# 自然图像 matting 的闭式解-CVPR

Dezeming Family

2023 年 2 月 10 日

DezemingFamily 系列文章和电子书**全部都有免费公开的电子版**，可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列电子书，可以从我们的网站 [https://dezeming.top/] 找到最新的版本。对文章的内容建议和出现的错误也欢迎在网站留言。

## 目录

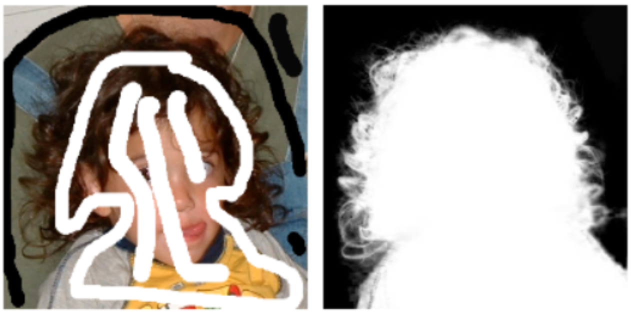
# 一　介绍

Matting 有许多种翻译方法和含义，比如抠图、融合等。这里的本意是指抠图，但是有些词组用“抠图”描述不太好听，我们为了更准确，直接用英文来描述。此外，由于图像 Matting 是一种非常重要的技术，因此有不少更通俗讲解的书籍，比如 [2]，都有介绍。本文只包含对论文 [1] 的讲解，而不包括整个技术历史脉络的描述。如果读者对 matting 有些不明白，那么很可能不太了解本文的动机。

本文 [1] 首先是发布在了 CVPR2006 上，然后后来在 2008 年又发布到了 TPAMI 期刊。本文是参考的 CVPR 版本，会在后面系列中再介绍基于 TPAMI 的。注意 TPAMI 中，用户输入的涂鸦可以不止使用黑白涂鸦，还可以在边界处绘制其他颜色的涂鸦来表示此处是边界区域。本文的公式序号是根据 CVPR 版本来描述的。

## 1 1　摘要

交互 matting，是基于有限的用户输入，来抽取一个前景图像的过程：



从计算机视觉的角度来看，这项任务极具挑战性，因为它是严重的 ill-posed——在每个像素上，我们必须从单个颜色测量中估计前景和背景颜色，以及前景不透明度（“阿尔法蒙版 (alpha matte)”）。当前的方法要么将估计限制在图像的一小部分内，基于已知的附近像素来估计前景和背景颜色，要么通过交替前景和背景色估计与 alpha 估计来执行迭代非线性估计。

在本文 [1] 中，我们提出了一种自然图像 matting 的闭式解。我们从前景和背景颜色的局部平滑度假设中导出了一个损失函数，并表明在得到的表达式中，可以通过分析消除前景和背景色，以获得 alpha 中的二次损失函数 (quadratic cost function)。这允许我们通过求解稀疏线性方程组来找到全局最优的 alpha matte。

此外，闭式公式允许我们通过分析稀疏矩阵的特征向量来预测解的性质，稀疏矩阵与光谱图像分割算法中使用的矩阵密切相关。我们表明，通过惊人的少量用户输入，可以在自然图像上获得高质量的 mattes。

## 1 2　引文

公式如下：

$$I_i = \alpha_i F_i + (1-\alpha_i) B_i, \ (1)$$

$$\boxed{\alpha_i \approx a I_i + b, \quad \forall i \in w, \ (2) \quad a = \frac{1}{F-B}, b = -\frac{B}{F-B}}$$

$$J(\alpha,a,b) = \sum_{j\in I}\left(\sum_{i\in w_j}(\alpha_i - a_j I_i - b_j)^2 + \varepsilon a_j^2\right), \ (3)$$

$$\begin{aligned} J(\alpha) &= \min_{a,b} J(\alpha,a,b). \\ J(\alpha) &= \alpha^T L\,\alpha, \end{aligned} \quad (4)$$

$$\sum_{k|(i,j)\in w_k}\left(\delta_{ij} - \frac{1}{|w_k|}\left(1+\frac{1}{\frac{\varepsilon}{|w_k|}+\sigma_k^2}(I_i-\mu_k)(I_j-\mu_k)\right)\right) \quad (5)$$

$$J(\alpha,a,b) = \sum_k \left\| G_k \begin{bmatrix} a_k \\ b_k \end{bmatrix} - \bar{\alpha}_k \right\|^2, \ (6)$$

$$\begin{aligned} (a_k^*, b_k^*) &= \operatorname{argmin} \left\| G_k \begin{bmatrix} a_k \\ b_k \end{bmatrix} - \bar{\alpha}_k \right\| && (7) \\ &= (G_k^T G_k)^{-1} G_k^T \bar{\alpha}_k && (8) \end{aligned}$$

$$\bar{G}_k = I - G_k (G_k^T G_k)^{-1} G_k^T$$

$$J(\alpha) = \sum_k \bar{\alpha}_k^T \bar{G}_k^T \bar{G}_k \bar{\alpha}_k,$$

$$\delta_{ij} - \frac{1}{|w_k|}\left(1+\frac{1}{\frac{\varepsilon}{|w_k|}+\sigma_k^2}(I_i-\mu_k)(I_j-\mu_k)\right)$$

自然图像 matting 和 compositing 在图像和视频编辑中很重要。一幅图像可以分为前景 **F** 和背景色 **B**，组合公式见公式 (1)，$\alpha_i$ 是前景 opacity。但是公式 (1) 的右边部分都是未知的。因此每个像素（3 通道 RGB 图）而言，有 3 个方程和 7 个未知量。

目前比较流行的方法是用户提供一个粗略的 trimap 作为初始点：

然后程序来求解背景和前景以及 opacity，这通常通过迭代非线性优化来完成，将 **F** 和 **B** 的估计与 $\alpha$ 的估计交替进行。实际上这意味着为了获得好的结果，trimap 中的未知区域必须尽可能小。因此，基于 trimap 的方法实际上在处理大量混合像素（比如头发丝或者轻纱、火焰等很透明的区域）的图像或前景对象有很多孔洞 (holes) 时遇到困难。

在这种具有挑战性的情况下，可能需要大量的经验和用户交互来构建能够产生良好 matte 的 trimap。trimap 界面的另一个问题是，用户无法直接影响图像中最重要的部分的蒙版：混合像素。

这篇文章中，我们提出了一种新的闭式解，来从自然图像中抽取 alpha matte。我们从对前景和背景的局部平滑假设 (local smoothness assumptions) 中导出一个损失函数，并表明在得到的表达式中，可以解析地消除 **F** 和 **B**，从而得到 $\alpha$ 中的二次损失函数 (quadratic cost function)。我们的方法产生的阿尔法蒙版是该损失函数的全局最优值，可以通过求解稀疏线性系统获得。

由于我们的方法直接计算 $\alpha$，而不需要对 **F** 和 **B** 进行可靠的估计，因此用户输入的数量（如稀疏的涂鸦）通常足以提取高质量的遮片。此外，我们的闭式公式可以通过检查稀疏矩阵的特征向量来理解和预测解的属性，稀疏矩阵与光谱图像分割算法中使用的矩阵密切相关。除了为我们的方法提供坚实的理论基础之外，这种分析还可以为用户提供关于应该在图像中放置涂鸦的位置的有用提示。

## 二　相关工作

Corel KnockOut 算法中，假设 **F** 和 **B** 是平滑的，根据已知的前景和背景像素来加权平均（颜色越相近权重值越高）。

Bayesian matting 算法中，使用定向高斯 (oriented Gaussians) 的混合来学习局部分布，然后将 $\alpha$、**F** 和 **B** 估计为给定该分布的最可能值。当前景和背景的颜色分布不重叠，并且 trimap 中的未知区域很小时，该方法可以很好地工作。但是如下图，当 trimap 比较稀疏时，可能会导致完全错误的蒙版：

Poisson matting 也需要 trimap 作为其输入的一部分，并通过使用 matte 梯度场和 Dirichlet 边界条件求解泊松方程来计算混合区域中的 alpha matte。在全局 Poisson matting 方法中，

通过采用合成方程 (compositing equation) 的梯度，忽略 **F** 和 **B** 中的梯度，matte 梯度场近似为 $\nabla\mathbf{I}/(\mathbf{F}-\mathbf{B})$。然后，通过求解梯度尽可能接近近似 matte 梯度的函数来计算 matte。每当 **F** 或者 **B** 在未知区域内不够平滑时，生成的 matte 可能不正确，可能需要交互地应用额外的局部操作到 matte 梯度场中，以获得满意的解决方案，这种交互式细化过程称为局部泊松 matting。正如我们将看到的，我们的方法对 **F** 和 **B** 的行为做出了较弱的假设，这通常会导致更精确的 mattes。

最近有些基于涂鸦或边框的方法，作为 min-cut 问题，将图像进行硬分割（$0-1$ 分割，而不是得到 $\alpha$ 值），这样很容易错过边缘区域：

虽然 Rother 通过在硬边界加一个窄带，拟合参数化 alpha 轮廓来执行边界抠图，这更类似于羽化 (feathering) 操作而不是全图 alpha matting，因为很宽的混合区域不能以这种方式处理：

我们的方法与 Levin 的 colorization 方法和 Grady 的随机步长 alpha matting 方法很相关。这两种方法都通过最小化二次损失将涂鸦约束传播到整个图像。在这里，我们应用了类似的策略，但我们的假设和损失函数进行了修改，以便更好地适应 matting 问题。

Wang 和 Cohen 从一些表示少量背景和前景像素的涂鸦开始，他们使用 belief propagation 迭代地估计图像中每个像素的未知值。虽然这种方法产生了一些不错的结果，但它的缺点是采用了昂贵的迭代非线性优化过程，可能会收敛到不同的局部最小值。

# 三　公式导出

我们首先介绍灰度图的 alpha matting 的闭式解的导出，后面会扩展到彩图。

前面说过，由于 matting 是严重约束不足的，所以需要一些额外的假设。我们假设在一个小窗口内，$\mathbf{F}$ 和 $\mathbf{B}$ 都是近似为常量，但由于 $\alpha$ 可能不是局部平滑的（局部不被近似为常量），因此输入图像 $I$ 可能也不是局部平滑的（这个假设会在后面稍稍放宽一点）。

这个假设允许我们把公式 (1) 表示 $\alpha$ 为作为图像 $I$ 的线性函数，见公式 (2)，可以看到，一个小窗口内的像素共用同一个 $a$ 和 $b$ 值。我们的目标是找到 $\alpha$、$a$ 和 $b$ 来最小化损失函数 (3)。公式 (3) 中可以看到，每个像素都有不同的线性系数，整个图像上的每个窗口 $w_j$ 中相当于各自被最小化该损失函数（窗与窗如果没有重叠，则两个窗之间就没有多少关联）。

我们可以看到公式 (3) 有一个对 $a$ 的正则化项，这是为了提高数值稳定性。比如，如果一个图像在第 $j$ 的窗口中是常量，没有先验，$a_j$ 和 $b_j$ 不能被唯一确定（如果没有正则化项，那么当 $w_j$ 中的 $I_i$ 值都相同时，那么在这个窗口内，$a_j$ 和 $b_j$ 有无数种组合可以使得最小化损失函数，因此没有闭式解）。而且这个对 $a$ 正则化可以使得 $\alpha$ 更平滑（$a_j$ 越小表示 $\alpha$ 越平滑，$a_j$ 为 0 就表示第 $j$ 个窗口内 $\alpha$ 是常量值）。

我们在实现中一般选择 $3\times 3$ 的窗，每个像素都有窗，所以相邻窗之间是重叠的。这使得相邻像素间的信息可以传递。损失函数是二次的，在 $N$ 个像素中有 $3N$ 个未知（$a$、$b$ 和 $\alpha$）。幸运的是，根据推导，$a$ 和 $b$ 是可以消去的，因此相当于二次损失函数有 $N$ 个未知量（推导过程在论文原文 [1] 的 Theorem 1 中有，属于比较简单的最小二乘优化问题，但是至于为什么可以先固定 $\alpha$ 然后优化 $a$ 和 $b$，我个人是有点疑问的）。公式 (4) 是我们的新优化项，只需要优化 $\alpha$ 即可。

# 四　彩色图像

公式如下：

$$\alpha_i \approx \sum_c a^c I_i^c + b, \quad \forall i \in w \tag{9}$$

$$J(\alpha, a, b) = \sum_{j \in I} \left( \sum_{i \in w_j} \left( \alpha_i - \sum_c a_j^c I_i^c - b_j \right)^2 + \varepsilon \sum_c a_j^{c^2} \right) \tag{10}$$

$$J(\alpha) = \alpha^T L \alpha. \tag{11}$$

Here $L$ is an $N \times N$ matrix, whose $(i, j)$-th element is:

$$\sum_{k|(i,j) \in w_k} \left( \delta_{ij} - \frac{1}{|w_k|} (1 + (I_i - \mu_k)(\Sigma_k + \frac{\varepsilon}{|w_k|} I_3)^{-1} (I_j - \mu_k)) \right) \tag{12}$$

彩图可以用线性模型 (9) 来表示（公式 (9) 的推导见论文 [1] 的 Theorem 2），这样可以放宽前面的 **F** 和 **B** 都是常量的假设。在一个窗口内，假设其中所有像素 **F** 和 **B** 是两种颜色的线性组合是足够的，换句话说：

$$F_i = \beta_i F_1 + (1 - \beta_i) F_2$$
$$B_i = \lambda_i B_1 + (1 - \lambda_i) B_2$$

这被叫做颜色线性模型 (color line model)。这样的模型很有用，因为它可以捕捉到例如具有恒定反照率的表面上的变化阴影，还有比如窗口包含两个颜色一致的区域之间的边缘，这两个区域都属于背景或前景。此外，Omer 和 Werman 证明，在许多自然图像中，RGB 空间中的像素颜色倾向于形成相对少量的细长簇。尽管这些簇不是直线，但它们在局部区域大致呈线性。

使用 $4D$ 线性模型 (9)，我们定义了公式 (10) 损失函数，并且约掉 $a^c$ 和 $b$，以得到公式 (11) 和 (12) 的二次损失函数。我们将 (5) 和 (12) 定义为 matting Laplacian。注意 $L$ 的每行元素之和为 0，因此 $L$ 的零空间包括常数向量（零空间，即 $L\mathbf{v} = \mathbf{0}$ 的所有 $\mathbf{v}$ 的集合），如果使用 $\epsilon = 0$，那么 $L$ 的零空间也会包含 $I$ 的每个颜色通道。

# 五　用户接口与光谱分析

公式如下：

$$\alpha = \operatorname{argmin} \alpha^T L \alpha, \quad \text{s.t. } \alpha_i = s_i, \forall i \in S \tag{13}$$

$$W_G(i, j) = e^{-\|I_i - I_j\|^2 / \sigma^2}, \tag{14}$$

$$W_M(i, j) = \sum_{k|(i,j) \in w_k} \frac{1}{|w_k|} (1 + (I_i - \mu_k)(\Sigma_k + \frac{\varepsilon}{|w_k|} I_3)^{-1} (I_j - \mu_k)) \tag{15}$$

用户可以用黑刷涂抹背景像素（$\alpha = 0$），用白刷涂抹前景像素（$\alpha = 1$）。为了计算出 alpha matte 来满足用户的涂鸦，我们求解公式 (13)。其中，$\alpha_i = s_i$ 是约束项，满足在涂鸦区域 $S$ 要满足用户标注的涂鸦值 $s_i$。

根据 Theorem 3，如果 **F** 和 **B** 在每个局部窗口都满足颜色线性模型，当 $L$ 构建中 $\epsilon = 0$ 则求解得到的 $\alpha^*$ 就是系统 (13) 的最优解（也就是保证公式 (10) 的值为 0）。

下面有些相对比较深的数学描述，分析了一些比较早期的算法，这个作者是有不少图论的背景的，比如图论中图像到图的映射关系，这里我尽量描述地通俗一些。

matting Laplacian matrix $L$ 是一个对称正定矩阵，可以写为 $L = D - W$ 的形式，即 $L$ 是用于分割的谱方法中使用的 graph Laplacian，但是亲和度函数 (affinity function，也可以称为相似性函数) 是新的（公式 (12)）。

对于 normalized-cuts 算法（用于超像素分割的硬边界分割）和我们的算法（亲和度函数分别见公式 (14) 和 (12)），我们把 $L$ 表示为 $D - W$ 就得到了新的亲和度公式 (15)。

考虑理想的边缘情况，在一个窗内仅有两种颜色，在亲和度公式 (15) 这种情况下，相同颜色之间的相似性会因为距离而下降；而不同颜色之间的相似性为 0（它的证明放在了本文作者的另一篇同名的报告中）。

matting 亲和性没有全局缩放参数 $\sigma$，而是使用均值和方差的局部估计。正如我们随后所展示的，这种自适应特性会显著提高性能，即 scaling 参数的局部调整改善了图像分割结果。

为了比较这两个亲和函数，我们检查了对应的 Laplacians 的最小特征向量，因为这些特征向量被光谱分割算法用于分割图像。由于最小的特征向量是图像常量，因此我们看第二小的特征向量。可以看到在复

杂的边界（前景有不少区域都是透明的）中，全局 $\sigma$ 的特征向量很难去捕获很好的边界。在这里 matting Laplacian 计算的 $\epsilon$ 值为 0.0001。

根据 matting Laplacian 的最小特征向量可以指导用户在何处画涂鸦。例如，在最小特征向量分段恒定的相同区域中，提取的 matte 也趋于分段恒定。如果特征向量图像中的一个分区的值是连贯的，在这个分区中的涂鸦足以将这个目标值（即你对这个区域希望它是背景还是前景，赋予的涂鸦颜色）传递到整个分区中。另外一方面，在图像上特征向量的值较少连贯的困难区域中，建议多做一些涂鸦。

更确切地说，alpha matte 可以通过检查 matting Laplacian 矩阵的最小特征向量来确认，因为公式 (13) 的最优解是更小的特征向量生成尽可能大的区域。事实上，可以限制权重，最优解会分配更大的特征向量，作为对应特征值的比值的函数。

## 六　源码实现

源码实现放在本栏目的其他文章中进行描述。

## 参考文献

[1] Levin, A. . ”A closed form solution to natural image matting.” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Computer Society, 2006.

[2] Radke R J . Computer vision for visual effects. Cambridge University Press, 2010.

[3] Szeliski R . Computer Vision: Algorithms and Applications[M]. Springer-Verlag New York, Inc. 2011.