

CS213, 2001年秋季

实验作业L4：代码优化分配。10月11日

到期：10月25日11:59PM

Sanjit Seshia (sanjit+213@cs.cmu.edu) 是这项任务的牵头人。

1 简介

这项作业涉及优化内存密集型代码。图像处理提供了许多可以从优化中受益的函数的例子。在这个实验室中，我们将考虑两个图像处理操作：旋转，将图像逆时针旋转 90° ，以及平滑，将图像"平滑"或"模糊"。

在这个实验中，我们将考虑把图像表示为一个二维矩阵 M ，其中 M_{ij} 表示 M 的 (i, j) 个像素的值。像素值是红、绿、蓝（RGB）值的三倍。我们将只考虑方形图像。让 N 表示图像的行（或列）数。行和列的编号是C型的，从0到 $N-1$ 。

鉴于这种表示方式，旋转操作可以很简单地实现为以下两种矩阵操作的组合。

- **转置**。对于每个 (i, j) 对， M_{ij} 和 M_{ji} 被互换。
- **交换行**。第 i 行与第 $N-1-i$ 行进行交换。

这种组合在图1中得到说明。

平滑操作是通过用它周围所有像素的平均值（在以该像素为中心的最大 3×3 的窗口中）替换每个像素值来实现的。请看图2。像素 $M[1][1]$ 和 $M[N-1][N-1]$ 的值如下。

$$M[1][1] = \frac{\sum_{i=0}^2 \sum_{j=0}^2 M[i][j]}{9}$$
$$M[N-1][N-1] = \frac{\sum_{i=N-2}^{N-1} \sum_{j=N-2}^{N-1} M[i][j]}{4}$$

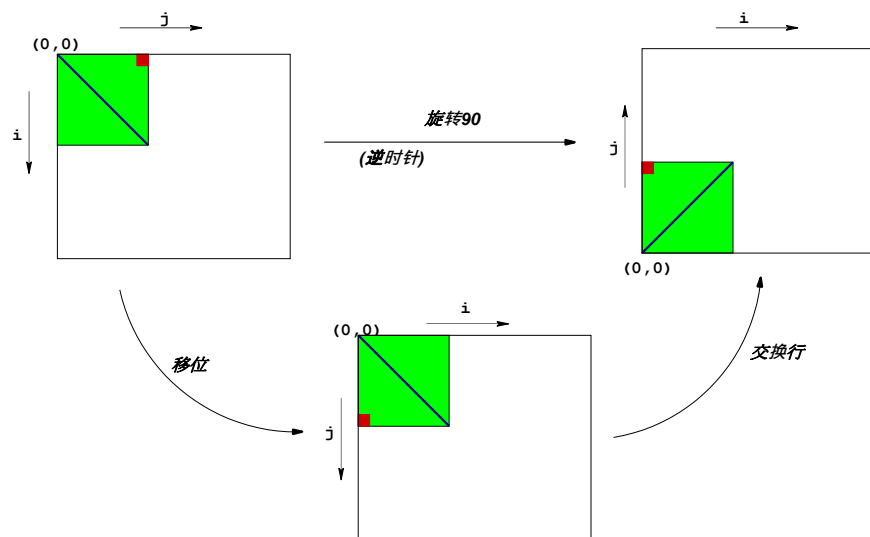


图1：将图像逆时针旋转90°

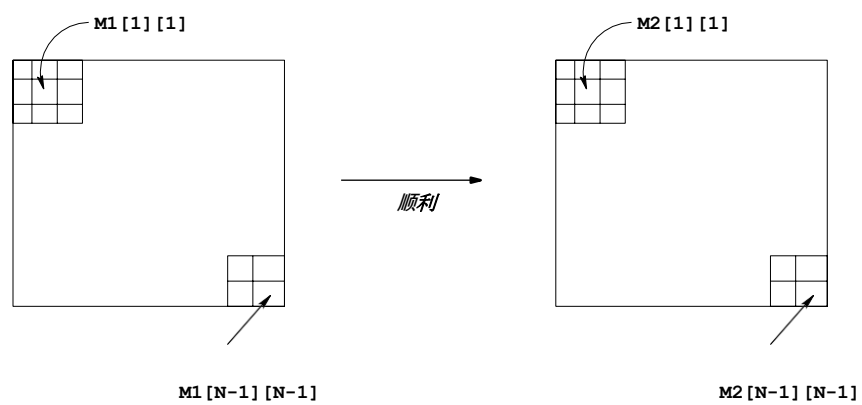


图2：对图像进行平滑处理

2 物流

你可以在一个最多两个人的小组中工作，解决这个作业的问题。唯一的"上交"将是电子的。对作业的任何澄清和修改都将在课程网页上公布。

3 发放指示

在此插入一段话，说明教师将如何向学生分发perflab-handout.tar文件。

首先，将perflab-handout.tar复制到一个你打算做工作的受保护的目录中。然后发出命令：`tar xvf perflab-handout.tar`，这将导致一些文件被解压到该目录中。你要修改和交出的唯一文件是`kernels.c`。`driver.c`程序是一个驱动程序，允许你评估你的解决方案的性能。使用命令`make driver`来生成驱动代码，并使用命令`./driver`来运行它。

看一下文件`kernels.c`，你会注意到一个C结构的团队，你应该在其中插入所要求的关于组成你的编程团队的一两个人的识别信息。马上就做，这样你就不会忘记。

4 实施概述

数据结构

核心数据结构处理图像表示。一个像素是一个结构，如下图所示。

```
Typedef struct {
    unsigned short red; /* R 值 */
    unsigned short green; /* G 值 */
    unsigned short blue; /* B值 */
} 像素。
```

可以看出，RGB值有16位表示（"16位颜色"）。一个图像I被表示为一个一维的像素阵列，其中 (i, j) 个像素是`I[RIDX(i, j, n)]`。这里n是图像矩阵的维度，RIDX是一个宏，定义如下。

```
#define RIDX(i, j, n) ((i)*(n)+(j))
```

这段代码见文件`defs.h`。

旋转

下面的C函数计算了将源图像src旋转90的结果，并将结果存储在destination图像dst中。dim是图像的尺寸。

```

void naive_rotate(int dim, pixel *src, pixel *dst) {
    int i, j.。

    for(i=0; i < dim; i++)
        for(j=0; j < dim; j++)
            dst[RIDX(dim-1-j,i,dim)] = src[RIDX(i,j,dim)]。

    返回。
}

```

上面的代码扫描源图像矩阵的行，复制到目标图像矩阵的列。你的任务是重写这段代码，利用代码运动、循环解卷和阻塞等技术使其尽可能快地运行。

这段代码见文件kernels.c。

顺利

平滑化函数将源图像src作为输入，并在目标图像中返回平滑化的结果。dst。这里是一个实现的一部分。

```

void naive_smooth(int dim, pixel *src, pixel *dst) {
    int i, j.。

    for(i=0; i < dim; i++)
        for(j=0; j < dim; j++)
            dst[RIDX(i,j,dim)] = avg(dim, i, j, src); /*平滑(i,j)的像素 */

    返回。
}

```

函数avg返回第(i,j)个像素周围所有像素的平均值。你的任务是优化smooth（和avg），使其尽可能快地运行。（注意：函数avg是一个局部函数，你可以完全摆脱它，用其他方式实现smooth。）

这段代码（以及avg的实现）在文件kernels.c中。

绩效措施

我们的主要性能指标是CPE或每元素循环数。如果一个函数对 $N \times N$ 大小的图像运行需要 C 个周期，那么CPE值就是 C/N^2

。表1总结了上述天真实现的性能，并将其与优化实现进行了比较。表中显示了5个不同的 N 值的性能。所有的测量都是在Pentium III Xeon Fish机器上进行的。

优化后的实现与天真的实现的比率（加速）将构成你的实现的得分。为了总结不同 N 值的总体效果，我们将计算这5个值的结果的几何平均值。也就是说，如果 $N=\{32, 64, 128, 256, 512\}$ 的测量速度是 R_{32} ， R_{64} ， R_{128} ， R_{256} ，和 R_{512} ，那么我们计算的整体性能为

$$R = \sqrt[5]{R_{32} \times R_{64} \times R_{128} \times R_{256} \times R_{512}}$$

测试案例	1	2	3	4	5	
方法 N	64	128	256	512	1024	地质学。平均值
奈何轮转 (CPE)	14.7	40.1	46.4	65.9	94.5	
优化的轮转 (CPE)	8.0	8.6	14.8	22.1	25.3	
速度提升 (天真/选择)	1.8	4.7	3.1	3.0	3.7	3.1
方法 N	32	64	128	256	512	地质学。平均值
天真流畅 (CPE)	695	698	702	717	722	
优化后的平稳 (CPE)	41.5	41.6	41.2	53.5	56.4	
速度提升 (天真/选择)	16.8	16.8	17.0	13.4	12.8	15.2

表1：优化后的CPE和比率与无知的执行情况对比

假设

为了方便起见，你可以假设 N 是32的倍数。你的代码必须对所有这样的 N 值都能正确运行，但我们将只对表1中的5个值进行性能测试。

5 基础设施

我们已经提供了支持代码，以帮助你测试你的实现的正确性，并测量它们的性能。本节描述了如何使用这个基础设施。任务的每一部分的确切细节将在下一节中描述。

注意：你要修改的唯一源文件是`kernels.c`。

版本管理

你将会编写许多版本的旋转和平滑例程。为了帮助你比较你所写的所有不同版本的性能，我们提供了一种"注册"函数的方法。

例如，我们提供给你的`kernels.c`文件包含以下函数。

```
void register_rotate_functions() {
    add_rotate_function(&rotate, rotate_descr);
}
```

这个函数包含一个或多个对添加旋转函数的调用。在上面的例子中。

`add_rotate` 函数在注册函数`rotate`的同时还注册了一个字符串`rotate_descr`，这是一个关于该函数作用的ASCII描述。参见文件`kernels.c`以了解如何创建字符串描述。这个字符串最多可以有256个字符。

文件`kernels.c`中为你的平滑内核提供了一个类似的函数。

驱动程序

你将编写的源代码将与提供的目标代码连接成一个驱动二进制。为了创建这个二进制文件，你将需要执行以下命令

```
unix> make driver
```

每次改变kernels.c中的代码时，你都需要重新制作驱动程序。要测试你的实现，你可以运行命令。

```
unix> ./driver
```

该驱动器可以在四种不同的模式下运行。

- **默认模式**，你的实现的所有版本都在这个模式下运行。
- **Autograder模式**，其中只运行rotate()和smooth()函数。这就是我们在使用驱动对你的交接进行评分时要运行的模式。
- **文件模式**，即只运行输入文件中提到的版本。
- **转储模式**，在这种模式下，每个版本的单行描述被转储到一个文本文件。然后你可以编辑这个文本文件，只保留那些你想用文件模式测试的版本。你可以指定在转储文件后是否退出，或者是否要运行你的实现。

如果运行时没有任何参数，驱动程序

将运行你的所有版本（默认模式）。其他模式和选项可以通过命令行参数指定给驱动程序，如下所示。

- g :只运行rotate()和smooth()函数（autograder模式）。
- f <funcfile> :只执行<funcfile>中指定的那些版本（文件模式）。
- d <dumpfile> :将所有版本的名称转储到一个名为<dumpfile>的转储文件中，一个版本有一行（转储模式）。
- q :在将版本名称转储到转储文件后退出。与-d同时使用。例如，要在打印完转储文件后立即退出，请输入./driver -qd dumpfile。
- h :打印命令行的用法。

团队信息

重要提示：在你开始之前，你应该在kernels.c中的结构中填写你的团队信息（小组名称、团队成员姓名和电子邮件地址）。这些信息就像数据实验室的信息一样。

6 任务详情

优化旋转 (50分)

在这一部分，你将优化旋转，以实现尽可能低的CPE。你应该编译驱动程序，然后用适当的参数运行它来测试你的实现。

例如，用所提供的naive版本（用于旋转）运行驱动程序会产生如下的输出。

```
unix> ./driver 团队名称
:bovik
成员 1: Harry Q. Bovik 电子邮
件 1: bovik@nowhere.edu
```

旋转。版本 = naive_rotate:天真的基线实现。

点状	64	128	256	512	1024	平均值
你的CPE	14.6	40.9	46.8	63.5	90.9	
基线CPEs	14.7	40.1	46.4	65.9	94.5	
加速	1.0	1.0	1.0	1.0	1.0	1.0

优化平稳 (50分)

在这一部分中，你将优化平滑，以实现尽可能低的CPE。

例如，用所提供的天真版运行驱动程序（用于平滑），会产生如下的输出。

```
unix> ./driver
```

平滑。版本 = naive_smooth:天真的基线实现。

点状	32	64	128	256	512	平均值
你的CPE	695.8	698.5	703.8	720.3	722.7	
基线CPEs	695.0	698.0	702.0	717.0	722.0	
加速	1.0	1.0	1.0	1.0	1.0	1.0

一些建议。看看为旋转和平滑生成的汇编代码。着重于使用课堂上所讲的优化技巧来优化内循环（在一个循环中被重复执行的代码）。与旋转函数相比，平滑函数的计算量更大，对内存的敏感度更低，所以优化的内容也有一定的不同。

编码规则

你可以写任何你想要的代码，只要它满足以下几点。

- 它必须是ANSI C语言，你不能使用任何嵌入式汇编语言语句。
- 它不能干扰时间测量机制。如果你的代码打印出任何不相干的信息，你也会受到惩罚。

你只能修改kernels.c中的代码。你可以在这些文件中定义宏、附加全局变量和其他程序。

评价

你的旋转和平滑的解决方案将各占你成绩的50%。每项的得分将基于以下几点。

- 正确性。对于导致驱动程序抱怨的错误代码，你将不会得到任何奖励。这包括在测试尺寸上正确操作

，但在其他尺寸的图像矩阵上不正确的代码。如前所述，你可以假设图像尺寸是32的倍数。

- CPE：如果你的旋转和平滑的实现是正确的，并且分别达到了平均CPE的阈值 S_r 和 S_s 以上，你将得到全分。如果你的正确实现比提供的天真实现更好，你将得到部分学分。

特定网站：作为教员，你需要决定你的满分门槛 S_r 和 S_s 以及你的部分学分规则。我们通常使用一个线性比例，如果学生真的尝试解决实验问题，那么最低比例为40%。

7 交付说明

特定网站：在此插入一段话，告诉每个团队如何交出他们的内核文件。例如，以下是我们在CMU使用的交接说明。

当你完成实验后，你将交出一个文件，`kernels.c`，其中包含你的解决方案。以下是如何交出你的解决方案。

- 确保你在`kernels.c`的团队结构中包含了你的识别信息。
- 确保`rotate()`和`smooth()`函数与你最快的实现相对应，因为当我们使用驱动对你的作业进行评分时，这些是唯一会被测试的函数。
- 删除任何多余的打印语句。
- 创建一个表格的团队名称。
 - " ID "，其中 ID 是你的安德鲁ID，如果你是单独工作，或
 - " $ID_1 + ID_2$ " 其中 ID_1 是第一个团队成员的Andrew ID， ID_2 是第二个团队成员的Andrew ID。

这应该与你在`kernels.c`结构中输入的团队名称相同。

- 要交出你的`kernels.c`文件，请输入。

进行交接 `TEAM=teamname`

其中`teamname`是上述的团队名称。

- 交稿后，如果你发现有错误，想提交一份修改稿，请输入

`make handin TEAM=teamname VERSION=2`

每次提交时都要不断递增版本号。

- 你可以通过查看以下内容验证你的交接

`/afs/cs.cmu.edu/academic/class/15213-f01/L1/handin`

你在这个目录中有列表和插入的权限，但没有读或写的权限。

好运!