



同濟大學
TONGJI UNIVERSITY

IPMV-Experiment-3

Lab0.2 Get familiar with Eigen

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

1. Get familiar with basic operation of eigen
2. Basic linear algebra operations
3. Inter-operate between OpenCV and Eigen

二、实验过程记录

1. Get familiar with basic operation of eigen
2. Basic linear algebra operations

```
cmake_minimum_required(VERSION 3.5.1)
project(lab11_eigentest)    #set project name

include_directories("/usr/local/include/eigen3")

add_executable(${PROJECT_NAME} main.cpp)

target_compile_options(${PROJECT_NAME} PRIVATE ${compile_options})
set_target_properties(${PROJECT_NAME} PROPERTIES CXX_STANDARD_REQUIRED ON CXX_STANDARD 17)
```

elp



```
CMakeLists.txt x main.cpp 2
CMakeLists.txt
1 cmake_minimum_required(VERSION 3.5.1)
2 project(lab11_eigentest)    #set project name
3
4 include_directories("/usr/local/include/eigen3")
5
6 add_executable(${PROJECT_NAME} main.cpp)
7
8 target_compile_options(${PROJECT_NAME} PRIVATE ${compile_options})
9 set_target_properties(${PROJECT_NAME} PROPERTIES CXX_STANDARD_REQUIRED ON CXX_STANDARD 17)
10
11
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
dot product 1: 12
dot product 2: 12
[1] + Done          "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MI
pb"
tianliangyao@tianliangyao-virtual-machine:~/work/lab00$
```

```
CMakeLists.txt  main.cpp 2 X
main.cpp > ...
1  #include <Eigen/Core>
2  #include <iostream>
3
4  int main() {
5      Eigen::Vector3d a(1, 1, 1);
6      Eigen::Vector3d b(3, 4, 5);
7
8      // Method 1: a.dot(b)
9      std::cout << "dot product 1: " << a.dot(b) << std::endl;
10
11     // Method 2: a.transpose() * b
12     std::cout << "dot product 2: " << a.transpose()*b << std::endl;
13
14     return 0;
15 }
```

```
CMakeLists.txt  main.cpp 2 X
main.cpp > main()
1  #include <iostream>
2  #include <Eigen/Dense>
3
4  int main() {
5      Eigen::Vector3d a(1, 1, 1);
6      Eigen::Vector3d b(3, 4, 5);
7      std::cout << "cross product:" << std::endl << a.cross(b) << std::endl;
8
9      return 0;
10 }
```

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

cross product:
1
-2
1
[1] + Done          "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<
uk"
tianliangyao@tianliangyao-virtual-machine:~/work/lab00$
```

创建一个 CMake 项目，将 Eigen 库包含目录加入到项目中。

在 main.cpp 文件中添加头文件 `#include <eigen3/Eigen/Core>`，使用 Eigen 名称空间。

创建一个 4x4 随机矩阵对象 `MatrixXd m` 和 4 维随机向量对象 `VectorXd v`，使用 `setRandom()` 函数对它们进行随机值初始化。

输出矩阵 `m` 查看初始化效果。

创建一个固定值矩阵对象做后续操作。

使用块操作 `block()` 替换矩阵 `m` 第 3、4 行第 1、2 列元素为零矩阵，再将此块元素设置具体值。

输出最终修改后的矩阵查看效果。

结果:学会使用 Eigen 库简单高效地创建和初始化矩阵向量对象, 利用块操作方便地修改矩阵元素, 初步掌握了 Eigen 库在线性代数基本运算中的应用。

总结:本实验利用 Eigen 库实现了矩阵和向量的创建、初始化和元素修改操作, 为后续更复杂计算奠定基础。Eigen 库利用 C++模板实现了线性代数运算, 大大提升了效率。

3.OpenCV 与 Eigen 连用

3.1 将 OpenCV Mat 转换为 Eigen 矩阵

使用 `cv::cv2eigen` 函数将 OpenCV Mat 转换为 Eigen 矩阵。注意 Eigen 矩阵元素通常为 `double` 类型,而 OpenCV 矩阵元素类型可能为整数或 OpenCV 定义类型,需要先进行类型转换。

该函数需要包含头文件:`#include <Eigen/Dense> #include <Eigen/Core> #include <opencv2/core/eigen.hpp>`

3.2 使用 Eigen 函数操作图像

利用 Eigen 矩阵的各种函数如乘法、向量运算等进行图像像素值的计算和处理。

3.3 (扩展任务)使用 Eigen 函数旋转图像 15 度

使用 OpenCV 绘制矩形或读取图片生成 Mat 类型图像,转换为 Eigen 矩阵,利用 Eigen 旋转函数生成逆时针旋转 15 度的图像结果。

3.4 (扩展任务)使用 Gaussian 模糊算法

读取图片,转换为 Eigen 矩阵。利用 Eigen 创建 5×5 高斯核,标准差为 10.0 的高斯模糊算子,对图像矩阵进行卷积运算。反向转换为 OpenCV 格式显示结果。

3.5 (扩展任务)Canny 边缘检测

读取图片,将其转换为 Eigen 矩阵。利用 Eigen 实现 Sobel 算子等算法,进行 Canny 边缘检测,生成边缘图像矩阵,转换回 OpenCV 格式显示结果。

总体运用 Eigen 库实现图像基本操作,如转换、旋转、滤波等,掌握 Eigen 在计算机视觉中的应用。

找了个 KITTI 数据集中的样例图,通过系列 OpenCV 图像预处理与 Eigen 计算操作,得到可视化结果如下:



三、心得体会

这次实验我使用 Eigen 库熟悉了线性代数基本运算,以及结合 OpenCV 进行图像处理的基本流程。使用 Eigen 创建和初始化矩阵、矢量对象,比较直观方便。setRandom()函数实现随机数初始化很便利。块操作方便地修改矩阵元素值,本身数据结构设计考虑了效率。OpenCV 与 Eigen 之间利用 cv2eigen 与 eigen2cv 函数进行无缝转换,实现两个库 seamless 连接,处理图像数据更高效。Eigen 提供丰富的线性代数函数库,如矩阵乘法、旋转变换等,直接使用帮助进行各种基础图像处理任务。配合 OpenCV 读写图像,利用 Eigen 函数处理像素值,得到旋转、滤波等预处理结果,流程清晰。扩展任务通过实现 Canny 边缘检测等算法,进一步加深理解 Eigen 库在计算机视觉中的应用。总体来说,这次实验让我学习到利用 C++模板设计的 Eigen 库在线性代数运算和图像处理中的强大功能。学习如何高效地选择不同库进行协同工作,为后续人工智能、计算机视觉等学习打下基础。未来需要进一步熟悉 Eigen 各类函数,在具体任务中灵活应用;同时学习源码理解其设计思想,能力获得提升。这个实验对我很有帮助。