

Defocus Matting on Background Blur Priors

Guilin Yao, Hongxun Yao

School of Computer Science and Technology, Harbin Institute of Technology, Harbin
Email: yaoguilin1980@sina.com, h.yao@hit.edu.cn

Received: Sep. 22nd, 2012; revised: Oct. 15th, 2012; accepted: Oct. 22nd, 2012

Abstract: In this paper, we present a matting method based on focused foreground and blurred background. We classify some pixels as priors to prevent the similarities between foreground and background colors. Firstly, we design a three-channel edge detector to roughly predict some edge pixels. Secondly, for each edge pixel, we estimate its blur degree by fitting an ideal second derivative filter response on the actual one along the gradient direction, and comparing its color and known foreground one. Thirdly, if this pixel is classified into a blurred background edge, we extend it along the gradient direction as blur priors according to the fitting errors to an ideal blurred edge. Finally, with the background blur priors, we run a general matting algorithm along with a trimap expansion method. The experimental results show that our background blur priors could generate much more precise alpha results than the state-of-art algorithms.

Keywords: Image Matting; Background Prior; Edge Detection; Blur Magnification

基于背景先验点的虚焦抠图方法

姚桂林, 姚鸿勋

哈尔滨工业大学计算机科学与技术学院, 哈尔滨
Email: yaoguilin1980@sina.com, h.yao@hit.edu.cn

收稿日期: 2012年9月22日; 修回日期: 2012年10月15日; 录用日期: 2012年10月22日

摘要: 本文针对图像抠图问题中背景虚焦时的情况, 将一些模糊的背景预划分成先验点, 使之成为绝对背景, 进而避免了与前景颜色相近背景的干扰, 使得抠图结果更为理想。首先, 采用边缘检测器进行边缘检测, 进而对每个边缘点进行分析, 根据其对于二阶算子的响应值与理想响应值之间的拟合误差, 以及它与已知前景之间的颜色差异, 判断其是否为被模糊的背景边缘点; 然后再计算它沿梯度正负方向两侧的像素与理想化边缘之间的拟合误差, 根据阈值可以将两侧的若干点扩展成背景先验点集, 即将它们划分为绝对背景点。最后利用与 trimap 扩展相结合的抠图方式, 得到最终的透明度图像。实验结果表明, 本文提出的背景先验点与抠图算法, 可以得到相对于以往的抠图算法更好的结果。

关键词: 图像抠图; 背景先验; 边缘检测; 模糊扩展

1. 引言

图像抠图问题为图像编辑中比较典型的一类问题, 目的是将前景物体从背景环境中分离出来, 并与另一幅新背景合成, 最终形成新的图像。从数学的角度上来说, 对于图像中的点 z , 观察到的图像颜色 I_z 可以采用前景图像的颜色 F_z 和背景图像的颜色 B_z 以及透明度 a_z 的线性组合来表示:

$$I_z = \alpha_z F_z + (1 - \alpha_z) B_z \quad (1)$$

其中 α_z 可以取在区间 $[0, 1]$ 中的任意值。如果 $\alpha_z = 1$ 或者 0 , 我们将点 z 称作绝对前景或者绝对背景。否则称 z 点为混合点。在自然图像中, 如何将每个点的 α 值准确的估计出来是将前景从背景中分离出来的关键技术之一。如果没有多余的附加条件, 从单一的输入图像中提取前景信息显然是无解问题, 因为公式(1)

中有 3 个已知量，却有 7 个未知量。为了解决这类问题，通常的方法需要用户提供 trimap 模板用来大体的指定前景、背景和未知区域，并且要求未知区域尽量小，以便简化问题，并求得精确的解；另外，很多方法需要对输入图像提供其它约束条件与附加先验假设。本文算法对输入图像的前景与背景做出了假设——即需要背景相对是一种模糊失焦的状态，而前景则是清晰的对焦状态。

利用附加信息中相对简单的方法为规定单一颜色的背景，统称为蓝屏抠图，或者将同一前景放置到 2 个已知且不同的背景中进行拍摄的“三角抠图”法^[1]。闪光抠图^[2]利用相机的闪光灯来生成闪光及非闪光下的“图片对”，强调闪光与非闪光图像的最大区别为在背景距离较远的条件下，它们的前景物体由于受到闪光的影响，颜色值会产生很大差异，而背景颜色则几乎不受影响。可合成式抠图^[3]需要提供新背景，将抠图与合成问题看作一种联合的优化问题。多相机抠图^[4]可以从“相机组”中获取多幅合成图像，以提取高质量的 α 图。虚焦抠图^[5]与本文相似，同样利用了虚焦的特性，约束条件为前景与背景模糊程度不同。它采用了多传感器照相机，利用 3 路传感器来同步采集视频流。该系统相对普通用户而言，对硬件的要求非常高。

本文所要求的背景虚焦、前景实焦的输入图像，相当于虚焦抠图^[5]中前景传感器的成像，而对设备的需求上相比虚焦抠图^[5]要宽泛的多，只需要单幅背景虚焦的照片，因此相应的实例也要多很多。图 1 显示了 4 幅前景实焦，而背景虚焦的例子，通过这 4 个例子中的局部图可以看出，它们均存在很多虚焦化的背景边缘，其颜色与前景颜色十分相似。如果对背景不做任何假设，只利用抠图公式(1)与 trimap 模板对这些



Figure 1. Examples in input images and local windows with blurred background
图 1. 背景虚焦的例子与局部窗口图

例子进行抠图，常会出现的问题是将虚焦化的背景误判成前景，或包含有前景的成分。因此，需要对虚焦的背景边缘进行“附加处理”，以使得它们被预先判别成绝对背景，即背景的“先验点”，这样才能对下一步的抠图计算有利。

本文算法针对背景虚焦这类问题，从已提供的 trimap 中，首先第 2 节通过边缘检测器，从全体像素 V 中划分出一部分边缘点集 V_E ，并对每个边缘点针对噪声阈值，对其设定一个最小可依赖模糊核宽度；其次第 3 节通过计算对每个边缘点沿着梯度方向对二阶算子的响应，与理想化模糊边界对二阶算子响应的拟合误差值，从边缘点集 V_E 中进一步提取属于背景并被模糊的边缘点集 V_b ，并计算出该边界的模糊程度；第 4 节对 V_b 中的点按照梯度方向进行扩展，通过计算实际模糊边缘与理想化模糊边界的拟合误差，将所有扩展到的点提取出其中的一部分，划分为属于绝对背景的点集 B_{ex} ；第 5 节将这些扩展到的背景先验点作为已知绝对背景，采用全局采样与局部采样相结合、以及采样与仿射相结合的方式，并利用 trimap 扩展的方式进行 α 抠图；第 6 节的实验结果表明，利用背景先验点与 trimap 扩展相结合的抠图方式，可大大提高图像抠图的质量，解决由以往的抠图算法所无法解决的图像实例。

需要说明的是，本章针对每个边缘点设置最小可依赖模糊核宽度的方式，与基于空间变化的 PSF 算法^[6]相类似；然而，本章的主导思想与该算法存在几点根本的不同：1) 理论依据不同：空间变化 PSF 法^[6]是针对 α ，即假设即 α 真实值由某个二值化的 α^b ，与某个固定的卷积核 K ，通过卷积运算而来，即 $\alpha = \alpha^b \otimes K$ ，而本文算法针对的是图像，假设模糊背景是由某个清晰背景由类似的方式通过卷积得到，一定程度上较 α 模糊的概念更具一般化性质，理论依据更为充分；2) 计算方法不同：空间变化 PSF 法^[6]在 α^b 边缘处计算模糊程度，这样会遗漏很多细节，本文则于图像边缘处计算，计算区域更为充分；3) 计算方式不同：空间变化 PSF 法^[6]是基于某个算法得到初始 α 后，再进行下一步处理，如果初始 α 不准确则会影响到二值化处理以及模糊程度的计算，而本文直接在图像上运行，不存在依赖上一步计算的问题；4) 最终目的的不同：空间变化 PSF 法^[6]用于为 α 提供先验值，由

于在某种程度上空间变化 PSF 法^[6]得到的先验并不一定准确,因而原文中该先验的权值较小(0.08),而本文算法则用于模糊背景边缘的识别,即将得到扩展点直接划分为绝对背景,可靠性非常好。

2. 图像边缘检测器

首先,我们采用基于局部模糊度的边缘检测与模糊估计^[7]的方式对原始输入图像 I 进行边缘检测。其中,设 I_b 为一个理想的模糊边界, g_1 与 g_2 分别为预先设定的一阶算子与二阶算子, θ 为图像梯度的正方向, σ_2^* 为图像二阶响应最小可依赖模糊度, I_b 对 g_1 和 g_2 的响应 G_1 与 G_2^0 如图 2 所示。于是,最终边缘检测器的输出,即判断图像上某点 z 属于边缘点集 V_E 的准则为:对于点 z , 设沿着 θ 方向所对应的邻域为 $z_{neb}^{\theta^+}$, 负方向的邻域 $z_{neb}^{\theta^-}$, 如果在最小可依赖模糊度 σ_2^* 对应的二阶响应上,点 z 与 $z_{neb}^{\theta^+}$ 、或 z 与 $z_{neb}^{\theta^-}$ 之间有零点(即乘积 ≤ 0), 并且 z 点与 $z_{neb}^{\theta^+}$ 和 $z_{neb}^{\theta^-}$ 处的二阶响应大小关系如式(2), 则判断 $z \in V_E$, 即

$$\begin{cases} G_2^0(z, \sigma_2^*) \cdot G_2^0(z_{neb}^{\theta^+}, \sigma_2^*) \leq 0, G_2^0(z, \sigma_2^*) \cdot G_2^0(z_{neb}^{\theta^-}, \sigma_2^*) \leq 0 \\ G_2^0(z, \sigma_2^*) > G_2^0(z_{neb}^{\theta^+}, \sigma_2^*), G_2^0(z, \sigma_2^*) \sigma_2^* < G_2^0(z_{neb}^{\theta^-}, \sigma_2^*) \\ G_2^0(z_{neb}^{\theta^+}, \sigma_2^*) < 0, G_2^0(z_{neb}^{\theta^-}, \sigma_2^*) > 0 \end{cases} \quad (2)$$

然后,我们在 3 个通道上分别计算边缘点,最后的边缘点集 V_E 为 3 个通道上边缘点的集合。图 3 的第 2 列给出了边缘检测的 4 个例子,而本文的边缘检测器相对于普通边缘检测不同的是,普通边缘检测方法在模糊的边缘处,可能会检测到大量的、或者宽度较

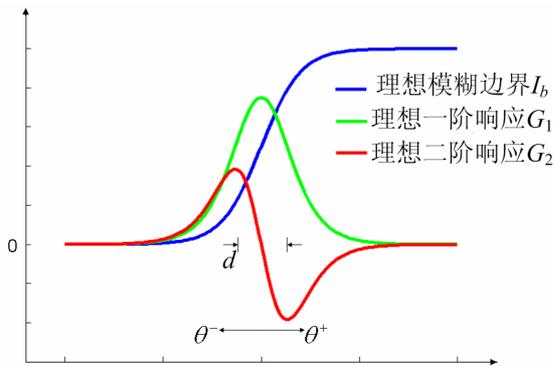


Figure 2. Responses of first and second derivative basis filters for an ideal blurred edge

图 2. 理想模糊边界分别对一阶算子和二阶算子的响应

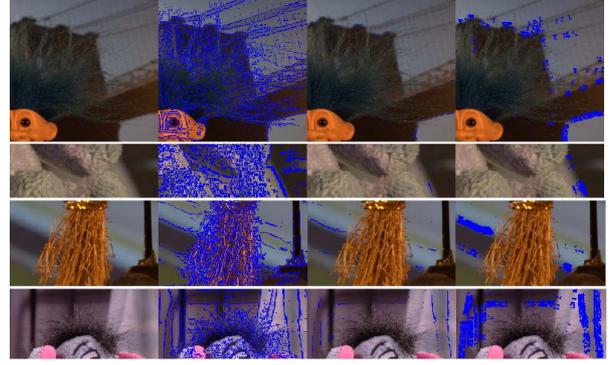


Figure 3. From left to right of each example: input image, edge set V_E from the edge detector, blurred background set V_b , and background prior set B_{ex}

图 3. 每个例子从左至右: 原始局部窗口, 由边缘检测器得到的边缘点集 V_E , 被模糊的背景边缘点集 V_b , 背景先验点集 B_{ex}

大的被模糊后的边缘,而本文的边缘检测器则只在二阶算子响应的“过零点”处寻找边缘,因此在模糊边缘处只寻找初始未被模糊的边缘,相对普通边缘检测方法更准确。

3. 模糊边缘的检测

为了检测在图像边缘处的模糊度,在以往的诸多算法均提出了不同形式的点扩散函数 g_b , 但是,在实际运用中采用最广泛的是高斯函数。我们首先按照扩大模糊的方法^[8]将边界处的模糊核定义为如下的高斯卷积核:

$$g_b(x, y, \sigma) = \frac{1}{2\pi\sigma_b^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_b^2}\right) \quad (3)$$

其中 σ_b 为该高斯模糊核的宽度,即对每个边缘点需要估计的值。如果该值较大则说明该边缘点属于被模糊的边缘点。考虑图 2 所示的一个沿 x 轴分布的理想化模糊边界 I_b , 假设 $I(x) = Au(x)$ 为图像边缘的理想化模型, $u(x)$ 为单位阶跃函数, A 为 $u(x)$ 的幅值,而对模糊边界有 $I_b(x) = I(x) \otimes g_b(x)$, 于是按照扩大模糊^[8]的方式,模糊边界 I_b 对二阶算子 $g_2^x(x, y, \sigma_2^*)$ 的期望响应,即理想化响应为

$$\begin{aligned} r_2^x(x, y, \sigma_2) &= I_b \otimes g_2^x = Au(x) \otimes g_2^x(x, y, \sigma_b^2 + \sigma_2^{*2}) \\ &= \frac{-Ax}{\sqrt{2\pi(\sigma_b^2 + \sigma_2^{*2})^{3/2}}} \exp\left(-x^2/2(\sigma_b^2 + \sigma_2^{*2})\right) \\ &= \frac{-Ax}{\sqrt{2\pi(d/2)^3}} \exp\left(-x^2/2(d/2)^2\right) \end{aligned} \quad (4)$$

其中 $(d/2)^2 = \sigma_b^2 + \sigma_v^2$ ，幅值 A 计算为

$$A = T(r_2^x) \cdot (2\pi e)^{1/2} (d/2)^2 \quad (5)$$

其中 $T(r_2^x)$ 为 r_2^x 的极值，若按照梯度正方向，则计算为 $\min(r_2^x)$ ，负方向计算为 $\max(r_2^x)$ 。

上述分析说明，若某边缘点 v 为被模糊的边缘点，那么它实际二阶算子的响应一定与“某个”理想的二阶响应 r_2^x 相符合，而寻找该理想 r_2^x 的方式，可以采用逐个对“不同形状”或“不同大小”下的 r_2^x 与实际响应 G_2 之间的差异获得，如图 4 所示。一般的，在得到理想响应 r_2^x 之后，对边缘点集 V_E 中的任意点 v ，沿着梯度方向，我们采用一个以 v 为中心、半径为 k 的窗口，计算边界对二阶算子的响应 r_2^x ，与实际二阶算子响应 G_2 之间的平均拟合误差，并求得使得拟合误差最小的窗口半径 k^* 。图 4 表示了在 3 个不同的窗口半径 k 下，对应的 3 个理想二阶算子 r_2^x ，与实际二阶算子响应 G_2 之间的差异，其中窗口半径为 k_2 对应的理想响应与实际响应符合的最好。

具体的说，对于边缘点 v ，设其梯度的正方向 θ^+ ，负方向为 θ^- ， $\sigma_2^*(v)$ 为边缘点 v 的最小可依赖模糊核宽度，于是，按照方向 $\theta \in \{\theta^+, \theta^-\}$ 发射两条射线，在以 v 为中心、窗口半径为 k 时，二阶算子响应 G_2^θ 与理想化响应 r_2 的平均拟合误差为

$$\text{err}_2(z, k, \theta) = \frac{1}{k} \sum_{l=1}^k \left(\left| G_2^\theta(z^\theta(l), \sigma_2^*(z)) - r_2(d_s(z, z^\theta(l)), \sigma_2^*(z)) \right| \right) \quad (6)$$

其中 $v^\theta(k)$ 为点 v 沿着 θ 的射线方向，与第 l 层 ($l \leq k$) 窗口最接近的像素，如图 5 所示； $d_s(v, v^\theta(l))$ 为点 v 与 $v^\theta(l)$ 的空间矢量距离，亦即(4)式中的 x 坐标值，在正方向 θ^+ 下， $d_s(v, v^\theta(l)) > 0$ ，负方向 θ^- 下 $d_s(v, v^\theta(l)) < 0$ ；而在 r_2 的 A 值中，根据梯度的方向，极值 $T(r_2^x)$ 分别计算为

$$\begin{cases} T(r_2^x, \theta^+) = \min_{l=1}^k \left(G_2^{\theta^+} \left(v^{\theta^+}(l), \sigma_2^*(v) \right) \right), \\ T(r_2^x, \theta^-) = \max_{l=1}^k \left(G_2^{\theta^-} \left(v^{\theta^-}(l), \sigma_2^*(v) \right) \right) \end{cases} \quad (7)$$

这里我们假设在窗口宽度 k 处为极值点的位置，因此 $d/2$ 即计算为窗口 k 的最远点 $v^\theta(k)$ 与 v 点的空间距离 $|d_s(v, v^\theta(k))|$ ， $\theta \in \{\theta^+, \theta^-\}$ 。图 5 表示了一个例子，由出发点 v 沿梯度正方向 θ^+ ，负方向为 θ^- 发

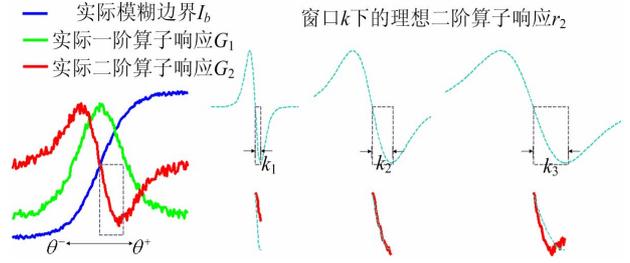


Figure 4. Comparisons for the ideal and actual second derivative filter responses on different window sizes
图 4. 不同窗口半径对应的理想二阶算子响应，与实际二阶算子响应之间的符合程度

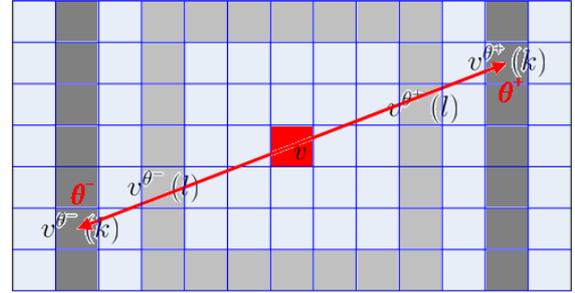


Figure 5. Generated points through two opposite directionally shooting rays from the source point
图 5. 由出发点沿梯度正负方向的射线所经过的点的示意图

出两条射线，到达与以 v 为中心的第 l 层窗口的点 $v^{\theta^+}(l)$ 与 $v^{\theta^-}(l)$ ，浅灰色表示第 l 层窗口，深灰色表示最外层的第 k 层窗口。本例中窗口宽度为 $k=5$ ，中间点 $l=3$ 。需要注意的是此假设与扩大模糊的方法^[8]的不同之处在于，后者将 G_2^θ 中的实际极值对应的位置设定为极值点，然而此方式的局限性在于其很容易受到噪声的干扰，如果出现某些错误的局部极值点，则会影响到对极值点位置的判断，因此这里我们采用实际极值点位置与理论位置相分离的方式，即假定在理论上极值点处于窗口宽度 k 处，而不考虑实际极值点的位置。如图 5 所示，在窗口半径为 k 下，极值点位置固定为 $v^{\theta^+}(k)$ 或 $v^{\theta^-}(k)$ ，而实际极值点的位置可能不在这两个点上。

由于我们假设在窗口宽度 k 处为极值点的位置，亦即 $|d_s(v, v^\theta(k))| = d/2$ ，而根据已知有 $d/2 > \sigma_2^*$ ，于是定义 k 的下界为

$$k^- = \arg \max_k \left(|d_s(v, v^\theta(k))| > \sigma_2^*(v) \right) \quad (8)$$

且我们不希望 σ_b 的值过大，因此定义 k 的上界为

$$k^+ = \arg \max_k \left(|d_s(v, v^\theta(k))| < \sigma_2^*(v) + \Delta\sigma \right) \quad (9)$$

其中通常取 $\Delta\sigma = 5$ 。于是 v 点按照方向 $\theta \in \{\theta^+, \theta^-\}$ ，实现最优拟合度的窗口宽度 k^* 计算为

$$k^*(v, \theta) = \arg \min_{k=k^-}^{k^+} (err_2(v, k, \theta)) \quad (10)$$

于是 v 点按照每个方向 θ ，其模糊核宽度 σ_b 与最小拟合误差计算为

$$\begin{cases} \sigma_b(v, \theta) = \sqrt{d_s^2(v, v^\theta(k^*)) - \sigma_2^{*2}(v)} \\ err_2(v, \theta) = Err(v, k^*, \theta) \end{cases} \quad (11)$$

同时，若 z 沿着梯度正负方向 $\theta \in \{\theta^+, \theta^-\}$ 为一段模糊边界，我们希望以方向 θ 为基准，向两侧变化一个微小 $\Delta\theta$ 的情况下，仍然为模糊边界，亦即其二阶算子 g_2^θ 仍然与其期望值 r_2 相符合。于是，对于正方向 $\theta^+ \in \{\theta^+ \pm n\Delta\theta\}$ ，反方向 $\theta^- \in \{\theta^- \pm n\Delta\theta\}$ ， $n = \{0, 1, 2\}$ ，各 5 个方向，我们对每对互为相反方向的 5 对“ θ 对” $\theta' \in \{\theta^+, \theta^-\}$ ，在 3 个通道 c 上，各计算其模糊核宽度 $\sigma_b(v, \theta', c)$ 与拟合误差 $err_2(v, \theta', c)$ 。这样对边缘点 v 的 3 个通道分别计算 5 对这样的值，总共有 15 对值。这样，定义 v 属于“被模糊”的背景边缘点集 V_b ，需要满足以下三个条件：该边缘点 a) 在每个颜色通道中，按每对相反方向上的 2 个拟合误差的和要尽量小；b) 在每个颜色通道中，每个方向的模糊核宽度要尽量大；c) 其颜色与已知前景要有一定的差别。具体的，需要满足以下条件：

$$\begin{cases} \sum_{c, \theta'} \delta(err_2(v, \theta^+, c) + err_2(v, \theta^-, c)) < T_E \\ \& \sigma_b(v, \theta', c) \geq \sigma_T \geq N_T \\ d_{color}(v, F_s) > t_c \end{cases} \quad (12)$$

其中 $T_E = 0.35$ 为相反方向上的误差和阈值， $\sigma_T = 3$ 为模糊边界的最小模糊核宽度阈值， $N_T = 12$ ， δ 为根据输入的判别式是否成立而返回 0 或 1 的布尔函数， $d_{color}(v, F_s)$ 按照 trimap 扩展算法^[9]定义，为点 v 与已知前景“学习区域” F_s 之间的平均颜色差异，这里我们假设按照 trimap 扩展算法^[9]，已将每幅图划分成若干局部窗口；阈值 t_c 定义与该算法的定义相似但略有差异：若 $t_b - t_f > 5$ 则 $\sigma_i = 0.1$ ，否则 $\sigma_i = 0$ 。上式说明，若边缘点 v 属于被模糊的背景边缘点 v ，在 3 个通道各 5 对 $\{\sigma_b, err_2\}$ 中，满足上述阈值设定的数对，需要大于一定的个数 N_T ；而且该点与前景的颜色差异

应大于一定的阈值，以确保 v_b 不会是前景点。

在得到被模糊的背景边缘点集 V_b 之后，我们记录每个这样的点 v_b 在 3 个通道 c 上的各 5 对、总共 30 个方向 θ' 上的 30 个模糊核宽度值 $\sigma_b(v_b, \theta', c)$ ，以供下一步边缘扩展时使用。图 3 的第 3 列表示在每个例子中，由边缘点集 V_E 经过本节的判别，得到的被模糊背景边缘点集 V_b 的示意图。可以看出 V_b 基本将属于前景或者混合区域的边缘从 V_E 中排除掉了，保留下来的只是背景上被模糊的边缘点集。

4. 模糊边缘的扩展

对于每个被模糊的背景边缘 $v_b \in V_b$ ，上一节已计算出 3 个通道上 15 对互为相反方向的、总共 30 个方向的模糊度 $\sigma_b(v_b, \theta', c)$ ，其中 $\theta' \in \{\theta^+, \theta^-\}$ ，正方向 $\theta^+ \in \{\theta^+ \pm n\Delta\theta\}$ ，反方向 $\theta^- \in \{\theta^- \pm n\Delta\theta\}$ ， $n = \{0, 1, 2\}$ ，因此可以按照每个模糊度 σ_b ，按图 6 所示，对每个模糊方向 θ' 进行“模糊扩展”得到点集 V_{ex} ，并按照一定的划分判别准则，将 V_{ex} 中的一部分子集划分为背景先验点集 B_{ex} ，作为绝对背景点。换一个角度说，如果前一节寻找的是模糊边缘对二阶算子响应的“过零点”，那么本节寻找的则是该过零点按照梯度的正方向或负方向，向两侧延伸得到的点。且与前一节不同的是，本节的运算域在模糊边缘 T_b 的实域上，而不是对算子的响应域。

与计算被模糊的边界方法类似，我们希望每个被模糊的背景边缘点 v_b ，按照每个模糊方向 θ' 的各个像素(例如图 4 中的实际模糊边缘 I_b 的灰度值，如果与按照 v_b 的计算的“理想化”模糊边界(例如图 2 中的理想模糊边缘 I_b)的期望值相似，并且如果这些像素的灰度值依照 θ^+ 或者 θ^- 的方向，满足绝对递减或者递增的方式，那么这些像素就能够被划分为绝对前景，否

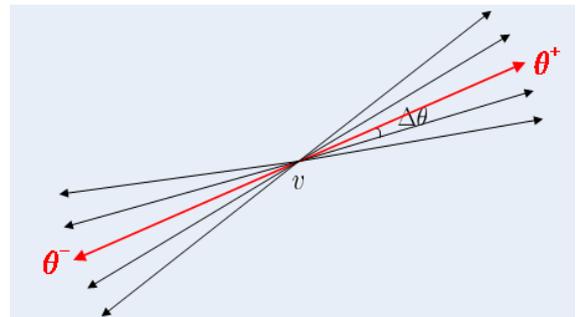


Figure 6. 10 rays from the source point
图 6. 由出发点分别向两侧发出的 10 条射线示意图

则按照实际情况，依旧划分为未知点。

一般的，给出一个沿 x 轴分布的未被模糊的理想边界，它表现为单位阶跃函数 $u(x)$ ，幅值为 A ，它对二阶算子 $g_b^x(x, y, \sigma_b)$ 响应的期望为 $r_b^x(x, y, \sigma_b) = Au(x) \otimes g_b^x(x, y, \sigma_b)$ ，该卷积虽然无法用初等函数的形式表示，但可以用逼近的方式来求解。对被模糊的边缘点 v_b ，设其沿着某个 θ' 的射线方向，到达以 v_b 为中心的层窗口对应的点集 $v_b^{\theta'}(k)$ ，其中 $k = 0, 1, \dots, k_{\max}$ ；令理想化边缘 $Au(x)$ 以点 $v_b^{\theta'}(k)$ 为模糊中心， f_w 为二阶算子 g_b 的模糊半径，每次按照与模糊中心距离为 l 、间隔 $\Delta l = 0.1$ 进行对算子 g_b 的乘积运算，最终对算子 g_b 所有元素的和求平均。于是，幅值为 A 的单位阶跃函数 $u(x)$ 对二阶算子 $g_b(x, \sigma_b)$ ，按照方向 θ' ，在以点 v_b 为中心的层窗口所对应的响应期望 r_b 为

$$r_b(v_b, k, \theta', \sigma_b) = \sum_{l=-f_w}^{f_w} A \cdot u(l + d_s(v_b, v_b^{\theta'}(k))) \cdot g_b(l, \sigma_b) / \sum_l g_b(l, \sigma_b), \quad (13)$$

其中 $d_s \geq (v_b, v_b^{\theta'}(k))$ 为点 v_b 与 $v_b^{\theta'}(k)$ 的空间矢量距离，若 θ' 为某个梯度正方向 $\theta^{+'}$ ，则 d_s 值大于 0，若 θ' 为某个梯度负方向 $\theta^{-'}$ ， d_s 值小于 0。

对于像素序列 $v_b^{\theta'}(k), k = 0, 1, \dots, k_{\max}$ ，幅值 A 与的最大窗口扩展半径 k_{\max} 的值分别计算为

$$\begin{cases} A = (\max(I^c(v_b^{\theta'}(k))) - \min(I^c(v_b^{\theta'}(k)))) + \Delta l \times 2 \\ k_{\max} = \arg \max_k (d_s(v_b, v_b^{\theta'}(k))) < f_w \end{cases} \quad (14)$$

其中 I^c 为图像像素值， c 为图像的第 c 个颜色通道，且被归一化至 $[0, 1]$ 区间， $\Delta l = \frac{3}{255}$ 为补偿值。在得到对 $g_b(x, \sigma_b)$ 的期望响应之后，可以计算以 v_b 为中心的层窗口，按 θ' 方向对应的点 $v_b^{\theta'}(k)$ ，其实际像素灰度值与期望值的拟合误差为

$$err_b(v_b, c, k, \theta') = \begin{cases} \left| \left(I^c(v_b^{\theta^{+'}}(k)) - \min_k(I^c(v_b^{\theta^{+'}}(k))) \right) - \left(r_b(v_b, k, \theta^{+'}, \sigma_b) - A/2 \right) \right| \\ \left| \left(\max_k(I^c(v_b^{\theta^{-'}}(k))) - I^c(v_b^{\theta^{-'}}(k)) \right) - \left(A/2 - r_b(v_b, k, \theta^{-'}, \sigma_b) \right) \right| \end{cases} \quad (15)$$

实际上，我们对被模糊的边缘点 v_b ，按照 3 个通道上 15 对互为相反方向的、总共 30 个方向的模糊度

$\sigma_b(v_b, \theta', c)$ ，其中 $\theta' \in \{\theta^{+'}, \theta^{-'}\}$ ，分别进行扩展。设每个扩展到的像素记为 v_{ex} ，于是它的平均拟合误差 $err_b(v_{ex})$ 定义为

$$err_b(v_{ex}) = \frac{1}{N_{ex}} \sum_{c, \theta'} \sum_{v_b | v_{ex} = v_b^{\theta'}(k)} err_b(v_b, k, \theta', c) \quad (16)$$

其中 N_{ex} 为所有上述扩展到能够到达点 z_{ex} 的次数。于是，由任一被模糊的背景边缘点 v_b 扩展得到的点 v_{ex} ，若被判别为绝对背景 B_{ex} ，需要满足以下三个条件：该扩展点 a) 平均拟合误差小于一定的阈值；b) 在其对应的任一组扩展像素序列中，若扩展方向为梯度的正方向，则其灰度值要大于上一级窗口对应的灰度值，否则要小于上一级窗口的灰度值；c) 其颜色与已知前景要有一定的差别。即

$$\begin{cases} err_{gb}(v_{ex}) < T_{ex} \\ I^c(v_b^{\theta^{+'}}(k)) \geq I^c(v_b^{\theta^{+'}}(k-1)) \\ I^c(v_b^{\theta^{-'}}(k)) \leq I^c(v_b^{\theta^{-'}}(k-1)) \\ d_{color}(v_{ex}, F_s) > t_{ff} \end{cases} \quad \forall v_b, c, k, \theta', v_{ex} = v_b^{\theta'}(k) \quad (17)$$

其中 $T_{ex} = \frac{8}{255}$ ， $d_{color}(v_{ex}, F_s)$ 和 t_{ff} 与 trimap 扩展算法^[9]定义相同。通过上述扩展方法，可以将背景上的一些由模糊边界 v 扩展的点 v_{ex} 划分至绝对背景 B_{ex} ，而此时的未知区域 $U^* = U - B_{ex}$ 。图 3 的第 4 列表示由被模糊的背景边缘点集 V_b ，经本节的背景扩展方法得到的背景先验点集 B_{ex} 。

5. 全局与局部相结合的抠图方法

既然已经获得了由背景模糊边界扩展而得到的先验点集 B_{ex} ，因此我们希望对于每个未知点，除了初始提供的 trimap 外，均可以利用这些背景先验点 B_{ex} 作为已知背景点，进行最终的 α 抠图计算。同时，我们可以继续利用 trimap 扩展^[9]结果，作为附加辅助输入，以使得本算法的结果更好。

进一步，如果背景先验点集 B_{ex} 的像素个数小于某阈值，即 $|B_{ex}| < T_N$ ，可以将这些背景先验点看作全局样本，每个未知点 z 均可以利用这些样本。为此，将这些新扩展的点集 B_{ex} 按照 GMM 聚类的方式，聚成 $N_C = 10$ 个类，并在每个类中随机抽取 $N_S = 3$ 个样本，于是至多有 30 个全局样本；即对于每个未知点 z ，

这些样本均起作用。

另外，我们对每个点都采用局部采样，为此我们采用鲁棒性算法^[10]中的方法，对于每个未知点 z ，在已知前景与背景轮廓上采用稀疏采样的方式，选取距离 z 点空间距离最近的各 10 个样本作为局部前景与背景样本。于是点 z 的前景样本为 10 个，而背景样本包括全局与局部样本，至多 40 个。与改进颜色模型^[11]算法相类似，给出未知点 $z \in U^*$ 以采集到及每个样本对 $\{F^i, B^j\}$ ，首先定义距离比值 R_d ，为了强调大多数未知点为绝对前景或背景，我们对每个前景或背景样本定义权值。注意与改进颜色模型算法^[11]不同的是，这里我们计算每个未知点 z 与已知前景或背景颜色欧式距离的最小值，这样会避免最终的权值 w 过小。其次，对每个样本对 $\{F^i, B^j\}$ 均算出一个相应的信念值 f ，为未知点 z 选出拥有最大信念值 f 的 3 对样本 $\{F^{i^*}, B^{j^*}\}$ ， $\bar{\alpha}_2$ 计算为由 $\{F^{i^*}, B^{j^*}\}$ 计算得到的 3 个 α 值的平均，各变量的定义与改进颜色模型算法^[11]类似。

下一步，我们采用拉普拉斯抠图矩阵作为“后处理”，即对 α 进行平滑性操作。最终的 α 求解为

$$\alpha = \arg \min \alpha L \alpha + \lambda (\alpha - \bar{\alpha}) D (\alpha - \bar{\alpha}) \quad (18)$$

其中 $\bar{\alpha}$ 为上一步颜色采样法得到的 α 值， L 为拉普拉斯抠图矩阵， $\lambda = 1$ 为平衡二者的权值， D 为对角阵，已知点所对应的对角线元素为恒定的较大值，对于未知点，其对角元素固定为 $\bar{\alpha}$ ，注意这里与传统算法不同的是，我们忽略了信念值 f 对最终 α 计算的影响，实验结果亦表明 f 对最终 α 的影响很小。

6. 实验结果

我们采用在线 α 评估系统^[12]所提供的图像。由于并非所有图像均具有前景实焦、背景虚焦的性质，因此我们从中选出 4 幅具有背景虚焦特性、且具有典型性的例子的局部窗口。图 7 显示了 4 个例子，每个例子中，左图为初始局部窗口图像；右图中深红与深蓝表示初始 trimap 的已知前景与背景，浅红色与浅蓝色区域表示由 trimap 扩展算法^[9]得到的绝对前景与背景，粉色区域表示本章算法得到的背景先验点集。其中，前 2 个例子为测试集中的图像，后 2 个为训练集中的图像，而初始 trimap 均为每个例子的最大 trimap。由于大 trimap 中包含的未知点最多，不确定的因素最大，对抠图算法的挑战性也最高，因此本算法采用大

型号的 trimap 以证明其对解决难点问题的有效性。

对于前 2 幅测试集中的图像， α 评估系统^[12]提供了由 21 个算法得到的 α 结果，但没有真实值。为了节省空间，我们仅从中选取了平均排名前 4 位的抠图算法在最大 trimap 下，所对应的 α 结果图，它们分别为 SVR 算法^[13]、共享算法^[14]、基于拉普拉斯抠图矩阵后处理的全局算法^[15]、基于 2 值分割与空间变化 PSF 的算法^[6]。图 8 针对这 2 个例子的局部，对包括上述 4 种算法，以及 trimap 扩展算法^[9]、本文背景先验点算法、本文先验点算法与 trimap 扩展方法^[9]相结合的算法等 7 种算法，分别进行了比较，其中第 1 幅图由于过大，我们相应截取了 2 个局部窗口。从图 8 中的 α 结果可以看出，第 1 幅图的 2 个局部中，由于背景中桥梁的模糊边缘与前景木偶头发的颜色极为相近，排在评估系统^[12]前列的 4 种算法以及 trimap 扩展算法由于只考虑到颜色，而没有考虑模糊背景的因素，而且所提供的 trimap 又没有足够的、与桥梁边缘相对应的已知背景参考点，因此它们均将一段较长且较暗的桥梁边缘、以及深褐色的桥梁内部判别为前景的一部分，即具有相当大的 α 值；而在本文得到的 2 种算法中，由于考虑到背景模糊边缘的因素，已经预先将一部分模糊的桥梁边缘以及内部划分为绝对背景，因此大大降低了桥梁部分对抠图算法的干扰；而且这些预划分的点可以为抠图部分提供较为充分的背景样本，使得得到的 α 值也相对准确，而结合了 trimap 扩展^[9]

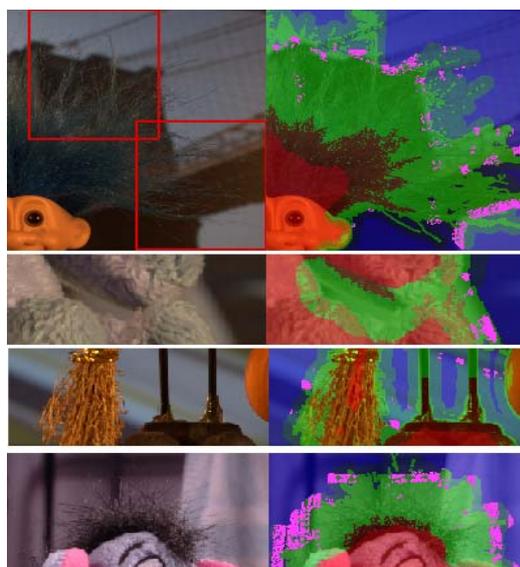


Figure 7. 4 examples of initial input
图 7. 4 个初始输入的例子

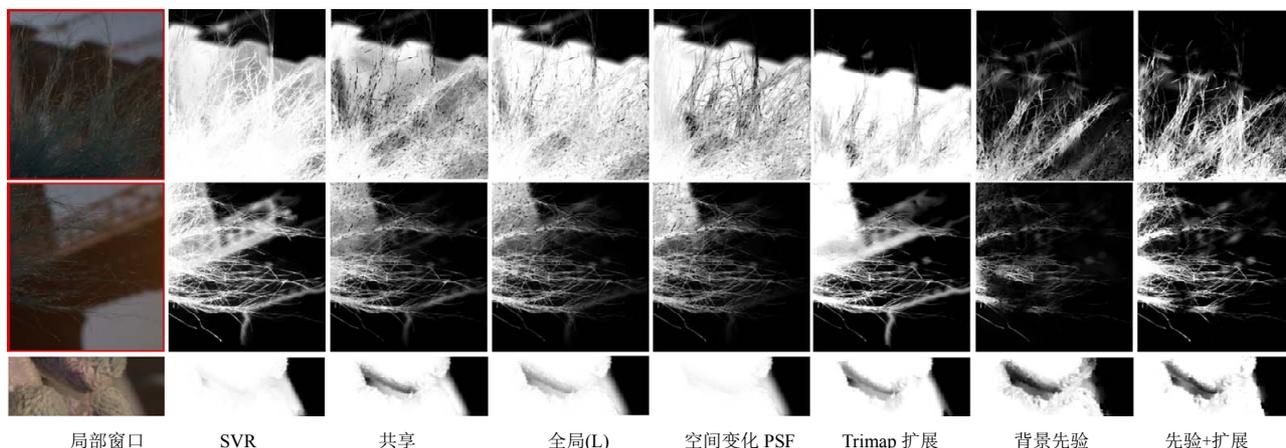


Figure 8. Alpha comparisons of 7 algorithms in 3 local windows of the test images
图 8. 测试图像的 3 个局部窗口中, 由 7 种算法得到的 α 结果的比较图

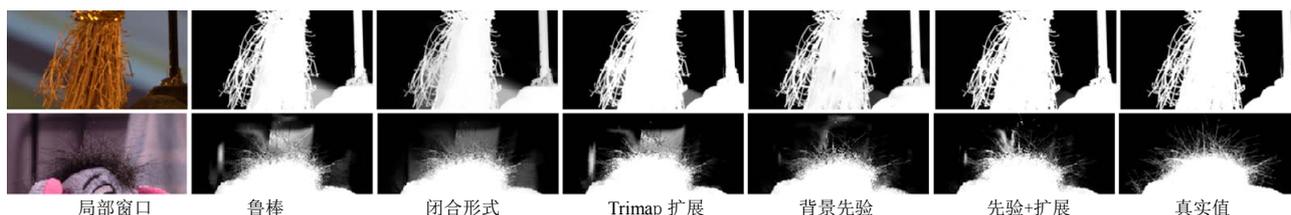


Figure 9. Alpha comparisons of 5 algorithms in 2 local windows of the training images
图 9. 训练图像的 2 个局部窗口中, 由 5 种算法得到的 α 结果的比较图

的本文算法, 由于样本更为充分, 结果则更为精确。第 2 幅动物图像也如此, 这里我们把一段背景的模糊石墩边缘进行了预判, 这同时对中间一段虽然不属于边缘的模糊背景产生了影响; 相比较其它 5 种算法, 本文对应的 2 种算法可以得到更好的 α 结果。

对于后 2 幅训练集中的图像, 评估系统^[12]提供了 α 实值, 但并没有其它全部算法得到的抠图结果, 然而我们可以从中选取比较有代表性的算法进行比较。这里我们选取鲁棒性算法^[10]、闭合形式算法^[16]、trimap 扩展算法^[9], 分别在最大 trimap 下, 得到的 α 结果图。从图 9 中的输入的图像窗口可知, 背景均存在一段模糊的背景边缘, 且颜色与前景比较接近, 按照背景先验点的方式可以使一部分模糊背景划分至绝对背景中, 而且对下一步的抠图算法提供了充分的样本, 而背景先验点方法与 trimap 扩展算法^[9]相结合, 可以获得更为充分的样本, 因此结果更好。从图 9 中的每个 α 结果可知, 本文算法相对其它 3 种算法, 对背景干扰的影响最低, 且与 α 真实值最为接近。

图 10 表示了对于 2 个训练集中的例子, 由 5 种不同算法得到的 5 个 α 结果图中, 根据真实值得到的

平均 SAD(绝对误差和)。从结果可以看出, 采用本文中单独的背景先验点法的误差不是最优结果; 而将先验点法与 trimap 扩展算法^[9]相结合, 可以得到相对于其它算法最优的 α 结果。因此也可以说, trimap 扩展算法^[9]是对本章算法的有力补充, 因为其可以为本章算法提供更为充分的样本。

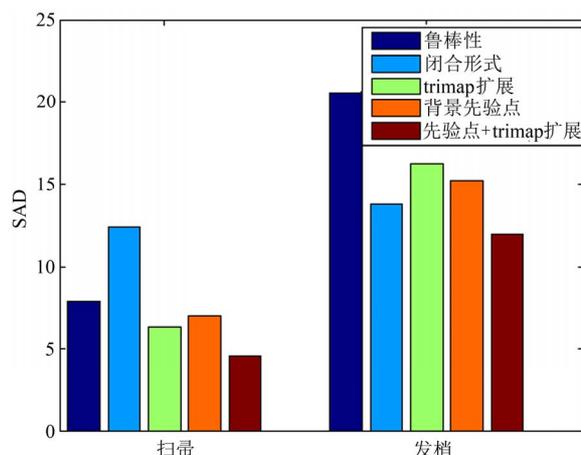


Figure 10. SAD comparisons of 5 algorithms in 2 examples of test set
图 10. 2 个训练集的例子中, 5 个算法得到的 α 与真实值的平均 SAD

7. 算法局限性与讨论

本文在提取被模糊的背景边缘部分, 以及模糊边缘的扩展部分, 各种阈值被设定为一个固定的值; 但它们根据例子可能需要随时调整, 这可以由已知区域由“学习”方式得到, 使得其自适应性更好; 另一个局限是需要背景模糊边缘尽量呈直线分布状态, 且不要过于杂乱与扭曲。因此对背景边缘分布规律的分析是可以进一步探讨与研究的问题。

8. 结论

本文针对图像抠图中前景实焦、背景虚焦这类问题, 对一些模糊的背景预划分成先验点, 使之成为绝对背景, 并为下一步的抠图提供更为准确的背景样本, 使得抠图结果更为理想。实验结果表明, 该背景先验点的方式, 可以使抠图过程更为精确, 并得到更好的抠图结果。

9. 致谢

本文由中国国家自然科学基金(No.61071180)以及关键项目(No.61133003)支持。

参考文献 (References)

- [1] A. R. Smith, J. F. Blinn. Blue screen matting. *ACM Transactions on Graphics (TOG)*, 1996: 259-268.
- [2] J. Sun, Y. Li, S. B. Kang and H.-Y. Shum. Flash matting. *ACM Transactions on Graphics (TOG)*, 2006, 25(3): 772-778.
- [3] J. Wang, M. Cohen. Simultaneous matting and compositing. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [4] N. Joshi, W. Matusik and S. Avidan. Natural video matting using camera arrays. *ACM Transactions on Graphics (TOG)*, 2006: 779-786.
- [5] M. McGuire, W. Matusik, H. Pfister, J. F. Hughes and F. Durand. Defocus video matting. *ACM Transactions on Graphics (TOG)*, 2005: 567-576.
- [6] C. Rhemann, C. Rother, P. Kohli and M. Gelautz. A spatially varying psf-based prior for Alpha matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010: 2149-2156.
- [7] J. Elder, S. Zucker. Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1998, 20(7): 699-716.
- [8] S. Bae, F. Durand. Defocus magnification. *Proceedings of Eurographics*, 2007.
- [9] G. L. Yao, H. X. Yao. Trimap expansion based Alpha matting via localized windows. Technical Report, 2012.
- [10] J. Wang, M. Cohen. Optimized color sampling for robust matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007: 1-8.
- [11] C. Rhemann, C. Rother and M. Gelautz. Improving color modeling for alpha matting. *Proceedings of British Machine Vision Conference (BMVC)*, 2009: 1155-1164.
- [12] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli and P. Rott. A perceptually motivated online benchmark for image matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009: 1826-1833.
- [13] Z. P. Zhang, Q. S. Zhu and Y. Q. Xie. Learning based Alpha matting using support vector regression. *IEEE International Conference on Imaging Processing (ICIP)*, 2012.
- [14] E. Gastal, M. Oliveira. Shared sampling for real time alpha matting. *Proceedings of Eurographics*, 2010, 29(2): 575-584.
- [15] K. He, C. Rhemann, C. Rother, X. Tang and J. Sun. A global sampling method for alpha matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011: 2049-2056.
- [16] A. Levin, D. Lischinski and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2007, 30(1): 228-242.