

Spectral Matting

Anat Levin Alex Rav-Acha Dani Lischinski

School of CS & Eng CSAIL

The Hebrew University MIT

光谱抠图

原文链接：<http://www.vision.huji.ac.il/SpectralMatting/spectralMatting.pdf>

译文出处：<https://github.com/hyifan/TranslatePapers>

摘要

我们提出光谱抠图 (spectral matting), 一种新的自然图像抠图的方法, 自动从适当定义的拉普拉斯矩阵 (suitably defined Laplacian matrix) 的最小特征向量计算一组基本的模糊 (fuzzy) 抠图分量 (matting components)。因此, 我们的方法扩展了以提取硬片段 (hard segments) 为目标的光谱分割技术 (spectral segmentation techniques), 使得能提取软抠图分量 (soft matting components)。然后, 可以将这些分量构建成块, 以便以无监督的方式或基于少量的用户输入, 轻松构建语义上有意义的前景遮罩 (matte)。

1. 介绍

数字抠图是从图像中提取前景对象的过程, 同时对对象覆盖的每个像素进行不透明度估计。此操作可以在新的背景上合成提取的对象, 因此在图像编辑、视频制作和电影特效中是一个非常宝贵的工具。

尤其是背景没有约束的自然图像抠图这一具有挑战性的案例受到了广泛的研究关注。认识到问题本质上是受约束的, 所有现有的方法都要求用户以三值图或一组笔画的形式提供额外的约束。因此, 是否 (或在多大程度上) 可以自动化抠图过程的问题, 具有相当大的理论和实践意义。

本文试图对这个问题提出一些新的见解。我们的工作受到光谱分割方法的强烈影响。这些方法通过检查图像的抠图拉普拉斯矩阵的最小特征向量来实现无监督图像分割。这项工作, 第一次扩展了从生产硬片段到软抠图分量这一想法。

光谱分割的方法, 如文献[12], 为了将一个 NP 完全优化问题转化为易于处理的问题, 通常需要近似估计实值 (real-valued) 特征向量。相反, 我们不是寻找一个不相交的图像分割法, 而是尝试恢复每个像素的部分前景覆盖。具体来说, 我们通过 Levin 等人介绍的抠图拉普拉斯矩阵的最小特性向量的线性转变来获得实值抠图分量。一旦获取成功, 这些抠图分量将为提取完整的前景遮罩而构建成块。

这个概念如图 1 所示。给定如图 1a 的输入图像, 可以使用无监督的不相交硬分割技术来生成图像, 如文献[18] (图 1b)。相反, 我们计算一组重叠的、少量的抠图分量, 如图 1d 所示。将其中三个分量 (用

红色框住的)组合在一起会生成女孩的前景遮罩,如图 1c 所示。

综上所述,我们的主要贡献是介绍了基本抠图分量的概念以及由此产生的第一个无监督抠图算法。当然,就像无监督分割技术一样,无监督抠图也是一个不适定的问题。因此,我们还描述了两个扩展,它们使用这些基本的抠图分量来构造一个特定的遮罩:(i)向用户提供几个可供选择的抠图选项;或(ii)让用户通过单击几下鼠标来指定其意图。

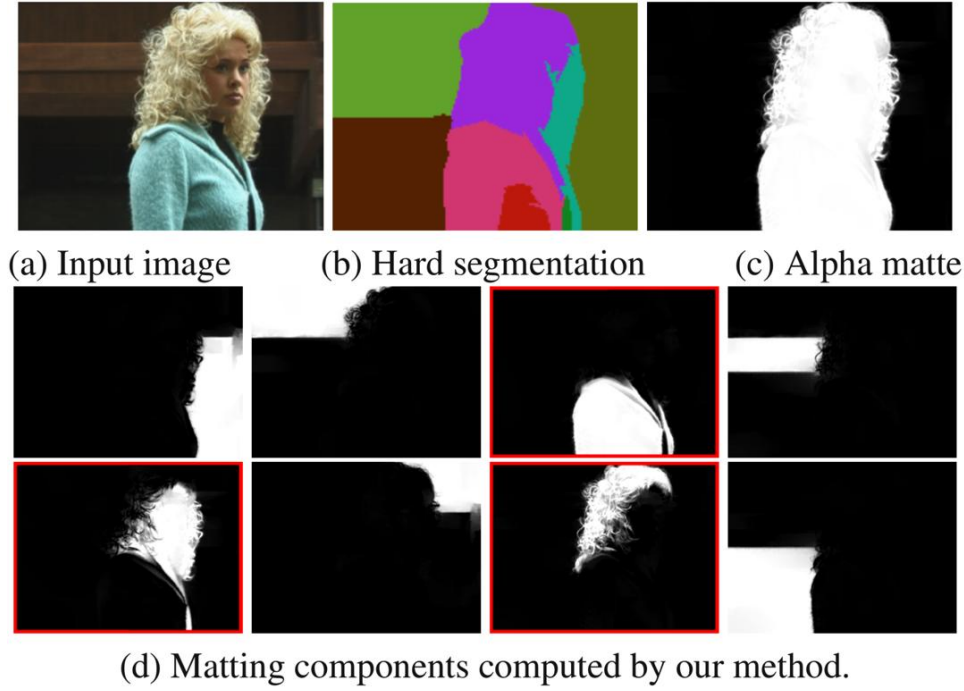


图 1 光谱分割和光谱抠图

2. 抠图分量 (Matting Components)

抠图算法通常假定输入图像中的每个像素 I_i 是前景色 F_i 和背景色 B_i 的线性组合:

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

这被称为合成方程(compositing equation)。在这项工作中,我们通过假设每个像素是 K 层图像 F^1, \dots, F^K 的凸组合来概括合成方程:

$$I_i = \sum_{k=1}^K \alpha_i^k F_i^k \quad (2)$$

K 个向量 α^k 是图像的抠图分量,它指定每个层对每个像素处观察到的最终颜色的少量贡献。抠图分量是非负的,每个像素的总和为 1。拥有这些分量的直观动机是,与过度分割图像中的单个低级片段类似,它们可用于构建更高级别、语义上有意义的前景遮罩,如图 1 所示。

一个理想的、虽然不是必需的关于抠图分量的属性是稀疏性(sparsity): 每个分量在尽可能多的图像像素上应该是完全不透明的,或者是完全透明的。这意味着不同层之间的过渡区域局限在少量像素,并且每个像素受少量层的影响。

本文，我们探究了抠图分量和抠图拉普拉斯矩阵的特性向量之间的关系。具体地说，我们表明在某些确定的假设下，抠图分量是由抠图拉普拉斯的最小特征向量构成的。然后，我们提出了一种通过找到合适的线性变换并将其应用于这些特征向量来计算抠图分量的方法。

3.光谱分析 (Spectral Analysis)

我们首先简要回顾光谱分割方法的基本理论。这些方法通常与图像的 $N \times N$ 亲和矩阵 A 相关联，例如 $A(i, j) = e^{-d_{ij}/\sigma^2}$ ，其中 d_{ij} 是像素之间距离的某种度量（例如颜色差分 and 几何距离）。然后可以定义拉普拉斯矩阵 $L = D - A$ ，其中 D 是对角矩阵 $D(i, j) = \sum_j A(i, j)$ 。 L 是一个对称的半正定矩阵，其特征向量捕获了大部分的图像结构。

考虑这样一个理想的情况，亲和矩阵 A 精确地捕捉到这样一个事实，即图像由几个不同的簇或连通分量 (connected components) 组成。也就是说，如果 $A(i, j) = 0$ 对于每一个 i, j ，使得 $i \in C, j \notin C$ ，则图像像素的子集 C 是图像的连通分量，并且不存在满足该属性的 C 的子集。让 m^C 表示分量 C 的指示向量 (indicator vector)，

$$m_i^C = \begin{cases} 1 & i \in C \\ 0 & i \notin C \end{cases},$$

那么 m^C 是 L 的特征值为 0 的特征向量。

现在假设图像由 K 个连通分量 C^1, \dots, C^K 组成，令 $\{1, \dots, N\} = \bigcup_{k=1}^K C^k$ ，其中 C^k 为像素的不相交子集。在这种情况下，指示向量 m^{C^1}, \dots, m^{C^K} 都是 L 的特征值为 0 的独立正交特征向量。然而，计算 L 的特征向量只能得到旋转 (rotation) 之前的这些指示向量。这是因为对于任何 $K \times K$ 旋转矩阵 R 的向量 $[m^{C^1}, \dots, m^{C^K}]R$ 也是 L 的零空间的一组基。

在真实图像中，亲和矩阵 A 很少能够在不同的像素簇之间完全分离。因此，拉普拉斯 L 通常不具有多个特征值为零的特征向量。然而，据观察， L 的最小特征向量在相干图像分量中趋于近似常数。从最小特征向量中提取不同分量的方法称为谱舍入法 (spectral rounding)，已引起广泛关注。最简单的方法是使用 k 均值算法对图像像素进行聚类，并利用摄动分析 (perturbation analysis) 将该算法的误差作为聚类内和聚类间连接性的函数进行约束。其他更近期的方法，启发了在这项工作中采取的方法，显式地搜索一个旋转矩阵，使特征向量尽可能接近二进制指示向量。

3.1 使用抠图拉普拉斯的光谱分析 (Spectral Analysis with the Matting Laplacian)

我们在这项工作中的目标是在硬分割和抠图之间进行类比，并证明可以从抠图拉普拉斯的最小特征向量中提取模糊抠图分量，类似于前面描述的硬簇提取。Levin 等人引入了抠图拉普拉斯，目的是在不明确估计等式 (1) 中前景和背景色的情况下评估遮罩的质量。它们表明，如果局部图像窗口 w 中背景和前景的颜色在 rgb 空间中形成两条不同的线，则 w 中的 α 值可以表示为颜色通道的线性组合：

$$\forall i \in w \quad \alpha_i = a^R I_i^R + a^G I_i^G + a^B I_i^B + b \quad (3)$$

因此,遮罩提取问题变成了在所有图像的窗口 w_q 中,寻找 α 遮罩使得与线性模型(3)的偏差最小化:

$$J(\alpha, a, b) = \sum_{q \in I} \sum_{i \in w_q} (\alpha_i - a_q^R I_i^R - a_q^G I_i^G - a_q^B I_i^B - b_q)^2 + \epsilon \|a_q\|^2 \quad (4)$$

其中 $\epsilon \|a_q\|^2$ 是 a 上的正则化项。线性模型系数 a, b 可以从方程(4)中消除,产生只含 α 的二次代价函数,

$$J(\alpha) = \alpha^T L \alpha \quad (5)$$

这里的 L 是抠图拉普拉斯,一个稀疏对称的半正定 $N \times N$ 矩阵,其条目是局部窗口中输入图像的函数,既不依赖未知的前景和背景颜色,也不依赖线性模型系数。 $L(i, j)$ 定义为:

$$\sum_{q | (i, j) \in w_q} \left(\delta_{ij} - \frac{1}{|w_q|} \left(1 + (I_i - \mu_q)^T \left(\sum_q + \frac{\epsilon}{|w_q|} I_3 \right)^{-1} (I_j - \mu_q) \right) \right) \quad (6)$$

其中 δ_{ij} 是 Kronecker delta, μ_q 是窗口 w_q 中围绕像素 q 的 3×1 平均值颜色向量, \sum_q 是同一窗口中的 3×3 协方差矩阵, $|w_q|$ 是窗口中的像素数, I_3 是 3×3 单位矩阵。

代价函数(5)有一个小的最小值,它是一个常数 α 向量,因此在文献[7]中描述的用户辅助框架中, $J(\alpha)$ 在用户约束下被最小化。Levin 等人观察到抠图拉普拉斯(6)的最小特征向量捕捉图像中像素的模糊聚类分配信息,甚至在考虑任何用户指定的约束之前。但是,它们除了将特征向量呈现给用户作为涂鸦放置的指导之外,没有使用其他特征向量。在这项工作中,我们证明了最小的特征向量跨越图像的单个抠图分量。

为了获得一些理解,我们从研究理想情况开始。为了证明光谱分析用于估计抠图分量的合理性,我们的目标是证明在合理的条件下,实际抠图分量属于抠图拉普拉斯的零空间。我们认为,如果存在像素 $i \in w$ 使得 $\alpha_i^k > 0$ 的,那么抠图分量 α^k 在局部图像窗口 w 中是活动的(active)。以下要求规定了在每一层局部颜色的条件,其中 $L\alpha^k = 0$ 。条件的严重性与本地窗口中活动层的数量有关。最不受限制的情况是只有一个层处于活动状态,其中局部颜色分布可能是任意复杂的。最受限制的情况是当一个窗口包含三个活动层(如 T 形连接)时,对于这种窗口,每个层的颜色必须局部一致。

要求 1

假设 $\alpha^1, \dots, \alpha^K$ 是图像 I 实际分解的 k 个抠图分量。如果每个局部图像窗口 w 满足下列条件之一,则向量 $\alpha^1, \dots, \alpha^K$ 位于抠图拉普拉斯 L 的零空间中(由等式 6 给出,其中 $\epsilon = 0$):

1. 单个分量 α^k 在 w 是活动的。
2. 两个分量 α^{k1}, α^{k2} 在 w 中是活动的,而 w 中对应层 F^{k1}, F^{k2} 的颜色位于 RGB 空间中的两条不同的线上。
3. 三个分量 $\alpha^{k1}, \alpha^{k2}, \alpha^{k3}$ 在 w 是活动的,每一层 F^{k1}, F^{k2}, F^{k3} 有恒定的颜色在 w , 三种颜色是线性无关的。

证明:

抠图代价函数(5)测量所有局部窗口上的遮罩与颜色通道的线性函数之间的偏差(4)。因此,为了

证明一个遮罩分量 α^k 满足 $L\alpha^k = 0$,对于每个局部窗口 w ,存在 a^R 、 a^G 、 a^B 、 b ,这样： $\alpha_i^k = a^R I_i^R + a^G I_i^G + a^B I_i^B + b$, $\forall i \in w$ 。下面我们为三种窗口类型中的每一种显示这个。

情况 1:

由于每一个图像像素的抠图分量之和为 1 ,因此单个活动分量 α^k 必须在 w 内等于 1。因此 ,通过设置 $a^R=a^G=a^B=0$ 和 $b=1$, 它很容易表示为图像的线性函数。

情况 2:

这个情况等价于文献[7]中的定理 2。

情况 3:

由于 F^{k1} 、 F^{k2} 、 F^{k3} 在 w 内是常数 ,且它们的颜色是线性无关的 ,因此存在 a^R 、 a^G 、 a^B 和 b ,使得 $\langle a, F^{k1} \rangle + b = 1$ 、 $\langle a, F^{k2} \rangle + b = 0$ 和 $\langle a, F^{k3} \rangle + b = 0$ 。当 $I = \alpha^{k1} F^{k1} + \alpha^{k2} F^{k2} + \alpha^{k3} F^{k3}$ 时 , 我们得到 $\langle a, I \rangle + b = \alpha^{k1}$, 因此 α^{k1} 是图像的线性函数。 α^{k2} 和 α^{k3} 也有类似的论据。

在标准的拉普拉斯的情况下 ,当计算抠图拉普拉斯的最小特征向量时 ,结果可能是不同抠图分量的线性组合 ,恢复单个分量相当于线性变换特征向量。应注意 ,与硬分割不同 ,抠图分量不是二进制向量 ,因此不一定是正交的。因此 ,虽然特征向量是正交的 ,但从特征向量到抠图分量的转换可能是一般的线性转换 ,而不是简单的旋转。

综上所述 ,上述讨论的主要结论是 ,当图像的抠图分量满足要求 1 的条件时 ,它们可以表示为 L 的零特征向量的线性组合。

在大多数真实图像中 ,要求 1 的假设并不完全成立 ,因此抠图拉普拉斯可能没有多个特征值为 0 的特征向量。然而 ,如果层足够明显 ,它们通常被 L 的最小特征向量捕获。例如 ,图 2 显示了真实图像的最小特征向量 ,所有特征向量都显示了模糊层边界。我们已经从经验上观察到 ,真实图像的抠图分量通常由抠图拉普拉斯的最小特征向量很好地覆盖。实际上 ,图 1d 所示的分量是以最小特征向量的线性组合获得的。



图 2 图 1a 抠图拉普拉斯的最小特征向量。这些特征向量的线性组合产生图 1d 所示的抠图分量。

3.2 从特征向量到抠图分量 (From Eigenvectors to Matting Components)

如上所述，恢复图像的抠图分量相当于找到特征向量的线性变换。回想一下，由于大多数图像像素通常是不透明的，因此每个图像像素的抠图分量总和应为 1，对于大多数图像像素，抠图分量应接近 0 或 1。因此，我们正在寻找一个特征向量的线性变换，它将产生一组接近二进制的向量。更正式地说，让 $E=[e^1,...,e^K]$ 是特征向量的 $N \times K$ 矩阵。我们的目标是找到一组 K 线性组合向量 y^k ，使最小化

$$\sum_{i,k} |\alpha_i^k|^\gamma + |1 - \alpha_i^k|^\gamma \quad \text{其中 } \alpha^k = Ey^k$$
$$\text{s.t. } \sum_k \alpha_i^k = 1$$
(7)

如果 $0 < \gamma < 1$ 被使用（在我们的执行中 $\gamma=0.9$ ），那么 $|\alpha_i^k|^\gamma + |1 - \alpha_i^k|^\gamma$ 是一个测量抠图分量稀疏性的鲁棒得分（robust score）。无需 $\alpha^k = Ey^k$ 通过二进制向量就可以最小化稀疏项，但随着向量 α^k 仅限于特征向量的线性组合，他们必须保持模糊层边界。尽管我们没有明确限制 0 至 1 之间的 α 值，但在实践中，由于稀疏惩罚（sparsity penalty），结果值往往在这一范围内。上一个代价函数是一个非凸函数，我们通过构建一个二阶近似序列（该序列涉及一个 $K^2 \times K^2$ 线性系统的解决方案），用 Newton 的方法来迭代地优化它。更多细节可以在文献[8]中找到。

由于代价函数（7）不是凸面的，Newton 过程的结果很大程度上取决于初始化的质量。初始化过程的一个有用方法是在抠图拉普拉斯的最小特征向量上应用 k 均值算法，并将结果簇的指示向量投影到特征向量 E 的跨度上：

$$\alpha^k = EE^T m^{C^k}$$
(8)

可以证明，由此产生的抠图分量之和为式（7）提供了一个合法的解决方案。

在实践中，我们通常使用比要恢复的抠图分量数量更多的特征向量。使用更多的特征向量可以获得更稀疏的分量。原因是更多的基元素跨越一组更丰富的向量（在极端情况下，如果使用所有 N 个特征向量，则可以生成任何二进制向量）。

4. 组分量 (Grouping Components)

到目前为止，我们已经展示了如何从抠图拉普拉斯提取抠图分量。然而，抠图分量通常不是我们的目标，因为人们最终是对恢复某个前景对象的完整的遮罩感兴趣。幸运地是，要获得完整的遮罩，只需指定某些分量属于前景。假设 $\alpha^{k_1}, ..., \alpha^{k_n}$ 被指定为前景组件，然后简单地将它们加在一起即可得到完整的前景遮罩：

$$\alpha = \alpha^{k_1} + \dots + \alpha^{k_n}$$
(9)

例如，图 1c 中的遮罩是通过将图 1d 中以红色突出显示的分量加在一起获得的。

对于下面讨论的应用，我们希望比较多个分组假设，从而测量得到的 α 遮罩的质量为 $J(\alpha) = \alpha^T L \alpha$ ，其中 L 为抠图拉普拉斯（6）。当要测试大量假设时，将每个假设乘以 L 可能太昂贵。然而，如果每个假设

只是一个抠图分量的总和，我们可以通过 L 预先计算抠图分量之间的相关性，并将它们存储在 $K \times K$ 矩阵 Φ 中，其中

$$\Phi(k, l) = \alpha^{k^T} L \alpha^l \quad (10)$$

遮罩代价函数可以被写成

$$J(\alpha) = b^T \Phi b \quad (11)$$

其中 b 是表示所选分量的 K 维二进制分量。因此，如果预先计算了 Φ ，则可以在 $O(K^2)$ 操作中而不是 $O(N)$ 操作中评估 $J(\alpha)$ 。

4.1 无监督抠图 (Unsupervised Matting)

给定一个图像和一组抠图分量，我们希望将这些分量拆分为前景和背景组，并拉出前景对象。如果分组标准只考虑低级提示，那么我们只需要搜索具有最佳抠图代价函数的分组，如等式 (11) 所定义。然而，抠图代价函数通常偏向于仅将非常量值分配给图像像素的一小部分的遮罩（在极端情况下，最佳遮罩是一个常量）。光谱分割文献提出了几个克服这种偏见的标准。一种方法是搜索商切割（例如，标准化切割[12]），该切割得分为切割成本与结果簇大小之间的比率。第二种方法是寻找平衡切割[6]，其中每个簇的大小被限制在图像大小的一定百分比以上。在这项工作中，我们遵循后一种方法，通过只考虑将至少 30% 的像素分配给前景和至少 30% 的像素分配给背景的分组来排除琐碎的解决方案。当抠图分量的 K 数很小时，我们可以列举所有的 2^K 假设，并使用等式 (11) 选择得分最高的假设。

图 3 显示了上述无监督抠图方法产生的一些结果。在每一个例子中，得分最高的假设确实对应于“正确的”前景遮罩，但考虑到我们的方法不试图执行任何高层次的图像理解，其他一些假设也相当合理。当然，无监督抠图失败的例子也不难找到。例如，当前景或背景对象由几个视觉上不同的分量组成时，以最小的抠图代价函数分配可能与我们的视觉感知不符。事实上，众所周知，在图像分割领域中，无监督的自下而上提示可以有效地对图像中的相干区域进行分组，但一般的图像分割问题本身就不明确，需要额外的信息。在实践中，例如在硬图像分割的情况下，前景/背景分配可以由几个附加的提示来指导，例如自顶向下的模型[1]、颜色统计[11]或运动和焦点提示。然而，在本文的其余部分中，我们将重点放在用户引导的抠图上。

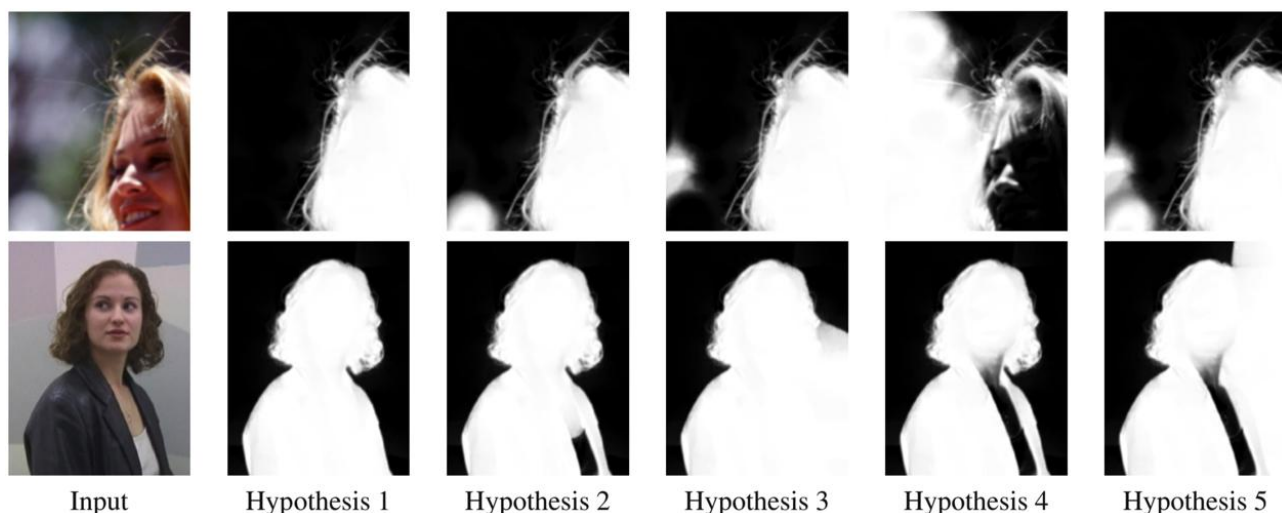


图 3 两张图片的无监督抠图结果。这些假设是根据它们的得分来排序的。

4.2 用户引导的抠图 (User-Guided Matting)

抠图分量在交互设置中也非常有用，在交互设置中，用户指导抠图过程以提取所需的前景遮罩。在这种设置中，某些分量的前景/背景分配由用户确定，从而减少了要测试的合法假设的数量。考虑到非常小的前景和背景约束，通常可以排除一些无关紧要的解决方案，因此不需要显式地将每个组的大小保持在某个阈值之上（如在不监督的情况下）。在这种情况下，我们可以将抠图代价函数（11）近似为成对的项的总和。这使我们能够将搜索最佳前景/背景分配近似为图中的最小切割问题，图中的节点是抠图分量，其边缘权重表示抠图惩罚（有关详细信息参见文献[8]）。在这个构想中，寻找最优分配不涉及指数搜索，并且可以有效地在时间多项式中找到分量的个数。因此，如果预先计算了抠图分量，则可以非常快速地计算出最佳遮罩，从而对用户输入进行交互式响应。我们算法的计算挑战相当于传统的光谱分割技术。具体来说，我们未优化的 matlab 实现需要几分钟的时间来计算图 4 中图像的抠图分量。然而，这个预处理步骤可以离线完成，一旦抠图分量可用，在给定用户约束的情况下，构造遮罩只需要几秒钟。

图 4 给出了几个示例，根据用户提供的少量前景（白色）和背景（黑色）标记从图像中提取前景遮罩。第二列显示通过上述方法提取的结果遮罩（涂鸦用于减少拆分假设的空间：当分量的区域包含白色涂鸦时，分量被约束为属于前景）。其余的列显示由前面的许多方法（7、16、4、14）从相同的输入生成的遮罩。这些以前的方法都不能从这么少的用户输入中恢复合理的遮罩。特别是，尽管我们的方法使用了与文献[7]相同的抠图拉普拉斯，但我们的结果与根据用户指定的约束直接最小化二次抠图代价函数（5）得到的结果非常不同。这种直接优化的主要缺点是，每当图像包含不受任何约束的不同连接分量时，二次代价函数（5）往往会为它们分配一些平均非不透明值，如图 5 中的简单示例所示。这个问题的核心是文献[7]的二次代价函数对前景和背景分布有很强的假设，但对 α 没有限制。因此，它搜索连续的解，而没有考虑到，对于大多数不透明的前景对象，遮罩在大多数图像上应该严格为 0 或 1。

一旦计算了图像的抠图分量，通过一组涂鸦或三值图设置硬约束并不是用户指定其意图的唯一方法。

抠图分量建议一种新的、更直接的用户交互模式，这种模式直到现在都不可能实现：在这种模式下，用户可以看到预先计算的抠图分量，并且可以简单地将其中一些分量标记为背景或前景。然后，标记的分量在最小切割问题中相应地受到约束。这种接口的优点如图 6 所示，其中较大的模糊头发区域不适合放置硬约束。因此，实际上我们可以期待最好的三值图使这些区域不受约束（图 6c）。文献[7]的最小二乘法用平均灰度值填充这些区域（图 6d）。相反，通过搜索与三值图一致的最劣质的抠图分量分配，我们获得了图 6e 中的遮罩。在这种情况下，不会观察到过度平滑，但某些模糊头发未被选择为属于前景。然而，如果允许用户直接选择三个附加分量（在图 6g 中以红色突出显示）作为前景，我们将获得图 6f 中的遮罩。

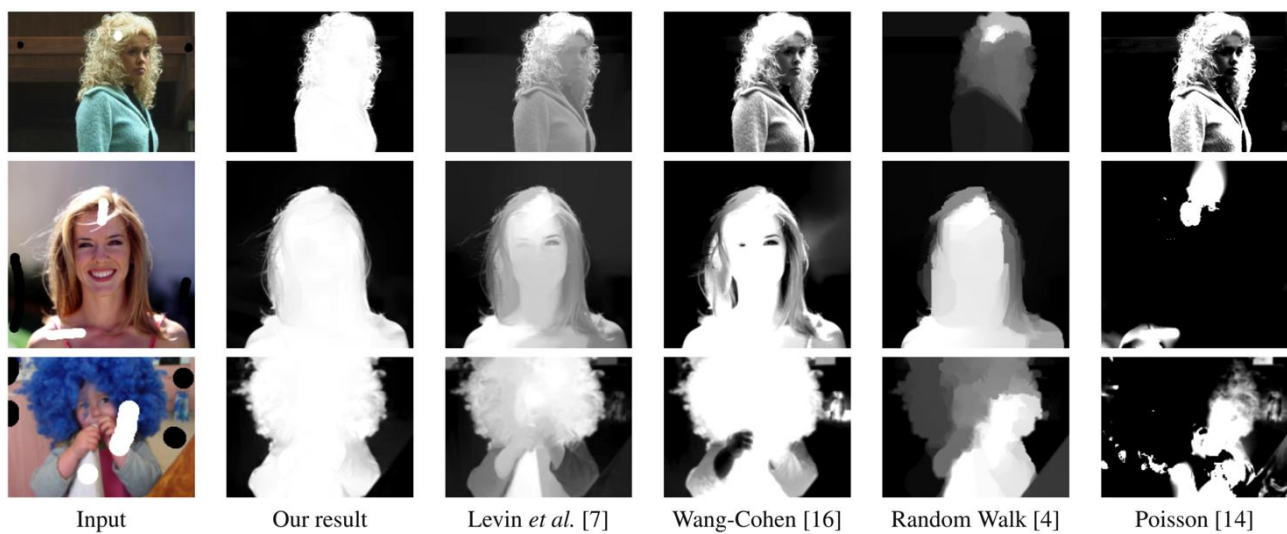


图 4 从最小用户输入中比较不同抠图方法产生的遮罩

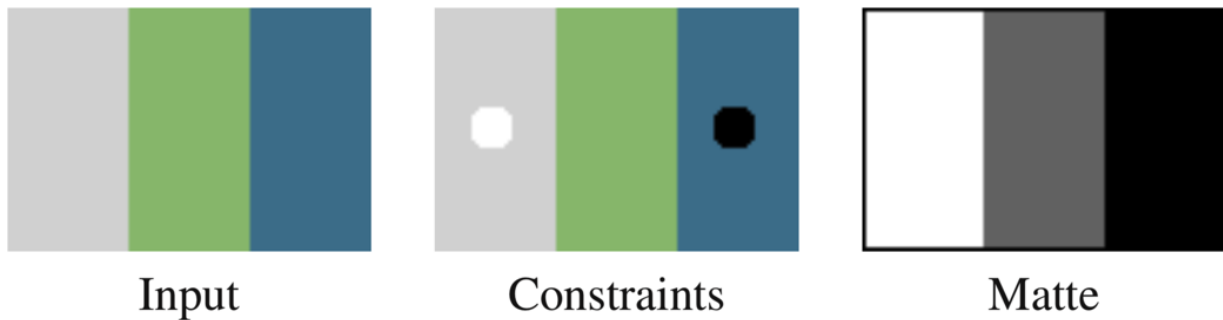


图 5 中间区域不受约束，Levin 等人的方法为其分配一个平均非不透明值

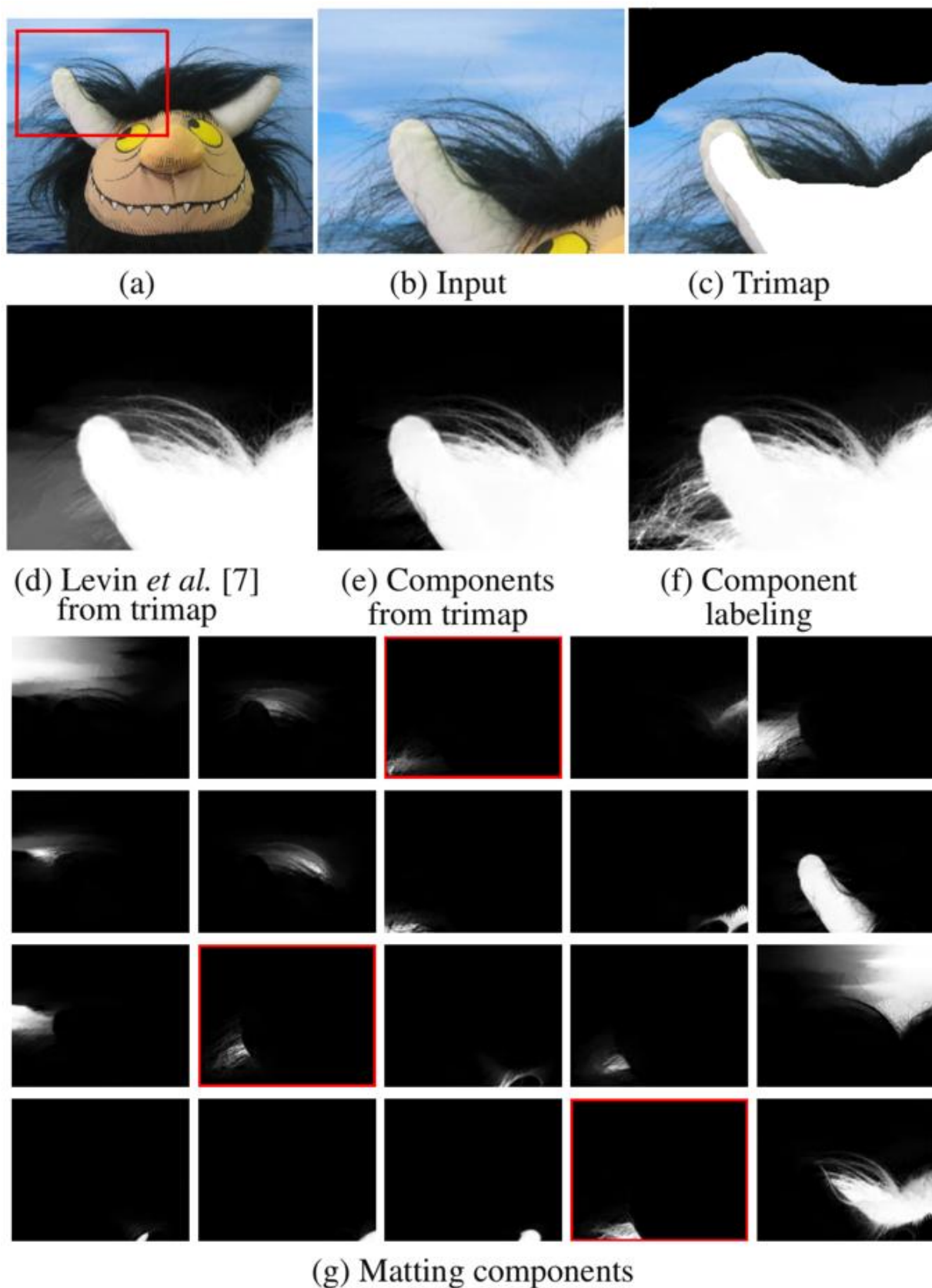


图 6 指示分量标签 (direct component labeling) 的好处

5.定量评估 (Quantitative evaluation)

为了定量评估我们的方法并与以前的方法进行比较，我们获取了真值数据。在一台电脑显示器前拍摄

了三个不同的玩偶，显示了七个不同的背景图像（图 7a）。然后，使用最小二乘框架为每个玩偶提取了一个真值遮罩。每幅图像被缩小到 560×820 像素，下面描述的测试是在从这些图像中裁剪的 200×200 窗口上进行的（重叠）。对于我们的方法，使用每个裁剪窗口的 70 个最小特征向量提取 60 个抠图分量。对于每个 200×200 的窗口，我们未优化的 matlab 实现（在 3.2Ghz CPU 上）的运行时间是几分钟。

为了设计使用抠图分量的遮罩提取法和以前的抠图算法之间的比较，我们需要处理两个不兼容的接口，并且还不清楚如何测量每种情况下涉及的用户工作量。虽然以前的方法被设计为使用硬约束（涂鸦或三值图），但我们的新方法通过分量选择实现了新的交互模式。因此，在我们的实验中，我们试图确定每一种方法在给定最佳用户输入的情况下能做得多好。所以，我们首先使用真值遮罩生成“理想的”三值图。该三值图中的未知区域是通过将真值遮罩值在 0.05 和 0.95 之间的所有像素放大 4 个像素来构造的。所得的三值图用作之前 4 种抠图算法的输入：Levin 等人、Wang 和 Cohen、random walk 抠图和泊松抠图。我们在每个实验中也运行了两次我们的方法：(i) 使用相同的三值图提供抠图分量的部分标签，然后进行最小切割计算，如第 4.2 节所述；和 (ii) 使用真值遮罩选择抠图分量的子集，使产生的遮罩与真值的距离最小，从而通过分量直接选择界面模拟理想的用户输入。不同方法产生的遮罩和真值遮罩之间的 SSD 误差（在不同背景 and 不同窗口上的平均值）如图 7b 所示。很明显，给定足够精确的三值图，与 Levin 等人的最小二乘法相比，我们的方法没有任何实际优势（当给定与输入相同的三值图时），Levin 等人的最小二乘法产生了数字最精确的遮罩。然而，当模拟分量的最佳标签时，我们的方法平均产生最精确的遮罩。

虽然我们的实验比较了理想输入产生的遮罩的质量，但更有趣的比较可能是测量每种方法提取满意遮罩所需的用户时间。理想情况下，我们还想测量（或在多大程度上）一个分量选择的界面是否比一个基于涂鸦的界面更直观。这样的比较涉及到一个非常重要的用户研究，我们将其留给将来的工作。

鉴于光谱抠图和硬光谱分割之间的强相似性，我们希望获得一些关于使用抠图分量与标准硬分割分量（也称为超级像素）相比潜在优势的直觉。当然，答案取决于应用程序。如果最终的输出是硬分割，那么抠图分量可能没有提供比标准硬分量更好的优势。另一方面，当目标是模糊遮罩时，最好像我们一样明确地构造抠图分量，而不是首先计算硬分割，然后对边界进行羽化（feather）。为了证明这一点，我们比较了这两种方法。我们首先从每个测试图像中提取抠图分量，然后选择抠图分量的子集，这将最小化与真值遮罩的距离。第二种方法是选择最接近真值的硬分量子集（我们使用了 Yu 和 Shi[18]的可用实现）。然后，我们在生成的硬遮罩上应用形态学操作（我们已经试验了几个恒定半径的腐蚀窗口），创建三值图并运行文献[7]的抠图算法。然而，由于腐蚀窗口的最佳半径在很大程度上取决于局部图像结构，并且随图像的变化而变化，因此不可能获得具有恒定半径窗口的理想三值图。这一问题在补充资料中有直观的说明。图 7c 显示了这两种方法的 SSD 错误（在不同背景 and 不同窗口上的平均值），这表明，与通过羽化硬分割获得的结果相比，通过选择最佳的抠图分量确实可以获得更精确的遮罩。

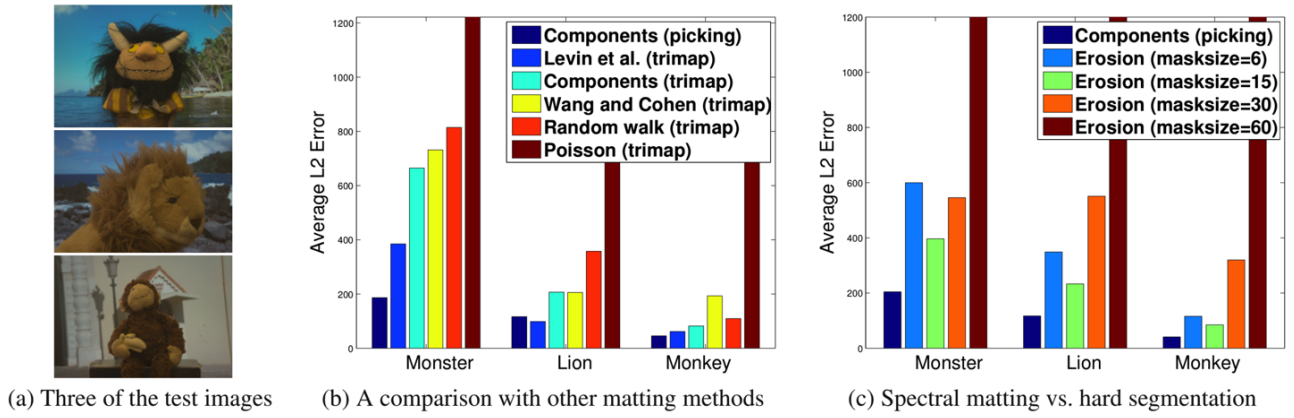


图 7 定量评估

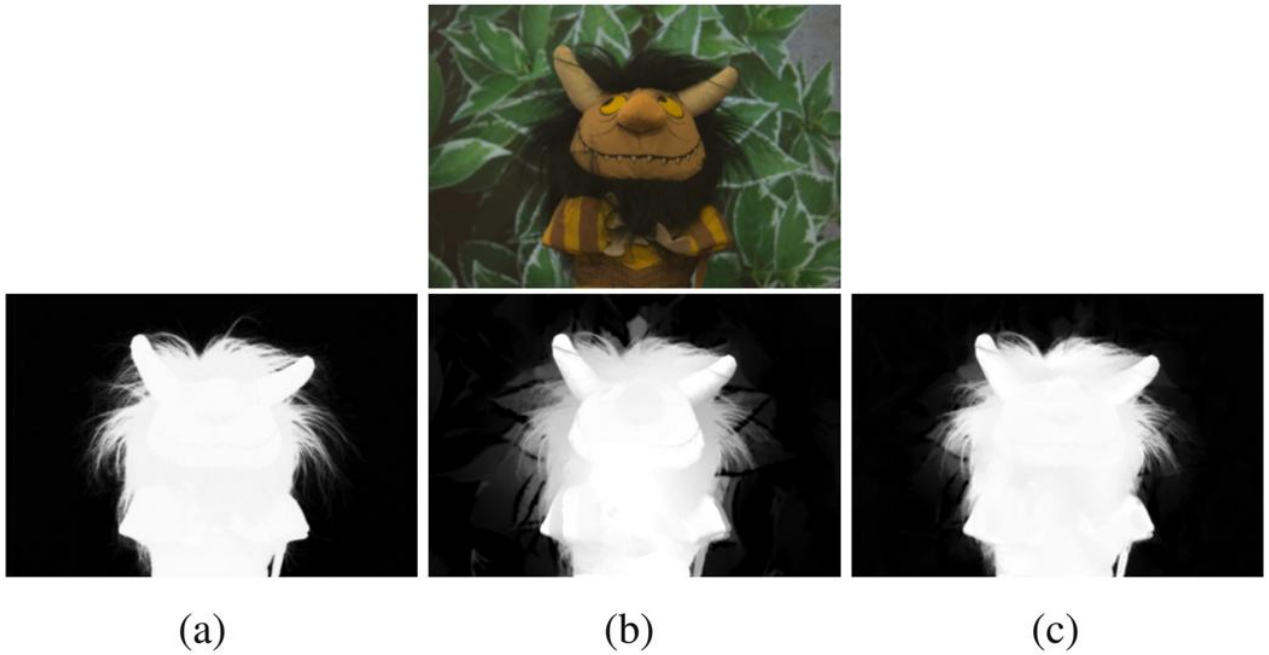


图 8

限制。第一行图像：输入图像；第二行图像：真值遮罩（a）、70 个最小特征向量生成的遮罩（b）、400 个最小特征向量生成的遮罩（c）。

6.讨论 (Discussion)

在这项工作中，我们推导了硬光谱图像分割和图像抠图之间的类比，并展示了如何从抠图拉普拉斯的最小特征向量自动获得基本抠图分量。这是一个新的有趣的理论结果，建立了两个先前独立的研究领域之间的联系。从实用的角度来看，抠图分量可以帮助自动执行遮罩提取过程，并减少用户的工作量。抠图分量还建议用户控制的提取遮罩的新模式：虽然在以前的方法中，结果是通过在遮罩的完全不透明或完全透明的图像区域中放置硬约束来控制的，但是我们的新方法可以为用户提供可选输出的简单直观预览，从而使用户能够直接控制遮罩的断片部分的输出。

局限：

我们的方法在视觉上包括有数量不多的不同分量的图像中自动提取遮罩的过程上是最有效的。然而，对于高度杂乱的图像，分量提取被证明是一个更具挑战性的任务。例如，考虑图 8 中的示例。输入图像由大量的小分量组成。在由 70 个最小特征向量构成的子空间上投影真值遮罩(图 8a)会导致近似值较差(图 8b)。回想一下，由于抠图分量是通过特征向量的线性组合获得的，因此它们不能比特征向量本身做得更好，因此图 8b 是我们希望从最多 70 个抠图分量中构建的最佳遮罩。因此，很明显，这一数量的分量不足以为该图像生成精确的遮罩。从 400 个最小特征向量中可以获得更好的遮罩(图 8c)，但即使这种遮罩也会留下改进的空间。由于计算的局限性，我们无法测试超过 400 个特征向量。我们从经验上观察到，如果在局部图像窗口中独立计算抠图分量，这个问题将显著地减少。我们目前正在研究将不同窗口中获得的分量拼接在一起的方法。

在光谱抠图中我们的主要挑战是确定给定图像的抠图分量的适当数量。这是所有光谱分割方法所共有的基本困难。虽然已经研究了自动选择分量数量的问题(例如文献[19])，但该参数仍然经常是手动调整的。对于本文中描述的应用程序，我们发现一个有用的策略是对图像进行过度分割，然后使用其他提示对分量进行分组。算法中的第二个自由参数是组成分量的最小特征向量数(该数应大于或等于分量数)。在实践中，我们观察到性能对这个数字不是很敏感，本文中的所有结果都是使用 70 个最小特征向量得到的。

未来方向：

将图像预分割为抠图分量的一个重要潜在优势是可以选择在每个分量中计算有意义的颜色或纹理直方图，或其他统计数据。直方图相似性可以为指导分量分组提供另一个重要提示。这种能力可能会显著改善使用颜色模型的抠图算法，如文献[16]。例如，在文献[16]中，当前的策略是使用涂鸦下的少量像素构建初始颜色模型。众所周知，这种糟糕的初始化会使算法对涂鸦位置中的小位移敏感。

鉴于人们对抠图问题越来越感兴趣，以及最近大量的抠图研究，未来的一个重要挑战似乎是设计不同用户交互模式和不同抠图算法之间的适当比较。在这项工作中收集的真值数据是实现这一目标的一步，但是为了评估每种方法产生良好结果所需的用户时间，需要进行适当的用户研究。

我们的代码和真值数据可从以下网址获得：www.vision.huji.ac.il/SpectralMatting