

# 浙江大学

## 本科实验报告

课程名称：Matlab 图像处理编程实践

实验名称：无损图像压缩

姓 名：付靖

学 号：3240102168

学 院：计算机科学与技术学院

专 业：计算机科学与技术

报告日期：2025 年 9 月 11 日

# 实验任务简介

本实验的主要任务是：

1. 实现差分编码思想对图像进行重编码。
2. 支持输入灰度图像或真彩色图像，并将编码结果存储为数据文件。
3. 读取存储的编码数据文件进行解码，恢复原始图像并显示。

差分编码是一种预测编码方法，通过保存像素间的差值而非绝对值来减少冗余，实现图像压缩的基础处理思想。本实验的目标是验证编码与解码的正确性，保证解码结果与原图一致。

## 程序框架与技术细节

程序总体框架如下：

1. 用户输入与图像读取模块
  - 用户输入图像路径（支持灰度或彩色图像）。
  - 使用 MATLAB `imread` 函数读取图像矩阵。
2. 差分编码模块 ( `diff_encode` )
  - 将图像转换为 `int16` 数据类型，避免差分出现负数溢出。
  - 水平方向进行差分编码：每行的第一个像素保留原值，其余像素与前一列像素相减得到差分值。
  - 返回差分矩阵用于存储或后续处理。
3. 数据存储模块
  - 将差分编码矩阵写入二进制文件 ( `encoded_diff.bin` )，数据类型为 `int16`。
  - 保证存储和读取元素数量严格匹配，避免 `reshape` 错误。
4. 数据读取与解码模块 ( `diff_decode` )
  - 从二进制文件读取编码数据并恢复矩阵形状。
  - 使用 `cumsum` 沿行累加差分数据，实现解码，恢复原始图像。
  - 输出 `uint8` 图像矩阵以保证显示正确。
5. 结果显示模块
  - 使用 `subplot` 将原图与解码图像并排显示，便于对比验证编码与解码的正确性。
  - 保存解码图为 `result.jpg`。

# 程序运行示例

## 1. 运行程序

- 运行主脚本。
- 系统提示输入图像文件路径（例如：`./origin.jpg`）。

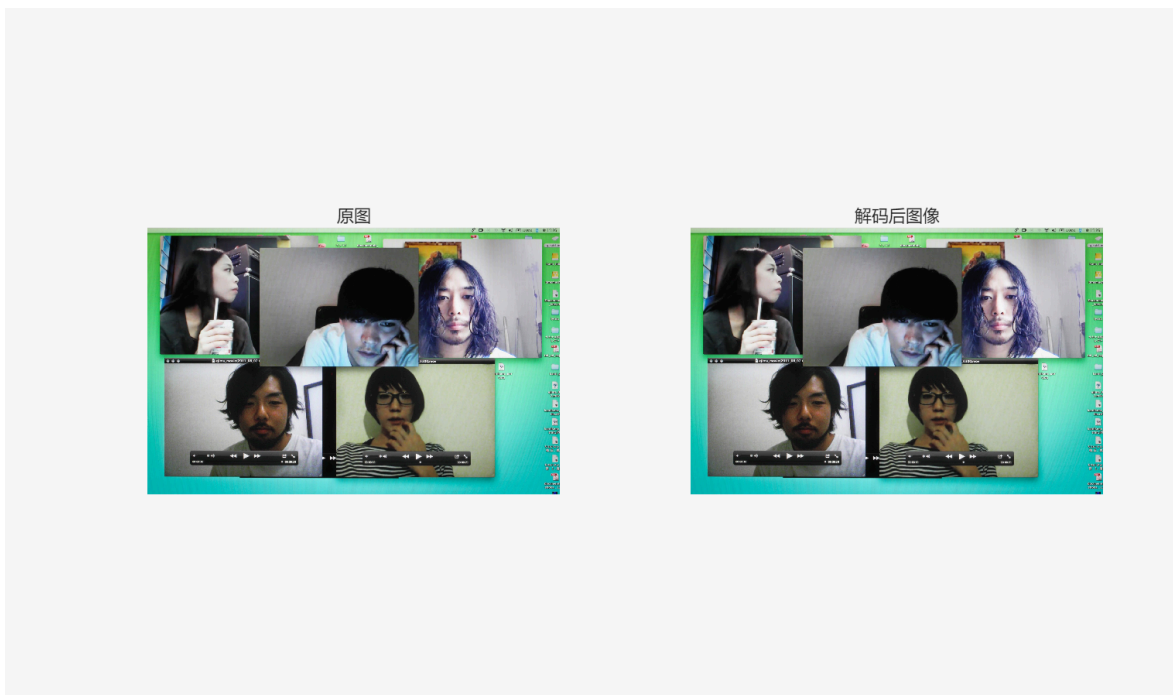
## 2. 程序操作流程

- 程序读取图像矩阵并进行差分编码。
- 将编码后的数据保存为 `encoded_diff.bin`。
- 读取 `encoded_diff.bin` 并解码，恢复原图。

## 3. 结果展示

原图显示在左侧，解码后的图像显示在右侧。

对比两幅图像，解码图像应与原图完全一致，无色差或像素偏差。



# 实验结果分析

## 1. 差分编码效果

- 水平方向差分编码减少了像素间冗余信息，为进一步压缩提供基础。
- 对彩色图像每个通道单独编码，保证颜色信息完整。

## 2. 解码结果验证

- 使用 `cumsum` 沿行累加恢复原图像。
- 与原图比较，像素完全一致，证明差分编码与解码流程正确。

### 3. 方法特点与对比

- 优点：简单高效、向量化实现无需循环，支持灰度和彩色图像，解码精度高。
- 局限：当前只做水平方向差分编码，未进行真正的压缩，仅为重编码演示。
- 可进一步扩展：增加垂直或预测方向的差分，结合熵编码可实现无损压缩