

马赛克瓷砖选色问题

摘要

现代装修采用的马赛克瓷砖装修的背景墙因具有装饰性强、使用寿命长、环保性高等优点而深受用户喜爱。马赛克背景墙以拼接的形式打造而成,样式各异,色彩多变,以丰富的图案给人以冲击,美感,为家居空间赋予了全新的感受。但是受工艺和成本的限制,瓷砖的颜色只是有限的几种,于是就面临着马赛克瓷砖的选色问题。本文将对该问题进行全面而深入的探讨。

针对问题一: 基于图像色彩领域已有的色差公式,欧几里德距离公式(见公式 1-1)、REDMEAN 公式(见公式 1-3)、CIEDE2000 公式(见公式 1-5),通过实验比较以及理论分析,我们根据精度最高最符合人眼视觉感受的 **CIEDE2000 色差公式**,建立了通过 **LAB 颜色空间** 色差选取近似颜色的算法模型。通过计算图像颜色与已有瓷砖颜色色差,并选出与图像颜色色差最小的颜色,作为最符合人眼视觉感受的近似色,最后将结果输出至结果文件(见附录)。

针对问题二: 一代模型我们建立了最小色差平方和概念来代表图像表现力。采用**遗传算法**,将添加颜色编码为染色体,通过遗传进化求得近似最优解(见表 2)。二代模型我们分析了一代模型的缺点和结果,做出两点改进,参考高销量背景图选取了更具**代表性的目标颜色集合**,参考心理学**斯蒂文斯幂定律**制定了更合理的效益函数代表图像表现力,将改进后的效益函数应用于遗传算法,求得了更符合实际情况的近似最优解(见表 3)。

针对问题三: 问题三是在问题二基础上的**成本效益**问题。我们利用问题二所得的增加颜色数与效益函数值关系做出折线图,进一步做出的每增加一个颜色对效益函数影响的台阶图,发现效益函数具有**边际效益递减**特征。通过分析效益函数图,得出结论在成本有限的情况下,可以优先考虑增加 **1、2 或 4** 种颜色(见表 4)。

最后我们对模型进行了中肯的评价和适当的推广。

关键词: CIELAB 颜色空间 CIEDE2000 色差公式 遗传算法 斯蒂文斯幂定律 边际效益

马赛克瓷砖选色问题.....	1
摘要.....	1
一、问题重述.....	3
1.1 问题背景.....	3
1.2 问题提出.....	3
二、问题分析.....	3
2.1 问题一的分析.....	3
2.2 问题二的分析.....	3
2.3 问题三的分析.....	4
三、模型假设.....	4
四、符号说明.....	4
五、模型的建立与求解.....	5
5.1 问题一模型的建立与求解.....	5
5.2 问题二模型的建立与求解.....	8
5.3 问题二模型的改进.....	11
5.4 问题三模型的建立与求解.....	18
六、模型的评价.....	19
6.1 模型优点分析.....	19
6.2 模型缺点分析.....	19
七、模型的推广.....	19
八、参考文献.....	20
附录.....	21

一、问题重述

1.1 问题背景

采用纯天然原料制成、绿色环保、使用寿命长、色彩丰富多变的马赛克背景墙已经成为了现代家居装修的“宠儿”，马赛克背景墙是由尺寸较小的马赛克瓷砖拼接组合而成，但受限于工艺和成本等因素，瓷砖的颜色不可能覆盖到所有色系，只能是有限多种。所以在拼接图案时要根据原图中的颜色找出颜色最相近的瓷砖代替。某工厂现只能生产22种颜色的马赛克瓷砖，为了尽可能的达到用户期待的装饰效果，需要开发一款能准确而迅速的找出颜色最接近的瓷砖的自动化识别软件，以减少人工选色的工作量，提高生产效率和精确度。

1.2 问题提出

问题一：设计出相近颜色选择的算法，从已给出的图像1（附件2）中的216种颜色，以及图像2（附件3）中的200中颜色，一一找出每种颜色最接近的瓷砖颜色，并将选出的瓷砖颜色的编号按照附件4的要求传输至结果文件。

问题二：如果该厂计划研发新颜色的瓷砖，在不考虑研发难度，仅考虑拼接图片的表现力的情况下，分析应该添加哪几种颜色的瓷砖。添加1种、2种、……、10种瓷砖时颜色RGB值应该如何选择。

问题三：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，在综合考虑成本和表现效果的前提下，分析计算得出新增的颜色种类及数量，说明理由并给相应的RGB编码值。

二、问题分析

2.1 问题一的分析

问题一要求对给出图像中的颜色找出颜色相近的瓷砖，即找出人眼效果色差最小的近似颜色。我们将分析三种求色差的公式，分别是欧氏距离公式、REDMEAN 公式以及 CIEDE2000 公式。通过查阅网上资料，CIEDE2000 是在 CIELAB 颜色空间上进一步解决感知非均匀问题的、即人眼效果近似度最高的色差公式。为了验证该公式的近似效果，我们需要用实例来更直观的观察近似的优越性，即用 Matlab 对示例图象进行像素处理。人眼对比，验证 CIEDE2000 公式是否优于其他色差公式，若是，则采用 CIEDE2000 公式对题中图像一、二所给颜色一一与已知 22 种颜色做色差计算，得出计算值最小的即为最近似颜色。

2.2 问题二的分析

第二问要求在原有的 22 种颜色上增加一种、两种至十种新的颜色，使得拼接的图像表现力最优。我们规定评价图像表现力的方法为计算给定图像颜色与瓷砖颜色的最小色差平方和，最小色差平方和越小代表图像和原图颜色越接近，表

现力越好。根据第一问的 CIEDE2000 算法，我们计算添加颜色后图像的最小色差平方和。分析可知该问题为优化问题，并且是 NP-hard 问题，依靠普通优化算法难以给出结果，因此我们将采用智能优化遗传算法来分析求解答案。之后对示例图和遗传算法的结果进行分析，改进模型。

2.3 问题三的分析

第三问要求综合考虑成本和表现效果，对于增加哪几种颜色提出建议。我们分析前面通过遗传算法得出的最优效益函数值和增加的颜色数的折线图，发现它们总体上符合边际效益函数性质。根据边际效益函数性质，可知本题表现效果随着颜色的增加而增大，但增长速度会不断降低。我们把每增加一个颜色对应的效益函数的增加值做成台阶图，可以清晰的看出每增加一个颜色，效益函数的增加量。通过比较台阶图并综合考虑成本因素，可得出增加颜色数的最优方案。

三、模型假设

本文作出如下假定：

1. 假设每个人对色差的主观感受相同，可以用通用公式衡量；
2. 假设近似出的瓷砖颜色以尽量接近原色为准，不考虑抽象的艺术审美；
3. 假设调查所得的高销量背景图具有代表性；

四、符号说明

表 1 符号标注

序号	变量名称	变量描述
1	N	瓷砖颜色数
2	N^D	目标颜色数
3	N^A	添加的瓷砖数
4	ΔE_{ij}	第 i 种目标颜色和第 j 种瓷砖颜色色差
5	ΔE_{ij}^A	第 i 种目标颜色和第 j 种添加瓷砖颜色的色差
6	C_i	第 i 个个体染色体
7	N^c	个体数
8	P_m	个体变异概率
9	R_E	精英比例
10	G	遗传代数
11	GAP	代沟 (生存竞争中胜利个体所占比例)
12	ω	效益函数
13	f_i	颜色频度
14	w_i	颜色权重

注:其余符号均以在文中作以解释；

五、模型的建立与求解

5.1 问题一模型的建立与求解

针对问题一，我们将利用欧式距离公式、REDMEAN 公式、CIEDE2000 公式这三种公式建立数学模型，通过近似图样结果，比对分析三者的近似效果。

5.1.1 欧式距离公式

很多日常使用的“颜色差异”，是直接通过在一个“设备无关”的色彩空间里，进行欧氏距离的计算得到的。给定一个 RGB（红绿蓝）的色彩空间，最简单的差异计算方式就是在这个三维空间里求两个点间的距离：

$$distance = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2} \quad (1-1)$$

但仅用简单的 RGB 距离表示色差，忽略了人眼对于不同色彩和明度的感受，以及各种影响因素，并不符合人眼对于色差的感受，用这种公式计算虽然能得到相近颜色，但对于人眼视觉效果可能并不接近，对于瓷砖装修来说应用效果不好，不能给人最好的视觉体验。

5.1.2 REDMEAN 公式

鉴于人眼对于红绿蓝三种颜色亮度感受大约（red 30% ， green 59% ， blue 11%），还有一种更好的近似公式：

$$\bar{r} = \frac{R_1 + R_2}{2} \quad (1-2)$$

$$\Delta C = \sqrt{\left(2 + \frac{\bar{r}}{256}\right) \times \Delta R^2 + 4 \times \Delta G^2 + \left(2 + \frac{255 - \bar{r}}{256}\right) \times \Delta B^2} \quad (1-3)$$

但这些公式都是仅对颜色 RGB 值做了不同变换，对于人眼对于色彩的主观感受描述还是不够。用这些公式来寻找相近颜色可能对于人眼的感受来说并不是最好的。

5.1.3 CIEDE2000 公式

我们搜集资料，考虑 CIEXYZ 颜色空间，这个模型采用三刺激值描述颜色（在颜色匹配中，当与待测色达到色匹配时所需要的三原色数量，称为三刺激值），颜色匹配实验即根据人眼的效果实验得出，因此能较好的符合人眼的视觉效果。三刺激值虽然考虑了颜色和明度的因素，但都与颜色感觉没有直接对应的关系，并不均匀，例如绿色区域敏感度较差，在绿色区域颜色变化较大时，人眼才能分辨出两个颜色的差异，但在蓝紫色区域，较小的变化就能引起视觉差异。因此用 CIEXYZ 颜色空间计算色差，对于人眼的感觉是不均匀的。

CIELAB 颜色空间，用 L、A、B 三个坐标轴表示颜色空间，L 轴代表明度，A 轴代表红绿，B 轴代表黄蓝，红绿和黄蓝更符合人眼构造对于色彩的识别，能

更好的符合人眼识别效果。

CIELAB 相比与 CIEXYZ 对于人眼感官更加均匀，不过后来人们发现 LAB 颜色空间，尤其是在饱和度较高的区域里，并没有设计时预想的那么“感官上均匀”。相同色差感觉的实际区域，不是一个球形，而是一个椭球。

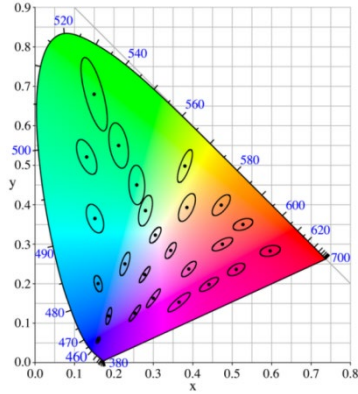


图 1.1 XYZ 色域图

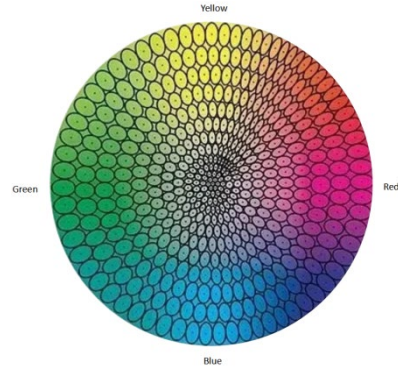


图 1.2 LAB 色域图

而 CIEDE2000 公式对于在 CIELAB 中直接用距离计算做了一些修正，优化对颜色的标定必须符合人眼的视觉规律，以及对颜色差异的评判，是 CIELAB 空间主要推广的色差公式。

CIEDE2000 色差公式是基于 CIELAB 颜色空间提出的，给定 CIELAB 颜色空间 L_1 、 A_1 、 B_1 和 L_2 、 A_2 、 B_2 中的一对颜色值，我们按照如下公式表示 CIEDE2000 色差公式的结果：

$$\Delta E_{00}(L_1, A_1, B_1, L_2, A_2, B_2) = \Delta E_{00} \quad (1-4)$$

给定两个 CIELAB 颜色值、以及参数加权因子 K_L ， K_C ， K_H 颜色差的计算过程主要分为以下三个步骤：

Step 1 :

计算彩度值 C'_i 以及色相值 h'_i ，其中 i 取 1 和 2，其中 C_{ab} 为标样彩度与测评样彩度的算术平均值， G 作为色相轴 A 轴的调整因子，是以彩度变量的函数值。详细过程见附录公式；

Step 2 :

计算明度差 $\Delta L'$ ，彩度差 $\Delta C'$ ，色相差 $\Delta H'$ ，详细过程见附录公式；

Step 3 :

根据 CIEDE2000 公式计算色差值：

$$\Delta E_{00} = \sqrt{\left[\left(\frac{\Delta L'}{K_L S_L} \right)^2 + \left(\frac{\Delta C'_{ab}}{K_C S_C} \right)^2 + \left(\frac{\Delta H'_{ab}}{K_H S_H} \right)^2 + R_T \left(\frac{\Delta C'_{ab}}{K_C S_C} \right) \left(\frac{\Delta H'_{ab}}{K_H S_H} \right) \right]} \quad (1-5)$$

S_L, S_C, S_H 分别为明度值 L, 彩度值 C 和色相值 H 的权重函数, 它们通过调整自身已达到影响色差公式对颜色均匀性评价的作用。参数因子 K_L, K_C, K_H 可根据不同的评价对象和评价条件选择不同数值, 参考条件下的参数因子数值均为 1。当观察条件改变时, 可以对各参数进行单独改变以适应视觉上每个颜色分量 (明度、彩度、色相) 的变化。详细推导过程见附录公式 1.2;

5.1.4 三种公式的近似效果

为尽量对比出三种公式的近似程度, 我们选择了一幅色彩丰富且具有明显轮廓的油画 (见下图 1.3)。为提高程序的运行效率, 我们将原图缩放至 100px*100px 并予以逐像素处理 (考虑到马赛克瓷砖拼图本就精度不高, 这种缩放可以接受), 即将像素块逐个与所给瓷砖颜色进行求色差计算, 将计算出的最近似颜色覆盖到原像素块上, 绘制出对应求色差公式的近似样图 (见下图 1.4 1.5 1.6)。

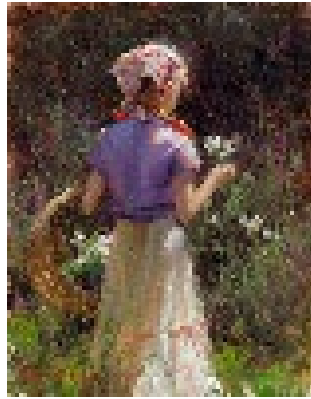


图 1.3 原图



图 1.4 欧式公式样图



图 1.5 REDMEAN 公式样图



图 1.6 CIEDE2000 公式样图

人眼观察三张近似图,我们发现原图背景色更近似于暗黑色,而由欧式公式, REDMEAN 公式近似出的背景色出现很多突兀的绿块,而且图中少女的棕色挎篮不清晰。可观察到图 1.6 背景色从感官上更符合原图,且挎篮部分轮廓清晰,少女头巾鲜亮色部分有明显的还原,可以看出 CIEDE2000 公式的近似效果最优。

5.1.5 问题一结果分析

使用 CIEDE2000 公式,结果在附录 1.1。

5.2 问题二模型的建立与求解

根据第一问的 CIEDE2000 算法,把已有的 22 块瓷砖和添加瓷砖的颜色 RGB 值转换为 LAB 坐标值,把图像中的颜色 RGB 值也转换为 LAB,计算其与所有瓷砖的色差的最小值,称为最小色差,并把图像中所有颜色的最小色差值求平方和,求得的最小色差平方和作为判断拼接图像表现力的标准。此问题为优化问题,并且是 NP-hard 问题,依靠普通优化算法难以给出结果,因此我们采用遗传算法。根据适者生存原则,用生物遗传模拟优化过程,通过群体逐代进化,来逼近问题的最优解。

5.2.1 采用遗传算法解决最优解问题

遗传算法是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型,是一种通过模拟自然进化过程搜索最优解的方法。该算法通过数学的方式,利用计算机仿真运算将问题的求解过程转换成类似生物进化中的染色体基因的交叉、变异等过程。在求解较为复杂的组合优化问题时,相对一些常规的优化算法,通常能够较快地获得较好的优化结果。

Step 1: 确定染色体编码方式

本模型采用十进制编码方式,把添加颜色点的 LAB 坐标按顺序拼接为染色体串。

$$C_i = [L_1, A_1, B_1, L_2, A_2, B_2, \dots, L_{N^A}, A_{N^A}, B_{N^A}] \quad (2-1)$$

Step 2: 初始化种群

随机生成基因长度的编码,对种群赋予初始值。由分析可知,最优解基因 LAB 值应处于目标颜色范围之间,因此应限制种群随机初始值处于这个范围中间。

Step 3: 确定适应函数

本题优化目标为使目标颜色的最小色差平方和最小。

1) 首先计算加入新颜色后对于图像中第 i 种颜色的新的最小色差:

$$\Delta E_i' = \min \left\{ \min_{1 \leq j_1 \leq N} \{ \Delta E_{ij_1} \}, \min_{1 \leq j_2 \leq N^A} \{ \Delta E_{ij_2}^A \} \right\} \quad (2-2)$$

2)求最小色差平方和 Q_k ， Q_k 为加入第 k 个体所代表的颜色后，新的最小色差平方和，对于 N^c 个个体，每个个体都求出加入后的色差和。 Q_k 越小代表个体适应度越好。

$$Q_k = \sum_{i=1}^n (\Delta E_i') \quad (2-3)$$

$$k = 1, 2, 3 \dots N^c$$

3)将 Q_k 归一化，记为 R_k ，作为适应度函数。使其处于 $[0, 1]$ 范围，并且具有正向性，越接近 1 个体适应性越好，即加入该个体代表的颜色后，新的最小色差平方和更小了。

$$R_k = \frac{\max_{1 \leq i \leq N^c} \{Q_i\} - Q_k}{\max_{1 \leq i \leq N^c} \{Q_i\} - \min_{1 \leq i \leq N^c} \{Q_i\}} \quad (2-4)$$

Step 4: 筛选优秀个体

使用上一步计算的适应度函数，把所有个体按适应度函数值排序，选出占比前 0.75 的优秀个体，优秀个体基因能传给下一代，而剩余的较差个体被淘汰。优秀个体结合产生占比 0.25 的后代。

Step 5: 遗传算子的设计

确定了染色体的编码方式和适应度计算后，考虑染色体的交叉和变异操作：

1、交叉操作：对于两个优秀个体的结合，采用随机分配两者权重，对其相应的基因进行加权组合，产生下一代基因。随机分配权重更符合遗传的特性，有更大的概率更快的逼近最优解。

$$C_k = C_a \cdot p + C_b \cdot (1 - p) \quad (2-5)$$

$$N^c' = N^c \cdot GAP \quad (2-6)$$

每次随机选择两个 select 选中的个体，将其染色体按随机比例组合。 a, b 为 $1 \sim N^c$ 中的随机整数， p 为 $0 \sim 1$ 的随机数， $K = N^c + 1, N^c + 2, \dots, N^c$ 。

2、变异操作：除了精英个体，每个优秀个体有 P_m 概率发生变异,变异个体随机一个基因变为取值范围中的随机值，即一个 LAB 分量变为随机值。变异有利于产生更优秀的变异个体，是种群进化的重要方式。

Step 6: 设定遗传算法相关参数

经过反复测试，得到一组较优的参数：

$$N^c = 50$$

$$P_m = 0.05$$

$$R_E = 0.05$$

$$GAP = 0.75$$

$$G = \begin{cases} 30, n \leq 3 \\ 30 + 5(n - 3), n > 3 \end{cases} \quad (2-7)$$

遗传代数 G 一开始取为定值 30，随着添加瓷砖数的增多，遗传算法逼近最优解的难度加大，容易在结束时还未达到最优解，因此设计了遗传代数 G 的动态增长。

5.2.3 一代模型的结果及分析

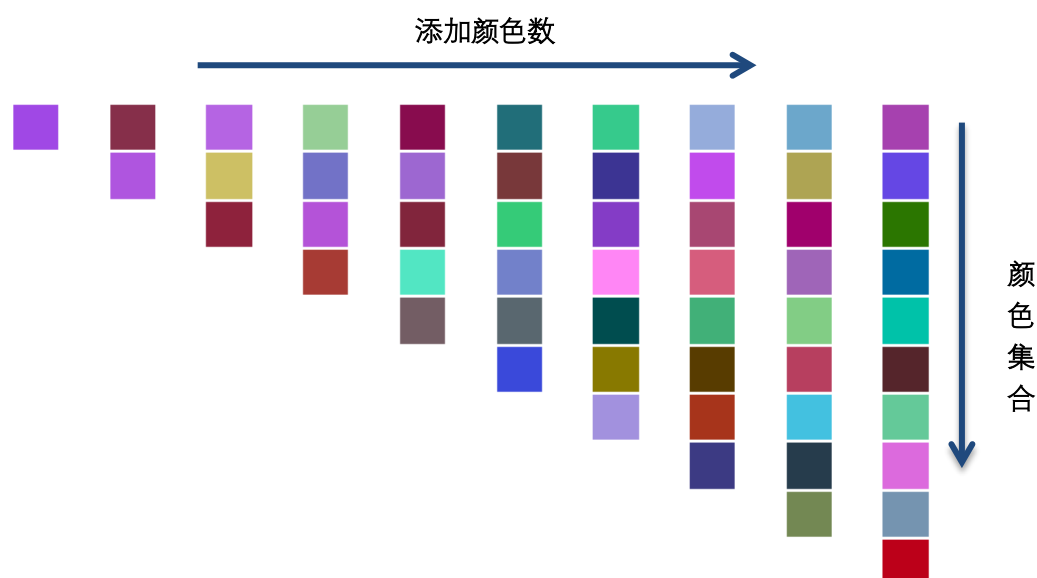


图 2.1 一代模型结果色卡图

表 2 一代模型结果

添加瓷砖的个数	添加瓷砖的颜色 RGB 值
1	(160,72,229)
2	(134,47,74),(175,85,223)
3	(181,100,227),(205,192,100),(142,34,60)
4	(150,206,150),(114,114,199),(180,83,216),(167,59,52)
5	(136,12,78),(157,103,209),(129,37,60),(82,230,195),(115,93,100)
6	(33,110,121),(120,56,58),(53,203,120),(114,129,202), (89,103,111),(58,73,218)
7	(54,202,140),(60,52,147),(132,60,198),(255,134,245), (0,77,79),(136,121,0),(162,145,222)
8	(149,172,219),(193,75,236),(168,71,114),(214,93,125), (65,176,120),(88,60,0),(167,52,27),(60,58,131)
9	(108,167,203),(174,164,83),(160,0,108),(159,101,184), (130,205,133),(183,63,95),(67,193,224),(38,60,76),(115,136,83)
10	(166,65,175),(101,71,228),(43,118,0),(0,107,161),(0,194,169), (85,37,43),(100,201,153),(220,106,221),(117,148,176),(188,0,25)

针对模型一，发现最佳添加颜色基本都为颜色较深的、偏向蓝紫色的颜色，这与图像 1、2 的颜色分布有关，图像 1、2 的颜色在蓝紫色区域密度更高，因此在计算最小色差平方和时，添加点也会更偏向蓝紫色区域。在添加 10 种瓷砖时，添加颜色基本都为深色，这显然使不符合实际应用场景。接下来的改进二代模型针对该问题，提出了解决方案。

5.3 问题二模型的改进

5.3.1 讨论图像一、二是否具有代表性

鉴于一代模型我们是使用附件图像 1 和图像 2 的颜色作为拼接图像颜色的代表，计算其与瓷砖颜色的最小色差平方和，来找出最优可增颜色。但图像 1 和 2 是否具有代表性和普遍性，需要我们进一步的讨论。

在判断所给样图是否具有代表性时，我们可以从这两方面分析：一是颜色分布要尽量均匀；二是颜色要能代表装修时常用墙面马赛克拼图的颜色。我们在此作出进一步探究。

Step1：分析第一点

首先考虑附件中给出的图像 1、2 的颜色集合是否均匀。我们将图像 1、2 的颜色分布绘制在 RGB 三维空间中发现图一图二都是在 RGB 空间内均匀分布的颜色，图一为等间隔分布，图二为随机均匀分布。考虑我们第一题获得的结论，

RGB 色彩空间是不均匀的，在数据上均匀分布的点，对于人眼来说并不均匀，可能出现对于某些颜色精度很高，而对另一些颜色精度很低的现象。

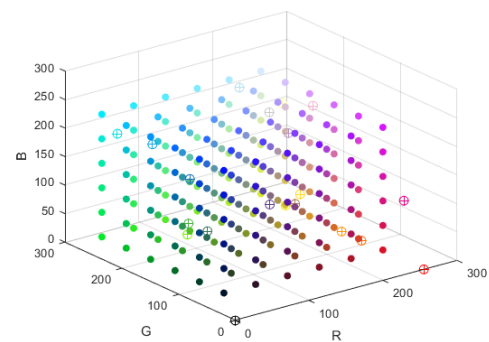


图 3.1 图像 1 在 RGB 空间内的分布图

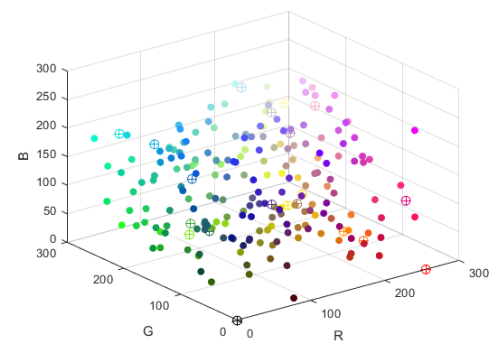


图 3.2 图像 2 在 RGB 空间内的分布

将图像 1 图像 2 绘制在对于人眼更均匀的 CIELAB 空间内，发现确实是不均匀分布的。对于蓝紫色的颜色精度很高，而对于黄绿色精度很低。

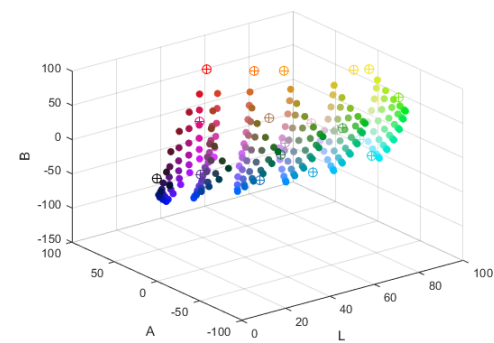


图 3.3 图像 1 在 CIELAB 空间内的分布

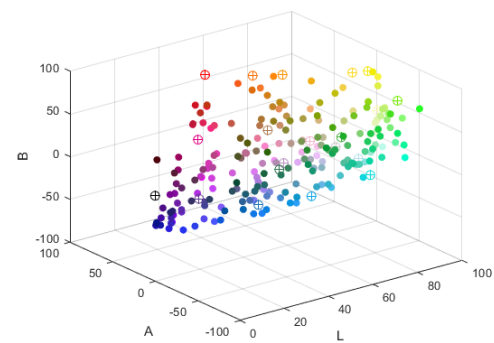


图 3.4 图像 2 在 CIELAB 空间内的分布

接下来我们生成一组对于人眼均匀分布的颜色集合，代替题目所给的样图，用其来代表普遍拼接图像的新颜色集合。类似图像 1 的生成方式，我们使用在 LAB 空间内均匀分布的点来代表均匀分布的颜色。

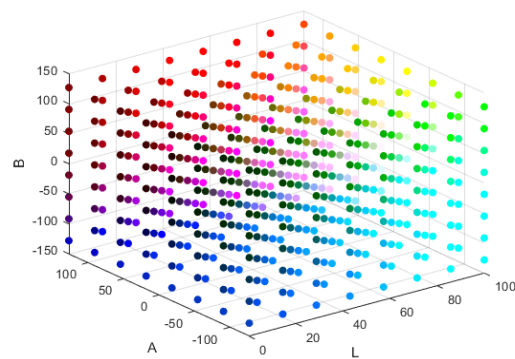


图 3.5 生成新的 LAB 空间均匀分布的颜色集合

Step 2: 分析第二点

1) 调查市场上热销背景图的颜色使用情况

我们设定马赛克瓷砖的主要应用场景是在家居装修时的墙面背景图,对于特定的使用场景,颜色的使用率并不是均匀的,即新生成的 LAB 空间颜色集合中存在部分颜色使用率偏低。对于频繁使用的颜色,需要更高的精度,而少量使用的颜色,色差较大也可以接受。

为了分析家居装修时常用的马赛克背景图的颜色分布,我们需要先找到装修时最常用的背景画。为综合考虑到选色的实用度,于是我们通过线下调查实体装修店和线上参考热销网店的背景图的销量数据,获取了 35 幅销较高的背景画,将其作为市面上最受欢迎,即市场需求量最高的背景画代表。

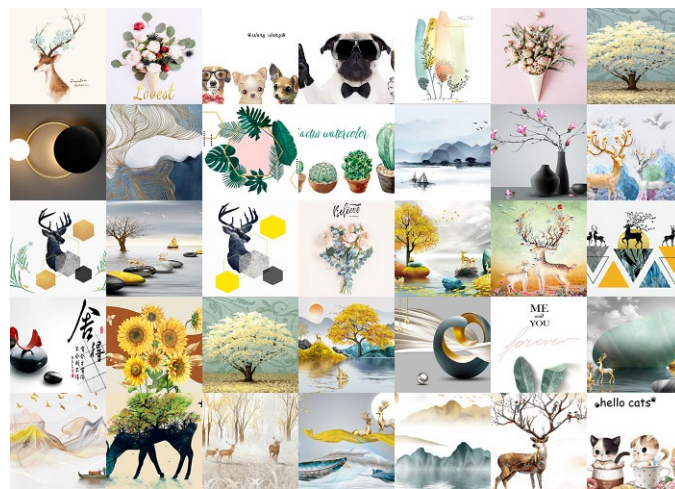


图 3.6 高销量背景图汇总

统计出的高销量背景画如上图(图 3.6)接下来我们使用 Matlab 软件依次读入这 35 张背景画,扫描以获取各颜色的出现次数,最终统计出各颜色的使用率。

2) 提高遗传算法运行效率

由于遗传算法的执行效率较低,为了提高执行效率,我们需要先减小颜色集合大小。使用前一步产生的均匀分布的颜色集合为匹配颜色,图像上每个像素点的颜色为目标颜色,计算匹配颜色和目标颜色的色差,选取色差最小的匹配颜色代表目标颜色。

考虑到匹配的瓷砖只有 22~32 种颜色,我们将图片上的所有颜色近似为 516 种颜色所带来的误差是可以接受的。接下来统计各颜色的使用率,在后续计算中出现率高的颜色在计算色差平方和时赋予更高的权重。而出现率相当低的颜色可以直接将其剔除出颜色集合,进一步提高遗传算法的效率。

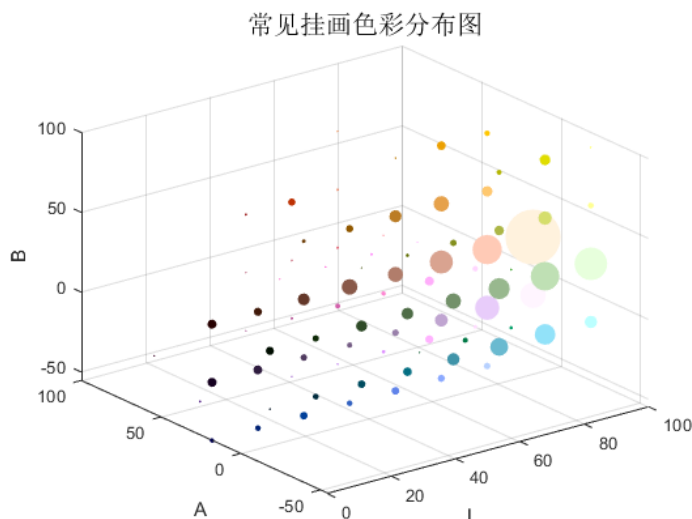


图 3.7 常用颜色频布图

依据统计结果，我们绘制了常用颜色的频度图（见图 3.7）。在 35 幅图 56000 个像素点中出现次数小于 10 次的颜色直接舍去，剩下的颜色频度以图中散点的大小直观呈现。

5.3.2 斯蒂文斯幂定律分析人对色差的刺激反应和色差大小的关系

只用色差公式来说明图像拼接表现力是不够的，色差最终要反应到人的刺激感觉上，例如对于不同颜色的相同色差，人感受到的主观刺激有可能不同。对于不同相近的颜色，人眼可能对边界模糊化，因此，除了色差公式，还要综合人体对于色差刺激的反应。

对于图像与瓷砖之间色差可以直接用最简单的视觉评价法，但这种方法没有标准可依，完全凭借观察者的经验评定颜色的好坏，对于不同个体主观性太大。在标准化的今天，这种方法得到的结果不能满足越来越严格的质量要求。应该用公式使其具体化。色差公式能从客观出发用颜色数据来评价色差。但所得数据与人眼的视觉评判还是有出入，因此有必要用心理物理学和一些数据可靠性方法对颜色评价方式进行改造。

利用颜色数据通过色差公式计算出来的色差值属于客观数据，把色差数据作为物理刺激，利用刺激与人主观感受的公式，可以把客观数据转换为人的主观感受。利用人感受到的刺激来评价色彩表现力能更好的满足顾客的需要。心理物理学把刺激物理量转化为人的刺激感觉量主要有三个公式，韦伯公式，费希纳法则，斯蒂文斯幂定律。其中斯蒂文斯幂定律综合了前两种公式的优势，对感受量与数学表达之间的关系进行了更详细的描述，涉及的感觉量范围更广，因此我们选择斯蒂文斯幂定律：

$$S = KI^r \quad (3-1)$$

式中, S 代表感觉强度, K 是个常数, I 代表刺激的物理强度, r 代表指数值。当 r 大于一个强度时, 感觉强度 S 具有持续上升的趋势; 当 r 小于一个强度时, S 具有费希纳法则的压缩趋势。

5.3.3 改进模型

综合运用上述提到的改进技术, 我们只需修改模型一遗传算法的适应度函数即可: (接公式 2-2) 添加效益函数

$$\omega_k = K \sum_{i=1}^n \left[\left(\Delta E_i' \right)^r \cdot w_i \right] \quad (4-1)$$

其中 K 和 r 为斯蒂文斯幂定律常数, w_i 为第 i 种目标颜色权重, 由目标颜色的权重计算:

$$w_i = \frac{\sqrt{f_i}}{\sum_{j=1}^{N^D} (\sqrt{f_j})} \quad (4-2)$$

权重与颜色在参考画中出现次数的开方成正比。这是考虑到一幅图中背景色占比通常非常大, 如果线性加权可能导致添加瓷砖的颜色都一味的靠近背景颜色。所以我们采用具有压缩趋势的函数对出现次数特别多的颜色给予一定抑制效果。我们先后尝试了对数函数, 指数数为 0.5 的幂函数, 发现指数为 0.5 的幂函数效果较好。

将效益函数归一化为 R_k 用于排名, R_k 具有正向性。

$$R_k = \frac{\max_{1 \leq i \leq N^c} \{\omega_i\} - \omega_k}{\max_{1 \leq i \leq N^c} \{\omega_i\} - \min_{1 \leq i \leq N^c} \{\omega_i\}} \quad (4-3)$$

由于适应度函数变得复杂了, 遗传算法收敛于最优解的难度也随之增大, 因此我们调整了部分参数。

$$N^c = 80$$

$$G = 30 + n \cdot 10$$

n 为添加颜色数

关于斯蒂文斯幂定律的常数选择, 我们取:

$$K=1; r=3$$

因为我们的目的是比较各个刺激值，所以参数 K 大小不影响比较结果可取为 1 简化中间计算，通过查阅资料，估计色差的刺激，我们将 r 值取为 3，在以后模型计算过程中，所得结果与实际符合较好，因此参数所取较为合理。

5.3.4 最优解的进化过程

随着遗传算法的进行，遗传代数不断增多，效益函数值逐步趋向于一个稳定值，即逼近于最优解。基于该模型下，分别添加 1~10 瓷砖，效益函数值的遗传进化过程如下图 3.8 所示。

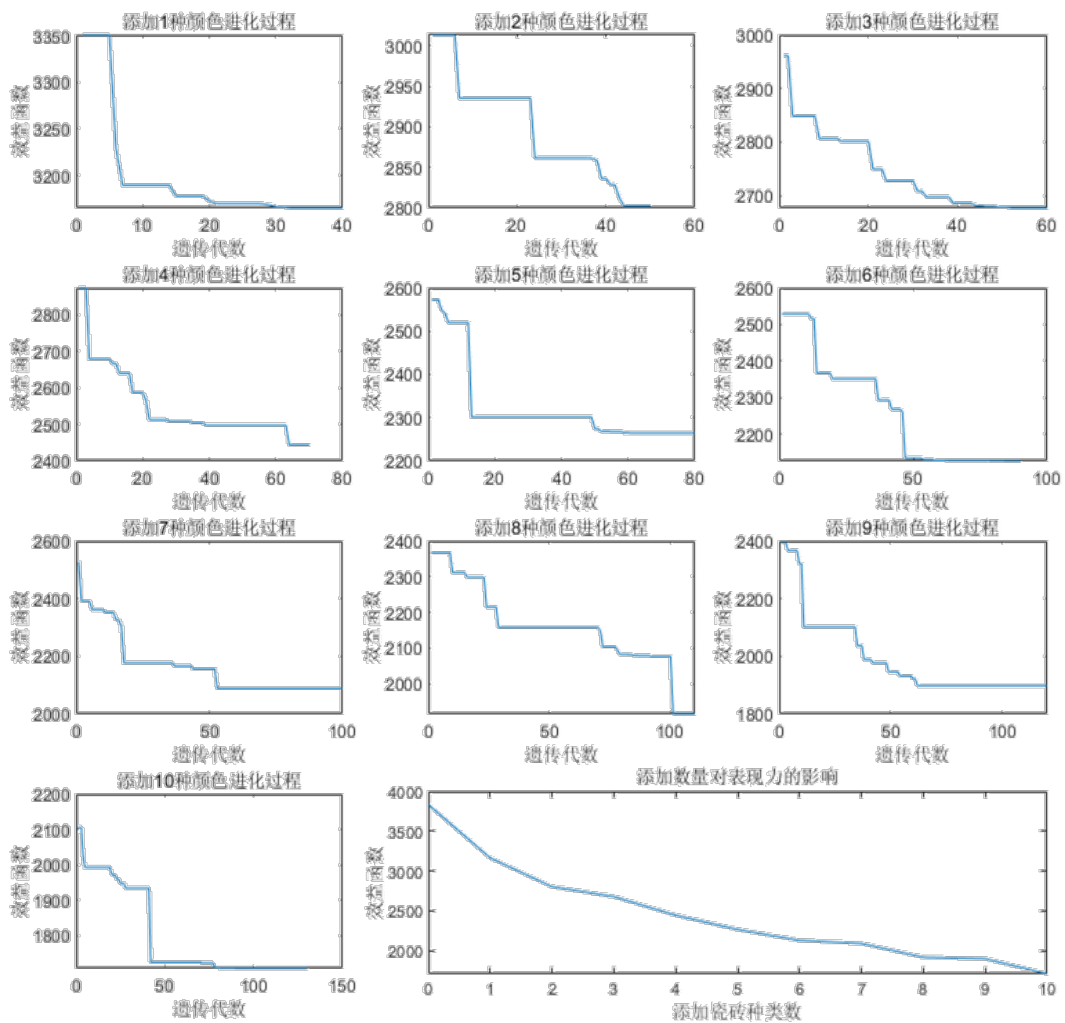


图 3.8 遗传算法最优解的进化过程

5.3.5 二代模型的结果及分析

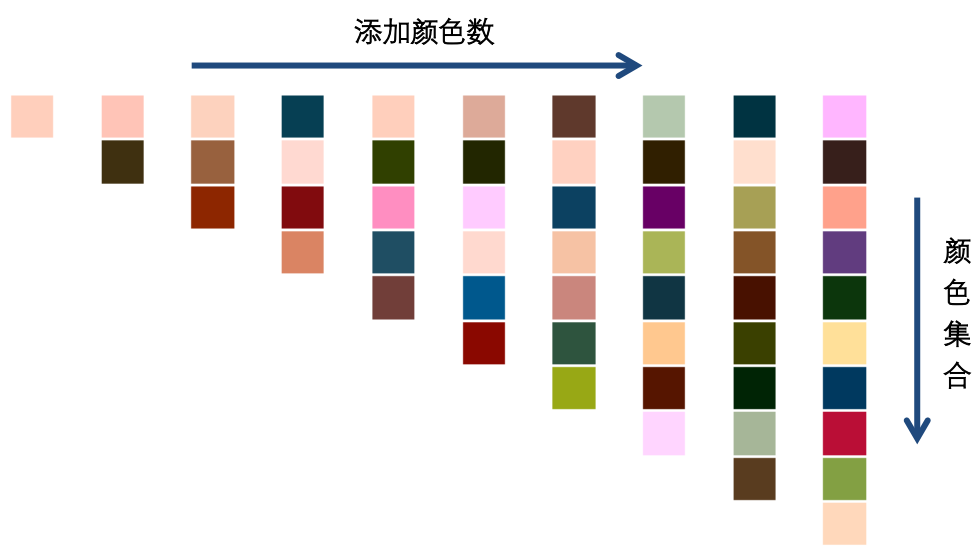


表 3 二代模型结果

添加瓷砖的个数	添加瓷砖的颜色 RGB 值
1	(255,207,188)
2	(255,196,183),(63,48,16)
3	(253,210,190),(152,97,62),(141,38,0)
4	(6,63,83),(255,217,209),(129,11,14),(218,132,99)
5	(255,207,188),(48,64,0),(255,142,193),(31,78,99),(113,62,57)
6	(221,170,153),(34,38,0),(255,203,255), (255,217,207),(0,88,141),(138,8,0)
7	(95,57,44),(255,209,193),(12,65,97),(246,194,164), (202,134,125),(46,84,62),(152,168,21)
8	(180,200,174),(48,31,0),(104,0,101),(170,181,87), (16,53,67),(255,200,143),(86,21,0),(255,213,255)
9	(0,51,65),(255,223,206),(167,160,85),(132,84,40), (72,17,0),(58,64,0),(0,36,5),(166,182,152),(89,60,31)
10	(255,182,255),(55,31,27),(255,161,139),(97,60,127),(12,54,12), (255,224,153),(0,57,95),(186,14,54),(131,160,67),(255,216,187)

相比于一代模型，二代模型的颜色分布显然更合理。添加一种颜色时优先选择常见背景图中的浅背景色。添加多种颜色时有深有浅，分布更平均。

5.4 问题三模型的建立与求解

利用第二问遗传算法求得的增加颜色数与效益函数关系的数据，绘制增加颜色数与效益函数的折线图。

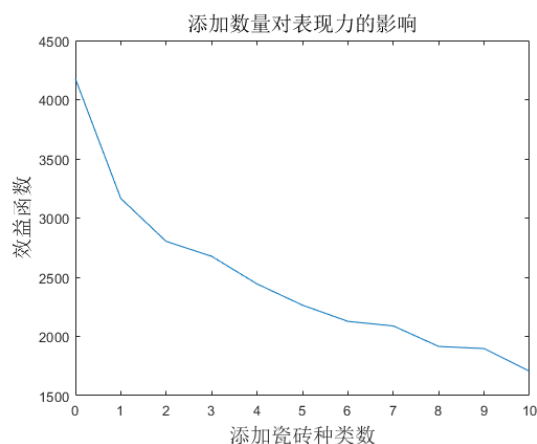


图 4.1 添加数量对表现力的影响

发现该折线图拥有边际效益递减的特征。边际效益是一个经济学概念，是卖主在市场上多投入一单位产量所得到的追加收入与所支付的追加成本的比较。当这种追加收入大于追加成本时，卖主会扩大生产；当这种追加收入等于追加成本时，卖主可以得到最大利润，即达到最大利润点。

为了能更直观的观察边际效益，我们绘制了颜色增加与效益函数减小值关系的台阶图。

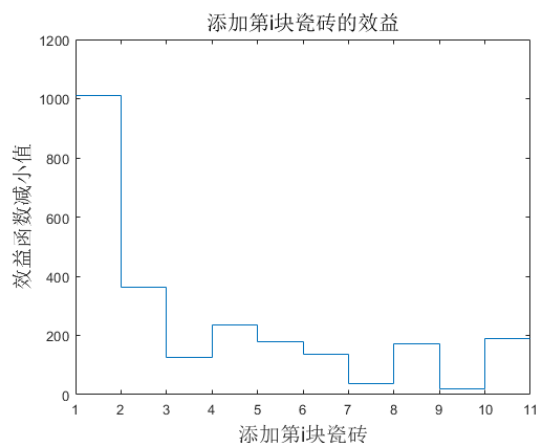


图 4.2 添加第 i 块瓷砖的效益

如需进一步做出决策，还需要做市场调查和成本分析，量化分析瓷砖颜色改进所能带来的经济收益，以及开发新颜色瓷砖带来的经济成本。

在现有资料不足的情况下，分析台阶图后，我们提出几条建议：

- 1) 添加颜色总能带来表现效果的提升，成本充足时可以尽量添加。
- 2) 随着添加颜色的增多，表现效果的提升会逐渐降低。在成本有限的情况下，可以优先考虑添加 1、2 或 4 种颜色，颜色选择如下表（表 4）：

表 4 建议添加颜色

建议瓷砖的个数	添加瓷砖的颜色 RGB 值
1	(255,207,188)
2	(255,196,183),(63,48,16)
4	(6,63,83),(255,217,209),(129,11,14),(218,132,99)
10	(255,182,255),(55,31,27),(255,161,139),(97,60,127),(12,54,12), (255,224,153),(0,57,95),(186,14,54),(131,160,67),(255,216,187)

六、模型的评价

6.1 模型优点分析

1、我们的模型对于匹配最相近颜色的问题，不仅从代数上考虑颜色相近，还综合人眼对于色差的识别和感受，在 LAB 坐标系中，采用目前精度最高的 CIEDE2000 色差公式计算色差，计算出的色差对于人眼视觉感受是均匀的。

2、对于问题二，我们搜索大量具有代表性的有大量用户使用的挂画图片，分析用户对于不同颜色的使用率，对于色差和进行加权，使计算公式更贴近实际使用情况。

3、求解第二问优化问题时采用了遗传算法，通过每代杂交和自然选择求得本问题的最优方案，具有更好的全局优化能力。

4、对于图像表现力，我们利用斯蒂文斯幂定律，考虑了色差对于人的不同刺激，更符合用户使用体验，更适合厂家决策。

6.2 模型缺点分析

1、所选样本挂画相对范围还是较小，没有最准确的用户所偏爱挂画颜色数据，计算可能与实际情况有较小的误差。

2、本模型所用算法较为复杂，对于大型图片的处理效率可能较低。

七、模型的推广

此次马赛克瓷砖选色算法的模型建立与解决有重要的现实意义。可利用此算法模型，逐像素分析各种图片，并在已有颜色集合中寻找适合人视觉的最佳匹配，自动生成马赛克瓷砖图。并能综合考虑表现度和成本给商家合理地建议，减少了人工选取马赛克瓷砖的任务量，并且增加了准确度，便于商家决策。若对模型稍加改进，可以用于图像识别，图像颜色处理等人工智能领域。还可用于色彩科学领域用于分析色彩对于人刺激的影响，可以辅助科学研究。

八、参考文献

- [1]王建刚, 基于 CIELAB 颜色空间的色差公式参数因子优化研究, [D]曲阜师范大学, 2014
- [2]武海丽, 黄庆梅, 苑馨方, 李瑞玲, 基于 S-CIELAB 和 iCAM 模型的图像颜色质量评价方法的实验研究, 北京理工大学光电学院颜色科学与工程国家专业实验室, 2010.12
- [3]Wikipedia, Color difference, 17 April 2021
https://en.wikipedia.org/wiki/Color_difference
- [4]Wikipedia, CIELAB color space, 9 April 2021
https://en.wikipedia.org/wiki/CIELAB_color_space
- [5]Wikipedia, CIE 1931 color space, 9 April 2021
https://en.wikipedia.org/wiki/CIE_1931_color_space
- [6]Wikipedia, Gamut, 17 April 2021
<https://en.wikipedia.org/wiki/Gamut>
- [7]Gaurav Sharma,^{1*} Wencheng Wu,² Edul N. Dalal²,
The CIEDE2000 Color-Difference Formula: Implementation Notes,
Supplementary Test Data, and Mathematical Observations

附录

1.1 题目一输出结果

图像一结果

1,1	60,16	121,10	182,13
2,11	61,5	122,10	183,13
3,11	62,5	123,22	184,13
4,11	63,5	124,11	185,13
5,11	64,12	125,11	186,13
6,11	65,12	126,6	187,3
7,7	66,16	127,5	188,13
8,11	67,15	128,5	189,13
9,11	68,15	129,7	190,13
10,11	69,15	130,16	191,13
11,11	70,12	131,6	192,22
12,11	71,12	132,6	193,17
13,7	72,12	133,5	194,3
14,7	73,1	134,5	195,13
15,6	74,11	135,5	196,13
16,6	75,11	136,12	197,22
17,6	76,11	137,20	198,22
18,6	77,11	138,16	199,9
19,5	78,11	139,15	200,9
20,5	79,1	140,15	201,10
21,7	80,11	141,15	202,8
22,16	81,11	142,15	203,22
23,16	82,11	143,12	204,22
24,16	83,11	144,20	205,14
25,5	84,11	145,3	206,14
26,5	85,7	146,13	207,19
27,5	86,7	147,13	208,18
28,12	87,6	148,11	209,8
29,12	88,6	149,11	210,22
30,16	89,6	150,11	211,4
31,15	90,6	151,3	212,4
32,15	91,5	152,13	213,19
33,15	92,5	153,13	214,19
34,12	93,7	154,13	215,2
35,12	94,16	155,11	216,20
36,12	95,16	156,11	
37,1	96,16	157,10	
38,11	97,5	158,10	
39,11	98,5	159,13	
40,11	99,5	160,22	
41,11	100,12	161,22	
42,11	101,12	162,22	
43,7	102,16	163,10	
44,11	103,15	164,10	
45,11	104,15	165,10	
46,11	105,15	166,22	
47,11	106,15	167,22	
48,11	107,12	168,22	
49,7	108,12	169,4	
50,7	109,13	170,15	
51,6	110,11	171,5	
52,6	111,11	172,18	
53,6	112,11	173,18	
54,6	113,11	174,20	
55,5	114,11	175,15	
56,5	115,10	176,15	
57,7	116,13	177,15	
58,16	117,11	178,15	
59,16	118,11	179,12	
	119,11	180,20	
	120,11	181,3	

图像二输出结果

60,11	1,12	121,15	182,13
61,6	2,15	122,15	183,17
62,15	3,6	123,13	184,19
63,6	4,6	124,22	185,8
64,12	5,5	125,12	186,9
65,6	6,12	126,15	187,8
66,5	7,11	127,18	188,13
67,12	8,7	128,22	189,14
68,7	9,6	129,14	190,4
69,5	10,16	130,10	191,22
70,12	11,6	131,14	192,9
71,1	12,7	132,22	193,17
72,11	13,16	133,11	194,17
73,5	14,15	134,13	195,4
74,11	15,6	135,10	196,8
75,6	16,5	136,10	197,8
76,11	17,12	137,13	198,13
77,7	18,5	138,22	199,13
78,16	19,5	139,3	200,9
79,11	20,5	140,13	
80,11	21,11	141,10	
81,11	22,7	142,18	
82,11	23,5	143,15	
83,12	24,7	144,10	
84,5	25,16	145,20	
85,15	26,5	146,13	
86,7	27,11	147,13	
87,11	28,6	148,22	
88,11	29,6	149,18	
89,16	30,5	150,19	
90,12	31,12	151,10	
91,5	32,5	152,4	
92,11	33,11	153,13	
93,11	34,7	154,3	
94,10	35,6	155,3	
95,11	36,5	156,15	
96,5	37,12	157,13	
97,11	38,5	158,19	
98,15	39,11	159,10	
99,6	40,7	160,13	
100,5	41,7	161,8	
101,11	42,15	162,3	
102,6	43,5	163,13	
103,5	44,11	164,22	
104,5	45,11	165,13	
105,11	46,11	166,9	
106,11	47,7	167,13	
107,10	48,11	168,9	
108,10	49,11	169,19	
109,5	50,6	170,22	
110,15	51,16	171,13	
111,15	52,16	172,17	
112,20	53,16	173,22	
113,5	54,11	174,4	
114,12	55,11	175,8	
115,5	56,7	176,19	
116,11	57,11	177,19	
117,10	58,16	178,19	
118,16	59,11	179,13	
119,11		180,8	
120,12		181,19	

1.2 CIEDE2000 推导公式^[7]

1. Calculate C'_i , h'_i :

$$\begin{aligned} C_{i,ab}^* &= \sqrt{(a_i^*)^2 + (b_i^*)^2} \quad i=1, 2 \\ \bar{C}_{ab}^* &= \frac{C_{1,ab}^* + C_{2,ab}^*}{2} \\ G &= 0.5 \left(1 - \sqrt{\frac{\bar{C}_{ab}^{*7}}{\bar{C}_{ab}^{*7} + 25^7}} \right) \\ a_i' &= (1+G)a_i^* \quad i=1, 2 \\ C_i' &= \sqrt{(a_i')^2 + (b_i^*)^2} \quad i=1, 2 \\ h_i' &= \begin{cases} 0 & b_i^* = a_i' = 0 \\ \tan^{-1}(b_i^* / a_i') & \text{otherwise} \end{cases} \quad i=1, 2 \end{aligned}$$

2. Calculate $\Delta L'$, $\Delta C'$, $\Delta H'$:

$$\begin{aligned} \Delta L' &= L_2^* - L_1^* \\ \Delta C' &= C_2' - C_1' \\ \Delta h' &= \begin{cases} 0 & C_1' C_2' = 0 \\ h_2' - h_1' & C_1' C_2' \neq 0; |h_2' - h_1'| \leq 180^\circ \\ (h_2' - h_1') - 360 & C_1' C_2' \neq 0; (h_2' - h_1') > 180^\circ \\ (h_2' - h_1') + 360 & C_1' C_2' \neq 0; (h_2' - h_1') < -180^\circ \end{cases} \\ \Delta H' &= 2 \sqrt{C_1' C_2'} \sin\left(\frac{\Delta h'}{2}\right) \end{aligned}$$

3. Calculate CIEDE2000 Color-Difference ΔE_{00} :

$$\begin{aligned} \bar{L}' &= (L_1^* + L_2^*) / 2 \\ \bar{C}' &= (C_1' + C_2') / 2 \\ \bar{h}' &= \begin{cases} \frac{h_1' + h_2'}{2} & |h_1' - h_2'| \leq 180^\circ; C_1' C_2' \neq 0 \\ \frac{h_1' + h_2' + 360^\circ}{2} & |h_1' - h_2'| > 180^\circ; (h_1' + h_2') < 360^\circ; \\ & C_1' C_2' \neq 0 \\ \frac{h_1' + h_2' - 360^\circ}{2} & |h_1' - h_2'| > 180^\circ; (h_1' + h_2') \geq 360^\circ; \\ & C_1' C_2' \neq 0 \\ (h_1' + h_2') & C_1' C_2' = 0 \end{cases} \end{aligned}$$

$$\begin{aligned} T &= 1 - 0.17 \cos(\bar{h}' - 30^\circ) + 0.24 \cos(2\bar{h}') \\ &\quad + 0.32 \cos(3\bar{h}' + 6^\circ) - 0.20 \cos(4\bar{h}' - 63^\circ) \end{aligned}$$

$$\Delta\theta = 30 \exp\left\{-\left[\frac{\bar{h}' - 275^\circ}{25}\right]^2\right\}$$

$$R_C = 2 \sqrt{\frac{\bar{C}'^7}{\bar{C}'^7 + 25^7}}$$

$$S_L = 1 + \frac{0.015(\bar{L}' - 50)^2}{\sqrt{20 + (\bar{L}' - 50)^2}}$$

$$S_C = 1 + 0.045 \bar{C}'$$

$$S_H = 1 + 0.015 \bar{C}' T$$

$$R_T = -\sin(2\Delta\theta) R_C$$

$$\begin{aligned} \Delta E_{00}^{12} &= \Delta E_{00}(L_1^*, a_1^*, b_1^*; L_2^*, a_2^*, b_2^*) \\ &= \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2 + R_T \left(\frac{\Delta C'}{k_C S_C}\right) \left(\frac{\Delta H'}{k_H S_H}\right)}. \end{aligned}$$

%% 求解问题一

```
tileColor = uint8([
    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;
    179, 226, 242;
    186, 149, 195;
]);
tilelab = rgb2lab(tileColor);
tileN = size(tileColor, 1);

imagergb = cell(1, 2);
imagelab = cell(1, 2);

imagefile = fopen(".\data\附件2: 图像1颜色列表.txt");
image1 = textscan(imagefile, "%d, (%d, %d, %d)", "HeaderLines", 1);
```



```
fclose(image1file);
imagergb{1} = uint8([image1{2}, image1{3}, image1{4}]);
imagelab{1} = rgb2lab(imagergb{1});

image2file = fopen(".\data\附件3: 图像2颜色列表.txt");
image2 = textscan(image1file, "%d, (%d,%d,%d)", "HeaderLines", 1);
fclose(image2file);
imagergb{2} = uint8([image2{2}, image2{3}, image2{4}]);
imagelab{2} = rgb2lab(imagergb{2});
```

%% 颜色匹配

```
dEs = cell(1,2);
for n = 1:2
    lab = imagelab{n};
    dEs{n} = zeros(tileN, size(lab,1));
    for i = 1:size(lab,1)
        for j = 1:tileN
            dEs{n}(j,i) = imcolordiff(lab(i,:), tilelab(j,:), ✓
"Standard", "CIEDE2000", "isInputLab", true);
        end
    end
end
```

%% 结果分析

```
minde = cell(1,2);
for n = 1:2
    [dE, index] = min(dEs{n});
    colorn = size(imagergb{n},1);
    minde{n} = [dE; index];
```

% 写入文件

```
file = fopen("./data/result"+ num2str(n) + ".txt", "w");
tmpfile = fopen("./data/tmp"+ num2str(n) + ".txt", "w");
fprintf(file, "序号, 瓷砖颜色编号\n");
for i = 1:colorn
    fprintf(file, "%d,%d\n", i, index(i));
    fprintf(tmpfile,("(%d,%d,%d)->(%d,%d,%d)\n", imagergb{n}(i,:), ✓
tileColor(index(i),:));
end
fclose(file);

pixel = zeros(colorn, 2, 3);
for i = 1:colorn
    pixel(i, 1, :) = double(imagergb{n}(i,:))./255;
    pixel(i, 2, :) = double(tileColor(index(i),:))./255;
end

imwrite(pixel, ".\image\image" + num2str(n) + ".png");
end

%% 拟合效果分析
tilematchn = {zeros(1,tileN), zeros(1,tileN)};
for n = 1:2
    for i = 1:size(minde{n},2)
        tilematchn{n}(minde{n}(2,i)) = tilematchn{n}(minde{n}(2,i)) + ✓
1;
    end
end

%% 色散点图

for n = 1:2
    figure;
    scatter3(imagergb{n}(:,1), imagergb{n}(:,2), imagergb{n}(:,3), 36, ✓
```

```
double(imagergb{n})./255, 'filled');
    hold on
    scatter3(tileColor(:,1),tileColor(:,2),tileColor(:,3),56, double✓
(tileColor)./255, '+');
    scatter3(tileColor(:,1),tileColor(:,2),tileColor(:,3),56, double✓
(tileColor)./255, 'o');
%     for i = 1:size(imagergb{n},1)
%         plot3([imagergb{n}(i,1),tileColor(minde{n}(2,i),1)],...
%             [imagergb{n}(i,2),tileColor(minde{n}(2,i),2)],...
%             [imagergb{n}(i,3),tileColor(minde{n}(2,i),3)], 'color', ✓
[195 195 195]./255)
%     end
end

for n = 1:2
    figure;
    scatter3(imagelab{n}(:,1),imagelab{n}(:,2),imagelab{n}(:,3), 36, ✓
double(imagergb{n})./255, 'filled');
    hold on
    scatter3(tilelab(:,1),tilelab(:,2),tilelab(:,3),56, double✓
(tileColor)./255, '+');
    scatter3(tilelab(:,1),tilelab(:,2),tilelab(:,3),56, double✓
(tileColor)./255, 'o');
%     for i = 1:size(imagergb{n},1)
%         plot3([imagergb{n}(i,1),tileColor(minde{n}(2,i),1)],...
%             [imagergb{n}(i,2),tileColor(minde{n}(2,i),2)],...
%             [imagergb{n}(i,3),tileColor(minde{n}(2,i),3)], 'color', ✓
[195 195 195]./255)
%     end
end

%% 色域图
for n = 1:2
```

```
figure;
hold on
cielabplot();
%     for i = 1:size(imagelab{n},1)
%         plot([imagelab{n}(i,2),tilelab(minde{n}(2,i),2)],...
%             [imagelab{n}(i,3),tilelab(minde{n}(2,i),3)],...
%             'color', [195 195 195]./255, 'linewidth', 0.01)
%     end

    for i = 1:size(imagelab{n},1)
        scatter(imagelab{n}(i,2), imagelab{n}(i,3), 15, double(
(tileColor(minde{n}(2,i),:)) ./ 255, 'filled')
    end
    scatter(tilelab(:,2),tilelab(:,3), 52, double(tileColor)./255 ,
'*');
end

%% 示例图片处理

for n = 1:3
    sample = imread(".\image\sample (" + num2str(n) + ").bmp");
    samplelab = rgb2lab(sample);
    output = zeros(size(samplelab));
    for i = 1 : size(samplelab,1)
        for j = 1:size(samplelab,2)
            de = zeros(1,tileN);
            for k = 1:tileN
                % de(k) = imcolordiff(reshape(sample(i,j,:),1,3),
tilelab(k,:), "Standard", "CIEDE2000", "isInputLab", true);
                % de(k) = Euclid(reshape(sample(i,j,:),1,3), tileColor
```

```
(k,:));  
        de(k) = RedMean(reshape(sample(i,j,:),1,3), tileColor);  
(k,:));  
    end  
    [~,index] = min(de);  
    output(i,j,:) = double(tileColor(index,:))./255;  
end  
end  
% imwrite(output, ".\image\sample (" + num2str(n) + ")out.bmp");  
% imwrite(output, ".\image\sample (" + num2str(n) + ")out2.bmp");  
imwrite(output, ".\image\sample (" + num2str(n) + ")out3.bmp");  
end
```

```
function d = RedMean(a, b)
% REDMEAN色差函数
a = double(a);
b = double(b);
r = (a(1) + b(1)) / 2;
d = sqrt((2+r/256)*(a(1) - b(1))^2 + ...
    4*(a(2) - b(2))^2 + ...
    (2+(255-r)/256)*(a(3) - b(3))^2);
d = uint8(d);
```

```
function d = Euclid(a,b)
% 欧几里得距离色差公式
a = double(a);
b = double(b);
d = sqrt(sum((a-b).^2));
d = uint8(d);
```

% 一代模型 遗传算法

%% 导入数据

```
mindeall = [minde{1},minde{2}];  
imagelaball = [imagelab{1}; imagelab{2}];
```

%% 遗传算法

```
NIND=50;           %个体数目  
ELITERATE = 0.05;  %精英率  
MAXGEN=30;         %最大遗传代数  
GGAP=0.75;        %代沟  
pm=0.05;          %变异概率
```

```
range = [max(imagelaball); min(imagelaball)];
```

%% 计算

```
for pointn = 1:10  
    if pointn > 3  
        MAXGEN = 30 + (pointn-3) * 5;  
    else  
        MAXGEN = 30;  
    end
```

% 种群初始化

```
chrom = zeros(NIND, 3*pointn);  
for n = 1:pointn  
    for i = 1:3  
        chrom(:, 3*(n-1)+i) = range(2, i) + (range(1, i) - range(2, i)) * rand(NIND, 1);  
    end  
end
```


% 迭代

```
bestsum = zeros(1, MAXGEN);
bestlab = zeros(MAXGEN, 3*pointn);
rankhistory = cell(1, MAXGEN);
chromhistory = cell(1, MAXGEN);
for gen = 1:MAXGEN
    [rank, bestdesum] = GaRanking(imagelaball, mindeall, chrom, ✓
pointn);
    [~, index] = max(rank);
    bestlab(gen, :) = chrom(index, :);
    rankhistory{gen} = rank;
    select = GaSelect(rank, GGAP, chrom);
    combine = GaCombine(select, NIND, pointn);
    mut = GaMut(combine, pm, range, ELITERATE, pointn);
    chrom = mut;
    chromhistory{gen} = chrom;
    bestsum(gen) = bestdesum;
end
```

% 保存数据 后续使用

```
save ("p2," + num2str(NIND) + ", " + num2str(MAXGEN) + ", " + ✓
num2str(pointn));
end
```

%% 绘图

```
figure;
plot(1:MAXGEN, bestsum);
title("进化过程", "Fontsize", 15);
```

```
xlabel("遗传代数", "Fontsize", 15);  
ylabel("色差和", "Fontsize", 15);
```

```
%% 新瓷砖绘图
```

```
for n = 1:10  
    figure;  
    scatter3(tilelab(:,1),tilelab(:,2),tilelab(:,3),56, double  
(tileColor)./255, 'o');  
    hold on;  
    for i = 1:size(BESTchrom{n},2)/3  
        scatter3(BESTchrom{n}((i-1)*3 + 1), BESTchrom{n}((i-1)*3 + 2), ✓  
BESTchrom{n}((i-1)*3 + 3), 80, lab2rgb(BESTchrom{n}((i-1)*3 + 1:(i-1)✓  
*3 + 3)), 'filled');  
    end  
    title("添加瓷砖", "Fontsize", 15);  
end
```

```
function [rank, bestdesum] = GaRanking(imagelaball, mindeall, chrom, ✓  
pointn)  
% 适应度评分, 0~1越大越好  
rank = zeros(size(chrom,1),1);  
de = zeros(pointn,1);  
for n = 1:size(chrom,1)  
    tmpsum = 0;  
    for i = 1:size(imagelaball,1) % a  
        for k = 1:pointn  
            de(k) = imcolordiff(imagelaball(i,:), chrom(n, (k-1)*3+1:✓  
(k-1)*3+3), "Standard", "CIEDE2000", "isInputLab", true);  
        end  
  
        t = min(de);  
        if t < mindeall(1,i)  
            tmpsum = tmpsum + t^2;  
        else  
            tmpsum = tmpsum + mindeall(1,i)^2;  
        end  
    end  
    rank(n) = tmpsum;  
end  
bestdesum = min(rank);  
  
if max(rank) - min(rank) == 0  
    for i = 1:size(rank)  
        rank(i) = 0.5;  
    end  
else  
    rank = (max(rank) - rank)/(max(rank)-min(rank));  
end
```

```
function select = GaSelect(rank, gap, chrom)
```

```
N = floor(size(rank, 1) * gap);
```

```
% topN
```

```
[~, index] = sort(rank, 'descend');
```

```
select = chrom(index(1:N), :);
```

```
function combine = GaCombine(select, NIND, pointn)
N = size(select,1);
combine = [select;zeros(NIND-N,size(select,2))];
for i = N+1:NIND
    p = rand();
    combine(i,:) = select(randi([1,N],1),:) * p + select(randi([1,N],1),:)* (1-p);
end
```

```
function mut = GaMut(combine, pm, range, ELITERATE, pointn)

mut = combine;
for i = ceil(size(combine,1)*ELITERATE):size(combine,1)
    if rand() < pm
        for j = 1:pointn
            k = randi([1,3],1);
            mut(i,3*(j-1) + k) = range(2,k) + (range(1,k) - range(2, ✓
k)) * rand();
        end
    end
end
```

% 二代模型改进, 示例背景图处理

%% 均匀分布的目标颜色

```
colormarklab = zeros(8*8*8, 3);
```

```
i = 1;
```

```
for l = linspace(0,100,8)
```

```
    for a = linspace(-128,127,8)
```

```
        for b = linspace(-128,127,8)
```

```
            colormarklab(i,:) = [1,a,b];
```

```
            i = i + 1;
```

```
        end
```

```
    end
```

```
end
```

```
colormarkrgb = lab2rgb(colormarklab);
```

%% 散点图

```
figure;
```

```
cielabplot();
```

```
scatter(colormarklab(256:320,2), colormarklab(256:320,3), 36, ✓
```

```
colormarkrgb(256:320,:), 'filled');
```

```
figure;
```

```
scatter3(colormarklab(:,1), colormarklab(:,2), colormarklab(:,3), 36, ✓
```

```
colormarkrgb, 'filled' )
```

```
xlabel("L");
```

```
ylabel("A");
```

```
zlabel("B");
```

%% 35副代表图

```
colormarkcount = zeros(size(colormarklab,1),1);

PICN = 35;
for i = 1:PICN
    picrgb = imread(".\image\pic\" + num2str(i) + ".PNG");
    picrgb = imresize(picrgb, 0.4, 'nearest');
    %figure;
    %imshow(picrgb)

    piclab3 = rgb2lab(picrgb);
    piclab = zeros(size(piclab3,1)*size(piclab3,2),3);
    for k1 = 1:size(piclab3,1)
        for k2 = 1:size(piclab3,2)
            piclab((k1-1)*size(piclab3,2) + k2, :) = piclab3(k1,k2,:);
        end
    end

    de = zeros(size(colormarklab,1),1);
    for k = 1:size(piclab,1)
        for j = 1:size(colormarklab,1)
            de(j) = imcolordiff(colormarklab(j,:), piclab(k,:), ✓
"Standard", "CIEDE2000", "isInputLab", true);
        end
        [~,index] = min(de);
        colormarkcount(index) = colormarkcount(index)+1;
    end
end
save("colormark2")

%% 绘制分布密度
figure;
scatter3(colormarklab(:,1), colormarklab(:,2), colormarklab(:,3), ✓
colormarkcount./max(colormarkcount) * 100 + 1, colormarkrgb, 'filled')
```



```
frecolor = colormark(colormarkcount > 10, :);
frecolorrgb = colormarkrgb(colormarkcount > 10, :);
frecolorcount = colormarkcount(colormarkcount > 10);

figure;
scatter3(frecolor(:,1), frecolor(:,2), frecolor(:,3), frecolorcount.✓
/max(colormarkcount) * 1000, frecolorrgb, 'filled')

title("常见挂画色彩分布图", 'FontSize', 15)
xlabel("L");
ylabel("A");
zlabel("B");

%% 拼图

allpic = [];
for i = 1:5
    tpic = [];
    for j = 1:7
        picrgb = imread(".\image\pic\" + num2str((i-1)*7+j) + ".PNG");
        %picrgb = imresize(picrgb, 0.7, 'nearest');
        tpic = [tpic, picrgb];
    end
    allpic = [allpic;tpic];
end
imwrite(allpic, ".\image\pic\all.PNG")
```

% 二代模型 遗传算法

%% 参数

NIND=80; %个体数目
ELITRATE = 0.05; %精英率
MAXGEN=30; %最大遗传代数
GGAP=0.75; %代沟
pm=0.05; %变异概率

range = [max(frecolor); min(frecolor)];

%% 权重函数

weight = sqrt(frecolorcount)./sum(sqrt(frecolorcount));
% weight = log(frecolorcount)./sum(log(frecolorcount));

%% 计算

for pointn = 1:10

 MAXGEN = MAXGEN + 10;

 % 种群初始化

 chrom = zeros(NIND, 3*pointn);

 for n = 1:pointn

 for i = 1:3

 chrom(:, 3*(n-1)+i) = range(2, i) + (range(1, i) - range(2, ✓
i)) * rand(NIND, 1);

 end

 end

 % 迭代

 bestsum = zeros(1, MAXGEN);

 bestlab = zeros(MAXGEN, 3*pointn);

 rankhistory = cell(1, MAXGEN);

```
chromhistory = cell(1,MAXGEN);
for gen = 1:MAXGEN
    [rank, bestdesum] = GaRanking2(frecolor, weight, minE, chrom, ✓
pointn, 3);
    [~, index] = max(rank);
    bestlab(gen,:) = chrom(index,:);
    rankhistory{gen} = rank;
    select = GaSelect(rank, GGAP, chrom);
    combine = GaCombine(select, NIND, pointn);
    mut = GaMut(combine, pm, range, ELITERATE, pointn);
    chrom = mut;
    chromhistory{gen} = chrom;
    bestsum(gen) = bestdesum;
end

% 保存数据 后续使用
save ("p2_ln_3," + num2str(NIND) + "," + num2str(MAXGEN) + "," + ✓
num2str(pointn));
end

%% 绘图
figure;
plot(1:MAXGEN, bestsum);
title("进化过程", "Fontsize", 15);
xlabel("遗传代数", "Fontsize", 15);
ylabel("色差和", "Fontsize", 15);
```

```
function [rank, bestdesum] = GaRanking2(imagelaball, weights, ✓  
mindeall, chrom, pointn, StevensR)  
% 二代模型适应度函数  
rank = zeros(size(chrom,1),1);  
de = zeros(pointn,1);  
for n = 1:size(chrom,1)  
    tmpsum = 0;  
    for i = 1:size(imagelaball,1)  
        for k = 1:pointn  
            de(k) = imcolordiff(imagelaball(i,:), chrom(n, (k-1)*3+1:✓  
(k-1)*3+3), "Standard", "CIEDE2000", "isInputLab", true);  
        end  
  
        t = min(de);  
        if t < mindeall(1,i)  
            tmpsum = tmpsum + t^StevensR * weights(i);  
        else  
            tmpsum = tmpsum + mindeall(1,i)^StevensR * weights(i);  
        end  
    end  
    rank(n) = tmpsum;  
end  
bestdesum = min(rank);  
  
if max(rank) - min(rank) == 0  
    for i = 1:size(rank)  
        rank(i) = 0.5;  
    end  
else  
    rank = (max(rank) - rank)/(max(rank)-min(rank));  
end
```

% 遗传算法结果分析

%% 初始化

BESTchrom = cell(10,1);

BESTdesum = zeros(10,1);

%% 添加0数据

tmpsum = 0;

for i = 1:size(frecolor,1)

 %tmpsum = tmpsum + mindeall(1,i)^2;

 tmpsum = tmpsum + minde(1,i)^3 * weight(i);

end

BESTdesum = [tmpsum; BESTdesum];

%% 加载数据

files = [

 "p2_ln_3, 80, 40, 1. mat";

 "p2_ln_3, 80, 50, 2. mat";

 "p2_ln_3, 80, 60, 3. mat";

 "p2_ln_3, 80, 70, 4. mat";

 "p2_ln_3, 80, 80, 5. mat";

 "p2_ln_3, 80, 90, 6. mat";

 "p2_ln_3, 80, 100, 7. mat";

 "p2_ln_3, 80, 110, 8. mat";

 "p2_ln_3, 80, 120, 9. mat";

 "p2_ln_3, 80, 130, 10. mat";

];

files2 = [

 "p2_sqrt_3, 80, 40, 1. mat";

 "p2_sqrt_3, 80, 50, 2. mat";

 "p2_sqrt_3, 80, 60, 3. mat";

```
"p2_sqrt_3, 80, 70, 4. mat";
"p2_sqrt_3, 80, 80, 5. mat";
"p2_sqrt_3, 80, 90, 6. mat";
"p2_sqrt_3, 80, 100, 7. mat";
"p2_sqrt_3, 80, 110, 8. mat";
"p2_sqrt_3, 80, 120, 9. mat";
"p2_sqrt_3, 80, 130, 10. mat";

];

for i = 1:10
    load(files2(i,:), "bestlab", "bestdesum");
    BESTchrom{i} = bestlab(end,:);
    BESTdesum(i + 1) = bestdesum;
end

%% 绘图
figure;

plot(0:10, BESTdesum(1:11))
title("添加数量对表现力的影响", "Fontsize", 15);
xlabel("添加瓷砖种类数", "Fontsize", 15);
ylabel("效益函数", "Fontsize", 15);
hold on

% subplot(1, 3, 1);
% P = polyfit(0:10, BESTdesum(1:11), 3);
% x = 0:0.01:10;
% y = polyval(P, x);
% plot(x, y, '-')
%
% subplot(1, 3, 2);
```

```
% Pd = polyder(P);
% plot(x, polyval(Pd,x), 'k')
%
% subplot(1,3,3);
% Pdd = polyder(Pd);
% plot(x, polyval(Pdd,x), 'k')
figure
Y = BESTdesum(1:end-1) - BESTdesum(2:end);
Y = [Y;Y(end)];
stairs(Y)

title("添加第i块瓷砖的效益", "FontSize", 15);
xlabel("添加第i块瓷砖", "FontSize", 15);
ylabel("效益函数减小值", "FontSize", 15);

%% 统一数据
labs = cell(10,1);
rgbs = cell(10,1);
for i = 1:10
    c = BESTchrom{i};
    labs{i} = zeros(i, 3);
    for j = 1:i
        labs{i}(j,:) = c((j-1)*3 + 1:(j-1)*3 + 3);
    end
    rgbs{i} = lab2rgb(labs{i}, 'OutputType', 'uint8');
end

%% 结果lab2rgb写入文件
f = fopen(".\data\p2ln.txt", "w");
for i = 1:10
    for j = 1:i
        fprintf(f, "(%d,%d,%d),", rgbs{i}(j,1), rgbs{i}(j,2), rgbs{i}(j,3))
```

```
(j,3));  
    end  
    fprintf(f, "\n");  
end  
fclose(f);  
  
%% 结果色卡  
resimg = ones(20,10,3) * 255;  
for i = 1:10  
    for j = 1:i  
        resimg(i*2, j, :) = rgbbs{i}(j, :);  
    end  
end  
resimg2 = imresize(resimg, 50, 'nearest');  
  
for i = 1:1000  
    for j = 1:500  
        if mod(i,50)==0 || mod(i,50)==1 || mod(i,50)==49 || mod(j,50)✓  
==0 || mod(j,50)==1 || mod(j,50)==49  
            resimg2(i, j, :) = [255,255,255];  
        end  
    end  
end  
  
imshow(resimg2./255);  
imwrite(resimg2./255, ".\image\p2res2.png");  
  
%% 10张遗传收敛图  
  
figure;  
subp = 1;
```


%% 手动载入数据

```
subplot(4, 3, subp);  
plot(1:MAXGEN, bestsum);  
title("添加"+num2str(subp)+"种颜色进化过程", "Fontsize", 10);  
xlabel("遗传代数", "Fontsize", 10);  
ylabel("效益函数", "Fontsize", 10);
```

```
subp = subp + 1;
```

%%

```
subplot(4, 3, [11, 12])  
plot(0:10, BESTdesum(1:11))  
title("添加数量对表现力的影响", "Fontsize", 10);  
xlabel("添加瓷砖种类数", "Fontsize", 10);  
ylabel("效益函数", "Fontsize", 10);
```