

# Lab 3 - 实现稀疏矩阵以及高斯赛达尔迭代法

---

本次实验内容为课程作业，计算成绩。你需要将**源代码、可执行程序（注明运行环境）和实验文档**上传至学在浙大，并将压缩包命名为 lab3-学号-姓名.rar/zip。

本次作业提交截止时间：2020年3月31日 23:59:59，逾期将要扣分。

## 环境要求与编程语言

---

计算摄影学课程假定同学们使用 Windows 10 操作系统并安装有（至少）Visual Studio 2015。

你也可以从 <http://ms.zju.edu.cn/> 免费下载 Visual Studio 2015/2017。

OpenCV 的官方网站是 <http://opencv.org/>，课程推荐使用 OpenCV 3。

请使用C++或者C语言作为编程语言。

## 稀疏矩阵

---

本实验要求自行实现一种稀疏矩阵。推荐的稀疏矩阵存储方式包括：

1. Compressed Row Storage
2. Compressed Sparse Column
3. 其他形式的稀疏矩阵存储方式

同学们请在实验报告中明确指出自己实现的稀疏矩阵方式，并保证实现的稀疏矩阵具备以下的几种最基本的功能：

1. `at(row, col)` : 根据 row 和 column 的系数来查询矩阵里面的元素的数值
2. `insert(val, row, col)` : 将 val 替换/插入到 (row, col) 这个位置去
3. `initializeFromVector(rows, cols, vals)` : 根据向量来初始化一个稀疏矩阵。其中 rows, cols, vals 皆为等长度的向量。rows 里面存的是行系数，cols 里面存的是列系数，vals 里面存的是数值。

4. 其余的基本功能可以参考Matlab里面的[sparse](#)函数，或者[Eigen Library](#)里面的[Sparse Matrix](#)的介绍。

备注：关于CRS与CSC的实现方式可以参考下面的网站：

- [http://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csr\\_matrix.html](http://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csr_matrix.html)
- [http://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csc\\_matrix.html](http://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csc_matrix.html)
- [http://en.wikipedia.org/wiki/Sparse\\_matrix](http://en.wikipedia.org/wiki/Sparse_matrix)
- [http://eigen.tuxfamily.org/dox/group\\_\\_TutorialSparse.html](http://eigen.tuxfamily.org/dox/group__TutorialSparse.html)

此外，有兴趣同学也可以尝试将自己的实现与[Eigen Library](#)的[Sparse Matrix](#)进行比较，看是自己的实现比较高效还是Eigen Library的实现比较高效。

## 稀疏矩阵的高斯赛达尔迭代法

---

本实验要求在自己实现的稀疏矩阵的表达式的基础上，实现高斯赛达尔迭代法（Gauss-Seidel Method），用于求解大规模的稀疏线性方程组。

关于高斯赛达尔方法的介绍已经在课件中进行了详细的介绍。请同学们参照课件进行代码的编写。

此外还有一些额外的辅助资料提供参考：

- <http://mathworld.wolfram.com/Gauss-SeidelMethod.html>
- [http://en.wikipedia.org/wiki/Gauss%E2%80%93Seidel\\_method](http://en.wikipedia.org/wiki/Gauss%E2%80%93Seidel_method)

作为验证实现的正确性，这里给出一个小的Test Case:

$$A = \begin{pmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{pmatrix}$$

$$b = (6, 25, -11, 15)^T$$

求解 得到的结果应该为：

$$x = [1, 2, -1, 1];$$

## Bonus: 实现共轭梯度法

---

有余力的同学可以参考相关资料实现共轭梯度法求解线性方程组。

维基百科上有比较详细的共轭梯度法的介绍：[Conjugate gradient method](#)