

马赛克瓷砖选色问题

摘要

本文主要研究马赛克瓷砖选色问题。为尽量选择与用户要求较为接近的颜色，增加用户体验感，需制订合理的选色方法及策略。通过对选色原理的分析，并制订相应的目标体系及约束条件，建立了以欧式定理为基础、遗传算法为主要方法的相似度模型，使用 matlab 与 excel 等工具求解，得到选取最相似瓷砖的优化方案。

针对问题一：以不会出现其他不必要的外界干扰因素为前提。作出假设：1. 在找出与每种颜色最接近的瓷砖颜色过程中，相同颜色的砖规格都相同，不会出现颜色偏差 2. 两种颜色的相似度能通过比较其 RGB 的差别判断，RGB 相差越小，相似度越大，即为颜色最相近的砖。同时，将求两种颜色的相似度的问题转化为比较其 RGB 的差别。而为了量化 RGB 的具体差别，把题目所给的 RGB 参数转化为坐标 (x,y,z) ，并利用欧式定理求出两种颜色对应的 RGB 坐标距离。最后，引入相对误差的概念。将其定义为欧氏距离除以其可能的最大值。并将相对误差从小到大一一比较，相对误差越小，相似度就越高，这样的瓷砖，就是最终想要的瓷砖。

针对问题二：问题一中，已经建立了相似度模型，以定义的相对误差来评价所要找的近似程度。问题二中，先对图像的表现力度进行了进一步理解，认为表现力度最重要的在于尽量 100% 的呈现。将第一问中得到得所有颜色对应的相似程度关系累加，得到一个简单的与原来图像的相似程度的比较。通过颜色相似到图像相似。将这种图像相似程度的计算定义为一个函数。要求得到函数图像的最大值。此时函数的因变量即为所增加的瓷砖的 RGB 的值设定为 X, Y, Z。以加一种瓷砖为例，所有的瓷砖颜色为原本 22 种，还有新的瓷砖 (X, Y, Z) 共有 23 个颜色。此时可以计算图像与这 23 个瓷砖的相似程度函数。可以通过启发式算法（模拟退火，遗传等）来计算最好的 X, Y, Z 取值。

针对问题三：问题二中，可以得出增加了 1 到 10 中瓷砖时最好的 RGB 效果。问题三在此基础上要综合考虑成本和表现成果。问题转化为相似程度与成本的多目标规划。设定一个 n 为增加瓷砖数，根据 n 得到的 n 的 xyz 的值，最后通过遗传算法求解。

关键字： RGB 颜色空间 颜色相似度 遗传算法

目录

一、问题重述	3
1.1 问题的背景	3
1.2 问题的提出	3
二、问题分析	3
2.1 问题一分析	3
2.2 问题二分析	4
2.3 问题三分析	4
三、模型的假设	4
四、符号说明	5
五、问题一的模型建立与求解	5
5.1 模型假设	5
5.2 模型建立	6
5.3 模型优化	7
六、问题二的模型建立与求解	7
6.1 模型假设	7
6.2 模型求解	9
七、问题三的模型建立与求解	11
7.1 模型假设	11
7.2 模型求解	11
八、模型评价	12
8.1 模型优点	12
8.2 模型缺点	13
参考文献	13
附录 A 问题一结果	14
附录 B Matlab 程序	26

一、问题重述

1.1 问题的背景

马赛克瓷砖是一种尺寸较小（常见规格为边长不超过 5cm）的正方形瓷砖，便于在非平整的表面铺设，并且容易拼接组合出各种文字或图案。但是受工艺和成本的限制，瓷砖的颜色只能是有限的几种。用户在拼接图案时，首先要根据原图中的颜色，选出颜色相近的瓷砖，才能进行拼接。某马赛克瓷砖生产厂只能生产 22 种颜色（见附件 1）的马赛克瓷砖。该厂要开发一个软件，能够根据原始图片的颜色，自动找出颜色最接近的瓷砖，以减少客户人工选色的工作量。该厂希望你们团队提供确定原始颜色与瓷砖颜色对应关系的算法。假设原始图像为 24 位真彩色格式，即 R、G、B 三个颜色分量均为 8 位，共有 $28 \times 28 \times 28 = 16777216$ 种颜色，对于任何一种指定的颜色，算法输出颜色最相近的瓷砖的颜色编号。

1.2 问题的提出

1) 附件 2 是图像 1 中的 216 种颜色，附件 3 是图像 2 中的 200 种颜色，请找出与每种颜色最接近的瓷砖颜色，将选出的瓷砖颜色的编号按照附件 4 的要求输出至结果文件。

2) 如果该厂技术革新，计划研发新颜色的瓷砖。那么，不考虑研发难度，只考虑到拼接图像的表现力，应该优先增加哪些颜色的瓷砖？当同时增加 1 种颜色、同时增加 2 种颜色、……、同时增加 10 种颜色时，分别给出对应颜色的 RGB 编码值。

3) 如果研发一种新颜色瓷砖的成本是相同的，与颜色本身无关，那么，综合考虑成本和表现效果，你们建议新增哪几种颜色，说明理由并给出对应的 RGB 编码值。

二、问题分析

2.1 问题一分析

根据题目要求，某马赛克瓷砖生产厂只能生产 22 种颜色的马赛克瓷砖。我们团队提供确定原始颜色与瓷砖颜色对应关系的算法，能够根据原始图片的颜色，自动找出颜色最接近的瓷砖。我们需要查阅相关资料，根据查阅到的数据对问题进行求解。

问题一中，要找出与每种颜色最接近的瓷砖颜色，通过分析题目，首先，我们对解题过程中可能会用到的相关数据进行了准备，包括根据题目要求确定如何判断瓷砖相似度，如何使结果更准确，相似度更高，我们其次进行数据处理将数据导入表格，将 RGB 颜色转化为坐标用，根据其空间距离的大小来分别找出与各个颜色最相似的瓷砖。最后整理结果，选出的瓷砖颜色的编号按照附件 4 的要求输出至结果文件。

2.2 问题二分析

根据题目要求，我们需要根据拼接图像的表现力，考虑优先增加哪些颜色的瓷砖，首先我们明确一点，什么是拼接图像的表现力，表现力有很多方面，颜色呈现还原度方面，还有颜色搭配方面，而此处简化只考虑单纯的颜色呈现还原度方面，而我们知道相似度越高，还原度也就越高，在第一题中，我们构建了求颜色间的相似度模型，第二题就可以相当于将颜色这一小维度转化到了一整张图片这一大维度上，故可看成大维度由许多个小维度组成，这一题相当于求一整张拼接图像与现实图片间的总的相似度，也就是许多个颜色间相似度的叠加。

总结一下，就是此处简化为单纯通过增强色彩的还原度来提高画面的逼真性，从而提高图像表现力，而我们知道颜色间的相似度越高，还原度也就越高，而在第一题中，我们构建了求颜色间的相似度模型，第二题便可以借鉴第一题的做法。第一题的研究对象是两个颜色间，相当于是单一维度，而第二天的研究对象是两个图片。但从颜色这一方面考虑，第二题的研究对象——图像相当于由许多个颜色组成，相当于第二题的研究对象是一个集合，集合里的许多个单一元素第一题的研究对象——颜色，就是就可以相当于将颜色这一小维度转化到了一整张图片这一大维度上，大维度由许多个小维度组成，是许多个小维度的叠加之后产生的效果，故求图像的表现力，也就是求一整张拼接图像与现实图片间的总的相似度，等同于是求许多个颜色间相似度的叠加。探究我们应该优先增加哪些颜色的瓷砖，便取决于增加不同的瓷砖时整张拼接图像的总相似度的大小，能使总相似度最大的瓷砖便是我们要添加的。

此题还要求添加多块瓷砖的情况，而上述我们只能求出添加一块瓷砖的情况，但是我们可以求出添加一块瓷砖的情况结果后将此作为已知条件，也就是现有 23 块瓷砖，再应该添加哪一块瓷砖，以此类推....

2.3 问题三分析

由问题三可知，新颜色瓷砖的成本是相同的，故总的成本随增加的砖块数变化而变化，此时不仅要考虑图像表现力，又要考虑增加的砖块数，我们能够想象到，砖的块数越多，表现力越大，但是成本在提高，故我们在两者都要考虑的情况下需要权衡，找到在表现力适度的情况下，成本也在正常范围内的砖块数 n ，即两者考虑下的总效果最好的情况结果。

三、模型的假设

1. 假设在找出与每种颜色最接近的瓷砖颜色过程中，相同颜色的砖规格都相同，不会出现颜色偏差，即没有其他不必要的外界干扰因素
2. 假设建立的相似度模型能准确反映颜色的相似性；

3. 假设在找出与每种颜色最接近的瓷砖颜色过程中，两种颜色的相似度能通过比较其 RGB 的差别判断，RGB 相差越小，相似度越大，即为颜色最相近的砖
4. 假设图像表现力只体现在图像还原度上，其他方面可忽略不计
5. 假设图像总的相似度可看成多种颜色相加形成的画的相似度，即由多种颜色的相似度相加得到图像的总相似度
6. 不考虑瓷砖与瓷砖连接处的缝隙的颜色差别，缝隙颜色忽略不计，假设不影响图像相似度及表现力
7. 假设实际设计能满足理论方案

四、符号说明

符号	符号说明
σ	综合相似度
δ	相似误差
τ	总相似度
ρ	平均相似度
k	比例系数/成本损耗度
w	总成本损耗
n	砖块数

五、问题一的模型建立与求解

5.1 模型假设

模型假设：为了简化计算，给出在找出与每种颜色最接近的瓷砖颜色过程中的合理假设：

假设一：在找出与每种颜色最接近的瓷砖颜色过程中，相同颜色的砖规格都相同，不会出现颜色偏差；

假设二：在找出与每种颜色最接近的瓷砖颜色过程中，不会出现误差与错误；

假设三：在找出与每种颜色最接近的瓷砖颜色过程中，两种颜色的相似度能通过比较其 HSV 的差别判断，HSV 相差越小，相似度越大，即为颜色最相近的砖。

5.2 模型建立

中心问题是如何将每种颜色与 22 种不同颜色的瓷砖间的颜色相似度定量的表示出来，从而找到颜色最接近的瓷砖颜色。

首先，对数据处理，将附件导入表格，同时将颜色的 RGB 用向量表示，其中 R, G, B 分别代表 XYZ 轴上的坐标

$$(x, y, z) = (R, G, B)$$

判断不同的颜色与 22 种瓷砖颜色间的相似度，我们采用两种方法组合讨论：

1：分别计算一种颜色与 22 种瓷砖的两个向量的欧几里得距离 (Euclidean)

由欧式距离公式 $Euclidean = \sqrt{\sum_{i=1}^k (x_i - x_j)^2}$ 可得此处

$$Euclidean = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (1)$$

计算结果，因为欧式距离最大是 $255\sqrt{3}$ ，故设相似误差 (δ)

$$\delta = \frac{Euclidean}{255\sqrt{3}} \quad (2)$$

将相似误差排序从小到大排序，相似误差越小，相似度就越高，这样的瓷砖，就是我们要找的颜色最相近的瓷砖。2：计算两向量的余弦相似度

$$sim(A, B) = \cos(\theta) = \frac{A \cdot B}{||A|| * ||B||} \quad (3)$$

$$A = (x_1, y_1, z_1) \quad B = (x_2, y_2, z_2) \quad (4)$$

3：最后综合两次计算结果，为了两次结果能统一起来，因为欧式距离最大是 $255\sqrt{3}$ ，而余弦最大为 1，故设综合相似度为

$$\sigma = 1 - \frac{Euclidean}{255\sqrt{3}} + sim(A, B) \quad (5)$$

最后将综合相似度 σ 排序从大到小排序，相似度越大，相似度最高的瓷砖，就是我们要找的颜色最相近的瓷砖。

我们可以取一下极限情况，当该颜色与瓷砖颜色无限的接近时，即两者一模一样时，综合相似度 σ 将趋近于一个常数，我们可以计算出，这个常数就是 2，即当两颜色越接近时，综合相似度 σ 越趋近于 2。

此处也可做推广，我们可以取多组计算结果，并分析实际的两颜色对比，可以得出一个颜色的综合相似度 σ 表，即综合相似度在什么范围时两颜色属于非常接近，什么范围又是比较接近，我们就可以通过这个颜色的综合相似度表来根据我们的需求，来选择合适的综合相似度范围内的瓷砖，这样如此一来，我们在生产瓷砖时，也可以根据实际来定量的生产，更有目的性，减少了一定的盲目性。

最终所求结果已在附录中显示，并且已导入附件表格中 (result1.xlsx)(result2.xlsx)

5.3 模型优化

1、在计算结果的基础上我们可以进一步优化，去考虑的是 RGB 差值的方差（方差为 0 说明为同种颜色，再次前提下欧式距离越大明暗差别越大），基于已经求出的差值 $X = \delta_i - \delta_j$ ，为进一步确定相似度，可在此基础上计算方差，通过方差的计算公式

$$s^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}$$

针对得到的差值 (0.1911, 0.2302, 0.227...) 导入数据进一步与对应相似颜色比较求出离散程度，当所求方差的值更趋近于 0，说明两种颜色相似程度更大。

2、由于用 RGB 的差值来判断两颜色的相似度误差较大，且不直观不好比较，但是我们发现颜色的 HSV 能够较为准确的表示两颜色的相似度，故我们优化了方案，如下：通过将 RGB 转化为 HSV，

$$\begin{aligned} V &= \frac{1}{\sqrt{3}}(R + G + B) \\ S &= 1 - \frac{\sqrt{3}}{V} \min(R, G, B) \\ H &= \begin{cases} \theta & G > B \\ 2\pi - \theta & G \leq B \end{cases} \end{aligned}$$

其中

$$\begin{aligned} \theta &= \cos^{-1} \frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \\ (x, y, z) &= (R, G, B) \end{aligned}$$

将颜色的 HSV 用向量表示 H, S, V 分别代表 XYZ 轴上的坐标，不同的颜色与 22 种瓷砖颜色间的相似度，我们再采用上述两种方法 (分别计算一种颜色与 22 种瓷砖的两个向量的欧式距离)(计算两向量的余弦相似度) 组合讨论。

六、问题二的模型建立与求解

6.1 模型假设

首先明确一点，什么是拼接图像的表现力，表现力有很多方面，颜色呈现还原度方面，还有颜色搭配方面，由于此题只考虑瓷砖的颜色，故我们只需考虑从色彩方面下手，至于色彩方面，由于通过色彩使画面更有冲击力比较抽象且不好定量分析，主观性较强，且需要改变不同的颜色，所以我们在此处也不考虑通过增强色彩来造成冲击力这一方法来提高图像表现力，故还剩下一点，就是通过增强色彩的还原度来提高画面的逼

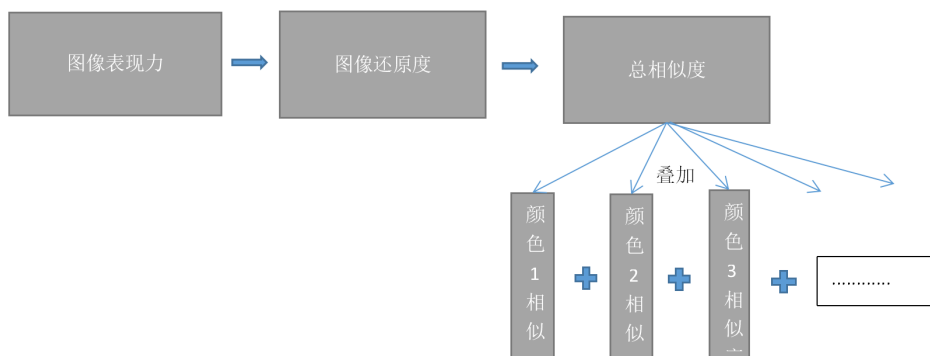


图 1 表现力分析

真性，从而提高图像表现力。此处简化只考虑单纯的颜色呈现还原度方面，而我们知道相似度越高，还原度也就越高。

题目要求出到底是添加哪一块瓷砖，我们便要列出一个有关添加的瓷砖的 RGB 向量 (x, y, z) 的函数，如何将图片的总相似度表示出来，这里便需要构建一个图片的总相似度函数，自变量就是添加的瓷砖的 RGB 的值，用向量表示，

$$(x, y, z) = (R, G, B).$$

根据上述的式 (1)(2)(3)(4)(5), 从颜色这一方面考虑，第二题的研究对象——图像相当于由许多个颜色组成，相当于第二题的研究对象是一个集合，集合里的许多个单一元素第一题的研究对象——颜色，就是就可以相当于将颜色这一小维度转化到了一整张图片这一大维度上，大维度由许多个小维度组成，是许多个小维度的叠加之后产生的效果，故求图像的表现力，也就是求一整张拼接图像与现实图片间的总的相似度，等同于求许多个颜色间相似度的叠加。可求出此处的总相似度

$$\tau = \sigma_1 + \sigma_2 + \dots = \sum_{i=1}^k \sigma_i \quad (6)$$

先考虑加一个瓷砖的情况，也就是现在有 23 块瓷砖，通过游历其他可能的瓷砖，带入到附件二三我们便能找到，使综合相似度最大的那一个瓷砖参数，便可得加一个瓷砖时的结果。我们接下来便可将求出的这块瓷砖当作已知的，求第二块瓷砖，也就是在原来 23 块瓷砖的基础上再添加一块瓷砖使总相似度最大，而后面的三块，四块..... 便可以此类推。

根据问题一的推广，此处我们也能再讨论一下，此处引进另一个概念——图像颜色平均相似度，我们能把图像看成一个包含无数种小颜色的大颜色，而每一种图像内部的瓷砖颜色与原来的对照颜色的综合相似度 σ 的平均值就可看作这个大颜色的综合相似度 σ ，但此处为了区别这是一个图像，我们用图像颜色平均相似度表示

我们便可构建函数模型

假设图像颜色平均相似度为 ρ

$$\rho = \frac{\sum_{i=1}^k \sigma_i}{N} \quad (7)$$

此处 N 加上增加砖块后总的砖块数

们亦可以想象一个极限情况，当 N 无限趋近于所有可能颜色砖的总数时，平均相似度为将趋近于两颜色无限接近时综合相似度的值，也就是 2，所以此处我们也可以在问题一的基础上再做一个推广，就是我们可以取许多的样本材料图像，算出图像颜色平均相似度为 ρ ，然后与实际图片比对，判断出图片还原度程度与图像颜色平均相似度为 ρ 的大小关系，然后列出对应表格。

这也可当作是一个图像还原度的评价机制，图像颜色平均相似度为 ρ 在什么范围时，可以被称为是还原度非常高，什么时候有时还原度较高，使评价比较更加方便，减少因肉眼的感官错觉而带来的误判与错误，增加了公平性。

而也可以得到还原度百分比的公式

$$\omega = \frac{\rho}{2} \times 100\% \quad (8)$$

因为前面我们通过取极限可知当图像与原图完全一样时，也就是还原度为百分之百时，图像颜色平均相似度 ρ 为为 2，故此处将实际的图像颜色平均相似度除以 2 即可得到还原百分比。

6.2 模型求解

将第一问中得到得所有颜色对应的相似程度关系累加，得到一个简单的与原来图像的相似程度的比较。

$$\tau = \sum_{i=1}^k \sigma_i$$

先以加一种瓷砖为例，所有的瓷砖颜色为原本 22 种，还有新的瓷砖 (X, Y, Z) 共有 23 个颜色。此时可以计算图像与这 23 个瓷砖的相似程度函数。通过遗传算法来计算最好的 X, Y, Z 取值。

首先产生随机群体 (50)，然后进行迭代 (运用交叉和变异)，生成进一步多的后代群体，计算后代群体的适应度，比对分析出最大值，在一次次进化中使算法一步步趋近最优解。

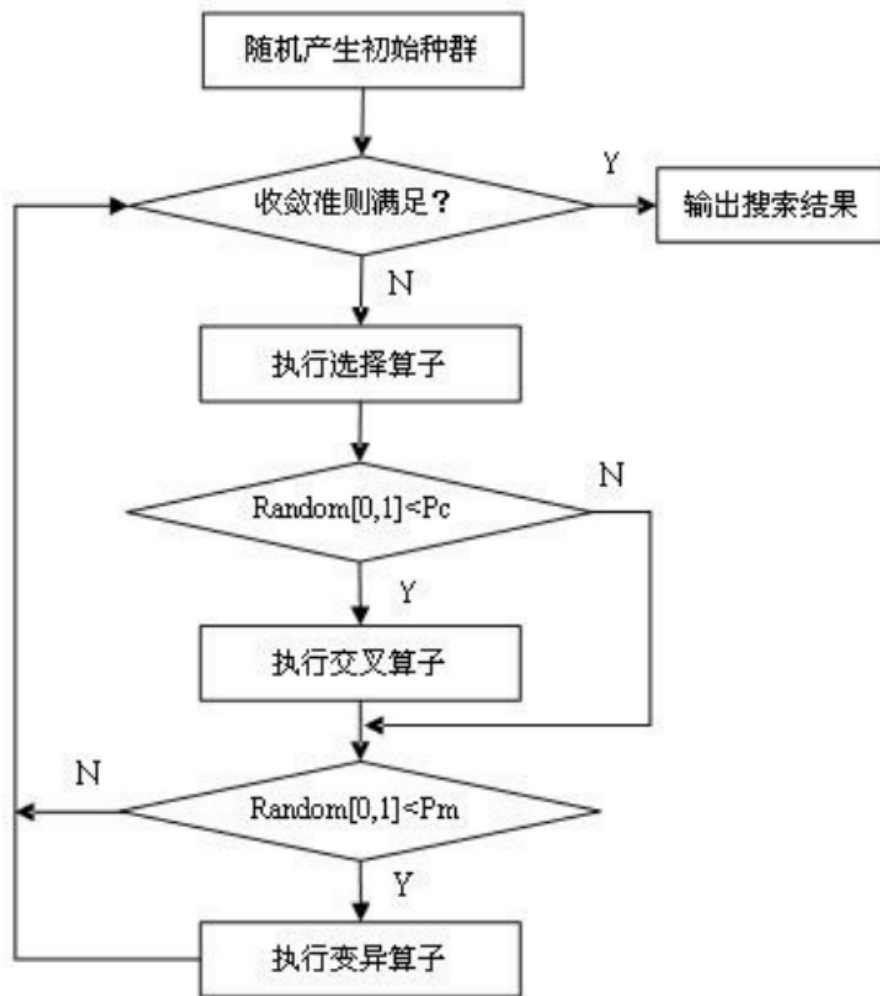


图 2 算法分析

根据程序运行我们大概在迭代了 30 此左右的情况下，算法趋近了最优解
可以得到我们添加的颜色其 RGB 值为

[215, 147, 126]

后续可以继续添加瓷砖，逐步添加样本个数，运行遗传算法，一步步求出最优解。
最终结果可得到：

增加颜色数量	对应RGB值	
1	[215, 147, 126]	
2	[23, 227, 13], [145, 162, 126]	
3	[70, 248, 223], [12, 227, 126], [57, 245, 157]	
4	[233, 186, 165], [127, 77, 16], [251, 210, 114], [192, 6, 158]	
5	[58, 115, 24], [169, 5, 150], [196, 49, 181], [110, 140, 115], [61, 119, 92]	
6	[215, 252, 38], [153, 199, 3], [197, 56, 106], [78, 202, 172], [2, 214, 76], [39, 91, 132]	
7	[44, 225, 89], [122, 70, 209], [79, 187, 170], [172, 107, 162], [250, 250, 254], [223, 188, 52], [206, 189, 94]	
8	[216, 233, 252], [158, 161, 141], [14, 81, 177], [12, 127, 219], [43, 45, 127], [179, 55, 60], [30, 125, 57], [57, 224, 10]	
9	[186, 227, 86], [243, 47, 52], [102, 252, 39], [227, 55, 112], [49, 95, 104], [92, 238, 209], [36, 198, 210], [0, 173, 174], [91, 2, 51]	
10	[107, 223, 234], [215, 197, 157], [44, 26, 77], [0, 89, 194], [66, 202, 237], [23, 0, 162], [42, 186, 151], [39, 89, 37], [157, 177, 196], [35, 38, 150]	

图 3 问题二求解

七、问题三的模型建立与求解

7.1 模型假设

问题三的中心是如何将成本与图像表现力放在一起考虑，怎样使二者产生有机联系，定量的分析二者放在一起考虑时的效果值。我们知道，唯一能使二者产生联系的便是砖块数 n

我们可建立如下模型：

假设每一块砖的成本损耗度 (k) ——此处的成本损耗度是相当于总的效果而言总成本损耗度 $w = k * n$ 此时我们可以设立一个以 n 为自变量的新的函数来表示总的效果 (用 $\varphi(x)$ 表示) 通过对各砖块数 n 值对应的 $\varphi(n)$ 的比较从而求出最优解。

$$\varphi = \tau - k \cdot n \quad (9)$$

7.2 模型求解

首先基于成本和表现力两者的同时考虑，我们此处需要定量分析，找到总的效果值。既然砖的成本都是相同的，我们可以假设一个比例系数 k ，取名为每一块砖的成本损耗度， k 与 n 的乘积可看作是降低总效果的一个因素值，即总成本损耗值，故我们不妨设

总的效果值 = 原来的总相似度 - 总成本损耗值

再具体考察 k 的含义，有如下几点影响 k 值大小的因素：

1、成本越大时， k 值越大，相反，则越小；

2、 k 值也取决于制造商的侧重点，若比起图像表现力更好更注重成本的低，则 k 值较大，但若比起成本的高低更注重图像的表现力，则 k 值较小，总之， k 的值随制造商的侧重变化而变化；

3、k 值同样也与总相似程度变化大小与实际的视觉变化相关，若总相似程度的微小变化引起了比较大的图像的呈现的视觉变化，则 k 值较小，相反，若总相似程度的较大变化只能引起图像呈现形式的微小改变，则说明 k 值较大。

由第二问知道此处的总相似度

$$\tau = \sigma_1 + \sigma_2 + \dots = \sum_{i=1}^k \sigma_i$$

k 值是一个与现实各种因素有关的常数，故我们在解答这个问题时可以根据实际情况带入一个合适的 k 值计算，根据结果计算，当 $\varphi(n)$ 取得最大值时即为效果最佳的情况，此时的 n 就是我们要找的添加砖的个数，

同时延续第二问，由第二问的解答我们可以得到 10 块砖以内的情况，若发现当 $n=10$ 时总的效果值仍在增加，则需要继续第二问的步骤，若在 $n=10$ 之前就找到了一个 $\varphi(n)$ 的极大值点，则可以近似认为此点所代表的 n 值就是我们所要求的，此时的 n 块添加的砖也是我们需要的几个。

八、模型评价

8.1 模型优点

1、在计算相似度时选择考虑了两种方法相结合的思想，使误差更小，相似性更准确，一种用的是欧氏距离，相当于算二者颜色间的“距离差”，可想象成“规格中的大小差”，一种用的是余弦相似度，相当于二者颜色间的“角度差”，可想象成“规格中的形状差”。

两种方法的结合更能说明二者颜色的真实相似度或相差多少，而不是单一的用一种方法，这样误差较大，会使结果不准确，从而造成工程上的麻烦，且选择将两种方法放在一个公式中计算，更加方便，同时也减少了分别算两种方法的结果后再接着考虑最终结果的麻烦，同时也降低了步骤繁杂而造成的误差；

2、在模型一的优化方案中提出了根据 RGB 的值来判断相似性误差较大，虽然方便，但是结果会不太准确，如用户选择了一个黄色，但是与标准的瓷砖颜色有一点点差异，此时用 RGB 判断就有困难，而若是转化成 HSV，H——色相，S——饱和度，V——明度，能更直观，更好的比较出两颜色间的差异，而不是 RGB 还需要三个指标叠加，十分不直观，且难以区分到底是哪有差别；

3、将总的图像相似度转化为图像中的各个颜色与对应瓷砖的相似度的叠加十分巧妙，使一二问产生了有机联系，也减少了重新再另创模型的麻烦，还能够起到检查原先第一问模型的合理性的作用；

4、k 值的引入将复杂问题简单化，线性化，同时 k 值的可变性也使该模型不局限于一种或几种情况下，能够根据现实的问题及外界因素的变化而改变，非常灵活，增强

了该模型的应用性，拓宽了其作用，使应用更加广泛。

8.2 模型缺点

1、问题二中要游历其他全部的可能颜色的瓷砖，虽说是用电脑算法，但是仍然十分麻烦，不简便，先算添加一个瓷砖，再接着算添加两个，三个瓷砖，到十个瓷砖，步骤繁琐，容易出错，增加误差；

2、 k 值虽能根据实际情况改变，很灵活，但是 k 值的大小不好掌握，很难根据情况确定 k 值的具体大小，这就需要分析大量的数据，考虑各种因素，才能大致判断再特定的情况下 k 值的范围，若要求更加精确，则需要耗费不小的人力物力，实际这样做有不小难度；

3、在定量分析图像表现力时由于其他各种因素太多且不好定量分析而只单纯考虑了颜色相似度，画面还原度这一种因素，而其他的如颜色冲击度，画面的颜色渐变，搭配度没有考虑在内，造成结果会存在偏差。

参考文献

- [1] 司守奎, 孙兆亮. 数学建模算法与应用 [M]. 第 2 版. 北京: 国防工业出版社, 2016.
- [2] 姜启源, 谢金星, 叶俊. 数学建模 [M]. 第 4 版, 北京: 高等教育出版社.
- [3] 成怡, 朱伟康, 徐国伟. 基于余弦相似度的改进 ORB 匹配算法, 2021-02.
- [4] 许应强, 施庆华, 曲永冬, 王翠芝, 朱攀, 叶愈, . 一种基于遗传算法的图像配准算法研究, 2016-11.

附录 A 问题一结果

表 1 图像 1

序号	瓷砖颜色编号
1	1
2	1
3	11
4	6
5	6
6	6
7	1
8	7
9	6
10	6
11	6
12	6
13	7
14	7
15	6
16	6
17	6
18	16
19	7
20	7
21	7
22	6
23	16
24	16
25	5
26	5
27	5
28	12
29	16
30	16
31	5

32	5
33	12
34	12
35	12
36	12
37	1
38	11
39	11
40	11
41	11
42	6
43	7
44	7
45	11
46	11
47	6
48	6
49	7
50	7
51	7
52	6
53	6
54	6
55	5
56	7
57	7
58	6
59	16
60	16
61	5
62	5
63	5
64	16
65	16
66	16

67	5
68	5
69	5
70	12
71	12
72	12
73	1
74	11
75	11
76	11
77	11
78	11
79	7
80	11
81	11
82	11
83	11
84	6
85	7
86	7
87	11
88	11
89	6
90	6
91	5
92	5
93	7
94	6
95	6
96	16
97	5
98	5
99	5
100	16
101	16

102	16
103	15
104	5
105	5
106	12
107	12
108	12
109	11
110	11
111	11
112	11
113	11
114	11
115	10
116	11
117	11
118	11
119	11
120	11
121	10
122	10
123	11
124	11
125	22
126	22
127	5
128	10
129	10
130	22
131	22
132	22
133	15
134	5
135	5
136	22

137	20
138	20
139	15
140	15
141	5
142	18
143	20
144	20
145	3
146	13
147	13
148	13
149	13
150	13
151	10
152	10
153	10
154	11
155	22
156	22
157	10
158	10
159	10
160	22
161	22
162	22
163	10
164	10
165	10
166	22
167	22
168	22
169	15
170	10
171	10

172	18
173	18
174	20
175	15
176	15
177	18
178	18
179	20
180	20
181	3
182	13
183	13
184	13
185	13
186	13
187	17
188	10
189	13
190	13
191	22
192	22
193	17
194	10
195	10
196	22
197	22
198	22
199	9
200	10
201	10
202	22
203	22
204	8
205	14
206	14

207	18
208	18
209	18
210	20
211	4
212	14
213	19
214	19
215	18
216	20

表 2 图像 2

序号	瓷砖颜色编号
1	12
2	5
3	6
4	6
5	7
6	12
7	11
8	7
9	16
10	16
11	6
12	7
13	16
14	12
15	6
16	7
17	12
18	5
19	12
20	5

21	11
22	7
23	5
24	7
25	6
26	5
27	11
28	16
29	7
30	5
31	12
32	5
33	11
34	7
35	7
36	5
37	16
38	12
39	6
40	7
41	7
42	5
43	7
44	11
45	7
46	7
47	7
48	6
49	11
50	6
51	16
52	16
53	16
54	11
55	11

56	7
57	11
58	16
59	11
60	11
61	6
62	15
63	6
64	12
65	6
66	5
67	16
68	7
69	5
70	5
71	1
72	11
73	5
74	11
75	6
76	6
77	7
78	6
79	11
80	11
81	11
82	11
83	5
84	5
85	12
86	5
87	11
88	11
89	16
90	20

91	5
92	11
93	11
94	7
95	11
96	15
97	11
98	5
99	22
100	10
101	11
102	22
103	15
104	5
105	11
106	22
107	10
108	10
109	10
110	15
111	5
112	22
113	10
114	20
115	10
116	11
117	10
118	22
119	11
120	18
121	15
122	18
123	13
124	22
125	20

126	18
127	22
128	22
129	10
130	10
131	10
132	22
133	22
134	11
135	10
136	22
137	10
138	22
139	3
140	10
141	10
142	18
143	19
144	10
145	20
146	13
147	10
148	22
149	18
150	18
151	10
152	4
153	13
154	3
155	10
156	19
157	22
158	19
159	10
160	13

161	22
162	3
163	22
164	22
165	13
166	17
167	13
168	10
169	10
170	22
171	13
172	17
173	8
174	14
175	22
176	19
177	19
178	19
179	13
180	22
181	21
182	13
183	10
184	19
185	8
186	9
187	8
188	13
189	14
190	4
191	8
192	8
193	17
194	17
195	14

196	8
197	8
198	13
199	13
200	9

附录 B Matlab 程序

```
%读取数据，需要把文件放到路径下
clc;
clear;
data1=xlsread('附件2: 图像1颜色列表');
data2=xlsread('附件3: 图像2颜色列表');
nowci=[0,0,0;255,255,255;255,0,0;246,232,9;72,176,64;
27,115,186;53,118,84;244,181,208;255,145,0;177,125,85;
92,59,144;11,222,222;228,0,130;255,218,32;118,238,0;
17,168,226;255,110,0;201,202,202;255,249,177;179,226,242;
249,255,214;186,149,195;]%%现在瓷砖RGB

%第一问为了方便表达之后意思，我直接用欧式距离来表示两种颜色的相似度，并进行归一化处理也就是都最后除以255^3，
MAX=sqrt(3*255^2);
[x,y]=size(data1);
for i=1:x
    flag1=1;
    flag2=sqrt((data1(i,2)-nowci(flag1,1))^2+(data1(i,3)-nowci(flag1,2))^2+(data1(i,4)-nowci(flag1,3))^2);
    for j=1:22 %%22个瓷砖颜色
        w=sqrt((data1(i,2)-nowci(j,1))^2+(data1(i,3)-nowci(j,2))^2+(data1(i,4)-nowci(j,3))^2);
        if(w<flag2)
            flag1=j;
            flag2=w;
        end
    end
    result1(i,1)=flag1;%%这个就是第一问的答案，附件2的对应结果
    result1(i,2)=flag2/MAX;
end
[x,y]=size(data2);
for i=1:x
    flag1=1;
    flag2=sqrt((data2(i,2)-nowci(flag1,1))^2+(data2(i,3)-nowci(flag1,2))^2+(data2(i,4)-nowci(flag1,3))^2);
    for j=1:22 %%22个瓷砖颜色
        w=sqrt((data2(i,2)-nowci(j,1))^2+(data2(i,3)-nowci(j,2))^2+(data2(i,4)-nowci(j,3))^2);
        if(w<flag2)
```

```

flag1=j;
flag2=w;
end
end
result2(i,1)=flag1;%%这个就是第二问的答案，附件3的对应结果
result2(i,2)=flag2/MAX;
end

%第二问的计算，遗传算法
%先加载数据；
clc;clear;
data1=xlsread('附件2: 图像1颜色列表');
data2=xlsread('附件3: 图像2颜色列表');
nowci=[0,0,0;255,255,255;255,0,0;246,232,9;72,176,64;
27,115,186;53,118,84;244,181,208;255,145,0;177,125,85;
92,59,144;11,222,222;228,0,130;255,218,32;118,238,0;
17,168,226;255,110,0;201,202,202;255,249,177;179,226,242;
249,255,214;186,149,195;]%%现在瓷砖RGB
MAX=sqrt(3*255^2);
[x,y]=size(data1);
result3=cell(10,2);
for j=1:10 %%我们需要求解10次，逐次增加瓷砖个数
popsize=50; %群体大小
chromlength=24*j; %字符串长度（个体长度）根据范围和精度决定，这个地方随着j的增加而增加；
pc=0.8; %交叉概率
pm=0.1;%变异概率
[c]=initpop(popsize,chromlength);
[objvalue]=calobjvalue(c,j); %计算目标函数
fitvalue=calfitvalue2(objvalue,data1,data2,nowci,j)/MAX;%计算群体中每个个体的适应度
q=0;
z=[];
for i=1: 200%50为迭代次数
[newpop]=selection2(c,fitvalue); %选择
pop=newpop;
[newpop]=crossover(pop,pc,j); %交叉
[newpop]=mutation(newpop,pm); %变异
[objvalue]=calobjvalue(newpop,j); %计算目标函数
fitvalue=calfitvalue2(objvalue,data1,data2,nowci,j)/MAX;%计算群体中每个个体的适应度
[bestpop,bestobjvalue,bestfitvalue]=best(newpop,fitvalue,objvalue);
    %求出群体中适应值最大的个体及其适应值
if q<bestfitvalue
q=bestfitvalue;
z=bestobjvalue;
end
c=newpop;
end
result3(j,1)={j};

```

```

result3(j,2)={z};
end

%calfitvalue2.m
function fitvalue=calfitvalue2(objvalue,data1,data2,nowci,m)
[x,y]=size(objvalue);
[x1,y1]=size(data1);
[x2,y2]=size(data2);
q=nowci;
MAX=sqrt(3*255^2);
for z=1:x
for n=1:m
q(22+n,1)= objvalue(z,1+(n-1)*3);
q(22+n,1)= objvalue(z,2+(n-1)*3);
q(22+n,1)= objvalue(z,3+(n-1)*3);
end
result1=0;
for i=1:x1
flag1=1;
flag2=sqrt((data1(i,2)-q(flag1,1))^2+(data1(i,3)-q(flag1,2))^2+(data1(i,4)-q(flag1,3))^2);
for j=1:22+m %%22个瓷砖颜色
w=sqrt((data1(i,2)-q(j,1))^2+(data1(i,3)-q(j,2))^2+(data1(i,4)-q(j,3))^2);
if(w<flag2)
flag1=j;
flag2=w;
end
end
result1=result1+flag2;
end
result2=0;
for i=1:x2
flag1=1;
flag2=sqrt((data2(i,2)-q(flag1,1))^2+(data2(i,3)-q(flag1,2))^2+(data2(i,4)-q(flag1,3))^2);
for j=1:22+m %%22个瓷砖颜色
w=sqrt((data2(i,2)-q(j,1))^2+(data2(i,3)-q(j,2))^2+(data2(i,4)-q(j,3))^2);
if(w<flag2)
flag1=j;
flag2=w;
end
end
result2=result2+flag2;
end
fitvalue(z)=(result1+result2)/416;
end

%calobjvalue2.m
function objvalue=calobjvalue2(pop,a)

```

```

[x,y]=size(pop);
for i=1:x
w=pop(i,:);
objvalue(i,:)=distance2(w,a);
end
end

%crossover.m
function newpop=crossover(pop,pc,j)
[px,py]=size(pop);
newpop=pop;
w=px;
for i=1:2:px
if rand<pc
w=w+1;
z=ceil(rand*8);
if z>7
z=7;
end
for q=1:j
k=(q-1)*24;
newpop(w,1+k:8+k)=[pop(i,1+k:z+k),pop(i+1,z+1+k:8+k)];
newpop(w,9+k:16+k)=[pop(i,9+k:9+z+k),pop(i+1,9+z+1+k:16+k)];
newpop(w,17+k:24+k)=[pop(i,17+k:17+z+k),pop(i+1,17+z+1+k:24+k)];
end
end
end

%初始化
function pop = initpop(popsizelength)
pop=round(rand(popsizelength));
end

%mutation.m
function newpop=mutation(c,pm)
[px,py]=size(c);
z=rand(px,py);
z(z>pm)=0;
z(z<=pm)=1;
c=c+z;
c(c>1)=0;
newpop=c;
end

%selection2.m
function newpop=selection2(pop,fitvalue)

```

```

[x,y]=size(pop);
for i=1:50
w=round(x*rand(1,4));
w=sort(w);
if w(1)==0
w(1)=1;
end
q=w(1);
for j=1:4
if w(j)==0
w(j)=1;
end
if fitvalue(q)>fitvalue(w(j))
q=w(j);
end
end
newpop(i,:)=pop(q,:);
end
end

%best.m
function [bestpop,bestobjvalue,bestfitvalue]=best(newpop,fitvalue,objvalue);
    %求出群体中适应值最大的个体及其适应值
w=1;
[x,y]=size(fitvalue);
for i=2:x
if(fitvalue(i)<fitvalue(w))
w=i;
end
end
bestpop=newpop(w,:);
bestobjvalue=objvalue(w,:);
bestfitvalue=fitvalue(w);
end

%第三问
function fitvalue=calfitvalue3(objvalue,data1,data2,nowci,m)
    %适应度是新瓷砖到416个颜色的平均距离 %m是瓷砖个数
[x,y]=size(objvalue);
[x1,y1]=size(data1);
[x2,y2]=size(data2);
q=nowci;
MAX=sqrt(3*255^2);
for z=1:x
for n=1:m
q(22+n,1)= objvalue(z,1+(n-1)*3);
q(22+n,2)= objvalue(z,2+(n-1)*3);

```

```

q(22+n,3)= objvalue(z,3+(n-1)*3);
end
result1=0;
for i=1:x1
flag1=1;
flag2=sqrt((data1(i,2)-q(flag1,1))^2+(data1(i,3)-q(flag1,2))^2+(data1(i,4)-q(flag1,3))^2);
for j=1:22+m %%22个瓷砖颜色
w=sqrt((data1(i,2)-q(j,1))^2+(data1(i,3)-q(j,2))^2+(data1(i,4)-q(j,3))^2);
if(w<flag2)
flag1=j;
flag2=w;
end
end
result1=result1+flag2;
end
result2=0;
for i=1:x2
flag1=1;
flag2=sqrt((data2(i,2)-q(flag1,1))^2+(data2(i,3)-q(flag1,2))^2+(data2(i,4)-q(flag1,3))^2);
for j=1:22+m %%22个瓷砖颜色
w=sqrt((data2(i,2)-q(j,1))^2+(data2(i,3)-q(j,2))^2+(data2(i,4)-q(j,3))^2);
if(w<flag2)
flag1=j;
flag2=w;
end
end
result2=result2+flag2;
end
fitvalue(z)=0.1*m+(result1+result2)/416; %%对面m前面的数字是代价，代价可以分为高中低0.5,0.1,0.02等等，之后选取10种中
end

```