



# 性能优化分析实验





# 常见代码优化方法

- 减少函数调用
- 提前计算
- 循环展开
- 并行运算
- 提高**cache**利用率





# 图像旋转

- 简单旋转

```
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)];
}
```

- 其中函数（或宏）**RIDX(i, j, dim)**将矩阵（图像）按行优先存放到一维数组时的下标转换函数，即，**RIDX(i, j, dim)**表示图像或矩阵（**dim\*dim**阶）的第**i**行第**j**列像素或元素的在一维数组中的下标。
- **pixel**定义：struct pixel{ int red; int green; int blue};

缺点：程序局部性不好，循环次数过多





# 第一次尝试：分块

- 尝试分成4\*4的小块，提高空间局部性

```
void rotate(int dim, pixel *src, pixel *dst)
{
    int i, j, ii, jj;
    for(ii=0; ii < dim; ii+=4)
        for(jj=0; jj < dim; jj+=4)
            for(i=ii; i < ii+4; i++)
                for(j=jj; j < jj+4; j++)
                    dst[RIDX(dim-1-j, i, dim)] = src[RIDX(i, j, dim)];
}
```

测试CPE（Cycles Per Element 每元素周期数）改进为  
1.8





## 第二次尝试：循环展开

- 采用**32\*32**分块，**4\*4**路循环展开，注意循环内部语句执行顺序

```
void rotate(int dim, pixel *src, pixel *dst)
{
    int i, j, ii, jj;
    for(ii=0; ii < dim; ii+=32)
        for(jj=0; jj < dim; jj+=32)
            for(i=ii; i < ii+32; i+=4)
                for(j=jj; j < jj+32; j+=4) {
                    dst[RIDX(dim-1-j, i, dim)] = src[RIDX(i, j, dim)];
                    dst[RIDX(dim-1-j, i+1, dim)] = src[RIDX(i+1, j, dim)];
                    dst[RIDX(dim-1-j, i+2, dim)] = src[RIDX(i+2, j, dim)];
                    dst[RIDX(dim-1-j, i+3, dim)] = src[RIDX(i+3, j, dim)];
                    dst[RIDX(dim-1-j-1, i, dim)] = src[RIDX(i, j+1, dim)];
                    ...
                }
}
```

测试CPE改进2.7





# 最后的尝试

- 考虑矩形分块**32\*1**，**32**路循环展开，并使**dest**地址连续，以减少存储器写次数

```
#define COPY(d,s) *(d) = *(s)
void rotate(int dim, pixel *src, pixel *dst)
{
    int i, j;
    for (i = 0; i < dim; i+=32)
        for (j = dim-1; j >= 0; j-=1) {
            pixel *dptr = dst+RIDX(dim-1-j, i, dim);
            pixel *sptr = src+RIDX(i, j, dim);
            COPY(dptr, sptr); sptr += dim;
            COPY(dptr+1, sptr); sptr += dim;
            ...
            COPY(dptr+31, sptr);
        }
}
```

测得CPE改进为3.5





## 推荐优化代码相关书籍

- 计算机体系结构：量化研究方法（中文第五版）
- 深入理解计算机系统（原书第三版）
- Write Great Code Volume 2: Thinking Low-level, Writing High-level --- Randall Hyde
- Software Optimization for High-Performance Computing: Creating Faster Applications  
---K. P.Wadleigh and I. L. Crawford.

