Some Techniques of Image Matting

10421038　　董子龙
浙江大学 CAD&CG 实验室
浙江大学计算机系

摘要：

图像和视频抠图(Matting)技术可以分成自动和半自动；根据背景的先验知识，又有蓝屏背景，已知背景，和自然背景扣图。报告介绍了自然背景下的半自动扣图，以及能获得类似结果的技术，如 Snapping。其中我实现了 Bayesian Matting。

一、 Introduction

Image Matting 是将图像的背景和前景分离的技术，广泛用于图像合成和影视特技制作中。最早的技术在如蓝色绿色等单色背景下将前景物体分离，之后通过已知自然背景包含和剔除前景物体的两幅图像求解，类似的是同一个前景物体在多个背景下的多幅图像，最新的技术都是处理自然背景的单幅图像的半自动学习方法，包括将 Image Matting 的技术应用到视频，可以将运动的物体抠出，并合成到任意场景中。报告主要介绍半自动学习技术。

所谓半自动，是因为实现用户必须在图像上预先指定三个区域(Trimap)：背景(黑色)、前景（白色）、未知区域（灰色），图 1(b)。根据已知确定的前景背景，半自动技术才能一边学习一边求解。当然也有其它的交互方式，不一定需要这么详致的信息，在 2.3 和 2.4 种都是特殊的更为简化的交互。

Matting 一般基于公式 1，C 为未知点的颜色，α 是透明度，B 是背景色，F 是前景色：

$$C = \alpha F + (1-\alpha)B \qquad (1)$$

(a)　　(b)

图 1

二、 Image Matting

2.1. A Bayesian Approach to Digital Matting [1] (CVPR 2001)

Bayesian Matting (BM)定义一个 Bayesian 框架来公式化 Matting 的参数，并求解一个最大后验问题。BM 的改进还在于，通过滑动的窗口，将已经计算出来的未知区域也作为样本数据，这样在过渡区域获得更好的效果。BM 的记过比之前所有的方法都要好。

2.1.1 Bayesian Framework

如公式 2，其中 *F*、*B*、α 分别是需要求解的背景色、前景色、和不透明度，C 是已经知道的图像上一点的颜色。在 C 已知前提下，求使得概率 *P* 最大的 *F*、*B*、α，就是我们需要做的事情。*L* 是对数函数，将乘法化为加法，简化计算。

$$\begin{aligned} & \arg\max_{F,B,\alpha} P(F,B,\alpha \mid C) && (2)\\ = \; & \arg\max_{F,B,\alpha} P(C \mid F,B,\alpha)\,P(F)\,P(B)\,P(\alpha)\,/\,P(C) \\ = \; & \arg\max_{F,B,\alpha} L(C \mid F,B,\alpha) + L(F) + L(B) + L(\alpha), \end{aligned}$$

对等式右侧的四项分别建模：

L(*C*|*F*, *B*, α)对应以σ_C为标准差，中心在α F-(1-α)B 的高斯分布，公式 3。

$$L(C \mid F,B,\alpha) = -\|C - \alpha F - (1-\alpha)B\|^2 / \sigma_C^2 . \quad (3)$$

L(*F*)也对应一个高斯分布(公式 6)，通过空间的耦合性，即相邻像素的颜色，计算期望(公式 7)和协方差矩阵(公式 8)。先以未知点为中心的圆采样，不断扩大搜索半径，直到采到足够的已知背景和前景点，其中所有的已经求解的点也加入样本，样本根据颜色聚类。每一个样本点对应一个权值(公式 4)。其中α_i是不透明度，g_i是以距离为参数一个高斯衰减函数。

$$w_i = \alpha_i^2 g_i \quad (4)$$

$$W = \sum_{i \in N} w_i \quad (5)$$

$$L(F) = -(F - \overline{F})^T \Sigma_F^{-1} (F - \overline{F}) / 2. \quad (6)$$

$$\overline{F} = \frac{1}{W} \sum_{i \in N} w_i F_i \quad (7)$$

$$\Sigma_F = \frac{1}{W} \sum_{i \in N} w_i (F_i - \overline{F})(F_i - \overline{F})^T \quad (8)$$

L(*B*)与 *L*(*F*)类似，只是把全 *w* 的α_i替换成$1-\alpha_i$。

L(α)被当作定值。

2.1.2 求解过程

求解的是一个迭代过程。先假设α确定，对公式 1 右侧求偏导数，得到 6 元一次方程组(公式 9)，转化为一个简单的解线形方程组问题。

$$\begin{bmatrix} \Sigma_F^{-1} + I\alpha^2/\sigma_C^2 & I\alpha(1-\alpha)/\sigma_C^2 \\ I\alpha(1-\alpha)/\sigma_C^2 & \Sigma_B^{-1} + I(1-\alpha)^2/\sigma_C^2 \end{bmatrix} \begin{bmatrix} F \\ B \end{bmatrix} = \begin{bmatrix} \Sigma_F^{-1}\overline{F} + C\alpha/\sigma_C^2 \\ \Sigma_B^{-1}\overline{B} + C(1-\alpha)/\sigma_C^2 \end{bmatrix}, \quad (9)$$

将公式 8 解得的结果投射到颜色空间(公式 10)，得到新的α，再代入公式 9。

$$\alpha = \frac{(C-B)\cdot(F-B)}{\|F-B\|^2}. \quad (10)$$

初始的α可以取附近像素的α平均值。如果样本有多个类，则对前背景的样本聚类一一对应求解，并取使公式 1 最大的解。

2.2. Poisson Matting [2] (Siggraph 2004)

Poisson Matting(PM）是在图像的亮度梯度场上求解一个泊松方程，确定α。PM 首先在全局上求解，得到的结果通过多种局部滤波方式选择性的改进。由于给了用户更多的交互手段，处理复杂的场景比 BM 更加稳定。

2.2.1 泊松方程

将公式 1 表示为公式 11，对公式 11 两边求偏导数得到公式 12。左侧对应的就是输入图像的亮度梯度，只是估计值。假定背景和前景都是比较光滑的，那么项$\alpha\nabla F + (1-\alpha)\nabla B$近似为零，公式 12 变换成公式 13，这是对$\alpha$的梯度估计，PM 就是在这个梯度场上解出$\alpha$。

$$I = \alpha F + (1-\alpha)B \quad (11)$$

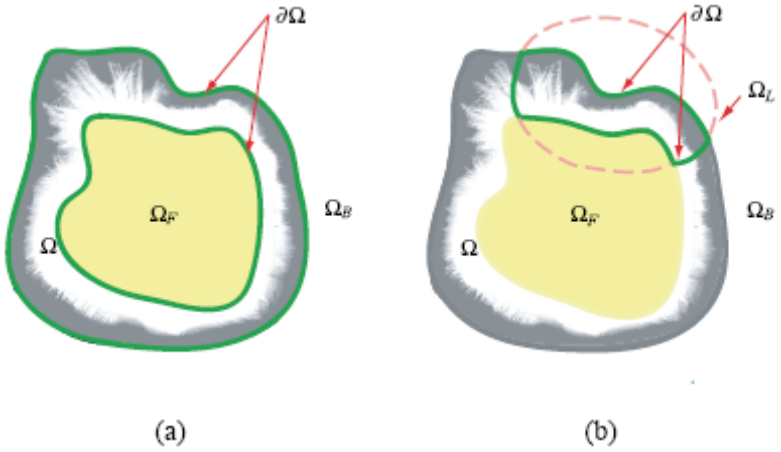
$$\nabla I = (F-B)\nabla\alpha + \alpha\nabla F + (1-\alpha)\nabla B \quad (12)$$

$$\nabla\alpha \approx \frac{1}{F-B}\nabla I \quad (13)$$

相应的泊松方程是公式 14，$\Delta = (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})$是拉普拉斯算子，$div$是散度运算。其狄利克雷边界条件$\alpha|_{\partial\Omega}$定义如公式 15，其中$\partial\Omega, \Omega_F, \Omega_B$代表未知区域的外边界，前景，背景，如图 2(a)。

$$\Delta\alpha = div(\frac{\nabla I}{F-B}) \quad (14)$$

$$\hat{\alpha}_p|_{\partial\Omega} = \begin{cases} 1 & p \in \Omega_F \\ 0 & p \in \Omega_B \end{cases} \quad (15)$$



(a) (b)

图 2

2.2.2 全局求解

这也是一个迭代优化，有三个步骤：

1) 初始化（*F*-*B*）项，因为 *F* 和 *B* 未知，分别近似为在 Ω_F, Ω_B 中最邻近的像素。

2) 计算 α，求解公式 14。

3) *F*，*B* 优化，使 $\Omega_F^+ = \{p \in \Omega \mid \alpha_p > 0.95, I_p \approx F_p\}$，即把几乎全部是前景色的像素加入到 Ω_F^+；类似的，$\Omega_B^+ = \{p \in \Omega \mid \alpha_p < 0.05, I_p \approx B_p\}$。然后将 Ω 中的 *F*，*B* 更新到在 Ω_F^+ 和 Ω_B^+ 与之最近的像素颜色。

重复步骤 2，3，直到 α 的变化足够小或 Ω_F^+ 和 Ω_B^+ 为空。

2.2.3 局部优化

局部优化事实上在全局求解结果不令人满意的区域加入用户的交互，使得对 α 梯度的估计更加准确，同时加入全局求解中被忽略的 $\alpha\nabla F + (1-\alpha)\nabla B$ 项，如图 2(b)区域 Ω_L。此时的边界条件是公式 16，α_g 是全局求解的结果：

$$\hat{\alpha}_p|_{\partial\Omega} = \begin{cases} 1 & p \in \Omega_F \\ 0 & p \in \Omega_B \\ \alpha_g & p \in \Omega \end{cases} \qquad (16)$$

2.3. GrabCut-Interactive Foreground Extraction using Iterated Graph Cuts [3](Siggraph 2004)

GrabCut 与上诉两种 Matting 不太一样，首先是在先验知识的获取上，只需要用户在前景物体周围画一个方框；其次，GrabCut 分成两个步骤，一是前景背景分割，二在膨胀的边界上在真正的 Matting，平滑过渡。GrabCut 在对比度较低的边界能获得比以往技术更好的结果，但是并不能处理复杂的情况，比如大面积的交错毛发，这是因为第一步的分割不能在这些区域获得好结果。

2.3.1 数据模型

GrabCut 的数据模型建立在 GraphCut 的基础上，只是用了一个高斯混合模型(Gaussian Mixture Model)代替原先的直方图，与 Bayesian Framework 类似。

对于输入图像$\mathbf{z} = (z_1,\ldots,z_n,\ldots,z_N)$，透明值表示成$\alpha = (\alpha_1, \cdots, \alpha_N)$，$0 \le \alpha_n \le 1$, 若是强分割，$\alpha_n \in \{0,1\}$，0 表示背景，1 是前景。GrabCut附加了一个向量$\mathbf{k} = (k_1,\ldots,k_n,\ldots,k_N)$, $k_n \in \{1,\ldots,K\}$ 到图像$\mathbf{z}$，代表来自背景或前景的一个GMM。于是，定义Gibbs能量：

$$\mathbf{E}(\underline{\alpha},\mathbf{k},\underline{\theta},\mathbf{z}) = U(\underline{\alpha},\mathbf{k},\underline{\theta},\mathbf{z}) + V(\underline{\alpha},\mathbf{z}) \qquad (17)$$

能量 E 包含了数据项和平滑项。数据项 U(公式 18)定义了α 的分布与输入 $\mathbf{z}$ 的匹配程度，其中 $D(\alpha_n,k_n,\underline{\theta},z_n) = -\log p(z_n \mid \alpha_n,k_n,\underline{\theta}) - \log \pi(\alpha_n,k_n)$。 $p(\cdot)$ 是一个高斯概率分布， $\pi(\cdot)$ 是混合权系数；通过对数函数将乘法和指数运算转化成加减法，全部展开为公式 19。 $\mu,\sum$ 分别是高斯分布的均值和协方差矩阵。

$$U(\underline{\alpha},\mathbf{k},\underline{\theta},\mathbf{z}) = \sum_n D(\alpha_n,k_n,\underline{\theta},z_n) \qquad (18)$$

$$D(\alpha_n,k_n,\underline{\theta},z_n) = -\log \pi(\alpha_n,k_n) + \frac{1}{2}\log\det\Sigma(\alpha_n,k_n) + \frac{1}{2}[z_n - \mu(\alpha_n,k_n)]^{\top}\Sigma(\alpha_n,k_n)^{-1}[z_n - \mu(\alpha_n,k_n)] \qquad (19)$$

于是，公式 19 相关的所有参数：

$$\underline{\theta} = \{\pi(\alpha,k),\mu(\alpha,k),\Sigma(\alpha,k),\ \alpha = 0,1,\ k = 1\ldots K\} \qquad (20)$$

最后是平滑项 V，约素相邻像素应该有相同的α：

$$V(\underline{\alpha},\mathbf{z}) = \gamma \sum_{(m,n)\in C} [\alpha_n \neq \alpha_m] \exp -\beta \|z_m - z_n\|^2 \qquad (21)$$

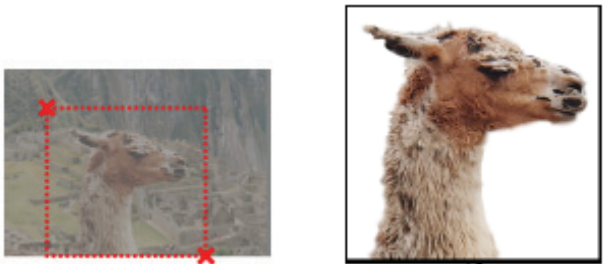
γ 是个常数，根据经验确定为 50。C 是像素之间的相邻关系，4 邻接或 8 邻接。$[\cdot]$ 取 0 或 1，根据谓词$\alpha_n \neq \alpha_m$。β 取大于 0 的值可以放松在高对比度区域的平滑限制，所以

$$\beta = \left(2\left\langle \|\mathbf{z}_m - \mathbf{z}_n\|^2 \right\rangle\right)^{-1} \qquad (22)$$

$\langle\cdot\rangle$ 表示样本的数学期望。这样取值，保证了公式 21 的指数运算在高低对比度间适当的切换。$\|\cdot\|$ 是像素颜色间的欧拉距离。

2.3.2 迭代最小化算法

GrabCut 在引入 GMM 的前提下，加入第二个改进，即迭代最小化算法。这使简化的交互成为可能，用户只需在前景物体周围拖动一个方框，如图 3(a)的红色虚线，方框之外的区域被认为是背景。



(a) (b)

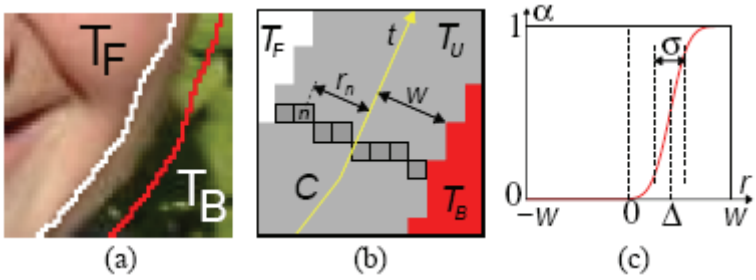
图 3

具体算法流程见[3]，图 3(b)是其中一个结果。

2.3.3 边界过渡

为了在边界上获得平滑过渡的效果，GrabCut首先将上一步得到的结果，生成新的三色图$\{T_B, T_U, T_F\}$，T_U是以边界轮廓线C为中心的宽 $2w$的窄带。

C首先参数化成t=1,….,T，因为 C 是闭合的，所以周期就是 T。T_U的每一个点 n对应一个索引$t(n)$。在图 4 (b)中， C就是黄色曲线，灰色区域是膨胀的边界区域，黑色边框包围的点有同一个t，称为一个α 剖面，r_n是点 n到C的距离。对每一个α 剖面都有一个平滑阶梯函数：$\alpha_n = g(r_n;\Delta_{t(n)},\sigma_{t(n)})$，如图 4（c）。



(a) (b) (c)

图 4

于是，我们要估计$\Delta_1,\sigma_1,...,\Delta_T,\sigma_T$，通过动态规划最小化以下 Gibbs 能量：

$$E = \sum_{n\in T_U} \tilde{D}_n(\alpha_n) + \sum_{t=1}^{T} \tilde{V}(\Delta_t,\sigma_t,\Delta_{t+1},\sigma_{t+1}) \qquad (23)$$

$$\tilde{D}_n(\alpha_n) = -\log \mathrm{N}\left(z_n;\mu_{t(n)}(\alpha_n),\Sigma_{t(n)}(\alpha_n)\right) \qquad (24)$$

$$\begin{aligned} \mu_t(\alpha) &= (1-\alpha)\mu_t(0) + \alpha\mu_t(1) \\ \Sigma_t(\alpha) &= (1-\alpha)^2\Sigma_t(0) + \alpha^2\Sigma_t(1) \end{aligned} \qquad (25)$$

$$\tilde{V}(\Delta,\sigma,\Delta',\sigma') = \lambda_1(\Delta-\Delta')^2 + \lambda_2(\sigma-\sigma')^2 \qquad (26)$$

最后，估计的前景色与已知的前景色比较，将差异最小的复制到当前区域，进一步保证平滑。

2.4. Lazy Snapping [4](Siggraph 2004)

Lazy Snapping 的模型基础也是 GraphCut，可以处理的情况与 GrabCut 相当。其特色在交互上，用户只需如图在前背景上简单的拖动曲线(如图 5)可以获得满意的效果，而且是实

时交互，用户可以随意的修改曲线的设定，同时进一步的优化修改交互工具也很简单稳定。这里简单说明一下 Lazy Snapping 的能量模型，更多的结果在[4]。

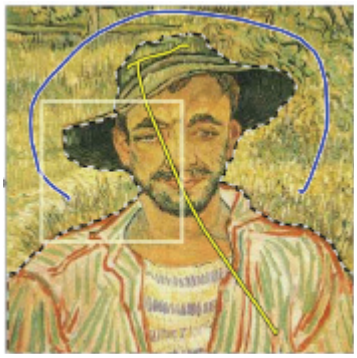


图 5

2.4.1 GraphCut 图像分割

图像被看成一个图，$G=\{V,\varepsilon\}$，V 是所有的节点，ε 是连接相邻节点的边。图像分割可以当作一个二元标记问题，每一个 $i\in V$，有唯一的一个 $x_i\in$ {前景为 1，背景为 0 }与之对应。所有的 x_i 集合 X 可以通过最小化 Gibbs 能量 $E(X)$获得：

$$E(X)=\sum_{i\in\mathcal{V}}E_1(x_i)+\lambda\sum_{(i,j)\in\mathcal{E}}E_2(x_i,x_j) \qquad (27)$$

同样的，根据用户画的曲线，我们有前景节点集 F和背景节点 B，未知节点集 U。首先用 K-Mean 方法将 F，B 的节点聚类，计算每一个类的平均颜色，$\{K_n^F\}$代表所有前景类的平均颜色集合，背景类是$\{K_n^B\}$。计算每一个节点 i 到每一个前景类的最小距离 $d_i^F=\min\left\|C(i)-K_n\right\|$，和相应的背景距离 $d_i^B=\min\left\|C(i)-K_n^B\right\|$，定义公式 28：

$$\begin{cases} E_1(x_i=1)=0 & E_1(x_i=0)=\infty & \forall i\in\mathcal{F} \\ E_1(x_i=1)=\infty & E_1(x_i=0)=0 & \forall i\in\mathcal{B} \\ E_1(x_i=1)=\frac{d_i^{\mathcal{F}}}{d_i^{\mathcal{F}}+d_i^{\mathcal{B}}} & E_1(x_i=0)=\frac{d_i^{\mathcal{B}}}{d_i^{\mathcal{F}}+d_i^{\mathcal{B}}} & \forall i\in\mathcal{U} \end{cases} \qquad (28)$$

前两组等式保证定义与用户输入一致，第三组等式意味着与前背景的颜色相近度决定着未知点的标记。

E_2定义为与梯度相关的一个函数：

$$\begin{aligned} E_2(x_i,x_j)&=|x_i-x_j|\cdot g(C_{ij}) \\ g(\xi)&=\tfrac{1}{\xi+1} \\ C_{ij}&=||C(i)-C(j)||^2 \end{aligned} \qquad (29)$$

E_2的作用是减少在颜色相近的像素之间，存在标记变化的可能，即使其只发生在边界上。

上述作用于像素的 GraphCut 因其效率并不适合实时交互。于是[4]在输入图像上进行预先作了过分割，然后把每一块当作一个像素来看待，颜色取均值，速度提升了 10 倍。

三、 Implementation

现在，我只在 Windows2000+Visual C++6.0 下实现了 Bayesian Matting 算法，结果和效率都基本和论文给出相似。程序界面如图 5，左侧窗口点右键打开图像(显示)，右侧窗口打开三色图(Trimap 不显示)，点击  运行算法，结果显示在右侧窗口中。

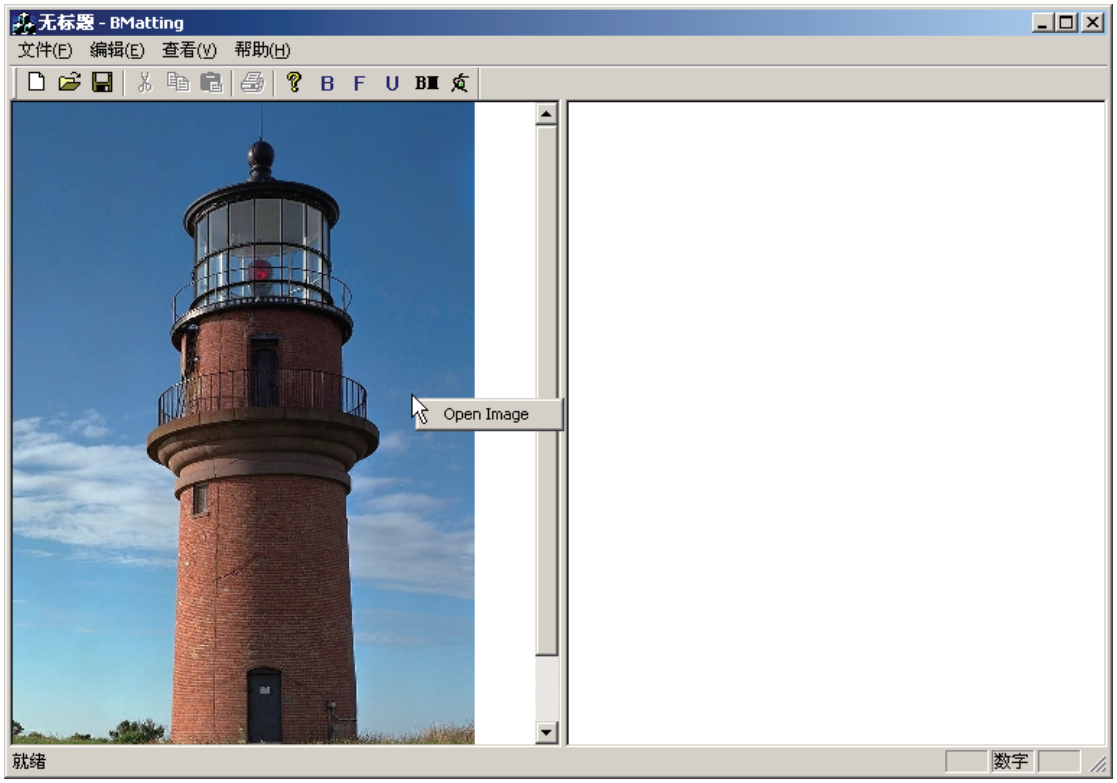


图 6

点击  ，可以修改程序参数：

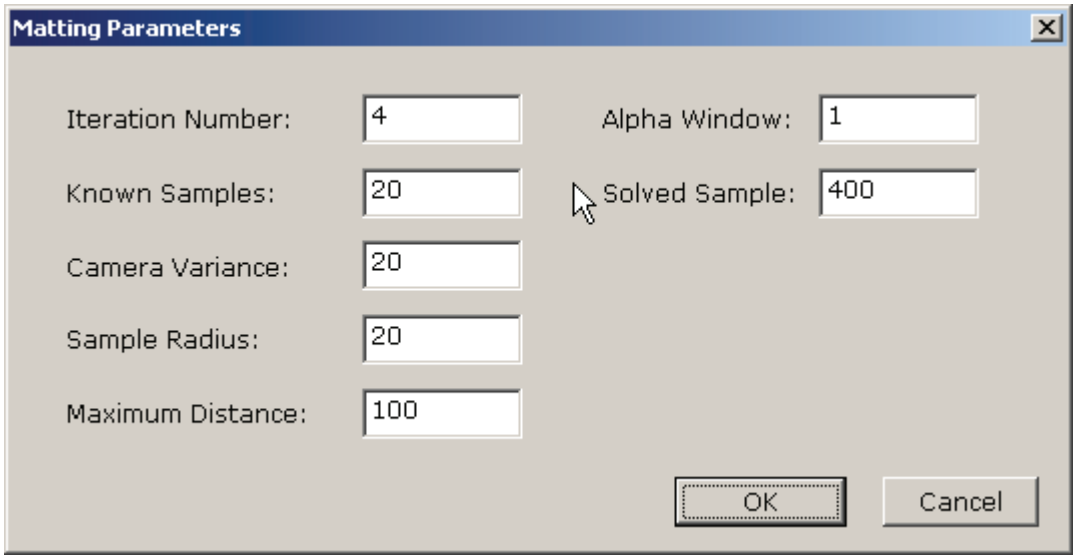


图 7

图 8，9，10，11，12 是试验结果，运行时间在 20 秒到 2 分钟，有些并不是最好的结果：

图 8

图 9

图 10

图 11

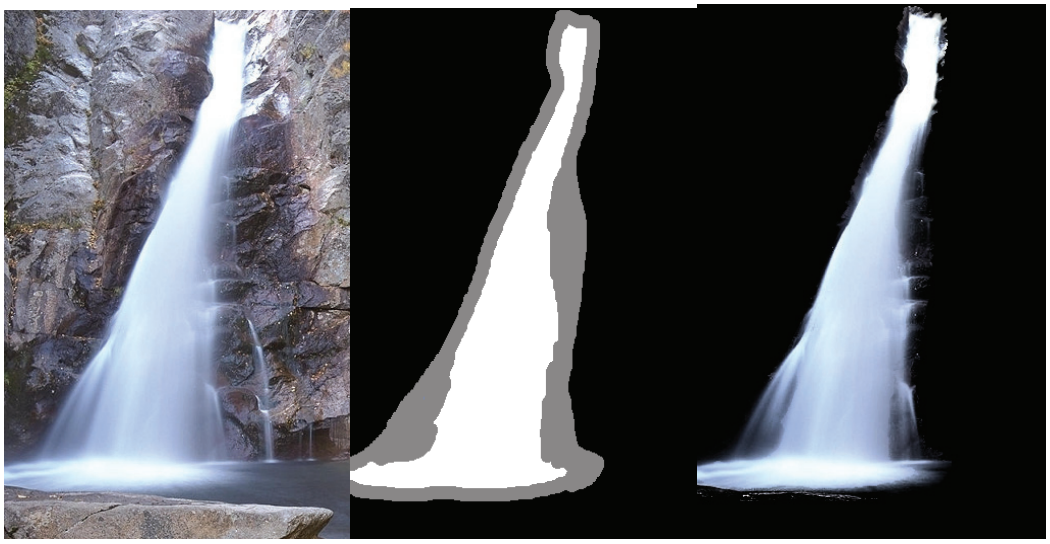


图 12

四、 References

[1] Chuang, Y.-Y., Curless, B., Salesin, D. H., Szeliski, R. 2001. A bayesian approach to digital matting. In *Proceedings of CVPR 2001, Vol. II, 264-271.*

[2] Jian Sun, Jiaya Jia, Chi-Keung Tang and Heung-Yeung Shum. Poisson Matting.In *Proceedings of ACM SIGGRAPH 2004, Vol 23, No. 3, April 2004.*

[3] ROTHER, C., BLAKE, A., AND KOLMOGOROV, V. 2004. Grabcut - interactive foreground extraction using iterated graph cuts. In *Proceedings of ACM SIGGRAPH 2004.*

[4] Yin Li, Jian Sun, Chi-Keung Tang, Heung-Yeung Shum, 2004, Lazy Snapping, In *Proceedings of ACM SIGGRAPH 2004.*