1 K-means 算法

1.1 算法简述

K-means 算法是一种迭代的、数值的、不确定的、无监督的聚类方法，它根据输入数据点之间的固有距离将它们分为多个类。K-means 算法是基于数据成分对之间的相似性或相异性指数。K-means 算法假设数据特征形成一个向量空间，并试图在其中找到自然聚类。在这种方法中，像素点围绕聚类中心聚集，聚类中心则通过最小化空间获得的。
K-means 聚类是一种将图像的 n 个像素聚类成 K 个簇的技术，（K<n，K 为正整数）。聚类中心在算法中是随机初始化的，聚类基于像素灰度强度和像素强度距离等相似性特征形成。在这种聚类算法中，数据通过计算每个组的强度迭代地进行聚类，并通过用最接近的一个像素对类中的每个像素进行分类来分割图像。使用 K-means 算法分割图像要遵循的各个步骤如下：
将图像作为输入并计算强度分布。

1) 用 k 个随机强度初始化质心；
2) 重复步骤，直到目标函数不再改变：
3) 基于其强度与质心强度的距离对点进行聚类，
4) 计算每个簇的新质心；

1.1.1 更新公式

聚类中心更新公式 $u_i = \frac{\prod_{j=1}^{N} P(u_i|x_j)^b x_j}{\prod_{j=1}^{N} P(u_i|x_j)^b}$

1.1.2 算法规格

1. 一个参数 k，聚类中心的数目，当然也有一些常规的参数，比如最大迭代次数 epochs, 容忍度 tol
2. 一个循环，判断目标函数是否变化足够小，以 F 范数（Frobenius norm）为度归

```
    while true,
    ...
    if norm(J_cur-J_prev, 'fro') < tol,
        break;
    end
    J_prev = J_cur;
end
```

3. 一条更新语句，更新各个类的聚类中心，根据每个样本应属的类别（欧式距离最小表征）

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}$$

```
    dist = sum(X.^2, 2)*ones(1, k) + (sum(C.^2, 2)*ones(1, m*n))'...
    - 2*X*C';
[~, idx] = min(dist, [], 2) ;
for i = 1:k,
   C(i, :) = mean(X(idx == i , :)); % 即求解各集合中新的聚类中心
end
```

1.1.3 算法流程

a 初始化聚类个数 k

b 循环，判断目标函数是否变化足够小，以 F 范式度量

c 推出循环，返回 dist(到聚类中心距离矩阵)，最新聚类中心，以及目标函数

1.2 与 Fuzzy C-Means 算法比较

硬聚类算法 K-means 相较于软聚类算法 Fuzzy C-Means 算法，更易快速迭代，达到全局最优；Fuzzy C-Means 虽然强调了各个样本和聚类中心的关系，但模糊隶属后更易受噪声干扰，因此更易陷入局部最优（因此对于 Fuzzy C-Means 的最基本优化方法是增加迭代次数）；相较于 Fuzzy C-Means，K-means 逼近全局最优的资源消耗大幅下降，在本实验中表现优于 Fuzzy C-Means 算法；

1.2.1 实验结果对比

本实验中对于 Kmeans 算法中 k 值以及 Fuzzy C-Means 算法中 k 和 b 值进行了大量实验。
对于 Kmeans，在 k=19 时可以达到最优分割效果，而实验三中，Fuzzy C-Means 在 k=19, b=4（k=19 条件下，b=4 时分割效果最好）时表现与 K-means 存在明显差距
对比结果如下

A. K-means 绘制图像
对最佳分割结果绘图

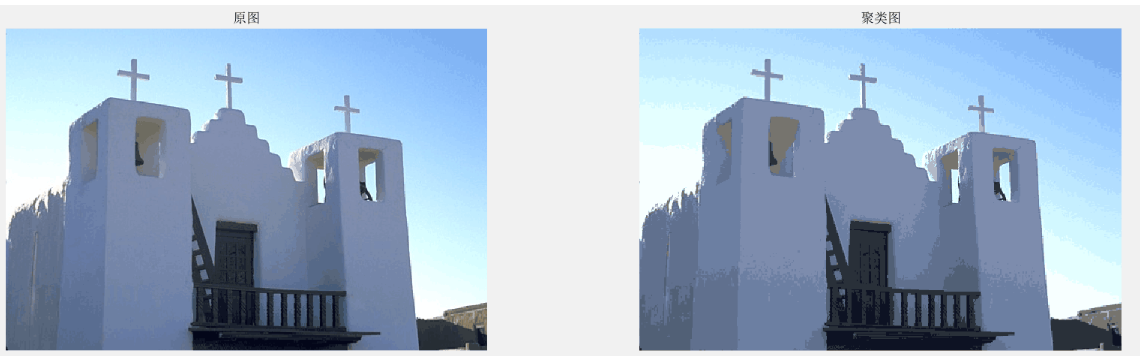

图 1: 原图像与分割后图像对比图

绘制函数迭代图

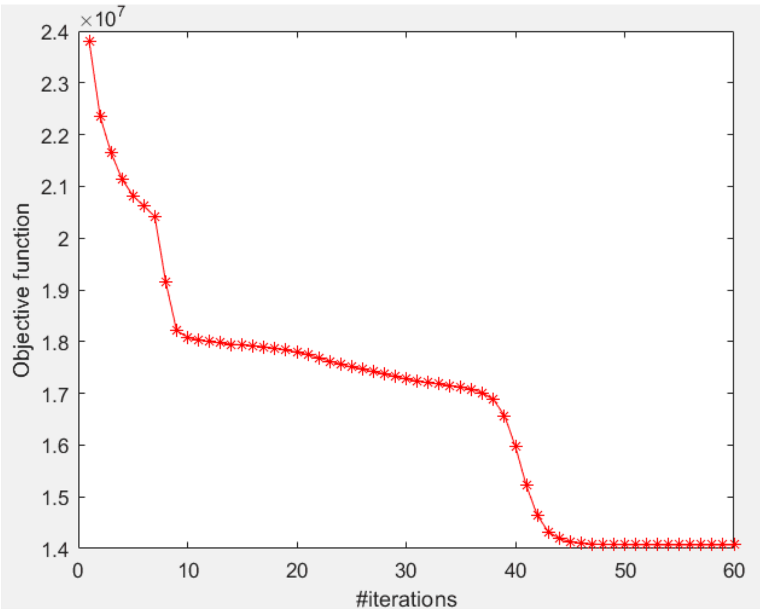

图 2: 目标函数迭代结果

B. 对比 Fuzzy C-Means

对 K=19,b=2 时分割效果绘图

观察分割后图像右上方渐变色天空可以看出，Fuzzy C-Means 算法对于本图中渐变色的分割层次明显少于 k 值相同的 K-means 算法

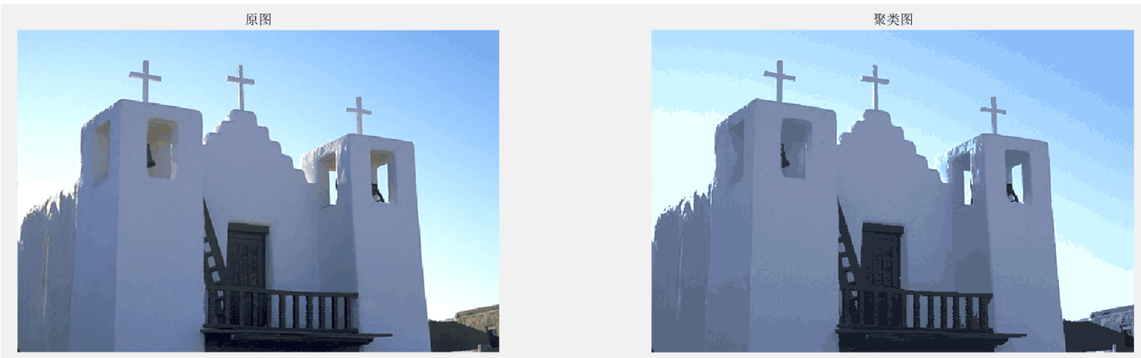

图 3: 原图像与分割后图像对比图

绘制函数迭代图

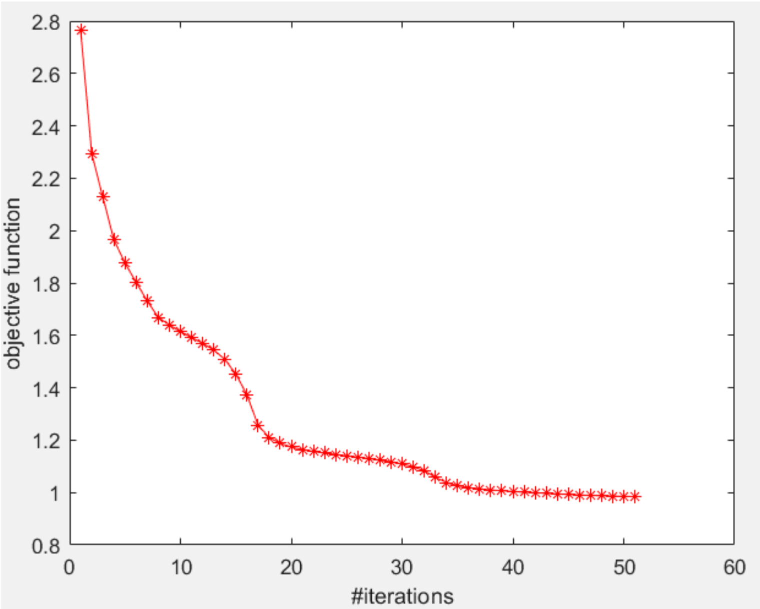

图 4: 从两种算法中目标函数的迭代次数也可看出，K-means 算法所需计算时间较少

除以上对比外，Fuzzy C-Means 算法受隶属度 b 值影响较大，在 k 取某些值时，经实验检测，b 取值过大将造成分割效果较差或者目标函数不收敛的情况。如，在 k=15, b=6 时，分割效果表现较差

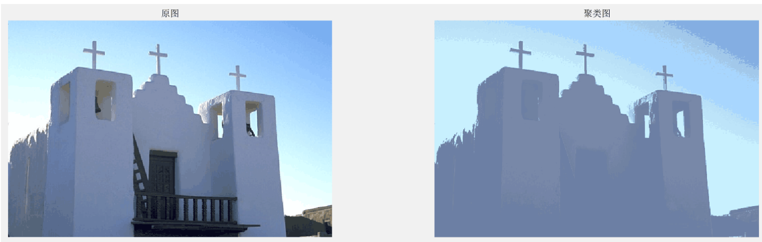

图 5: 目标函数迭代结果

而在 k=15, b=4 时，分割效果显著提高

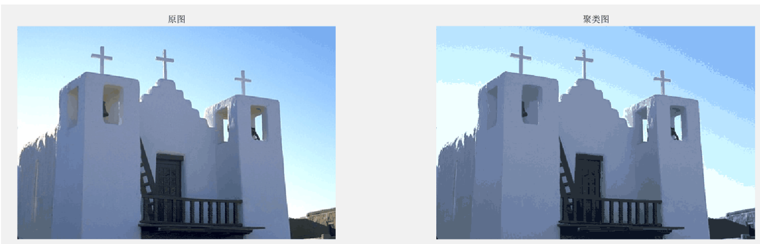

图 6: 目标函数迭代结果

注：
由于时间有限，本实验仅列出了最佳效果下的 K-means 分割图像以及目标函数，K-means 和 Fuzzy C-Means 算法更多分割效果图详见文件夹 imgKmeans 和 imgFCM

1.3 主要功能实现相关代码展示

```
function [C, label, J] = kmeans(I, k)
[m, n, p] = size(I);
%将二维压缩成一维，转换精度
X = reshape(double(I), m*n, p);
rng('default');
%初始化聚类中心
C = X(randperm(m*n, k), :);
%初始化容忍度tol
J_prev = inf; iter = 0; J = []; tol = 1e-2;
%%
%更新语句
while true
    iter = iter + 1;
    dist = sum(X.^2, 2)*ones(1, k) + (sum(C.^2, 2)*ones(1, m*n))' - 2*X*C';
    [~, label] = min(dist, [], 2) ;
    for i = 1:k
       C(i, :) = mean(X(label == i , :));
    end
    %计算目标函数
    J_cur = sum(sum((X - C(label, :)).^2, 2));
    J = [J, J_cur];
    fprintf('#iteration: %03d, objective function: %f\n', iter, J_cur);
    %判断目标函数变化是否足够小
    %F范数为度归
    if norm(J_cur-J_prev, 'fro') < tol
        break;
    end
    J_prev = J_cur;
end
```

2 补充说明

本实验所使用图片来源于 UC-Berkely，Computer Vision Lab (https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html)