

优化式着色算法的混色机制

张可为^{*①②} 申群太^①

^①(中南大学信息科学与工程学院 长沙 410008)

^②(湖南工程学院电气信息学院 湘潭 411101)

摘要: 基于局部关系模型的优化式着色方式,难以定量描述其全局特性信息诸如着色结果与用户输入间关系。该文首先提出一种优化式着色框架,证明两种常用算法为其特例而且等价,然后将每个像素的着色结果转化为各涂抹颜色的组合形式,并通过数学分析推导出组合系数非负且归一化,从而证明优化着色具有混色机制,其结果由涂抹颜色混色而成,并推导出混色权重与首达概率的等价性,最后基于混色机制对现有算法提出改进。实验结果表明混色机制的引入既有助于提高着色效果对涂抹数量与位置的鲁棒性,也便于实时挑选涂抹颜色。

关键词: 图像处理; 优化式着色; 数学分析; 算法等价; 混色

中图分类号: TN911.73

文献标识码: A

文章编号: 1009-5896(2013)07-1612-07

DOI: 10.3724/SP.J.1146.2012.01454

Blending Mechanism of Optimization-based Colorization

Zhang Ke-wei^{①②} Shen Qun-tai^①

^①(College of Information Science & Engineering, Central South University, Changsha 410008, China)

^②(College of Electrical & Information Engineering, Hunan Institute of Engineering, Xiangtan 411101, China)

Abstract: Optimization colorization methods base on a specific local relation model, it can not afford to describe global characteristics such as direct connection between visual result and user-provided scribbles. Hence a general frame is put forward, and then it is proved that two seemingly different methods are the same case of it, therefore they are equivalent. Next, it is derived that colorization result can be synthesized a weighted sum of chrominance of all kinds of scribbles where coefficients are all nonnegative and normalized and it leads to the conclusion that optimization-based colorization is an exact process of chrominance blending, and moreover, equivalence between individual chrominance blending weight and the pixel's first-reach probability to relevant scribble is determined. The mechanism inspires some improvement to existing algorithm. Experiment results indicate that introduction of blending mechanism benefits visual result to user's scribbles not only in robustness to amount and position but convenience in selecting desirable colors in-time.

Key words: Image processing; Optimization-based colorization; Mathematics analysis; Equivalence among algorithms; Chrominance blending

1 引言

利用着色对黑白图像添加色彩可以获得更丰富的视觉表现力。最初的着色方法先将图像分割成块再分块上色,但自动分割算法难以精确定位复杂的区域边界,为此一些交互式分割方法被引入,如 Liu 等人^[1]利用随机游走分割法进行着色。但基于分割的着色结果呈严格的颜色分块,块间缺乏颜色过渡带,效果显得生硬,特别是在处理物体内部平缓颜色渐变区域问题更严重。另一类方法通过从彩色参考图像实现颜色迁移实现图像着色^[2],该技术适用于不同颜色区域具有不同灰度纹理的图像。由于要求参考图

像既具有所需的全部颜色,各颜色区域纹理还必须与对应的待着色区域相似,所以用户难以直接控制着色结果,对结果中存在问题的区域也难以调整。近年来出现的优化式着色方法既可以直接选色,也便于修改,还避免了分割法出现的颜色分块,所以得到广泛的应用,既可以作为一类新的着色机制实现着色,也可以用于颜色迁移算法的改进^[3,4]。该方法通过假设相邻像素满足指定的局部关系模型,把用户的涂抹作为约束条件,设计一个代价函数作为最小化对象,其最佳着色分布对应于函数最优解,从而将着色问题转化为约束优化问题,如 Levin 等人^[5]假定每个像素的颜色可以表示成周围像素的加权平均和,由此构建一个全局代价函数;文献[6]采用半监督学习方法通过假设灰度相近的相邻像素其颜色差异越小,构建出另一个全局代价函数。尽管

2012-11-12 收到, 2013-01-06 改回

湖南省教育厅科研基金(08C220)资助课题

*通信作者: 张可为 d10zhangkw@yahoo.com.cn

优化式算法应用广泛，但是这种基于局部关系模型的扩展方式，难以提供所需要的全局特性信息，使得算法难以结合本身特性进行改进，譬如该类算法存在着色效果对涂抹的数量与位置敏感等问题，但现有改进方法只能通过用户添加大量涂抹，或者通过借助于其他方法来弥补，如加入边缘信息^[7]、非局部均值信息^[8]、纹理信息^[9,10]，不仅加重了算法的复杂度，而且改进质量还依赖于检测精度。本文首先将优化式着色纳入同一个框架内，推导出其混色机制，进而提出一种简单有效的改进算法。

2 一种优化式着色框架

首先根据决定着色效果的两个因素提出优化式着色的一种框架，进而将两种常用优化着色算法纳入该框架中并证明其等价性。

2.1 问题描述

图像着色表现为像素的灰度标量用3维颜色矢量取代。由于不同的颜色可能具有相同的灰度值，所以着色问题不具有唯一解。为了降低这种不确定性应加入颜色约束，在优化式着色中由用户添加颜色涂抹加以实现。

图像着色一般在YUV或YCbCr颜色空间进行，主要是为了在着色过程中将色度与灰度分开以避免混扰，其中Y为灰度通道，UV(或CbCr)为两色度通道，优化式着色的任务就是结合图像Y通道信息以及涂抹色度信息为所有像素配置色度。

由于高质量图像的着色效果要求悦目自然，除颜色约束外，一般还要求色度变化与灰度变化一致：均匀灰度区域的色度变化小，而边缘等灰度剧烈变化的区域色度变化的可能性大。为此，设计一个全局的代价函数来衡量着色结果与上述一致性要求的差异，同时将颜色涂抹作为约束条件，其最小化解作为最佳的色度分布。尽管需要确定的色度通道有两个(UV或CbCr)，但两色度通道相互独立，且计算方法相同，因而统一采用向量 \mathbf{c} 表示，最佳的 \mathbf{c} 值分布通过最小化过程实现：

$$J(\mathbf{c}) = \frac{1}{2} \sum_{i=1}^n \sum_{j \in N_i} \mathbf{A}(i, j) (\mathbf{c}(i) - \mathbf{c}(j))^2$$

$$\text{s.t. } \mathbf{c}(i) = \mathbf{b}(i), i \in S$$

其中 $J(\mathbf{c})$ 为全局代价函数，用于衡量 \mathbf{c} 分布与一致性假设的差异； S 为涂抹像素集合；向量 \mathbf{b} 表示图像着色前的初始色度； n 为像素总数； N_i 表示与像素 i 位置相邻的像素集合，即像素 i 周围一圈8个像素(i 位于图像边缘时相邻像素会减少)， j 为该集合中一个像素， \mathbf{A} 为相似度矩阵，元素 $\mathbf{A}(i, j)$ 用于衡量像素 i 和 j 间相似程度，只有当 i 和 j 位置为近邻时 $\mathbf{A}(i, j)$ 才允许为非0值，计算方法采用宽度为 σ 的

高斯核权值函数^[6]

$$\mathbf{A}(i, j) = \exp\left\{\frac{-(\mathbf{y}(i) - \mathbf{y}(j))^2}{2\sigma^2}\right\} \quad (1)$$

其中 $\mathbf{y}(i)$ 表示像素 i 的灰度。由于 $\mathbf{A}(i, j) = \mathbf{A}(j, i)$ ，所以 \mathbf{A} 为对称阵。

上述约束优化问题可利用拉格朗日乘子法求解，即色度 \mathbf{c} 对应于式(2)的最小化解

$$J_1(\mathbf{c}) = \frac{1}{2} \sum_{i=1}^n \sum_{j \in N_i} \mathbf{A}(i, j) (\mathbf{c}(i) - \mathbf{c}(j))^2 + \lambda \sum_{i \in S} (\mathbf{c}(i) - \mathbf{b}(i))^2 \quad (2)$$

式中第2项称为数据约束项，用误差平方和形式表示约束条件， λ 为拉格朗日乘子，用于衡量约束程度， $\lambda \rightarrow +\infty$ 对应于在着色过程中涂抹像素的色度始终保持不变。

为方便求解 $J_1(\mathbf{c})$ ，首先将其转化为矩阵形式。

取对角阵 $\mathbf{\Lambda}$ ，其对角元素满足 $\mathbf{\Lambda}(i, i) = \begin{cases} 1, & i \in S \\ 0, & i \notin S \end{cases}$ ，

像素的度为 $\mathbf{d}(i) = \sum_{j=1}^n \mathbf{A}(i, j)$ ，对角阵 $\mathbf{D} = \text{diag}(\mathbf{d}(1),$

$\mathbf{d}(2), \dots, \mathbf{d}(n))$ 。这样式(2)可转化成矩阵形式

$$J_1(\mathbf{c}) = \mathbf{c}^T (\mathbf{D} - \mathbf{A}) \mathbf{c} + \lambda (\mathbf{c} - \mathbf{b})^T \mathbf{\Lambda} (\mathbf{c} - \mathbf{b})$$

$J_1(\mathbf{c})$ 的最小化需要满足 $\partial J_1 / \partial \mathbf{c} = 0$ ，即 $(\mathbf{D} - \mathbf{A}) \mathbf{c} + \lambda \mathbf{\Lambda} \mathbf{c} - \lambda \mathbf{\Lambda} \mathbf{b} = 0$ 。

若 $\lambda \rightarrow +\infty$ ，

$$\begin{aligned} \mathbf{c} &= \lambda ((\mathbf{D} + \lambda \mathbf{\Lambda}) - \mathbf{A})^{-1} \mathbf{\Lambda} \mathbf{b} \\ &= (\mathbf{I}_n - (\mathbf{D} + \lambda \mathbf{\Lambda})^{-1} \mathbf{A})^{-1} ((\mathbf{D} + \lambda \mathbf{\Lambda})^{-1} (\lambda \mathbf{\Lambda}) \mathbf{b}) \\ &= (\mathbf{I}_n - (\mathbf{I}_n - \mathbf{A}) \mathbf{W})^{-1} \mathbf{b} \end{aligned} \quad (3)$$

其中 \mathbf{I}_n 为 n 维单位矩阵； $\mathbf{W} = \mathbf{D}^{-1} \mathbf{A}$ 。

2.2 优化式着色框架与常用着色算法的关系

现有的一种常用优化式着色方法是把着色问题转化为机器学习中的半监督学习^[7]，通过设定下列最小化代价函数后求解获得最佳的 \mathbf{c} 值。

$$J_2(\mathbf{c}) = \frac{1}{2} \mu \sum_{i \in I} \sum_{j \in N_i} \mathbf{A}(i, j) ((\mathbf{c}(i) - \mathbf{c}(j))^2 + \sum_{k \in S} (\mathbf{c}(k) - \mathbf{b}(k))^2)$$

其中 $\mu \rightarrow 0^+$ 。

它与本文所建着色框架的关系见定理1。

定理1 基于半监督学习的着色算法为新着色框架下 $\lambda \rightarrow +\infty$ 情形。

证明 将 $J_2(\mathbf{c})$ 改写为

$$J_2(\mathbf{c}) = \mu \left\{ \frac{1}{2} \sum_{i \in I} \sum_{j \in N_i} \mathbf{A}(i, j) ((\mathbf{c}(i) - \mathbf{c}(j))^2 + \frac{1}{\mu} \sum_{k \in S} (\mathbf{c}(k) - \mathbf{b}(k))^2) \right\}$$

式中 $1/\mu \rightarrow +\infty$ ，与式(2)比较， $J_2(\mathbf{c})$ 与 $J_1(\mathbf{c})$ 在 $\lambda \rightarrow +\infty$ 情况下具有相同的最小化解，表明半监督学习法着色算法为新着色框架下 $\lambda \rightarrow +\infty$ 情形。

证毕

另一种广泛应用的优化式着色方法——Levin 着色算法，假定着色结果中每个未涂抹像素的色度与周围像素的加权平均和一致，由此构建一个全局代价函数衡量着色结果与该假定的差异，仍将用户的涂抹作为约束条件。其最佳的色度分布表现为式(4)误差平方和函数的最小化解

$$\left. \begin{aligned} J_2(\mathbf{c}) &= \sum_{i=1}^n \left(\mathbf{c}(i) - \sum_{j \in N_i} \mathbf{W}(i, j) \mathbf{c}(j) \right)^2, \\ \mathbf{W}(i, j) &= \mathbf{A}(i, j) / \sum_{j \in N_i} \mathbf{A}(i, j) \\ \text{s.t. } \mathbf{c}(k) &= \mathbf{b}(k), \quad k \in S \end{aligned} \right\} \quad (4)$$

定理 2 Levin 着色算法亦为新着色框架下 $\lambda \rightarrow +\infty$ 情形。

证明 将式(3)改写为 $\mathbf{c} - (\mathbf{I}_n - \mathbf{A})\mathbf{W}\mathbf{c} = \mathbf{b}$ ，像素的色度具有以下特点：

(1) 对于种子像素 $i \in S$ ，由于 $\mathbf{A}(i, i) = 1$ ，得 $\mathbf{c}(i) = \mathbf{b}(i)$ 。

(2) 对于非种子像素 $i \notin S$ ， $\mathbf{A}(i, i) = 0$ ， $\mathbf{b}(i) = 0$ ，得 $\mathbf{c}(i) - \sum_{j \in N_i} \mathbf{W}(i, j) \mathbf{c}(j) = 0$ 。由于式中 $\mathbf{W}(i, j)$ 只在 $j \in N_i$ 具有非零值，可将其改写为 $\mathbf{c}(i) = \sum_{j \in N_i} \mathbf{W}(i, j) \mathbf{c}(j)$ ，代入式(4)后，非负函数 $J_2(\mathbf{c})$ 等于其最小值 0，说明 \mathbf{c} 为满足 Levin 假定的最佳的色度分布。证毕

推论 Levin 着色算法与基于半监督学习的着色算法等价。

3 优化式着色算法的混色机制

由于优化式着色基于局部模型加以实现，没有提供全局特性方面的信息，使得系统输出(着色结果)与输入(颜色涂抹)的关系不明确，本节分析并推导出其隐含的混色机制，从而定量描述着色结果与颜色涂抹的联系。

将涂抹色度用向量 $\mathbf{e} = [e_1, e_2, \dots, e_l]^T$ 表示，则初始色度 \mathbf{b} 可转化为涂抹的色度表示， $\mathbf{b} = \mathbf{M}\mathbf{e}$ ，其中 \mathbf{M} 为 $n \times m$ 矩阵， $\mathbf{M}(i, j) = \begin{cases} 1, & \mathbf{b}(i) = e_j \\ 0, & \mathbf{b}(i) \neq e_j \end{cases}$ 。将其代入式(3)后，图像色度 \mathbf{c} 就转化为 \mathbf{e} 的组合表达式： $\mathbf{c} = \mathbf{X}\mathbf{e}$ 。其中 \mathbf{X} 与 \mathbf{M} 存在以下关系：

$$\mathbf{X} = (\mathbf{I}_n - (\mathbf{I}_n - \mathbf{A})\mathbf{W})^{-1} \mathbf{M} \quad (5)$$

而 \mathbf{X} 称为表达式 $\mathbf{c} = \mathbf{X}\mathbf{e}$ 的组合系数，其行对应于像

素编号，列对应于涂抹色度编号。

对于 \mathbf{X} 中位于涂抹像素 $i \in S$ 处的行，因为 $\mathbf{A}(i, i) = 1$ ，存在 $\mathbf{X}(i, j) = \mathbf{M}(i, j)$ ，即

$$\mathbf{X}(i, j) = \begin{cases} 1, & (i \in S) \& (\mathbf{b}(i) = e_j) \\ 0, & (i \in S) \& (\mathbf{b}(i) \neq e_j) \end{cases} \quad (6)$$

而对 \mathbf{X} 中其他行(对应于非涂抹像素 $i \notin S$)的元素，因为 $\mathbf{A}(i, i) = 0$ ，根据式(5)，得

$$\mathbf{X}(i, k) = \sum_{j \in N_i} \mathbf{W}(i, j) \mathbf{X}(j, k), \quad i \notin S \quad (7)$$

即任意一个非涂抹像素 i 在 \mathbf{X} 中任意一列 k 的取值均可以表示为其周围像素在该列取值的加权平均和。而 \mathbf{W} 称为加权系数矩阵。

3.1 优化式着色的混色特性分析

定义 1 混色指图像每个像素的色度均可以表示成涂抹色度的加权平均和，且权重值均处于 $[0, 1]$ 之间。

具体说来，若图像中任意像素 i 的色度均可表示为涂抹色度的加权和， $\mathbf{c}(i) = \sum_{r=1}^l \mathbf{X}(i, r) \mathbf{c}_r$ ，其中权重 $\mathbf{X}(i, r)$ 满足条件 $0 \leq \mathbf{X}(i, r) \leq 1$ ，而且 $\sum_{r=1}^l \mathbf{X}(i, r) = 1$ ，则图像色度 \mathbf{c} 称为涂抹色度的混色。在对应的矩阵形式 $\mathbf{c} = \mathbf{X}\mathbf{e}$ 中，则要求 \mathbf{X} 必须为非负元素的行归一化矩阵。

为简化混色机制分析，首先定义 2 维点阵 (lattice) 上的离散调和函数和调和向量。

定义 2 取 $\mathbf{L} = \mathbf{D} \cup \mathbf{B}$ 表示有限平面点阵，其中 \mathbf{B} 作为边界点集合， \mathbf{D} 作为内部数据点集，且为连通集， $\mathbf{D} \cap \mathbf{B} = \emptyset$ ，若函数 f 在所有内部点的取值均可以表示为周围邻居点函数值的加权平均和，且权值均为正数，则函数 f 称为离散调和函数。相应地，调和函数在 \mathbf{L} 上的函数值按顺序组成的向量称为调和向量。

显然，若将图像作为 \mathbf{L} ，其中涂抹像素作为边界点，未涂抹像素作为内部点，则根据式(7)，矩阵 \mathbf{X} 中每个列向量均为调和向量。

首先分析调和向量中元素的允许取值范围。

定理 3(最大、最小值定理) 调和向量中元素取值介于边界点的最大值、最小值之间。

证明 假设调和向量中元素的最大值不是出现在边界点，而是出现在一内部点，则根据调和向量定义，该点可表示为周围邻居点的加权平均和，由于权值为正数且归一化，则其所有邻居点均应该为最大值。以此类推，所有的点(包括边界点)均应该取最大值，这意味着边界点取值应该大于其实际取值，假设不成立，所以元素取值应不大于边界点最大值。

同样, 若调和向量中元素的最小值出现在一内部点, 则根据调和向量的定义, 其所有邻居点均应该为最小值。以此类推, 所有的点(包括边界点)均应该取最小值, 也不符合边界点的实际取值, 所以元素取值应不小于边界点最小值。 证毕

定理 4 矩阵 \mathbf{X} 中所有元素的取值介于 $[0,1]$ 之间, 即满足 $0 \leq \mathbf{X}(i, k) \leq 1, \forall i, \forall k$ 。

证明 根据式(6)可知矩阵 \mathbf{X} 中每个列向量的最大边界值为 1, 最小边界为 0, 根据最大、最小值定理, \mathbf{X} 中所有元素的取值介于 $[0,1]$ 之间。 证毕

为了分析 \mathbf{X} 矩阵的行特性, 首先分析调和向量的叠加特性。

定理 5(线性叠加定理) 对具有相同加权系数的调和向量进行线性叠加, 结果仍为调和向量。

证明 用 \mathbf{v}_1 和 \mathbf{v}_2 表示两调和向量, \mathbf{G} 为共用的加权系数矩阵, 则对所有内部点 i , 存在 $\mathbf{v}_1(i) = \sum_{j \in N_i} \mathbf{G}(i, j)\mathbf{v}_1(j)$, $\mathbf{v}_2(i) = \sum_{j \in N_i} \mathbf{G}(i, j)\mathbf{v}_2(j)$ 。设计函数 $\mathbf{v}_3 = \alpha\mathbf{v}_1 + \beta\mathbf{v}_2$, 其中 α, β 为任意常数, 则

$$\begin{aligned} \mathbf{v}_3(i) &= \alpha \sum_{j \in N_i} \mathbf{G}(i, j)\mathbf{v}_1(j) + \beta \sum_{j \in N_i} \mathbf{G}(i, j)\mathbf{v}_2(j) \\ &= \sum_{j \in N_i} \mathbf{G}(i, j)(\alpha\mathbf{v}_1(j) + \beta\mathbf{v}_2(j)) \\ &= \sum_{j \in N_i} \mathbf{G}(i, j)\mathbf{v}_3(j) \end{aligned}$$

所以 \mathbf{v}_3 也是调和向量。以此类推, 更多调和向量的叠加结果也为调和向量。 证毕

根据定理可推出矩阵 \mathbf{X} 的行具有以下特性。

定理 6 矩阵 \mathbf{X} 每行之和均为 1。

证明 将矩阵 \mathbf{X} 的列向量表示为向量组 $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_l$, 而所有列向量的相加结果用向量 \mathbf{u} 表示。由于每个列向量均为调和向量, 且加权系数阵相同(均为 \mathbf{W}), 故根据定理 5, \mathbf{u} 也为调和向量。另外, 由于在每个边缘点(即涂抹像素)上有且仅有一种颜色, 故 \mathbf{u} 在边界点取值均为 1, 所以根据上述最大值、最小值定理, 所有内部点(对应于未涂抹像素)编号处的行元素累加结果也为 1, $\mathbf{u}(i) = 1, \forall i$ 。 证毕

定理 7 优化式着色算法实现混色式着色。

证明 结合定理 5 及定理 6, 矩阵 \mathbf{X} 满足混色定义。 证毕

3.2 优化式着色的混色权重分析

优化式着色作为一种新型混色式算法, 其混色权重具有什么样的物理意义? 为此, 首先引入首达概率的定义并讨论其特性。

定义 3 首达概率指从某像素出发随机游走, 首次遇到的涂抹像素属于某种颜色涂抹的概率。

若定义任意像素 i 首达任意一种涂抹 r 的概率为 $\mathbf{P}(i, r)$, $\mathbf{P}(i, r)$ 显然满足 $0 \leq \mathbf{P}(i, r) \leq 1$ 。如果 i 为

涂抹像素, 由于每个涂抹像素首次遇到的涂抹像素必然为自己, 不可能为其他颜色涂抹, 所以其首达概率取值可分为两种情况:

$$\mathbf{P}(i, j) = \begin{cases} 1, & (i \in S) \& (\mathbf{b}(i) = e_j) \\ 0, & (i \in S) \& (\mathbf{b}(i) \neq e_j) \end{cases}$$

与式(6)比较, 首达概率 \mathbf{P} 与 \mathbf{X} 在边界点(涂抹像素)具有相同取值。

定理 8 首达概率为一个调和向量。

证明 由于任意非涂抹像素 i 做随机游走直至首达涂抹 r 的整个游走过程为齐次马尔科夫链, 故整个转移过程可表示为组合形式: 先一步转移至邻居像素 j , 然后由邻居像素随机游走首达涂抹 r 。若用 $\mathbf{Q}(i, j)$ 表示像素 i 向邻居像素 j 的一步转移概率, 组合方法可根据 Chapman-Kolmogorov 方程, 表示为 $\mathbf{P}(i, r) = \sum_{j \in N_i} \mathbf{Q}(i, j)\mathbf{P}(j, r)$ 。由于 $\mathbf{Q}(i, j)$ 与 $\mathbf{W}(i, j)$ 成正比, 且 $\sum_{j \in N_i} \mathbf{Q}(i, j) = 1$, 故 $\mathbf{Q}(i, j) = \mathbf{W}(i, j)$, 相应地, $\mathbf{P}(i, r) = \sum_{j \in N_i} \mathbf{W}(i, j)\mathbf{P}(j, r)$, $i \notin S$ 。表明首达概率为一个调和向量。 证毕

定理 9(唯一性定理) 具有相同加权系数的两调和向量, 如果边界点取值相同, 则两向量完全相同。

证明 用 \mathbf{v}_1 和 \mathbf{v}_2 表示边界点取值相同的两调和向量。设计向量 $\mathbf{v}_3 = \mathbf{v}_1 - \mathbf{v}_2$, 则根据线性叠加定理, \mathbf{v}_3 也为调和向量。而且由于 \mathbf{v}_1 和 \mathbf{v}_2 在边界点元素上取值相同, 使得 \mathbf{v}_3 在边界点元素处均为 0 值。根据上述最大、最小值定理, \mathbf{v}_3 的所有元素上取值为 0, 即 $\mathbf{v}_1 = \mathbf{v}_2$ 。 证毕

定理 10 在优化型着色所对应的混色式着色中, 每种颜色涂抹所占的混色权重等价于像素游走至该种涂抹的首达概率。

证明 由于首达概率与着色框架混色权重具有相同加权系数(\mathbf{W}), 且边界条件相同(\mathbf{M}), 根据唯一性定理, 该混色权重等价于首达概率。 证毕

4 基于混色机制的优化式着色算法改进

要获得满意的着色效果, 一方面要求所选择的颜色符合用户的意愿, 另一方面要求图像悦目自然。前者可以通过对涂抹颜色做精挑细选实现, 后者主要体现在尽量保证色度变化与灰度变化一致, 着重表现在控制颜色过渡带的质量。混色机制在涂抹颜色挑选上的应用放在实验部分讨论, 本节主要基于混色机制提出一种过渡带控制算法。

颜色过渡带指图像中人眼可以觉察的颜色过渡区域, 区域内每个像素上的混色权重差异不会很大(如小于 0.9)。在自然图像中, 同一个颜色对象内出

现的颜色过渡带往往反映颜色对象本身的颜色分布特征, 区域较宽且灰度变化均匀。而在不同颜色对象之间(如不同物体间)会出现快速变化的颜色过渡带, 主要有几个原因: 运动模糊效应、光学模糊效应以及离散化伪影等, 这些效应所形成的过渡带较窄, 混色权重经过很窄的区域内就完成 0 与 1 之间的快速转化。实验发现优化式着色在缓慢变化区域着色后效果较好, 但在距离涂抹较远的对象间区域容易出现过宽的过渡带, 其原因可采用概率法混色机制进行分析: 高质量的着色图像要求每个颜色对象的色度主要取决于该对象内的涂抹, 该涂抹的混色权重远大于其他涂抹, 只是经过边缘过渡带快速衰减, 同时相邻对象的涂抹的混色比重迅速上升, 从而实现颜色快速渐变。但对于优化式着色来说, 其混色权重等价于首达概率, 大小取决于两个因素: 转移步数和单步转移概率。如果一些像素至内部涂抹的距离远大于至外部对象中涂抹距离的话, 即使单步转移概率较大, 但转移步数增加会降低这些像素至内部涂抹的首达概率, 从而降低了该涂抹的混色权重, 同时其他涂抹的权重相应增加, 导致这些像素处于过渡状态, 结果形成过宽的过渡带。依靠用户在交界处附近两边设置大量涂抹来加以抑制的办法费时费力, 下面通过调节混色权重来控制过渡带宽度, 从而改善对象边缘处的着色质量。

设计归一化指数函数作为调节函数, 如果图像中不存在对象内颜色渐变, 则调节函数为

$$X_h(i, r) = \frac{X^h(i, r)}{X^h(i, 1) + X^h(i, 2) + \dots + X^h(i, l)} \quad (8)$$

其中 h 称为混色因子, 允许取值为任意一个正数。

由于对象间的过渡带上的像素颜色基本上取决于相邻两个对象颜色的混色结果, 所以调整作用常可以简化为两色的混色情形, 另外, 由于两混色权重互补, 可将式(8)简化为

$$X_h(i, r) = \frac{X^h(i, r)}{X^h(i, r) + (1 - X(i, r))^h}$$

定理 11 在 $X_h(i, r) = \frac{X^h(i, r)}{X^h(i, r) + (1 - X(i, r))^h}$

中, 混色权重 $X(i, r)$ 经调整后满足

$$\left. \begin{aligned} X_{h+1}(i, r) &> X_h(i, r), & X(i, r) > 0.5 \\ X_{h+1}(i, r) &= X_h(i, r), & X(i, r) = 0.5 \\ X_{h+1}(i, r) &< X_h(i, r), & X(i, r) < 0.5 \end{aligned} \right\} \quad (9)$$

证明

$$\begin{aligned} & X_{h+1}(i, r) - X_h(i, r) \\ &= \frac{X^{h+1}(i, r)}{X^{h+1}(i, r) + (1 - X(i, r))^{h+1}} \\ &\quad - \frac{X^h(i, r)}{X^h(i, r) + (1 - X(i, r))^h} \\ &= (2X(i, r) - 1) \cdot X^h(i, r) \cdot (1 - X(i, r))^h \\ &\quad / [(X^{h+1}(i, r) + (1 - X(i, r))^{h+1}) \\ &\quad \cdot (X^h(i, r) + (1 - X(i, r))^h)] \end{aligned}$$

由于 $X(i, r)$ 取值处于区间[0,1]内, 故存在 3 种取值情形:

当 $X(i, r) > 0.5$, $X_{h+1}(i, r) - X_h(i, r) > 0$; 若 $X(i, r) = 0.5$, $X_{h+1}(i, r) - X_h(i, r) = 0$; 反之, 如果 $X(i, r) < 0.5$, $X_{h+1}(i, r) - X_h(i, r) < 0$ 。证毕

定理 11 表明, 混色权重调整有多种变化趋势: 随着 h 增大, 混色中的主要颜色(混色权重大于 0.5)会增强, 而次要颜色(混色权重小于 0.5)会削弱。

需要注意的是, 如果图像中存在缓变的对象内颜色过渡带, 采用式(8)会造成该过渡带变窄, 严重时形成色斑。解决的方法是首先将构建正常颜色过渡带的两种涂抹的混色权重相加后作为一个整体权重, 进行上述调整, 增强它们与其它涂抹的区隔性, 然后再按原来的比例进行分配从而体现两者间正常的渐变特性。譬如第 $j, j+1$ 种涂抹间存在对象内渐变区域, 可将式(8)修正为

$$\begin{aligned} X_h(i, j) &= (X(i, j) + X(i, j+1))^h / [X^h(i, 1) + \dots \\ &\quad + (X(i, j) + X(i, j+1))^h + \dots + X^h(i, l)] \\ &\quad \cdot \frac{X(i, j)}{X(i, j) + X(i, j+1)} \end{aligned} \quad (10)$$

整个改进算法主要实现步骤如下:

步骤 1 混色权重求解。采用式(5)求解。

步骤 2 权重映射。采用式(8)或式(10)将权重 X 转化为 X_h 。

步骤 3 色度混合。采用 $c = X_h e$ 实现色度混色。

5 实验结果与数据分析

通过实验检测改进算法的着色性能, 包括过渡带控制以及涂抹颜色选择两方面。

为检验混色因子对过渡带宽度的控制作用, 提供了 3 个 h 值下的着色结果, 该结果采用混色权重分布加以体现, 而且为便于考察算法对涂抹的位置与数目的鲁棒性, 设置了 4 种涂抹方式, 如图 1 所示。

图 1(a)为 4 种不同的涂抹图像(其中第 1 种颜色涂抹均位于灰色条纹的左侧, 第 2 种涂抹在图中加

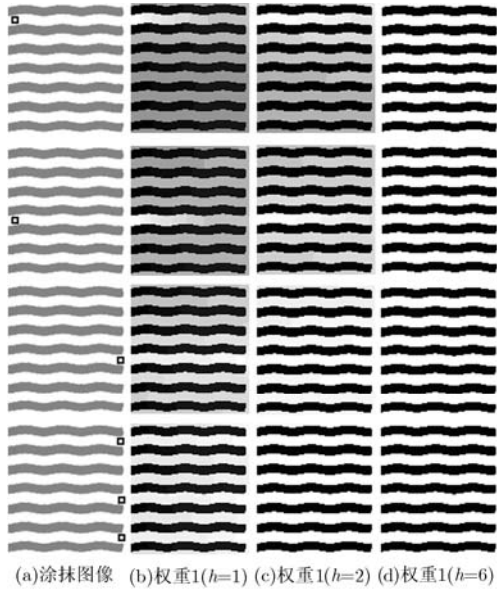


图 1 混色因子 h 值对着色效果的影响

黑框指示), 而各种情况下颜色涂抹的混色权重分布见图 1(b),1(c),1(d), 由于混色权重归一化, 使得每个像素上各颜色的混色权重和一定为 1, 所以第 2 种涂抹的混色权重存在与第 1 种涂抹权重互补关系, 故在此省略。权重分布通过归一化灰度图表示: 像素灰度值越大表示指定涂抹的颜色所占的混色比例越大, 而其它涂抹的颜色所占比例越小。黑色对应于 0, 表示像素不含该种颜色, 白色对应于 1, 表示像素颜色完全由该颜色组成。图 1 提供了以下信息: (1) $h=1$ 时(对应于传统算法, 如 Levin 算法以及基于半监督学习的着色算法), 若第 1 种涂抹为单笔涂抹时会存在严重的混色权重的越界现象, 而且对涂抹位置以及数量变化敏感, 表明传统算法的着色结果对涂抹位置与数量敏感。敏感的原因在于这些算法在对象交界处的区隔性较弱, 在距涂抹位置较远处第 2 种涂抹的混色权重下降情况严重, 同时造成第 1 种涂抹颜色的混色权重增加, 使得大量像素上的混色权重差异降低, 会形成过宽的颜色过渡带, 着色质量不高。(2)不管是哪一种涂抹方式, 随着 h 值的增加, 越界程度均逐步降低, $h=6$ 时均可以获得满意的权重分布, 对应于良好的着色效果, 表明新算法对涂抹位置与数量具有高鲁棒性。这是因为随着 h 增大, 根据定理 11, 混色中的主要颜色的新权重值 X_h 快速增加, 而次要颜色权重会快速减少, 这样, 像素上的各混色权重迅速趋向于两极(0 和 1), 原来处于过宽过渡带的大量的像素脱离过渡条件, 使得过渡带变窄, 从而有效地改进了着色质量。

对具有缓变过渡带的图像调整示例见图 2(3 种涂抹分别位于灰色条纹左边、右边和指示方框内)。提供了 h 调整前后第 1、2 类涂抹(分别位于灰色条

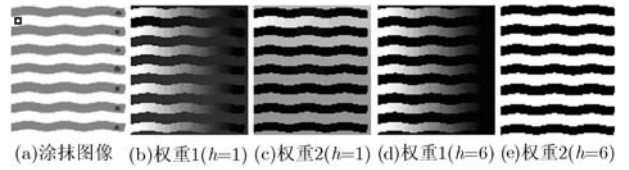


图 2 混色因子 h 值对含渐变区域图像着色的作用

纹左、右边)的混色权重归一化灰度图。从图中看到, $h=1$ 时这两种涂抹间存在权重渐变区, 同时 3 种颜色的混色权重具有相互渗透的现象, 尤其是前两种涂抹越界至第 3 类涂抹的主导区域内的渗透现象更为严重。而在 $h=6$ 时, 越界现象已得到有效抑制, 同时前两种涂抹间区域具有更好的权重渐变特性。

图 3 给出其它一些着色示例(其中第一行中权重子图体现加菲猫肤色所占的混色权重, 第二行中权重子图对应于车体颜色所占的混色权重)。同样, 调整混色因子能够有效控制过渡带宽度。

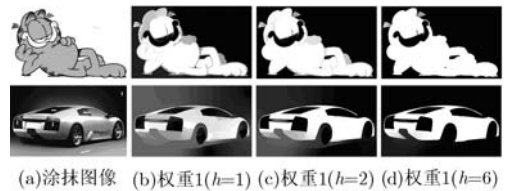


图 3 利用混色因子 h 控制着色的其它示例

除了要减轻对涂抹位置与数量的敏感度外, 要获得满意的着色结果还取决于选择的涂抹颜色是否满足用户的意愿。用户要选出中意的颜色涂抹, 常用的方法是通过改变涂抹颜色提供一系列颜色的着色效果后作为候选样本, 通过直观比较进行选择。传统算法下每一份候选样本都是通过重新着色得到, 而基于混色机制的新算法采取不同的机制, 所以选色效率有异。首先列出传统算法(以 Levin 算法为例)与改进算法的一次着色时间(数据来自于普通 PC 机实现: Dual core 1.8 MHz 的 CPU, 2 G 内存, Windows 7 专业版操作系统, 运算平台为 MATLAB 2009a. 图像尺寸(pixel): 800×1400)。传统算法的一次着色时间为 31.3805 s, 而改进算法的运行时间包括 3 部分: 求解混色权重所需时间 31.5384 s, 权重映射花费时间 1.6167 s 以及混色时间 0.2535 s。尽管新算法比传统算法步骤更多, 但两者在单次着色花费的时间相近, 主要是因为两者的运行时间主要都花费在矩阵求逆上, 其时间复杂度为 $O(n^3)$ 。要得到新颜色涂抹下的着色结果, 在调整好涂抹颜色后(可以在调色板上点选实现), 传统算法基于重新着色方式, 计算机每一次都需要运行 31.5384 s, 相反, 改进算法每次运行时间只需要

0.2535 s, 原因在于只需运行混色步骤即可实现, 时间复杂度仅为 $O(n)$ 。所以涂抹颜色改动后的着色结果可实时反馈给系统供用户在线观察, 整个选色过程可交互式进行, 相比之下, 传统算法难以满足实时要求, 而且用户在调色板上调整好涂抹颜色后还须在图像中重新绘制添加, 更增加了用户的时间和精力开销。

6 结束语

由于不同的优化式着色算法基于不同形式的局部模型, 一方面造成难以在理论层面上进行算法比较, 另一方面, 也难以提供全局特性信息, 为此将数学分析引入到着色特性的研究中。首先提出了一种优化着色框架, 并推导出最佳分布为一个调和向量, 在此基础上证明了 Levin 的基于加权平均和的优化算法与基于半监督学习的着色算法等价; 并进一步对优化着色算法的混色机制进行推导, 首先引出图像色度与涂抹色度的组合形式, 进而确定出混色权重的取值范围 $[0,1]$, 并推导出组合系数具有归一化特性, 从而证明了图像中每个像素的颜色由各初始颜色涂抹按比例权重混色而成, 而且论证了各颜色的混色权重的物理意义——像素随机游走首先到达该颜色涂抹的概率。最后根据混色机制设计了一种改进算法, 既可以在线实时选择涂抹的颜色, 也可以通过调整不同对象间的颜色过渡带有效地提高算法对涂抹数量与位置的鲁棒性。

参考文献

- [1] Liu Xiao-ming, Liu Jun, and Feng Zhi-lin. Colorization using segmentation with random walk[C]. Proceedings of International Conference on Computer Analysis of Images and Patterns, Berlin, Germany: Springer-Verlag, 2009: 468-475.
- [2] Welsh T, Ashikhmin M, and Mueller K. Transferring color to grayscale images[J]. *ACM Transactions on Graphics*, 2002, 21(3): 277-280.
- [3] 朱黎博, 孙韶媛, 谷小婧, 等. 基于色彩传递与扩展的图像着色算法[J]. *中国图象图形学报*, 2010, 15(2): 200-205.
Zhu Li-bo, Sun Shao-yuan, Gu Xiao-jing, et al. Image colorization based on color transfer and propagation[J]. *Journal of Image and Graphics*, 2010, 15(2): 200-205.
- [4] An Xiao-bo and Pellacini F. User-controllable color transfer [J]. *Computer Graphics Forum*, 2010, 9(2): 263-271.
- [5] Levin A, Lischinski D, and Weiss Y. Colorization using optimization[J]. *ACM Transactions on Graphics*, 2004, 23(3): 689-694.
- [6] Liu Bei-bei and Lu Zhe-ming. Image colourisation using graph-based semi-supervised learning[J]. *IET Image Processing*, 2009, 3(3): 115-120.
- [7] Chen Ying, Yao Heng, and Zhao Yan. Image colorization with diffusion tensor and edge enhancing[J]. *Journal of Information & Computational Science*, 2011, 8(14): 3329-3336.
- [8] Yao Chen, Yang Xiao-kang, Wang Jia, et al. Patch-driven colorization[J]. *Optical Engineering*, 2010, 49(1): 1-5.
- [9] Kawulok M and Smolka B. Texture-adaptive image colorization framework[J]. *EURASIP Journal on Advances in Signal Processing*, 2011, DOI: 10.1186/1687-6180-2011-99.
- [10] Sheng Bin, Sun Han-qiu, Chen Shun-bin, et al. Colorization using the rotation-invariant feature space[J]. *IEEE Computer Graphics and Applications*, 2011, 31(2): 24-35.

张可为: 男, 1966年生, 博士生, 讲师, 研究方向为图像处理、智能控制。
申群太: 男, 1945年生, 教授, 博士生导师, 研究方向为智能控制、信息处理。