****

IPMV-Experiment-1

lab0 preparing expenvironment and

introduction to opencv

课程名称： 图像处理与机器视觉

实验地点： 嘉定校区智信馆131

指导教师： Lei Jiang, Rui FAN

姓 名： 姚天亮

学 号： 2150248

一、LAB Task

· Preparing environment

· Clone opencv/eigen/sophus/geographiclib source code

· Configure environment

· Write a simple project to verify the environment

二、实验环境配置

**1.在虚拟机中安装ubuntu linux**

我们的实验需要以下最低硬件要求：

Intel/AMD x86\_64 兼容CPU；

CPU至少2核，运行频率2GHz以上；

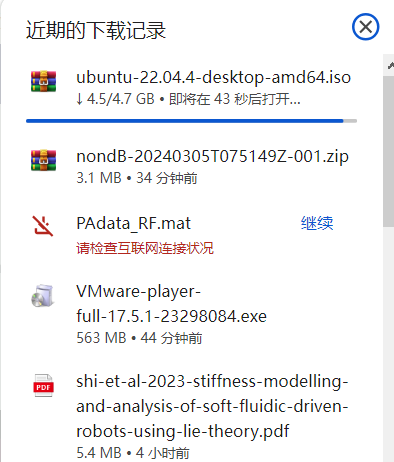
原生 Linux 至少 8GB RAM，如果您打算使用虚拟机，建议 12GB+ RAM；

本地 Linux 至少 20GB 可用磁盘空间，虚拟机至少 40GB；

如果您需要在裸机计算机上安装Linux，则需要32GB USB磁盘用于操作系统安装。

强烈建议您使用 Ubuntu Linux 22.04.x LTS 作为实验环境的主要操作系统。如果您使用的是 Microsoft Windows Vmware 工作站，则可以在此处下载安装 Vmware。

等待下载：



完成安装 Vmware 工作站后，使用以下配置设置新的虚拟机：

2+ CPU 核心；

6GB+内存；

40GB 磁盘空间；

为 Ubuntu Linux 22.04.x 安装映像（iso 文件）分配 CDROM；

启动虚拟机以开始安装过程。安装开始时选择正常安装。必须选择英语作为新系统的语言。如果你选择中文，它会使用中文作为你的主目录的名称，这将在你将代码放在工作目录下时导致严重的问题。

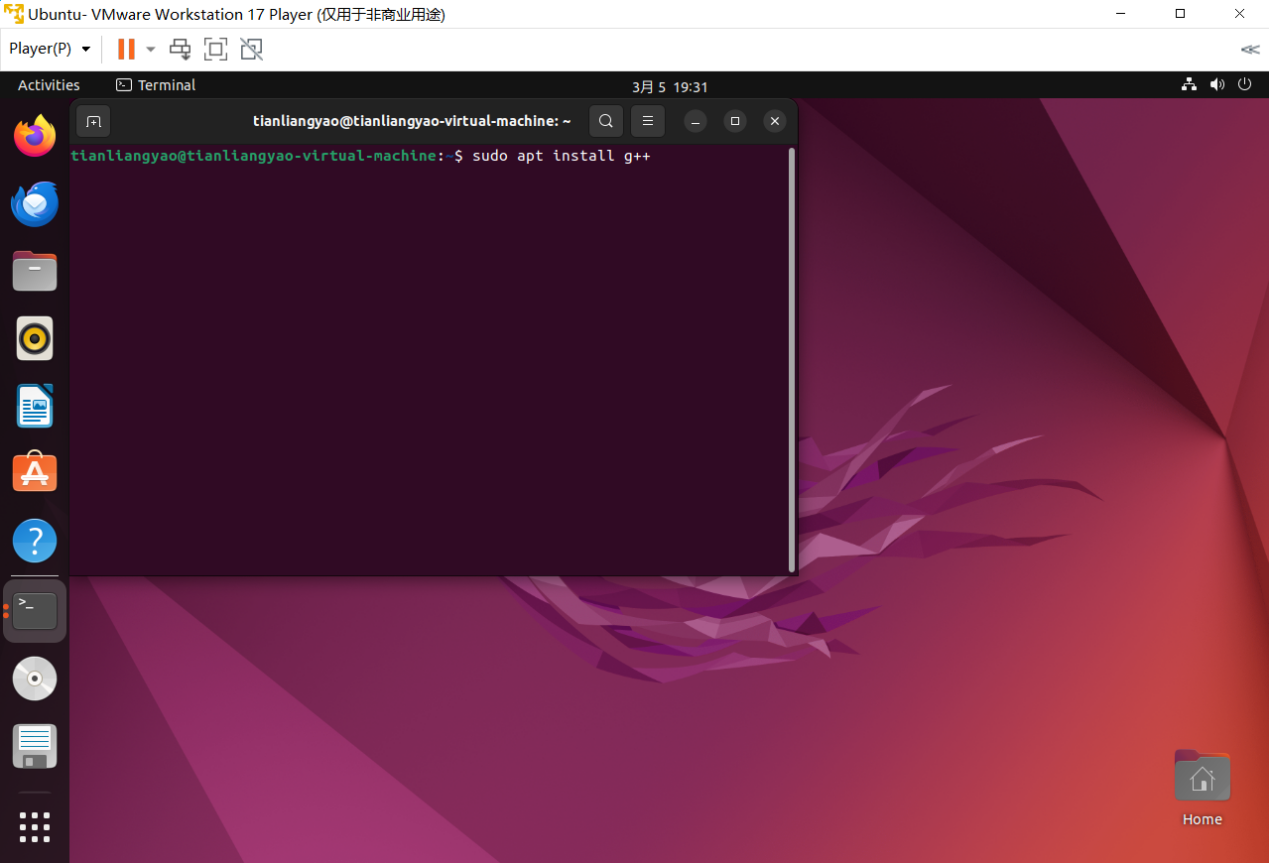
在安装过程中，我在后面新扩充的的G盘中添加，发现无法导入虚拟机。

**后在蒋老师的帮助下**，在原始自带的F盘中导入，成功安装。

**2.在Ubuntu Linux下安装一些有用的工具**

进入Ubuntu Linux桌面后，按CTRL+ALT+T打开终端，之后，安装如下库：

· git· gcc· make· wget· unzip· g++· cmake· vim



按要求添加下载即可

1. 使用git克隆opencv/eigen/sophus/geographiclib库

克隆opencv源代码

打开终端，使用以下命令创建工作目录。

mkdir -p ~/work/cv

此命令将在您的主目录下创建一个空白目录。然后使用以下命令进入该工作目录：

cd ~/work/cv

[opencv/eigen/sophus/geographiclib 源代码可以在https://git.tongji.edu.cn/image\_pro\_and\_cv/example\_code](https://git.tongji.edu.cn/image_pro_and_cv/example_code)中找到 请 注意，这些存储库包含这些项目的所有历史记录，当前（或 HEAD）版本是最新的，但不适合我们的实验。我们需要选择指定的版本（或者git系统中的BRANCH/TAG）。使用以下命令克隆 OpenCV 源代码：

git clone https://git.tongji.edu.cn/image\_pro\_and\_cv/example\_code/opencv.gitgit -C opencv checkout 4.x

克隆完成后，准备编译opencv，内容如下：

cd opencvmkdir -p buildcd buildcmake ..

等待cmake完成编译配置，然后

make -j4

注意：“-j4”表示使用 4 个线程来编译代码，请根据 vmware 设置中的 CPU 内核调整此数字。这个过程会持续半个小时，耐心等待。完成后，使用以下命令安装二进制文件：

sudo make install

默认情况下 OpenCV 将安装到 /usr/local 目录，所有文件将被复制到以下位置：

/usr/local/bin - executable files/usr/local/lib - libraries (.so)/usr/local/cmake/opencv4 - cmake package/usr/local/include/opencv4 - headers/usr/local/share/opencv4 - other files (e.g. trained cascades in XML format)

请参考[https://docs.opencv.org/4.9.0/d7/d9f/tutorial\_linux\_install.html](https://docs.opencv.org/4.9.0/d7/d9f/tutorial_linux_install.html" \t "_blank)

克隆特征源代码

返回 ~/work/cv，然后继续克隆 eigen 源代码。使用以下命令：

git clone https://git.tongji.edu.cn/image\_pro\_and\_cv/example\_code/eigen.git

克隆完成后，执行以下命令：

cd eigenmkdir -p buildcd buildcmake ..make -j4sudo make install

克隆sophus源代码

返回 ~/work/cv，然后继续克隆 sophus 源代码。使用以下命令：

git clone -b 1.22.10 --single-branch https://git.tongji.edu.cn/image\_pro\_and\_cv/example\_code/Sophus.git

克隆完成后，执行以下命令：

cd Sophusmkdir -p buildcd buildcmake ..make -j4sudo make install

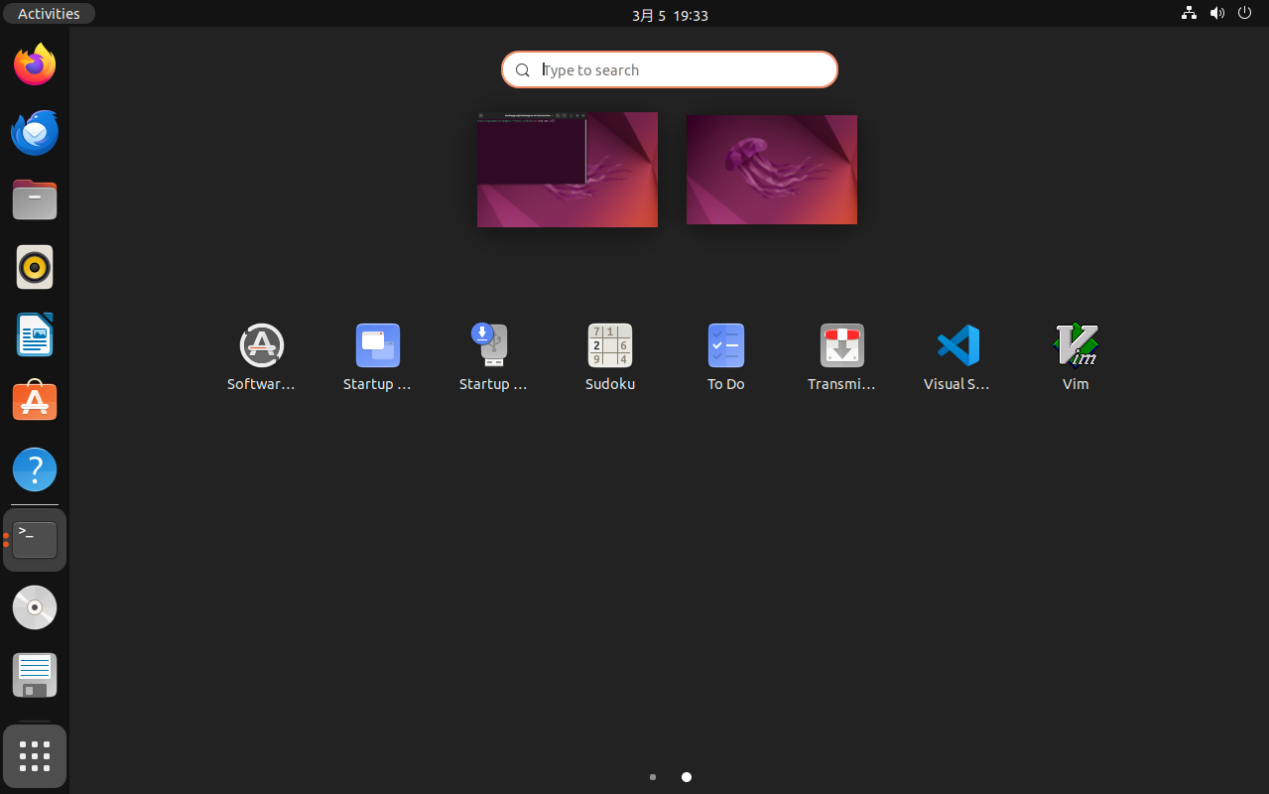
克隆geogelib源代码

返回〜/work/cv，然后继续克隆geologicallib源代码。使用以下命令：

git clone https://git.tongji.edu.cn/image\_pro\_and\_cv/example\_code/geographiclib.git

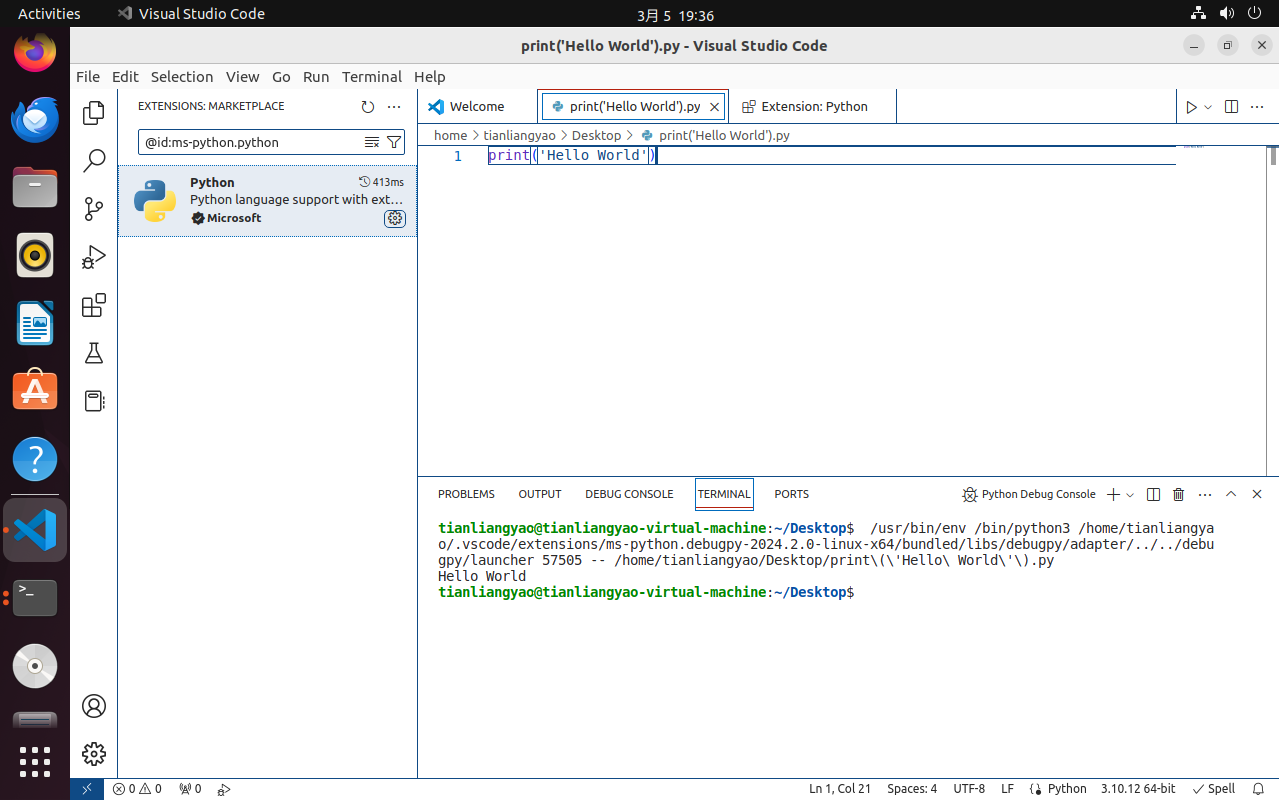
克隆完成后，执行以下命令：

cd geopraphiclibmkdir -p buildcd buildcmake ..make -j4sudo make install



我的visualstudio配置成功了，导入了最熟悉的python（深度学习做多了）

我利用python成功输出了Hello World，本次实验完成!



三、心得体会

这次实验帮助我熟悉了Linux系统，并掌握了在Ubuntu环境下配置OpenCV等代码库的过程。虽然遇到了一些小问题，但通过耐心等待和蒋磊老师的指导都得到了解决。

具体来说,有以下几点体会:

操作系统的选择很关键。Linux环境对编程和科研工作非常友好，但对新手来说需要一个适应过程。我之前在Windows下做过一些Python开发，这次切换到Linux就有点陌生。幸好Ubuntu系统界面比较直观，通过一步步跟着指导能较快上手。

配置开发环境需要耐心和细心。编译安装OpenCV、Eigen等庞大的C++库需要源码下载、cmake配置、多线程编译等繁琐步骤。过程中如果遇到错误需要积极查阅文档，寻求他人帮助。编程能力固然重要，但在科研领域，具备这种动手能力和解决问题的毅力同样关键。

实践是检验真理的唯一标准。经过将近2个小时的配置，当最后能在新环境里运行Hello World程序时,我获得了难能可贵的成就感。今后需要多动手实践，在不断的实践中积累经验。

合作的重要性。尽管本次实验是个人操作，但我获得了老师和同学的帮助。在编程和科研道路上，我们要学会相互支持、团结协作，发挥集体的智慧和力量。

总之，这是一次非常宝贵的实践机会。展望未来，我会在这个扎实的基础上，努力钻研图像处理与计算机视觉的方方面面，学好这门理论与实践并重的课程。