

西北农林科技大学

Northwest A&F University

信息工程学院

实验报告

实验名称: 图像增强

专业班级: 计算机 141

学号: 2014012537

姓名: 刘朝洋

指导教师: 杨龙

2016~2017 学年第一学期

使用 L^AT_EX 撰写于 2016 年 12 月 7 日

目录

0	实验目的	2
1	实验要求	2
0	实验过程	3
0	实验总结	5

0 实验目的

1. 掌握图像平滑的基本概念，以及常用的方法；
2. 了解不同滤波方法的原理，并可以编程实现。

0 实验要求

1. 使用 C++ 编程实现中值滤波，但不能直接调用 OpenCV 中的滤波函数；
2. 将处理结果与 OpenCV 中的平滑处理函数`medianBlur()`的处理结果做对比。

0 实验过程

中值滤波的原理很简单，它把以某像素为中心的小窗口内的所有像素的灰度按从小到大排序，取排序结果的中间值作为该像素的灰度值。

首先是放置窗口，这里我们通过改变开始的索引值来放置窗口。

```
1 for (int m = 1; m < M - 1; ++m)
2     for (int n = 1; n < N - 1; ++n)
```

注意，这里是从第 1 个元素开始，而不是第 0 个元素；倒数第 1 个元素结束，而不是最后一个元素。问题就是我们无法从第 0 个元素开始，因为在这种情况下，过滤窗口的左半部分是空的。为了解决这个问题，我们需要在处理之前，对图像的扩展，方法如下图2.1所示：

```
1 void medianfilter(element* image, element* result, int N, int M)
2 {
3     // Check arguments
4     if (!image || N < 1 || M < 1)
5         return;
6     // Allocate memory for signal extension
7     element* extension = new element[(N + 2) * (M + 2)];
8     // Check memory allocation
9     if (!extension)
10        return;
11    // Create image extension
12    for (int i = 0; i < M; ++i)
13    {
14        memcpy(extension + (N + 2) * (i + 1) + 1, image + N * i, N * sizeof(element));
15        extension[(N + 2) * (i + 1)] = image[N * i];
```

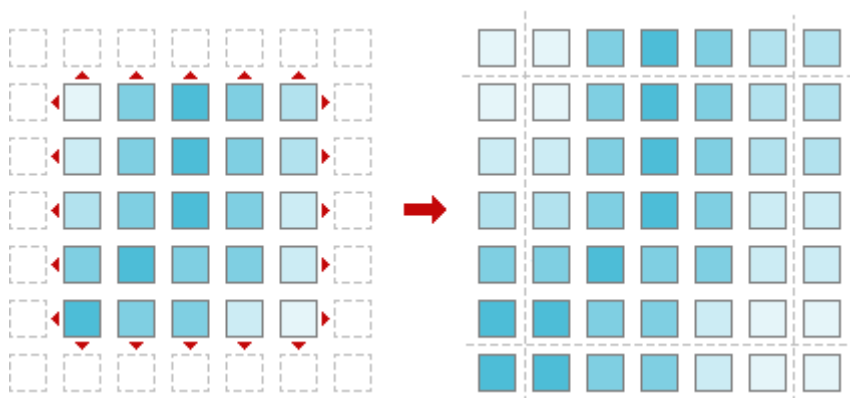


图 3.1: 图像扩展

```
16     extension[(N + 2) * (i + 2) - 1] = image[N * (i + 1) - 1];
17 }
18 // Fill first line of image extension
19 memcpy(extension, extension + N + 2, (N + 2) * sizeof(element));
20 // Fill last line of image extension
21 memcpy(extension + (N + 2) * (M + 1), extension + (N + 2) * M, (N + 2) * sizeof(element));
22 // Call median filter implementation
23 _medianfilter(extension, result ? result : image, N + 2, M + 2);
24 // Free memory
25 delete[] extension;
26 }
```

0 实验总结