# 计算机视觉实验报告（第五次作业）

## 实验目的

本实验旨在实现一个基于OpenCV的图像视差图计算程序。通过使用不同的相似性度量方法（包括归一化互相关NCC、平方差之和SSD和绝对差之和SAD），程序能够计算出左右两个图像之间的视差图。

## 实现说明

1. **MNIST数据集介绍：**

* 本次实验所用的数据集Cones由Daniel Scharstein、Alexander Vandenberg-Rodes 和 Rick Szeliski 创建。它们由具有复杂几何形状的高分辨率立体序列和像素精确的地面实况视差数据组成。地面实况差异是使用一种采用结构照明的新技术获得的，不需要校准光投影仪。

1. **立体匹配算法的核心思想：**

* 立体匹配算法的核心思想是利用一对视角略有不同的图像（左视图和右视图），通过找到两幅图像中对应像素点的视差来计算深度信息。视差是指同一场景点在左、右视图中投影位置的水平偏移量。主要步骤包括：
  + 图像块匹配：将图像划分为小块，针对每个像素块在另一幅图像的对应搜索区域内寻找最佳匹配块。
  + 相似性度量：使用某种相似性度量方法来评估两个图像块之间的相似程度。常用的度量方法有：
    - NCC（Normalized Cross-Correlation）：归一化互相关，衡量两个图像块的线性相关性，值在[-1, 1]之间，值越大，块越相似。
    - SSD（Sum of Squared Differences）：平方差之和，计算两个图像块对应像素差值的平方和，值越小，块越相似。
    - SAD（Sum of Absolute Differences）：绝对差之和，计算两个图像块对应像素差值的绝对值和，值越小，块越相似。
  + 视差计算：对于每个像素块，在搜索范围内找到相似性度量值最佳的匹配块，记录对应的视差值。

1. 实验整体流程介绍：

* 读取图像：

使用cv2.imread以灰度模式读取左右视图图像。

* 定义相似性度量函数：

分别实现NCC、SSD和SAD三种相似性度量函数。

* 选择相似性度量函数：

根据用户指定的方法名称，选择相应的相似性度量函数。

* 计算视差图：
* 初始化视差图数组。
* 确定块的大小和视差范围。
* 遍历图像中的每个像素，以当前像素为中心定义一个左图像块。
* 在右图像的相应区域内移动匹配块，计算左图像块与右图像块之间的相似性。
* 记录相似性最高的视差值。
* 将视差值存储在视差图数组中。
* 显示和保存视差图：

将生成的视差图保存到指定目录。

使用OpenCV显示结果，并根据需要调整显示比例。

## 结果截图

原始输入的左图和右图：



1. 使用归一化互相关NCC作为相似性度量方法时，获得的视差图如下：

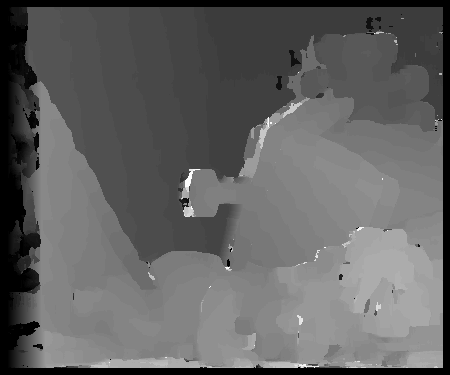


图 1 NCC+BlockSize=15

1. 使用平方差之和SSD作为相似性度量方法时，获得的视差图如下：



图 2 SSD+BlockSize=3



图 3 SSD+BlockSize=5

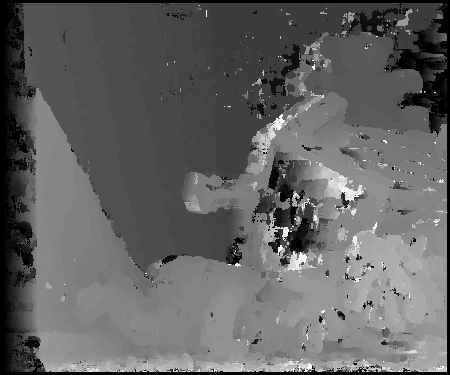


图 4 SSD+BlockSize=7

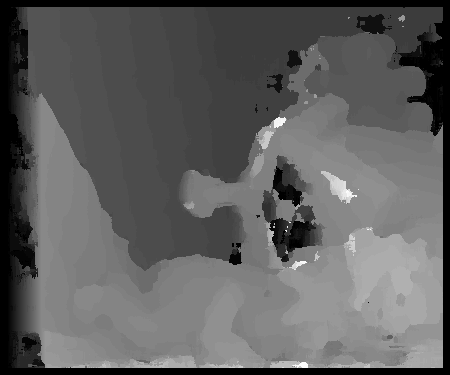


图 5 SSD+BlockSize=15

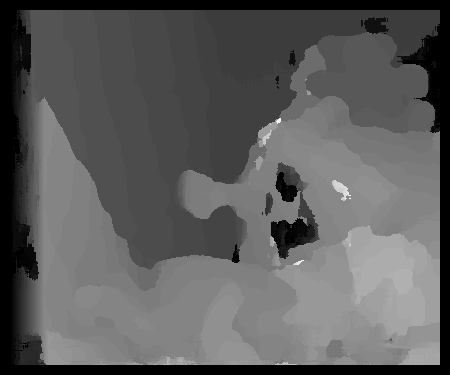


图 6 SSD+BlockSize=21

* 在实验初期使用非常小的窗口尺寸（例如3），可以清晰地看到物体的轮廓，但深度图受到大量噪声的严重影响。当将窗口尺寸略微增加到5或7时，噪声明显减少。随着窗口尺寸进一步扩展到15或21，噪声几乎降至不可察觉的水平，使图像不仅更平滑，而且具有明显的深度感，可以清楚地区分近处和远处的物体。

1. 使用绝对差之和SAD作为相似性度量方法时，获得的视差图如下：

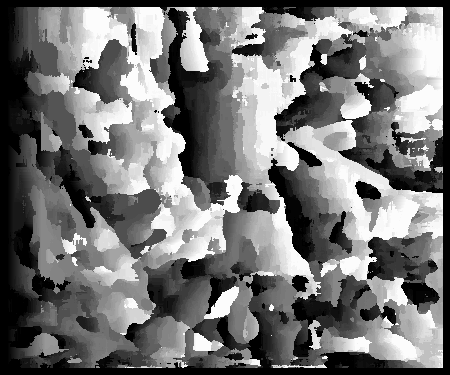


图 7 SAD+BlockSize=15

## 运行说明

要运行此程序，请确保您的环境中已安装Python和OpenCV库。将程序保存为.py文件，并在包含图像文件的目录中运行。程序需要两个图像文件，分别为imgdata/im2.png和imgdata/im6.png。执行程序后，将生成一个名为disparity\_result的目录，并将视差图保存为disparity\_map.png。可通过修改method变量的值（'ncc', 'ssd', 'sad'）来切换不同的相似性度量方法，

## 实验小结

通过实现和运行此视差图计算程序，我们能够观察到不同相似性度量方法对视差计算的影响。实验结果表明，选择合适的相似性度量方法对提高视差图的准确性至关重要。此外，程序的性能通过时间测量得到验证，展示了算法的效率。最终，实验加深了对图像处理和计算机视觉基本概念的理解，特别是在立体视觉和视差估计方面。实验表明，SAD（绝对差之和）算法的性能较差。在使用3x3等小窗口尺寸时，SAD、SSD（平方差之和）和NCC（归一化互相关）算法都会生成大量的噪声点。相较之下，SSD生成的噪声点稍微少于SAD，而NCC生成的噪声点最多，显著影响了远处物体的清晰度。基于多次试验，建议使用SSD或NCC算法，配以7x7到15x15的中等块大小，以获得更精确的结果，或使用SAD算法和较小的3x3窗口尺寸，以在较低计算需求下获得可接受的结果。