

## 代表性调色板提取及图像重着色

夏子勋<sup>1)</sup>, 杜正君<sup>1,2)\*</sup>, 刘晓静<sup>1)</sup>

(青海大学计算机技术与应用系 西宁 810016)  
(清华大学计算机科学与技术系 北京 100084)  
(dzj@qhu.edu.cn)

**摘要:** 基于调色板的图像编辑技术是近年的热门研究方向, 在海报制作、服装设计、电影、短视频编辑等方向有着广泛的应用. 通过计算凸包提取图像调色板是一个较新技术. 然而, 其目前仍然存在 2 个问题: 一是忽略了凸包内部的颜色分布, 使得调色板整体上缺乏代表性; 二是插值权重缺乏稀疏性, 难以实现针对性的局部编辑. 针对上述问题, 提出一种新颖的调色板提取算法, 并在此基础上实现高效的图像重着色编辑. 首先, 提取图像在 RGB 空间的凸包并简化; 其次, 通过聚类算法捕捉凸包内部的颜色分布, 构造代表性调色板; 最后, 在 RGB 空间对调色板颜色进行四面体剖分, 并对图像像素进行插值. 为了验证该算法的有效性, 从互联网上获取了 40 余幅图像组成数据集进行实验, 对插值权重的稀疏性、图像重着色效果等进行了对比分析和用户调研. 大量实验结果表明, 该算法提取的调色板具有更好的代表性, 插值权重具有更好的稀疏性, 实现了更精确的局部编辑.

**关键词:** 调色板; 重着色; 凸包; 代表性; 空间剖分

中图分类号: TP391.41 DOI: 10.3724/SP.J.1089.2023.19430

## Representative Palette Extraction and Image Recoloring

Xia Zixun<sup>1)</sup>, Du Zhengjun<sup>1,2)\*</sup>, and Liu Xiaojing<sup>1)</sup>

(Department of Computer Technology and Application, Qinghai University, Xining 810016)  
(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

**Abstract:** Palette-based image recoloring has recently drawn much attention, which is widely used in poster making, fashion design, film, short video editing, etc. The state-of-the-art method extracts the image palette by calculating the convex hull. However, there are two main problems in this technique: one is that the color distribution inside the convex hull is always ignored, resulting in less representative palettes; the other is that the interpolation weights lack sparsity, which makes it difficult to achieve targeted local editing. This paper proposes a novel palette extraction algorithm to address these problems. Firstly, calculate the convex hull of the image in the RGB space and simplify it. Secondly, extract the color distribution inside the convex hull to build the representative color palette. Finally, perform tetrahedral subdivision of palette colors in RGB space and interpolation of image pixels. More than 40 images are obtained from the Internet to evaluate the proposed algorithm's effectiveness, comparative analysis and user study on interpolation sparsity and recoloring effects are also presented. Experiments show that the palette extracted by the proposed algorithm is more representative, the interpolation weights are sparser and achieve more accurate local recoloring than state-of-the-art methods.

收稿日期: 2021-10-31; 修回日期: 2022-02-16. 基金项目: 青海省自然科学基金青年项目(2023-ZJ-951Q), 青海省科技厅应用基础研究项目(2021-ZJ-717), 国家自然科学基金(61863031, 61862053). 夏子勋(1999—), 男, 硕士研究生, CCF 学生会会员, 主要研究方向为图像视频处理; 杜正君(1989—), 男, 博士研究生, 讲师, 硕士生导师, CCF 会员, 论文通信作者, 主要研究方向为计算机图形学、图像视频处理; 刘晓静(1978—), 女, 硕士, 教授, 硕士生导师, CCF 会员, 主要研究方向为图像视频处理、非物质文化遗产保护.

**Key words:** palette; recoloring; convex hull; representative; spatially partition

基于调色板的图像编辑是当前热门的研究方向, 广泛应用于电影、艺术创作、广告设计、短视频编辑等多个方向. 这类方法操作简单、计算高效, 用户只需要对调色板进行简单的操作就可以自然地修改图像的颜色.

为了实现高效、简洁的图像编辑, 调色板的代表性尤为重要, 代表性较强的调色板能够直观地反映图像的颜色分布, 进一步辅助用户有针对性地调整颜色; 反之, 代表性较差的调色板往往导致复杂、烦琐的操作, 一定程度上加重了用户的编辑负担. 目前, 针对图像调色板的提取主要有 2 类方法, 即基于聚类的方法和基于几何凸包的方法. 其中, 基于聚类的方法提取的调色板完全来自于图像像素本身, 对于本身需要多种颜料混合的图像(如油画)显得力不从心, 这是因为代表性颜色来自于几种绘画颜料而非图像本身. 基于几何凸包的方法提取的调色板可以天然地提取构成图像的基色, 并且在计算效率上和重着色效果上更胜一筹, 目前已成为图像重着色领域的优秀方法.

基于几何凸包的方法通过计算图像在 RGB 空间的简化凸包来构建调色板, 并通过重心坐标插值的方法将图像像素自然地表示为调色板颜色的线性组合, 实现了直观、高效的图像编辑. 但该方法目前仍存在 2 个主要的缺陷亟待进一步改进.

(1) 图像中某些代表性颜色可能位于凸包内部而被忽略, 使得提取的调色板未能反映图像的主要颜色分布. 如果用户需要对这些颜色进行编辑, 那么不得不对大量的调色板颜色进行修改, 增加了用户的使用成本.

(2) 基于几何凸包方法的插值权重不够稀疏, 即大部分像素点关于所有调色板颜色的插值权重均大于 0. 修改任意调色板颜色可能导致大量像素的颜色发生变化, 无法实现精确的局部编辑.

针对问题 1, 受图像变形技术的启发, 本文在几何凸包方法的基础上对图像进行聚类, 以捕捉凸包内部的代表性颜色, 形成最终的代表性调色板. 针对问题 2, 本文对提取的简化凸包顶点和聚类中心进行四面体剖分, 并在每一个四面体内部对像素进行插值. 因此, 每个像素点仅与所在的四面体的 4 个顶点相关, 极大地提升了插值权重的稀疏性.

## 1 相关工作

### 1.1 基于调色板的图像编辑

调色板在相关文献中主要存在 3 种定义: (1) 调色板表示图像中的代表性颜色, 这些颜色大致反映了图像整体的颜色分布; (2) 从传统的油画艺术出发, 认为调色板包含固定数量的颜色用以调和生成五彩斑斓的图像; (3) 将颜色查找表定义为调色板, 使用较少的颜色还原高质量图像. 第 3 种定义主要用于在有限的颜色空间内对图像颜色进行近似和压缩. 本文涉及算法主要面向前 2 种定义: 第 1 种定义一般通过聚类算法提取图像的代表性颜色; 第 2 种定义一般通过求解几何凸包获取图像调色板.

Chang 等<sup>[1]</sup>率先使用改进的 *K*-Means 算法捕获图像的代表性颜色作为调色板. 该算法首先将图像投影到 RGB 空间, 并将像素值归一化到 [0,1], 再将 RGB 空间划分为  $16 \times 16 \times 16$  的网格, 且在每个网格选取一个代表性的像素后, 对这些代表性像素进行聚类, 得到固定数目的聚类中心作为图像调色板. 然后, 使用径向基函数(radial basis function, RBF)插值将调色板的颜色变化映射到图像的颜色变化上. 除此之外, 该算法实现了亮度感知的图像重着色, 使得图像编辑前后的相对亮度保持不变. Zhang 等<sup>[2-3]</sup>对 Chang 等<sup>[1]</sup>的方法进行了加速, 并使用能量优化的方法求解图像像素关于聚类中心的插值权重.

Tan 等<sup>[4-6]</sup>受到传统绘画艺术的启发, 认为图像的像素颜色可以通过多种基本颜色混合而成, 故将图像在 RGB 空间中的简化凸包作为调色板. 凸包中的像素点可以通过均值坐标插值(mean value coordinates, MVC)<sup>[7]</sup>等广义重心坐标实现精确、快速的插值, 即将图像像素表示为凸包顶点(调色板颜色)的凸组合. 该方法与几何处理中的自由变形<sup>[8-12]</sup>技术相似, 通过移动控制点实现网格变形. 在图像编辑任务中, 用户只需要对调色板进行简单的修改即可实现直观的图像重着色. 但该方法计算的插值权重往往稀疏性(局部性)较差, 修改局部的颜色往往导致图像整体出现不自然的色彩变化. 为了解决此问题, Wang 等<sup>[13]</sup>通过顶点优化的方法对 Tan 等<sup>[4]</sup>的调色板进行了改进, 其对几何凸包施加了 2 个约束条件, 即重建约束和代表性约束. 其中, 重建约束要求凸包能够尽可能地完全包

裹图像像素,从而实现精确的重建;代表性约束要求凸包紧致地包裹像素,使得凸包顶点尽可能地接近图像像素,以此实现良好的代表性和局部性。但二者是相互矛盾的,因此,其总的损失定义为二者的加权和。最后,逐个地移动凸包顶点,在降低损失的同时,实现调色板重建误差和代表性误差的平衡。该方法一定程度上提升了调色板的代表性和插值权重的稀疏性,但针对颜色数量较少的调色板却无法从根本上同时提升代表性和局部性。Du 等<sup>[14]</sup>进一步将基于几何凸包的调色板提取方法扩展到视频场景,通过构建 RGBT 空间的简化多面体生成视频调色板,实现了自然的颜色渐变效果。

综上所述,基于凸包的方法在图像颜色复杂时容易忽略位于凸包内部的代表性颜色。基于凸包方法提取的图像调色板如图 1 所示,可以看到图像调色板包含了前景中的 7 种主要颜色,但忽略了位于凸包内部的背景灰色,使得用户对于背景部分的颜色编辑变得困难。

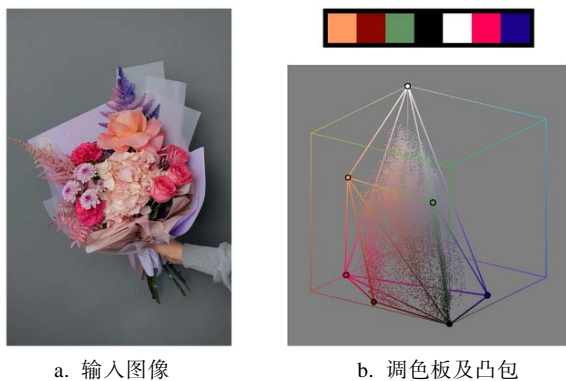


图 1 基于凸包方法提取的图像调色板

深度学习技术在图像重着色中同样得到了广泛的应用。Cho 等<sup>[15]</sup>提出了一种内容感知的自动重着色的深度学习框架 PaletteNet,其输入包括原始图像和目标调色板。对于给定的图像和对应的调色板实现了自然的颜色主题变化。然而,该框架仅支持用户提供 6 种主题颜色的调色板,并且无法实现针对性的局部编辑。Akimoto 等<sup>[16]</sup>将深度学习应用到插值权重估计任务中,对于给定的图像和调色板,通过 U-Net 自动地预测图像像素关于调色板颜色的混合权重,但其估计的权重仍然缺乏良好的局部性。

基于深度学习的方法往往需要用户提供调色板,而对于普通用户而言手动提取图像调色板并非易事,且其难以实现局部性良好的图像编辑。本文在几何凸包的基础上进一步引入聚类算法,自

动地提取图像的主要基础颜色和代表性颜色作为图像的调色板,并在此基础上通过简单的插值实现高效、自然的图像编辑任务。

## 1.2 基于编辑传播和风格迁移的图像编辑

编辑传播<sup>[17-19]</sup>是常见的颜色编辑算法,旨在将局部的颜色编辑传播到几何、纹理相似的其他区域上。然而,该算法往往需要用户绘制大量的笔触,加重了用户编辑图像的负担;同时,它往往将编辑传播形式化为二次能量函数优化问题,因此,计算开销较大,往往需要数十分钟才能完成一次编辑任务。与编辑传播算法相比,本文算法通过修改调色板颜色驱动图像颜色编辑,用户交互简单、直观,且可以实时反馈图像编辑效果。

风格迁移<sup>[20-22]</sup>是实现图像重着色的另一种经典算法,它需要用户提供一幅参考图像,目标图像自动学习参考图像的纹理特征,在参考图像和目标图像之间建立全局映射,并将这种颜色特征全自动地迁移到目标图像上。与风格迁移算法相比,基于调色板的图像编辑算法为用户提供了更多的自由度,用户可以根据实际需求随意地修改调色板,以实现特定的编辑目标,而风格迁移不需要任何用户交互。

## 2 本文算法

本节描述代表性调色板的构建以及基于调色板的图像编辑,如图 2 所示,本文算法包括以下 3 个步骤:

Step1. 将图像投影到 RGB 空间,并计算对应的简化凸包得到初始调色板。

Step2. 在 RGB 空间中对图像进行聚类,得到凸包内部的代表性颜色。

Step3. 对凸包顶点和聚类中心进行四面体剖分,得到代表性调色板;使用重心坐标对四面体内部的图像像素进行插值,从而提升插值权重的稀疏性。

### 2.1 构建初始调色板

本文使用 Tan 等<sup>[4]</sup>提出的算法构建初始调色板,将输入图像视为 RGB 空间的点集,计算该点集的三维凸包,并将凸包顶点作为图像调色板。该算法计算的原始凸包往往包含成百上千个顶点,无法直接作为调色板对图像进行编辑,因此,使用缩边的方法<sup>[4]</sup>对原始凸包进一步简化。

凸包简化示意图如图 3 所示,其中试图将边  $(v_i, v_j)$  缩减为一个顶点  $v$ 。具体而言,需要在凸包外部寻找一个顶点  $v$ ,使得  $v$  连接  $(v_i, v_j)$  的邻接点所构成的新凸包向外扩张并完全包裹  $(v_i, v_j)$ ,则

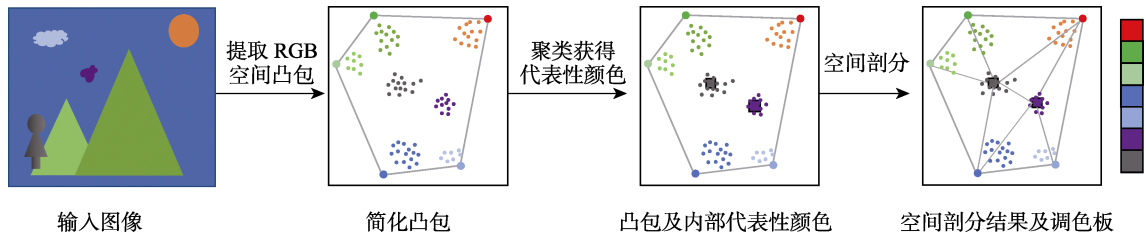


图 2 本文算法框架

将边  $(v_i, v_j)$  及对应的 2 个三角面片从原始凸包中删除, 在凸包体积扩张的同时实现了凸包的拓扑简化. 为了寻找这样一个使得体积增加最小的顶点  $v$ , Tan 等<sup>[4]</sup>对能量函数

$$v = \arg \min \sum \frac{A}{3} n(v - v_i) \quad (1)$$

进行优化. 其中, 等式右边表示缩边导致的凸包扩张增加的体积,  $A$  表示  $(v_i, v_j)$  关联的每个三角面片的面积;  $n$  表示每个三角面片的法向;  $v$  是待求解的顶点坐标. 在实际运行中, 每次枚举凸包的所有边, 并选择体积增加最小的边收缩, 当凸包体积增长到一定程度, 则算法结束.

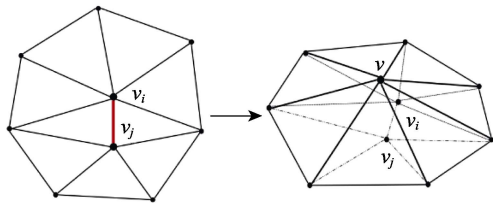


图 3 凸包简化示意图

上述算法对原始凸包进行了简化, 得到了顶点数目较少的凸包. 但随着缩边的持续进行, 凸包的体积将不断增大, 那些位于凸包内部的代表性颜色将无法加入到调色板中, 不能参与到后续的图像编辑任务. 为了使调色板具有更好的代表性, 本文通过聚类引入那些被忽略的、位于凸包内部的代表性颜色, 进一步优化初始调色板.

### 2.2 构建代表性调色板

本文使用 MeanShift 算法<sup>[23]</sup>对图像进行聚类, 以提取图像内部的代表性颜色. Cheng<sup>[24]</sup>通过核函数对 MeanShift 算法进行优化. 相对于  $K$ -Means<sup>[25]</sup>等聚类算法需要手动设定类别数目, MeanShift 算法可以自适应地决定最终的类别数目.

本文仍然将图像视为 RGB 空间的点集, 初始情况下, 将每一个像素点视作一个独立的聚类中心; 然后, 对每一个像素点进行均值漂移, 经过多轮均值漂移, 每个像素点最终会收敛到一个固定

的位置; 最后, 将位置接近的像素点归到同一类. 具体而言, 对于任意的一个像素点, 假定其在 RGB 空间的坐标为  $c_0 = p_0(R_0, G_0, B_0)$ , 以  $r$  为半径作一个球  $s_0$ , 计算落在球  $s_0$  内的所有像素点的均值  $c_1 = (R_1, G_1, B_1)$ ; 然后以  $c_1$  为球心  $r$  为半径作球  $s_1$ , 并再次计算球内像素点的均值; 不断重复该过程, 直至  $s_{k+1}$  的球心  $c_{k+1}$  距离  $s_k$  的球心  $c_k$  足够接近, 则认为算法收敛, 并停止均值漂移. 对 MeanShift 算法中聚类中心漂移过程形式化地表示为

$$\begin{cases} c_0 = p_0 \\ c_{k+1} = c_k + \frac{1}{M} \sum_{i=1}^M (p_i - c_k) \end{cases} \quad (2)$$

其中,  $p_1, p_2, \dots, p_M$  表示位于球心为  $c_k$  半径为  $r$  的球内的所有像素点.

半径  $r$  的估计对 MeanShift 聚类尤为重要. 通常, 半径越小, 最终得到的类越多; 反之, 最终得到的类越少. 为了自适应地选取合适的半径, 本文首先在图像中随机选取若干像素点, 计算所有点对的距离并对其进行排序, 排序后的序列标记为  $D$ . 假设所有点对数目为  $N$ ,  $q$  是一个介于 0~1 的比值, 则选取  $D$  中位置为  $Nq$  处的距离  $r = D[Nq]$  作为均值漂移的半径.

直接在所有像素上进行 MeanShift 聚类的效率非常低下. 为了提升速度, 本文对原始图像进行采样, 仅考虑采样点集上的点作为初始的聚类中心进行均值漂移. 聚类后得到了一些新的代表性颜色, 这些颜色在 RGB 空间中基本位于第 2.1 节得到的凸包内部, 并且某些颜色可能跟凸包顶点的距离非常接近, 使得调色板冗余. 为此, 保留所有初始调色板的颜色, 并删除聚类结果中与初始调色板相似的颜色. 即算法会删除那些距离凸包顶点的距离小于  $r$  的颜色.

结合 MeanShift 聚类算法, 本节在几何凸包调色板的基础上增加了凸包内部的代表性颜色, 对初始调色板进行了扩充和优化. 下面进一步介绍如何基于代表性调色板实现图像重着色编辑.

### 2.3 图像重着色编辑

通过第 2.1 节和第 2.2 节得到了输入图像的简化凸包,并在简化凸包内部增加了一些代表性颜色.但二者相互独立,为了将二者有机地结合起来,对凸包顶点和筛选后的聚类中心进行 Delaunay 四面体剖分,并进一步通过重心坐标插值将图像像素跟调色板颜色关联,以实现图像重着色编辑.

为了便于描述,将调色板的颜色集合标记为  $V$ ,输入图像的像素点集标记为  $P$ .为了实现图像编辑,对输入图像的所有像素点进行插值,即

$$p = \sum_{i=1}^N w_p^i V_i \quad (3)$$

将任意像素点  $p \in P$  表示为调色板  $V$  的凸组合.其中,  $w_p^i$  表示像素点  $p$  关于调色板颜色  $V_i$  的混合权重且满足  $\sum_{i=1}^N w_p^i = 1$ .实际的图像编辑过程中,混合权重只在初始时计算一次,之后保持不变,用户只需要修改调色板  $V$  的一种或多种颜色,并将修改后的调色板  $V'$  代入公式  $p' = \sum_{i=1}^N w_p^i V'_i$ ,即可得到编辑后的图像颜色.

针对混合权重计算,目前主要分为基于优化和基于插值的方法.其中,基于优化的方法得到的混合权重具有良好的光滑性,但往往计算效率较低,计算单幅图像往往需要数十分钟.基于插值的方法,具有代表性的包括 MVC 算法<sup>[13]</sup>,它不仅计算高效且实现平滑的插值效果;Zhang 等<sup>[25]</sup>虽然提出局部重心插值坐标应用到网格模型的局部编辑,但要求提供大量的控制点.因此,当调色板数量较少时,现有算法难以实现稀疏的插值,即像素点关于大部分的调色板颜色的插值权重均大于 0,修改某些调色板颜色可能导致图像大面积的颜色变化,难以实现有效的局部编辑.区别于 Tan 等<sup>[4]</sup>和 Wang 等<sup>[13]</sup>将调色板表示为凸包或非凸多面体的方式,本文将所有的调色板颜色组织起来,并对这些颜色(三维顶点集合)进行四面体剖分.这样图像像素点集将被自然地分割到多个四面体内部,经过 Delaunay 四面体剖分后,每个像素只与所在的四面体的 4 个顶点相关,因此,每个点的混合权重最多只有 4 个元素的值大于 0,天然地满足稀疏性要求,从而实现了良好的局部编辑.

针对混合权重计算,目前主要分为基于优化和基于插值的方法.其中,基于优化的方法得到的混合权重具有良好的光滑性,但往往计算效率较低,计算单幅图像往往需要数十分钟.基于插值的方法,具有代表性的包括 MVC 算法<sup>[13]</sup>,它不仅计算高效且实现平滑的插值效果;Zhang 等<sup>[25]</sup>虽然提出局部重心插值坐标应用到网格模型的局部编辑,但要求提供大量的控制点.因此,当调色板数量较少时,现有算法难以实现稀疏的插值,即像素点关于大部分的调色板颜色的插值权重均大于 0,修改某些调色板颜色可能导致图像大面积的颜色变化,难以实现有效的局部编辑.区别于 Tan 等<sup>[4]</sup>和 Wang 等<sup>[13]</sup>将调色板表示为凸包或非凸多面体的方式,本文将所有的调色板颜色组织起来,并对这些颜色(三维顶点集合)进行四面体剖分.这样图像像素点集将被自然地分割到多个四面体内部,经过 Delaunay 四面体剖分后,每个像素只与所在的四面体的 4 个顶点相关,因此,每个点的混合权重最多只有 4 个元素的值大于 0,天然地满足稀疏性要求,从而实现了良好的局部编辑.

对调色板空间剖分后,得到了一组空间四面体,而针对四面体的插值具有良好的定义.具体而

言,在调色板空间剖分后的某个四面体  $ABCD$  内部,任意像素点  $p$  对应顶点  $A$  的插值权重可以简单地表示为

$$w_p^A = T(pBCD)/T(ABCD) \quad (4)$$

其中,  $T(pBCD)$  表示由  $pBCD$  共 4 个顶点构成的四面体的体积;  $T(ABCD)$  表示四面体  $ABCD$  的体积.类似地,可以得到点  $p$  关于四面体其他 3 个顶点的混合权重  $w_p^B$ ,  $w_p^C$  和  $w_p^D$ .

至此,已经得到了输入图像的代表性调色板,进一步将调色板表示为空间剖分的四面体网格,并实现了高效、稀疏的插值,保证了图像编辑的优良局部性.

## 3 实验结果与分析

本文实验环境为 Windows10, AMD R7-5800H, 16GB 内存.图 4 展示了本文算法实现的图像重着色用户界面.为了验证参数的敏感性,首先,对 MeanShift 算法使用的参数  $q$  进行了验证;其次,为了说明本文算法插值权重的优良稀疏性,将其与文献[1,13]算法进行了对比.为了说明本文算法支持更好的局部编辑,将其与文献[1,13,15]算法的图像重着色效果进行对比分析,并通过用户调研对本文算法进行了评估.实验结果和用户调研均表明,本文算法提取的调色板具有更好的代表性,相较于其他对比算法,本文插值权重更稀疏,更加有利于图像的局部编辑,并且具有更好的重着色效果.

### 3.1 参数评估

在调色板提取算法中,如前文所述,  $q$  决定了 MeanShift 聚类算法执行过程的半径  $r$ ,进一步决定了最终调色板的颜色数量.通常,越小的  $q$ ,会得到越多的聚类中心,从而在后续的重建中得到更精细的调整能力,但调色板颜色数量较多、过于分散、不够直观,且难以实现快速地调整图像颜色;而越大的  $q$ ,则会得到越少的聚类中心,使得最终的调色板颜色较少,缺乏代表性.

为了验证参数的敏感性,本文分别设置  $q = 0.1, 0.2, 0.4$ ,将其应用到聚类过程中提取图像的代表性颜色,并在表 1 中展示了 2 幅图像的调色板提取结果.

如表 1 所示,调色板的第 1 行颜色从凸包中获取,第 2 行为聚类算法提取的代表性颜色;几何结



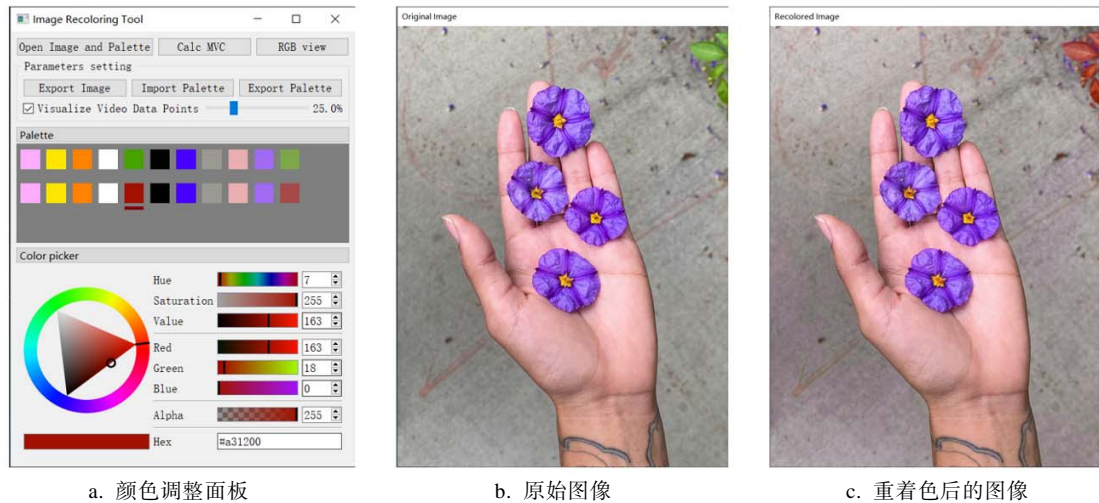


图 4 图像重着色用户界面

表 1 不同的  $q$  对于最终调色板的影响

输入图像	提取结果	$q = 0.1$	$q = 0.2$	$q = 0.4$
	调色板			
	几何结构			
	调色板			
	几何结构			

构中的圆点表示简化凸包的顶点, 方块表示聚类算法提取的代表性颜色. 由此可以看出, 当  $q = 0.1$  时, 提取了较多的代表性颜色, 使得整体的调色板颜色数量较多且某些颜色比较相似, 在第 1 个示例图像中, 除了凸包包含的 7 种颜色外, 还额外产生了 8 个聚类中心, 总共生成了包含 15 种颜色的调色板, 其中某些颜色相似度较高. 类似地, 第 2 个示例图像也生成了包含 11 种颜色的调色板. 而当  $q = 0.4$  时, 最终的调色板颜色较少, 一些凸包内部的代表性颜色未能被有效地提取. 根据经验, 对大部分图像而言, 当  $q = 0.2$  时, 能

够较好地平衡调色板的颜色数量与调色板颜色的代表性, 因此, 本文的后续实验结果均采用  $q = 0.2$  进行颜色聚类.

### 3.2 插值权重的稀疏性对比

通常, 插值权重越稀疏越好. 当修改调色板的颜色时, 希望修改最接近调色板颜色的像素而不至于影响到其他部分. 本文算法在每个四面体内部对像素进行插值, 每个像素的混合权重只与所在的四面体的 4 个顶点相关, 每个调色板颜色也只影响到与之共点的四面体内部的像素. 因此, 整体上保证了插值权重的稀疏性. 为了直观地展示插值权

重的稀疏性, 本文将每个调色板颜色影响的像素显示出来. 具体地, 若像素  $p$  在调色板颜色  $V_i$  的影响范围之内, 当且仅当  $p$  关于  $V_i$  的插值权重  $w_p^i > 0$ , 进一步绘制图层中的每个像素点  $p$  的颜色为

$$c_p = w_p^i V_i \quad (5)$$

通常, 某个像素若其出现在较少的图层, 则认

为其插值权重的稀疏性较好; 反之, 则认为其插值的稀疏性较差, 因为其关联了更多的调色板颜色.

本文算法提取的调色板和 Wang 等<sup>[13]</sup>提取调色板的图像分层结果对比如表 2 所示, 针对每个示例, 分别提供了输入图像, Wang 等<sup>[13]</sup>和本文算法的图层分解结果, 并在每个图层下方展示了该图层对应的调色板颜色.

表 2 Wang 等<sup>[13]</sup>和本文算法的图层分解结果对比

输入图像	提取算法	分层结果										
		1	2	3	4	5	6	7	8	9	10	11
man	Wang 等 <sup>[13]</sup>											
	本文											
lady	Wang 等 <sup>[13]</sup>											
	本文											
ball	Wang 等 <sup>[13]</sup>											
	本文											

在 man 图像示例中, Wang 等<sup>[13]</sup>算法将天空分解为 5 个图层, 如第 1 行第 2 列(黄色)、第 3 列(深蓝色)、第 4 列(白色)、第 5 列(淡蓝色)和第 7 列(红色)所示. 因此, 当用户调整这 5 种颜色中的任意颜色均会导致天空的颜色发生变化, 使得整个编辑过程不够直观. 而本文算法只将天空分解为单个图层(第 2 行第 8 列), 意味着可以针对性地修改天空的颜色而不影响到其他部分. 类似地, 如示例 lady 中第 3 行的前 3 列所示, Wang 等<sup>[13]</sup>算法将绿色背景分解为 3 个图层(黄、白、绿), 而本文算法可以将背景分解得更为干净(第 4 行第 7 列). 在示例 ball 中, Wang 等<sup>[13]</sup>的算法得到了深蓝(第 5 行第 3 列)、淡蓝(第 5 行第 5 列)和白(第 5 行第 6 列)共 3 个背景图层, 而本文算法同样能够有效地分离出背景部分(第 5 行第 7 列). 从整体上看, 本文算法

的插值权重更稀疏, 更加有利于用户进行针对性的局部颜色编辑.

### 3.3 重着色效果对比

本小节展示了不同算法在图像重着色任务上的效果对比. 表 3 和表 4 展示了本文算法与几何算法的实验效果对比, 表 5 展示了本文算法与深度学习算法的实验效果对比. 需要说明的是, 由于不同算法得到的调色板的表达能力不尽相同, 无法通过一致的调色板颜色修改来比较不同实验结果, 因此, 在对比实验中, 本文对每幅图像提出确定的修改意图, 并以目标为导向对调色板进行修改, 最后分析和比较重着色效果.

如表 3 所示, 在第 1 行示例中, 用户期望将右上角的植物从绿色改为褐色. 文献[1]算法在修改了植物颜色的同时也改变了花朵的颜色和花蕊的



表 3 本文算法与几何方法的重着色效果对比 1

名称	输入图像	文献[1]	文献[13]	本文
hand				
lady				
flag				
flower				

颜色, 文献[13]算法使得地面的颜色发生了不自然的变化, 而本文算法则较好地实现了编辑意图. 在第 2 行示例中, 用户期望将女士额头上方一小撮棕色头发修改成蓝色以进一步突出染发效果. 文献[1]和文献[13]的算法均引入了不自然的脸部颜色变化. 如第 3 行所示, 用户有 2 个编辑意图: 一是将彩旗中黄色条带改为粉红, 二是将女孩的牛仔裤从浅蓝调成灰蓝. 文献[1]算法使得图像整体偏暗, 文献[13]算法使得背景和脸部均出现了不自然的颜色变化, 本文算法得到的结果则更加自然. 在第 4 行示例中, 用户期望将紫色的花朵修改为黄色, 文献[1]算法使得编辑后的图像整体偏亮, 文献[13]算法修改了桌面的颜色. 相比之下, 本文算法不仅实现了精准的局部编辑, 而且不改变图像整体的亮度.

如表 4 第 1 行所示, 用户期望将绿色的气球和

长裤统一修改为紫色, 文献[1]和文献[13]的算法使得面部发生了不自然的颜色变化. 类似地, 在第 2 行中, 在修改图像的背景过程中, 文献[1]和文献[13]的算法使得壁画中的部分发生了不自然的颜色变化, 而本文算法实现了精确的局部编辑. 在第 2 行和第 3 行的示例中, 文献[1]算法与本文算法编辑效果相当, 但文献[13]算法仍然难以实现精确的局部编辑.

表 5 所示为本文算法与深度学习框架 PaletteNet<sup>[15]</sup>的对比结果. 在第 1 行示例中, 用户试图将绿叶修改为红叶, 本文算法精确地完成了这个编辑任务, 而文献[15]算法使得整个天空的颜色由蓝变红. 相似地, 在第 2 行示例中, 用户仅希望将红色部分的透光玻璃修改为紫色, 而文献[15]算法却意外地将绿色的玻璃变成了粉红色. 在第 3 行示例中, 文献[15]算法使得整个画面变为蓝色,



表 4 本文算法与几何方法的重着色效果对比 2

名称	输入图像	文献[1]	文献[13]	本文
man				
painting				
glass				
moon				
ball				

而叶片上的红色反而消失. 在第 4 行示例中, 本文算法与文献[15]算法均将书本变为绿色, 但后者同时修改了花的颜色与背景. 总体上看, 文献[15]算法使得重着色后的图像整体上契合给定调色板的主题, 但无法实现有效的局部编辑.

从实验结果来看, 基于凸包算法提取的调色板有局部性欠缺, 导致图像编辑出现了不自然的颜色变化; 基于聚类算法的调色板修改往往使得图像整体的亮度发生变化; 基于深度学习的方法仍然难以实现有效的局部编辑; 本文算法在实现了代表性调色板提取的同时, 实现了高效、局部性良好的图像编辑.

### 3.4 用户调研

由于图像重着色算法没有客观的评价指标,

难以进行有效的定量分析, 本文通过用户调研对图像调色板提取、重着色效果以及用户界面进行了评估. 总共有 20 名用户被邀请参与调研, 90% 的用户年龄为 22~29 岁. 其中 14 名用户有使用 Adobe Photoshop, Adobe Premiere, Adobe After Effects 等软件处理图像的经验, 15 名用户有对图像处理算法研究的经验. 用户调研的任务包括: (1) 要求用户对不同算法提取的调色板的代表性进行评价; (2) 给定具体的重着色任务以及不同算法的重着色效果, 要求用户对不同重着色的效果进行比对; (3) 对重着色的效果和用户界面给出评价.

总体上, 用户调研得到了比较积极的反馈. 其中, 80% 以上的用户认为本文算法提取的调色板具有更好的代表性; 85% 以上的用户认为本文重着色

表 5 本文算法与深度学习方法的重新着色效果对比

名称	输入图像	PaletteNet <sup>[15]</sup>	本文
garden			
window			
leaf			
book			

算法能够实现自然的图像编辑, 并且具有更优良的局部性; 75%的用户认为本文提供的用户界面简洁、明了、方便使用, 能够快速地实现针对性的图像编辑; 超过 80%的用户愿意将本文提供的图像重着色工具推荐给他人使用。

#### 4 结 语

为了增强调色板的代表性和图像编辑的局部性, 本文对基于凸包的调色板提取算法进行了有效的改进. 首先, 在 RGB 空间提取图像的简化凸包得到初始调色板, 并通过聚类算法在凸包内部提取代表性颜色得到优化后的图像调色板; 然后, 对调色板颜色顶点进行空间剖分, 并使用重心坐标对图像像素进行插值, 显著地提高了插值权重的稀疏性. 实验结果和用户调研表明, 本文算法提取的调色板具有更好的代表性和编辑局部性, 重新着色效果优于对比的几种算法.

整体上来讲, 虽然本文算法提取的调色板具有更好的代表性和局部性, 但仍存在一些缺点, 有待未来做进一步改进: 一方面, 本文使用的插值算法仅具有  $C^0$  连续性, 在少数情况下无法获得足够平滑、自然的插值结果, 因此, 高阶连续的插值方

法值得进一步研究; 另一方面, 本文算法不能做到物体级别的颜色编辑. 如图 5 中一个红色的果盘上盛了一些红色的树莓, 因二者颜色非常接近, 在不借助语义信息的情况下无法对其有效的区分, 本文算法难以做到单独修改红色的树莓而不影响果盘的颜色. 为此, 结合语义信息的图像重着色将是一个有趣的研究方向.



图 5 颜色高度相似果盘和树莓无法区分着色

#### 参考文献(References):

[1] Chang H W, Fried O, Liu Y M, et al. Palette-based photo recoloring[J]. ACM Transactions Graphics, 2015, 34(4): Article

- No.139
- [2] Zhang Q, Xiao C X, Sun H Q, *et al.* Palette-based image recoloring using color decomposition optimization[J]. *IEEE Transactions on Image Processing*, 2017, 26(4): 1952-1964
- [3] Zhang Q, Nie Y W, Zhu L, *et al.* A blind color separation model for faithful palette-based image recoloring[J]. *IEEE Transactions on Multimedia*, 2022, 24(99): 1545-1557
- [4] Tan J C, Lien J M, Gingold Y. Decomposing images into layers via RGB-space geometry[J]. *ACM Transactions on Graphics(TOG)*, 2017, 36(1): Article No.7
- [5] Tan J C, Echevarria J, Gingold Y. Efficient palette-based decomposition and recoloring of images via RGBXY-space geometry[J]. *ACM Transactions on Graphics (TOG)*, 2018, 37(6): Article No.262
- [6] Tan J C, Echevarria J, Gingold Y. Palette-based image decomposition, harmonization, and color transfer[OL]. [2021-10-31]. <https://arxiv.org/abs/1804.01225>
- [7] Ju T, Schaefer S, Warren J D. Mean value coordinates for closed triangular meshes[J]. *ACM Transactions on Graphics*, 2005, 24(3): 561-566
- [8] Sederberg T W, Parry S R. Free-form deformation of solid geometric models[C] //Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1986: 151-160
- [9] Lipman Y, Levin D, Cohen-Or D. Green coordinates[J]. *ACM Transactions on Graphics (TOG)*, 2008, 27(3): 1-10
- [10] Joshi P, Meyer M, DeRose T, *et al.* Harmonic coordinates for character articulation[J]. *ACM Transactions on Graphics (TOG)*, 2007, 26(3): Article No.71
- [11] Liu Jing, Cheng Zhiquan, Li Jun, *et al.* A harmonic field free-form deformation algorithm[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2011, 23(12): 2054-2058(in Chinese)  
(刘晶, 程志全, 李俊, 等. 一种基于调和函数的自由变形算法[J]. *计算机辅助设计与图形学学报*, 2011, 23(12): 2054-2058)
- [12] Du Zhengjun, Zhang Hui. As-rigid-as-possible mesh deformation based on volumetric graph[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2016, 28(2): 218-227(in Chinese)  
(杜正君, 张慧. 体积图控制的近似刚性的网格变形[J]. *计算机辅助设计与图形学学报*, 2016, 28(2): 218-227)
- [13] Wang Y L, Liu Y F, Xu K. An improved geometric approach for palette - based image decomposition and recoloring[J]. *Computer Graphics Forum*, 2019, 38(7): 11-22
- [14] Du Z J, Lei K X, Xu K, *et al.* Video recoloring via spatial-temporal geometric palettes[J]. *ACM Transactions on Graphics (TOG)*, 2021, 40(4): Article No.150
- [15] Cho J, Yun S, Lee K, *et al.* PaletteNet: image recolorization with given color palette[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. Los Alamitos: IEEE Computer Society Press, 2017: 1058-1066
- [16] Akimoto N, Zhu H C, Jin Y H, *et al.* Fast soft color segmentation[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2020: 8274-8283
- [17] Levin A, Lischinski D, Weiss Y. Colorization using optimization[C] //Proceedings of ACM SIGGRAPH 2004. New York: ACM, 2004: 689-694
- [18] An X B, Pellacini F. AppProp: all-pairs appearance-space edit propagation[C] // Proceedings of ACM SIGGRAPH. New York: ACM, 2008: Article No.40
- [19] Xu K, Li Y, Ju T, *et al.* Efficient affinity-based edit propagation using K-D tree[J]. *ACM Transactions on Graphics (TOG)*, 2009, 28(5): Article No.118
- [20] Bonneel N, Sunkavalli K, Paris S, *et al.* Example-based video color grading[J]. *ACM Transactions Graphics*, 2013, 32(4): Article No.39
- [21] Luan F J, Paris S, Shechtman E, *et al.* Deep photo style transfer[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 6997-7005
- [22] Huang Y F, Qiu S, Wang C B, *et al.* Learning representations for high-dynamic-range image color transfer in a self-supervised way[J]. *IEEE Transactions on Multimedia*, 2021, 23: 176-188
- [23] Fukunaga K, Hostetler L. The estimation of the gradient of a density function, with applications in pattern recognition[J]. *IEEE Transactions on Information Theory*, 1975, 21(1): 32-40
- [24] Cheng Y Z. Mean shift, mode seeking, and clustering[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995, 17(8): 790-799
- [25] Zhang J Y, Deng B L, Liu Z S, *et al.* Local barycentric coordinates[J]. *ACM Transactions on Graphics*, 2014, 33(6): Article No.188