CS213,2001年秋季
实验室分配L4：代码优化分配：10月11日
截止日期：10月25日晚上11：59

SanjitSeshia(sanjit+213@cs.cmu.edu)是这个任务的负责人。

1 介绍

此任务处理优化内存密集型代码。图像处理提供了许多可以从优化中获益的函数示例。在这个实验室中，我们将考虑两种图像处理操作：旋转，它逆时针旋转图像90°，和平滑，它“平滑”或“模糊”一个图像。

对于这个实验室，我们将考虑将图像表示为二维矩阵M，其中$M_{i,j}$表示M的（i、j）像素的值。像素值是包含红色、绿色和蓝色(RGB)值的三元组。我们将只考虑正方形的图像。设N表示图像的行数（或列）。行和列以-样式从0到N-1进行编号。

鉴于这种表示，旋转操作可以很简单地实现为以下两个矩阵操作的组合：

- *转位符：对于每一个（i、j）对，*$M_{i,j}$和$M_{j,i}$是互换的。
- 交换行：第i行与第N-1-i行交换。

这种组合如图1所示。

平滑操作是通过用其周围所有像素的平均值替换每一个像素值（以该像素为中心的最多3×3像素）来实现的。考虑一下图2。像素M2[1][1]和M2的值如下：

$$M2[1][1] = \frac{\sum_{i=0}^{2}\sum_{j=0}^{2} M1[i][j]}{9}$$

$$M2[N-1][N-1] = \frac{\sum_{i=N-2}^{N-1}\sum_{j=N-2}^{N-1} M1[i][j]}{4}$$

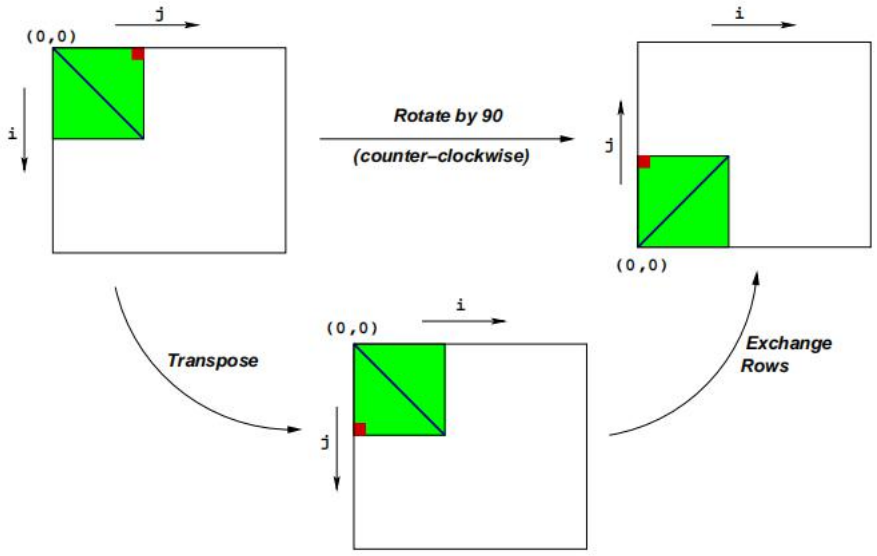

图1：图像旋转90°逆时针方向的

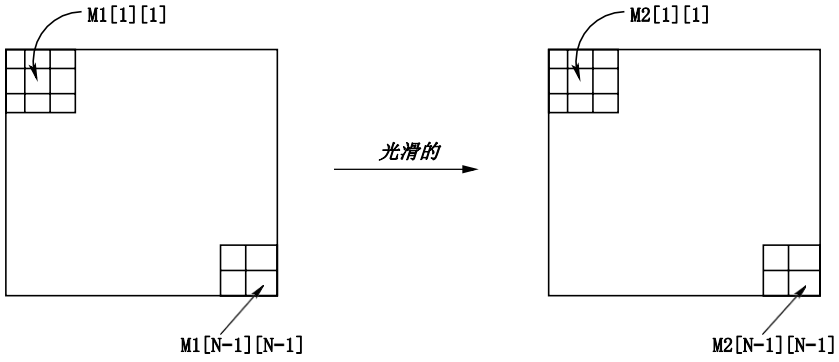

图2：平滑一个图像

2 物流

你可以用最多两个人来解决这个任务的问题。唯一的“手持手机”将是电子化的。对作业的任何澄清和修改都将张贴在课程网页上。

3 分发说明

特定地点：在这里插入一段，解释教师将如何向学生分发perflab-handout.tar文件。

首先将perflab-handout.tar复制到您计划在其中执行工作的受保护的目录中。然后发出命令：tarxvfperflab-handout.tar。这将导致许多文件被解压缩到目录中。唯一要修改和提交的文件是neele.c。驱动程序。c程序是一个驱动程序的程序，它允许您评估解决方案的性能。使用命令使驱动程序生成驱动程序代码，并使用该命令运行它。/驱动程序。

查看文件内核。c您会注意到一个C结构团队，您应该在其中插入所请求的关于包含编程团队的一两个人的标识信息。**马上这么做，这样你就不会忘记了。**

4 实施概述

数据结构

核心数据结构处理图像表示。一个像素是一种结构，如下图所示：

```
打字机结构{
    无符号短红色；/*R值*未签名的绿色*G值：在
    辐射效应中*无符号的蓝色短写*B值*/
}像素；
```

可以看出，RGB值有16位表示（“16位颜色”）。图像I表示为一维像素数组，其中（i、j）像素为I[RIDX（i、j、n）]。这里n是图像矩阵的维数，RIDX是一个宏，定义如下：

```
#定义RIDX（i、j、n）（(i)*(n)+(j)）
```

有关此代码，请参见文件defs.h。

旋转

下面的C函数计算出将源图像src旋转90的结果°并将结果存储在目标图像dstdim是图像的尺寸。.

```
空naive_rotate(intdim，像素*src，像素*{inti，j；

  为(i=0；i<dim；i++)为(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。您的任务是重写此代码，以使用代码运动、循环展开和阻塞等技术，使其尽可能快地运行。

有关此代码，请参见文件neeles.c。

平滑

平滑函数以源图像src作为输入，并在目标图像中返回平滑结果
dst.以下是一个实现的一部分：

```
空naive_smooth(intdim，像素*src，像素*{inti，j；

  为(i=0；i<dim；i++)为(j=0；
    j<dim；j++)
      深[RIDX(i、j、dim)]=avg(暗、i、j、src)；/*平滑第（i、j）像素*/

  返回；
}
```

函数avg返回（i、j）个像素周围所有像素的平均值。您的任务是优化平滑（和平均），以尽可能快地运行。(注意：该函数avg是一个本地函数，您可以完全摆脱它，以以其他方式实现平滑。)

这段代码(以及avg的一个实现)在文件内核。c中。

绩效指标

我们的主要性能指标是CPE或每个元素的周期。如果一个函数需要C周期来运行大小为N×N的图像，则CPE值为C/N^2. 表1总结了上面所示的朴素实现的性能，并将其与优化的实现进行了比较。显示了5个不同的N值的性能。所有的测量都是在奔腾III至强鱼机上进行的。

优化实现的比率比幼稚实现的比率（加速）将构成您的实现的一个分数。为了总结在不同的N值上的总体效应，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}的测量加速速度为R_{32}，R_{64}，R_{128}，R_{256}，和R_{512}然后我们计算整体性能为

$$R \stackrel{\surd}{=} \frac{}{R_{32}\times R_{64}\times R_{128}\times R_{256}\times R_{512}}$$

测试用例	1	2	3	4	5	
方法 N	64	128	256	512	1024	Geom。平均
朴素旋转(CPE)	14.7	40.1	46.4	65.9	94.5	
优化旋转(CPE)	8.0	8.6	14.8	22.1	25.3	
Speedup（幼稚/选择）	1.8	4.7	3.1	3.0	3.7	3.1
方法 N	32	64	128	256	512	Geom。平均
朴素光滑(CPE)	695	698	702	717	722	
优化光滑(CPE)	41.5	41.6	41.2	53.5	56.4	
Speedup（幼稚/选择）	16.8	16.8	17.0	13.4	12.8	15.2

表1：优化与优化之间的cpe和比率。朴素的实现

假设

为了使生活更轻松，你可以假设N是32的倍数。您的代码必须对所有这样的N值正确运行，但我们将仅对表1中所示的5个值测量其性能。

5 基础设施

我们提供了支持代码，以帮助您测试实现的正确性，并衡量它们的性能。本节介绍如何使用此基础设施。下一节将描述分配的每个部分的具体细节。

注意：您唯一要修改的源文件是nerels.c。

验证

您将编写许多版本的旋转和平滑的例程。为了帮助您比较您所编写的所有不同版本的性能，我们提供了一种“注册”功能的方法。

例如，我们提供的文件neele.c包含以下函数：

```
voidregister_rotate_functions(){add_rotate_function(&旋转，
    rotate_descr)；
}
```

此函数包含一个或多个要添加旋转函数的调用。在上面的例子中，
添加旋转函数注册旋转与字符串旋转descr，这是函数的ASCII描述。请参见文件nerle.c以查看如何创建字符串描述。这个字符串最多可长256个字符。

在文件内核。c中也为平滑内核提供了类似的函数。

司机

您将编写的源代码将与我们提供的一个驱动程序二进制文件的对象代码链接。要创建此二进制文件，您将需要执行该命令

Unix>制造驱动程序

每次更改nerels.c中的代码时，您都需要重新生成驱动程序。要测试实现，您可以运行以下命令：

unix>./driver

该驱动程序可以以四种不同的模式运行：

- 默认模式，其中运行所有版本的实现。
- *自动分级模式，其中只运行旋转的()和平滑的()函数。*这是我们在使用驱动程序对您的手持进行分级时将运行的模式。
- 文件模式，其中只运行在输入文件中提到的版本。
- 转储模式，其中每个版本的一行描述被转储到一个文本文件中。然后，可以编辑此文本文件以只保留使用文件模式测试的版本。您可以指定是在倾倒文件后退出，还是要运行实现。

如果没有任何参数就可以运行，则驱动程序将运行您的所有版本（默认模式）。其他模式和选项可以通过对驱动程序的命令行参数来指定，如下所示：

-g：仅运行旋转的()和平滑的()函数（自动分级模式）。

-f<功能文件>：仅执行在<功能文件>（文件模式）中指定的版本。

-d<垃圾文件>：将所有版本的名称转储到一个名为<垃圾文件>的转储文件中，将一行转储到一个版本（转储模式）。

-q：将版本名称转储到转储文件后退出。与-d一起使用。例如，要在打印转储文件后立即退出，请键入。/drived-qd垃圾程序文件。

-h：打印命令行的用法。

团队信息

重要提示：在开始之前，应在norle.c中填写有关团队的信息（组名称、团队成员名称和电子邮件地址）。这个信息和数据实验室的信息一样。

6 分配详细信息

优化旋转（50分）

在本部分中，您将优化旋转，以实现尽可能低的CPE。您应该编译驱动程序，然后使用适当的参数运行它来测试实现。

例如，使用提供的原始版本（用于旋转）运行驱动程序将生成如下所示的输出：

Unix>。/驱动工具名称：bovik
电子邮件1：bovik@nowhere.edu

旋转：版本=naive_rotate：朴素的基线实现：

昏暗的	64	128	256	512	1024	平均
您的CPEs	14.6	40.9	46.8	63.5	90.9	
基线CPEs	14.7	40.1	46.4	65.9	94.5	
高速公路	1.0	1.0	1.0	1.0	1.0	1.0

优化平滑（50分）

在这部分中，您将优化平滑，以实现尽可能低的CPE。

例如，使用提供的原始版本（为了平滑）运行驱动程序将生成如下所示的输出：

unix>./driver

平滑：版本=naive_smooth：朴素的基线实现：

昏暗的	32	64	128	256	512	平均
您的CPEs	695.8	698.5	703.8	720.3	722.7	
基线CPEs	695.0	698.0	702.0	717.0	722.0	
高速公路	1.0	1.0	1.0	1.0	1.0	1.0

一些建议。查看为旋转和平滑而生成的装配代码。重点是使用类中涵盖的优化技巧来优化内环（在循环中重复执行的代码）。与旋转函数相比，平滑度对计算更密集，内存敏感性更低，因此优化的味道有些不同。

编码规则

您可以编写任何您想要的代码，只要它满足以下条件：

- 它必须是在ANSIC中。您可能不能使用任何嵌入式汇编语言语句。
- 不得干扰时间测量机制。如果您的代码打印了任何无关的信息，您也将受到惩罚。

您只能修改nerels.c中的代码。您可以在这些文件中定义宏、其他全局变量和其他过程。

评价

你的解决方案将占你成绩的50%。每个项的分数将基于以下内容：

- 正确性：你将不会得到任何信用的错误的代码，导致司机抱怨！这包括在测试大小上正确操作，但在其他大小的图像矩阵上操作错误的代码。如前所述，您可以假设图像尺寸是32的倍数。

- CPE：如果你是正确的旋转和平稳的实现，并达到超过阈值S的平均CPE，你将获得充分的赞扬$_{T}$和S_s各自地您将获得比提供的幼稚实现更好的正确实现的部分学分。

  具体网站：作为讲师，你需要决定你的全部信用阈值S_{s}和S_s以及你对部分信用额度的规则。我们通常使用线性量表，如果学生真的试图解决实验室问题，其最小值约为40%。

7 手动说明

特定于站点的：在这里插入一段话，告诉每个团队如何处理他们的内核。c
提出例如，这里是我们在CMU上使用的手持指令。

当您完成实验室后，您将提交一个包含解决方案的文件nerle.c。以下是如何提交您的解决方案：

- 确保您已经在norles.c的团队结构中包含了标识信息。
- 确保旋转()和平滑()函数对应于您最快的实现，因为当这些是我们使用驱动程序为您的分配分级时将测试的唯一函数。
- 删除任何无关的打印语句。
- 创建表单的团队名称：
  - “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$_1$是第一团队成员的AndrewID和ID吗$_2$是第二个团队成员的安德鲁·ID。

  这应该与您在nerles.c中的结构中输入的团队名称相同。
- 要交入nerels.c文件，请键入：

  制作手机TEAM=团队名称

  其中，团队名称是上面描述的团队名称。
- 插入操作后，如果发现错误并希望提交修改后的副本，请键入

  制作手动团队=团队名版本=2

  在每次提交时，请不断增加版本号。
- 你可以通过查看来验证你的手机

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

  您在此目录中有列表和插入权限，但没有读写权限。

祝你好运！