 存放算法工程文件
6. scripts.sh - 主程序脚本
7. upload.txt - 最终运行结果

目录结构示意图如下

```
.
├── dataset
│   ├── fishEye
│   ├── gt
│   ├── mvpic_ofFish.py
│   ├── mvpic_ofPano.py
│   ├── officialData
│   ├── pano
│   ├── panoSplit
│   └── panoSplit8
├── scripts
│   ├── colmap
│   ├── contextDesc
│   └── pipeline.pdf
├── scripts.sh
├── software
│   ├── colmap
│   └── openMVG
├── upload
│   ├── cnn
│   ├── dsp
│   ├── final
│   ├── fishEye
│   ├── org
│   ├── org8
│   └── selectBest.py
└── upload.txt
```

```
└─ workspace
   └─ cnn
   └─ dsp
   └─ fishEye
   └─ org
   └─ org8
```

## Getting Started

---

运行之前，需要修改两处路径为绝对路径：

<code>./scripts/colmap/matchForCnn.data</code>	第 1 行
<code>./scripts/colmap/scriptsOfCnn/matching_pipeline.m</code>	第 10-13 行

将测试数据置于以下目录(数据组织方式与预赛相同)：

```
./dataset/officialData/
```

在根目录下运行：

```
sh scripts.sh
```

若运行有问题，请及时联系！

## Contact Us

---

如果有问题请及时联系。

Tel: 156 5075 8779 (常同学), 182 9288 5460 (蒋同学)

Email: [cxy19960919@163.com](mailto:cxy19960919@163.com), [1143958845@qq.com](mailto:1143958845@qq.com)

星期一, 08. 七月 2019 05:14下午