Project2 Phase1

TA：李景行　　Mail: jhanglee@hnu.edu.cn

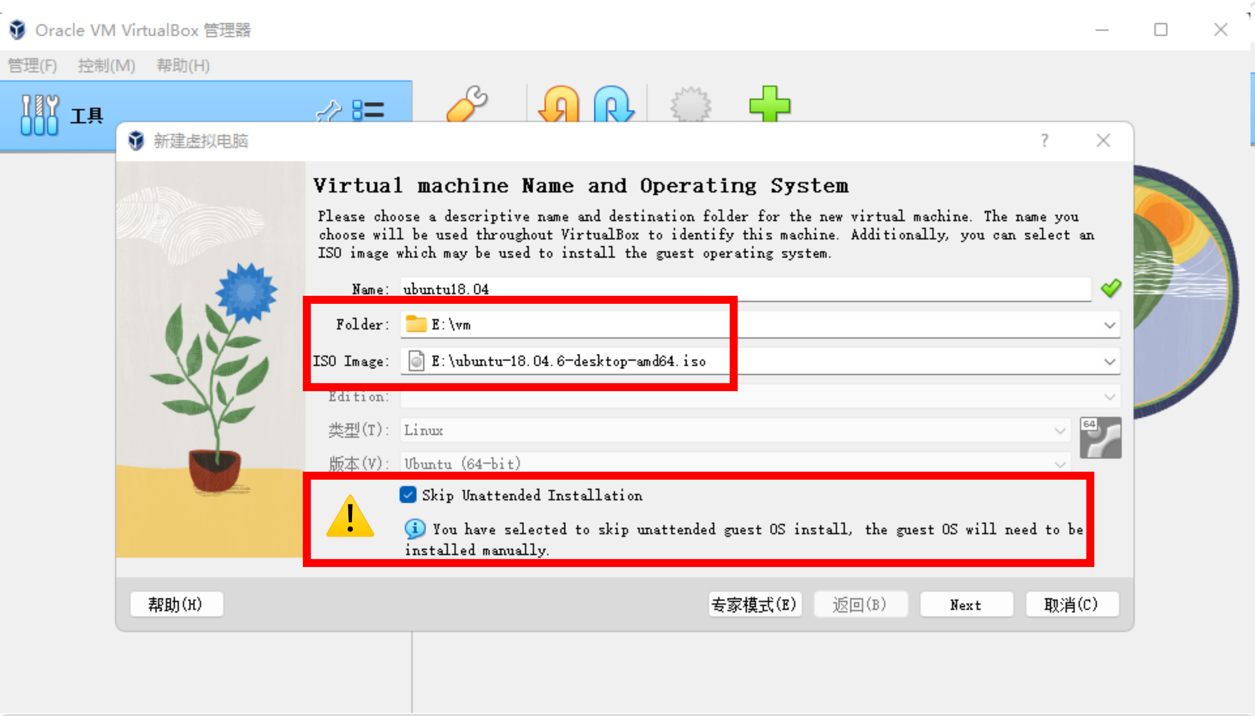
一. 安装虚拟机和Ubuntu 18.04 系统

虚拟机的安装包和 Ubuntu18.04 的镜像文件会提供给大家

1. 安装教程

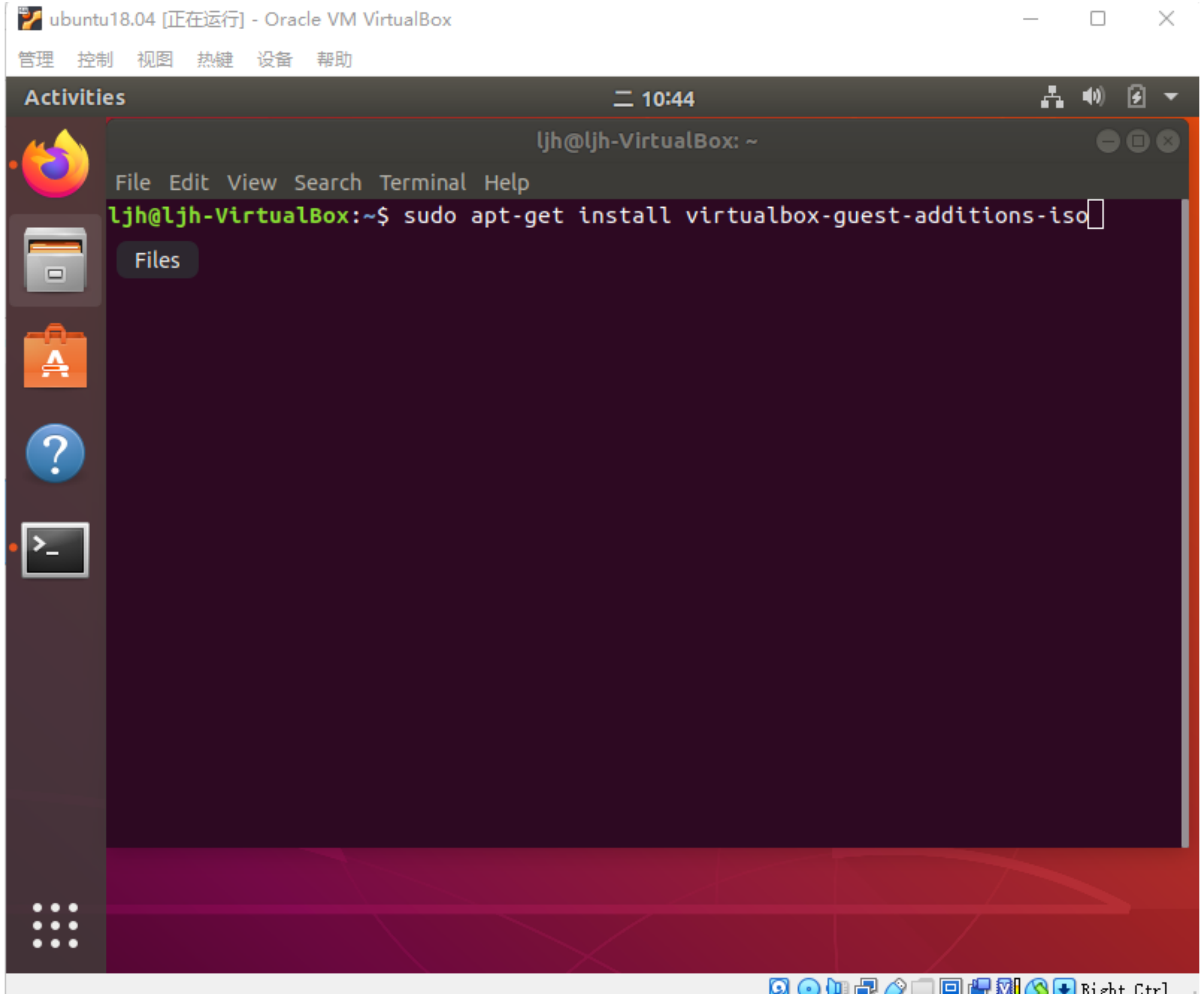
几个关键步骤：

⚠️ 注意：Ubuntu 安装时的语言设置请选择英文，否则目录的名字会变成中文，不方便后续操作


Oracle VM VirtualBox 管理器
管理(F) 控制(M) 帮助(H)
工具
新建虚拟电脑
Virtual machine Name and Operating System
Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.
Name: ubuntu18.04
Folder: E:\vm
ISO Image: E:\ubuntu-18.04.6-desktop-amd64.iso
Edition:
类型(T): Linux
版本(V): Ubuntu (64-bit)
Skip Unattended Installation
You have selected to skip unattended guest OS install, the guest OS will need to be installed manually.
帮助(H)
专家模式(E)
返回(B)
Next
取消(C)


Oracle VM VirtualBox 管理器
管理(F) 控制(M) 帮助(H)
工具
新建虚拟电脑
Hardware
You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.
内存大小(M):
11264 MB
4 MB
16384 MB
Processors:
4
1 CPU
8 CPU
Enable EFI (special OSes only)
帮助(H)
返回(B)
Next
取消(C)


Oracle VM VirtualBox 管理器
管理(F) 控制(M) 帮助(H)
工具
新建虚拟电脑
Hardware
You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.
内存大小(M):
11264 MB
4 MB
16384 MB
Processors:
4
1 CPU
8 CPU
Enable EFI (special OSes only)
帮助(H)
返回(B)
Next
取消(C)



解决虚拟机在安装后不能全屏的问题

2. 安装ROS系统

melodic

ROS Melodic Morenia

ROS Melodic Morenia is the twelfth ROS distribution release. It was released on May 23rd, 2018.

Contents

1. Platforms

ROS Melodic Morenia is primarily targeted at the Ubuntu 18.04 (Bionic) release, though other Linux systems as well as Mac OS X, Android, and Windows are supported to varying degrees. For more information on compatibility on other platforms, please see REP 3: Target Platforms. It will also support Ubuntu 17.10 Artful and Debian Stretch.

⚠️ 注意：ROS 安装时选择 melodic 版本，

安装方法见 ROS wiki，链接如下：

Link: http://wiki.ros.org/melodic/Installation/Ubuntu

3. ROS 基础知识

- 请先跟着ROS wiki 上 <1.1初级教程 (中文)> 操作一遍，了解ROS的一些基本用法，链接如下:

  Link: http://wiki.ros.org/cn/ROS/Tutorials

- 可以选择 VS code 编辑代码，安装方法见链接:

  Link: https://zhuanlan.zhihu.com/p/430939275

4. 编译 tag_detector 的预先准备

OpenCV

eigen

Project ID: 15462818

二. 程序架构

```cpp
/* THIS IS THE POSE ESTIMATION FUNCTION YOU NEED TO IMPLEMENT */
// pts_id: id of each point
// pts_3: 3D position (x, y, z) in world frame
// pts_2: 2D position (u, v) in image frame
void process(
  const vector<int> &pts_id,
  const vector<cv::Point3f> &pts_3,
  const vector<cv::Point2f> &pts_2,
  const ros::Time& frame_time)
{
```

```cpp
{
    /* Pose estimation using OpenCV's PnP function.
     * THIS IS YOUR WORK
     * The result will be used as reference. */
    // Use cv::solvePnPRansac() and cv::Rodrigues().
```

1. 程序架构

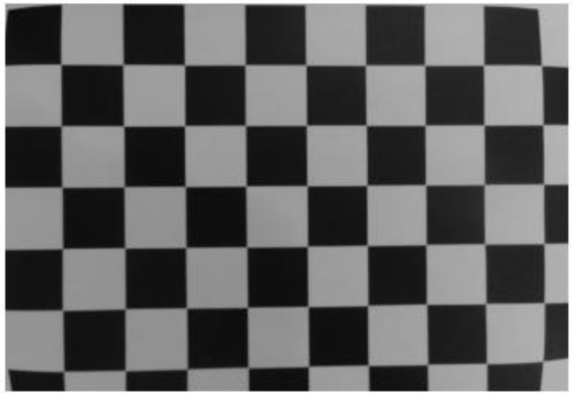
```
    // Apply undistortion if needed. Use cv::undistortPoints().
```

Lec5_机器视觉

湖南大学 HUNAN UNIVERSITY | 机器人视觉感知与控制技术国家工程研究中心

Radial Distortions

- Wide angle lenses have radial distortions
  - Straight lines become curves
- Formulation:
  - $r^2 = u^2 + v^2$
  - $u^{dist} = u(1 + k_1 r + k_2 r^2 + k_3 r^3 + \cdots)$
  - $v^{dist} = v(1 + k_1 r + k_2 r^2 + k_3 r^3 + \cdots)$



1. 程序架构

```
    // Step 1: Work out the initial pose estimates using the linear solution.
```

- Suppose $\boldsymbol{H} = \boldsymbol{K}(\boldsymbol{r_1} \quad \boldsymbol{r_2} \quad \boldsymbol{t})$

$$\lambda\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

- If the SVD of $(\tilde{\boldsymbol{h}}_1 \quad \tilde{\boldsymbol{h}}_2 \quad \tilde{\boldsymbol{h}}_1 \times \tilde{\boldsymbol{h}}_2) = \boldsymbol{USV^T}$, then the solution is

$$\boldsymbol{R} = \boldsymbol{UV^T}$$

$$\lambda u = h_{11}x + h_{12}y$$
$$\lambda v = h_{21}x + h_{22}y$$
$$\lambda = h_{31}x + h_{32}y$$

Given an obs
of feature $(x_i$

- To find the translation:

$$\boldsymbol{t} = \tilde{\boldsymbol{h}}_3 / \|\tilde{\boldsymbol{h}}_3\|$$

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_iu_i & -y_iu_i & -u_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -x_iv_i & -y_iv_i & -v_i \end{bmatrix} \begin{bmatrix} h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0$$

Solve this Ax = 0
problem using SVD

Stack pairs of observations

1. 程序架构

- If the SVD of $(\tilde{\boldsymbol{h}}_1 \quad \tilde{\boldsymbol{h}}_2 \quad \tilde{\boldsymbol{h}}_1 \times \tilde{\boldsymbol{h}}_2) = \boldsymbol{USV}^T$, then the solution is

$$\boxed{\boldsymbol{R} = \boldsymbol{UV}^T}$$

- To find the translation:

$$\boxed{\boldsymbol{t} = \tilde{\boldsymbol{h}}_3 / \|\tilde{\boldsymbol{h}}_3\|}$$

1. 程序架构

```
        // Step 2: Refine the pose estimates using non-linear optimization
```

机器人视觉感知与控制技术
国家工程研究中心

Nonlinear 3D-2D Pose Estimation

$\boldsymbol{\theta}$: $Euler\ Angles\ \epsilon\ R^3$ $\quad$ $\boldsymbol{t}$: $Translation\ \epsilon\ R^3$ $\quad$ $\pi(\cdot)$: $projection\ function$

1) Nonlinear least square

$$\min_{\boldsymbol{\theta},\boldsymbol{t}} \sum_i \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \pi\left(\boldsymbol{K}\cdot\left(\boldsymbol{R}(\boldsymbol{\theta})\cdot\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \boldsymbol{t}\right)\right)\right\|^2 = \min_{\boldsymbol{\theta},\boldsymbol{t}} \sum_i \|\gamma_i(\boldsymbol{\theta},\boldsymbol{t})\|^2$$

2) Linearization

$$\gamma_i(\boldsymbol{\theta},\boldsymbol{t}) \approx \gamma_i\overbrace{(\boldsymbol{\theta_0},\boldsymbol{t_0})}^{\textbf{Initial values}} + \underbrace{\frac{\partial\gamma_i}{\partial\boldsymbol{\theta_i},\boldsymbol{t_i}}|_{\boldsymbol{\theta_0},\boldsymbol{t_0}}}_{2\ \times 6\ matrix} \cdot \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix} = \gamma_i(\boldsymbol{\theta_0},\boldsymbol{t_0}) + \underbrace{\boldsymbol{J_i}}_{2\ \times 6\ matrix}\begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix}$$

The problem becomes:

$$\min_{\delta\boldsymbol{\theta},\delta\boldsymbol{t}} \sum_i \left\|\gamma_i(\boldsymbol{\theta_0},\boldsymbol{t_0}) + \boldsymbol{J_i}\begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix}\right\|^2$$

机器人视觉感知与控制技术
国家工程研究中心

Nonlinear 3D-2D Pose Estimation

$\boldsymbol{\theta}$: $Euler\ Angles\ \epsilon\ R^3$ $\quad$ $\boldsymbol{t}$: $Translation\ \epsilon\ R^3$ $\quad$ $\pi(\cdot)$: $projection\ function$

3) Take derivative and set it to zero

$$\sum_i \boldsymbol{J}_i^T\boldsymbol{J}_i\begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix} + \sum_i \boldsymbol{J}_i^T\gamma_i(\boldsymbol{\theta_0},\boldsymbol{t_0}) = 0 \Rightarrow \overbrace{\sum_i \boldsymbol{J}_i^T\boldsymbol{J}_i}^{\mathbf{A}}\begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix} = \overbrace{-\sum_i \boldsymbol{J}_i^T\gamma_i(\boldsymbol{\theta_0},\boldsymbol{t_0})}^{\mathbf{b}}$$

4) Solve for the incremental states and update the optimization variables

$$\begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix} = \boldsymbol{A}^{-1}\boldsymbol{b} \quad \Rightarrow \quad \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta_0} \\ \boldsymbol{t_0} \end{bmatrix} + \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix}$$

5) Iterate from step 1)

2. C++ 求解SVD

eigen

Project ID: 15462818

使用Eigen 库：进行svd分解，形如 $A = U * S * V^T$。

```
JacobiSVD<MatrixXd> svd(J, ComputeThinU | ComputeThinV);

U = svd.matrixU();

V = svd.matrixV();

A = svd.singularValues();
```