实验一 图像抠图 A Closed Form Solution to Natural Image Matting 实验报告

黄翘楚 计84 2018011363

公式推导

计算α

用F表示前景图，B表示背景图，α表示前景不透明度，图像抠图的公式为

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

对于彩色图片，假设在一个窗口内，前景色和背景色是线性的，那么在任意一个窗口内，有

$$\begin{aligned} F_i &= \beta_i^F F_1 + (1-\beta_i^F)F_2 \\ B_i &= \beta_i^B B_1 + (1-\beta_i^B)B_2 \end{aligned}$$

其中F_i, B_i均包含3个元素，表示3个通道的值，带入化简可得

$$[(F_1-F_2)(B_1-B_2)(F_2-B_2)] \cdot \begin{bmatrix} \alpha_i\beta_i^F \\ \beta_i^B - \alpha_i\beta_i^B \\ \alpha_i \end{bmatrix} = I_i - B_2$$

令$H_j = [(F_1-F_2)(B_1-B_2)(F_2-B_2)]$，则有

$$\begin{bmatrix} \alpha_i\beta_i^F \\ \beta_i^B - \alpha_i\beta_i^B \\ \alpha_i \end{bmatrix} = H_j^{-1} \cdot \begin{bmatrix} I_i^1 - B_2^1 \\ I_i^2 - B_2^2 \\ I_i^3 - B_2^3 \end{bmatrix}$$

其中上标表示通道，假设H_j^{-1}最后一行为$[a_j^1 \quad a_j^2 \quad a_j^3]$，可以得到

$$\begin{aligned} \alpha_i &= a_j^1(I_i^1 - B_2^1) + a_j^2(I_i^2 - B_2^2) + a_j^3(I_i^3 - B_2^3) \\ &= \sum_c a_j^c I_i^c + b_j \end{aligned}$$

因此，cost function为

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

令

$$G_k = \begin{bmatrix} I_1^1 & I_1^2 & I_1^3 & 1 \\ I_2^1 & I_2^2 & I_2^3 & 1 \\ & & & \cdots \\ I_9^1 & I_9^2 & I_9^3 & 1 \\ \sqrt{\epsilon} & 0 & 0 & 0 \\ 0 & \sqrt{\epsilon} & 0 & 0 \\ 0 & 0 & \sqrt{\epsilon} & 0 \end{bmatrix}, c_k = \begin{bmatrix} a_j^1 \\ a_j^2 \\ a_j^3 \\ b_j \end{bmatrix}, \bar{\alpha}_k = \begin{bmatrix} \alpha_{k_1} \\ \alpha_{k_2} \\ \cdots \\ \alpha_{k_9} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

则有

$$J(\alpha,a,b)=\sum_{k=1}^{n}|G_k c_k-\bar{\alpha}_k|^2$$

其中n为窗口数。为最小化函数，求偏导并令其等于0

$$\frac{\partial J(\alpha,a,b)}{\partial c_k}=2G_k^T(G_k c_k-\bar{\alpha}_k)=0$$

计算可得

$$c_k=(G_k^T G_k)^{-1}(G_k\bar{\alpha}_k)$$

带入后可得

$$\begin{aligned}G_k c_k-\bar{\alpha}_k&=[G_k(G_k^T G_k)^{-1}G_k^T-I]\bar{\alpha}_k\\&=\bar{G}_k\bar{\alpha}_k\end{aligned}$$

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

因此有

$$\begin{aligned}J(\alpha)&=\sum_{k=1}^{n}|G_k c_k-\bar{\alpha}_k|^2\\&=\sum_{k=1}^{n}(G_k c_k-\bar{\alpha}_k)^T(G_k c_k-\bar{\alpha}_k)\\&=\sum_{k=1}^{n}\bar{\alpha}_k^T\bar{G}_k^T\bar{G}_k\bar{\alpha}_k\\&=\sum_{k=1}^{n}\bar{\alpha}_k^T L_k\bar{\alpha}_k\\&=\alpha^T L\alpha\end{aligned}$$

其中$L_k=\bar{G}_k^T\bar{G}_k$

根据用户的输入，求解目标为

$$\begin{gathered}\min_{\alpha} J(\alpha)=\alpha^T L\alpha\\\alpha_i=1,\quad s.t.\,\alpha_i\in FG\\\alpha_i=0,\quad s.t.\,\alpha_i\in BG\end{gathered}$$

根据作者给出的思路，需要求解

$$\begin{gathered}\min_{\alpha} J(\alpha)=\alpha^T L\alpha\\s.t.\,(\alpha-\alpha_i)^T D_c(\alpha-\alpha_i)=0\end{gathered}$$

其中D_c为confidence对角矩阵，FG,BG上的点为1，其余为0。其Lagrange函数的解为

$$(\lambda D_c+L)\alpha-\lambda D_c b_c=0$$

b_c中FG的点为1，BG上的点为0，其余点为0.5

因此，算法的流程为，遍历每个窗口，计算每个窗口的G_k，进一步计算得到$\bar{G}_k$,从而得出每个窗口的L_k，根据每个L_k拼凑出L，再解最后的Lagrange方程得到α

计算F,B

求解出α后，需要进一步求解F和B。根据论文中的描述，需要满足

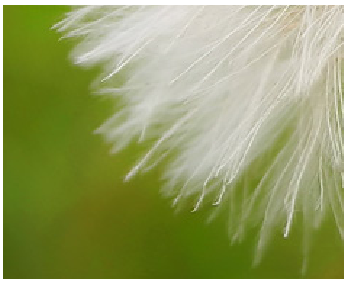
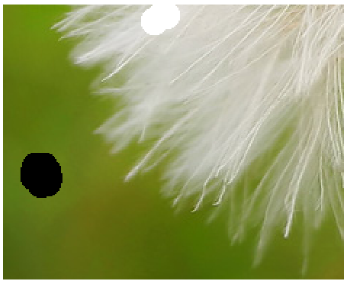
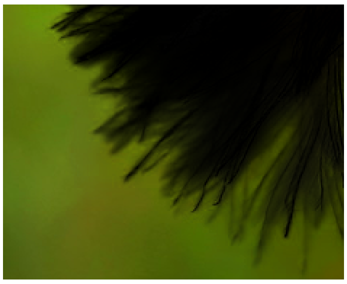
$$\min \sum_{i\in I}\sum_{c}(\alpha_i F_i^c + (1-\alpha_i)B_i^c - I_i^c)^2 \\ + |\alpha_{i_x}|((F_{i_x}^c)^2 + (B_{i_x}^c)^2) + |\alpha_{i_y}|((F_{i_y}^c)^2 + (B_{i_y}^c)^2)$$

其中下标i_x表示x方向导数，i_y表示y方向导数

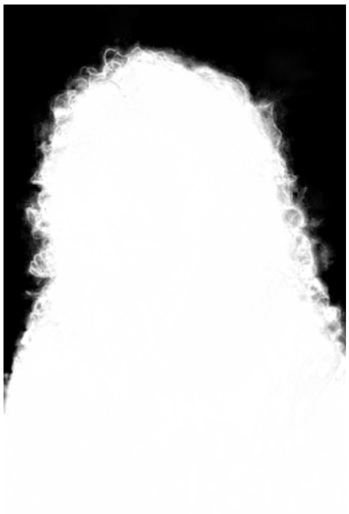
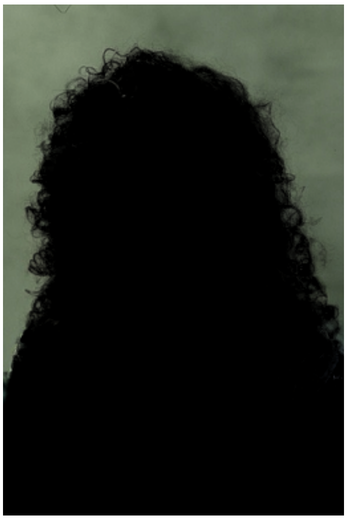
效果展示

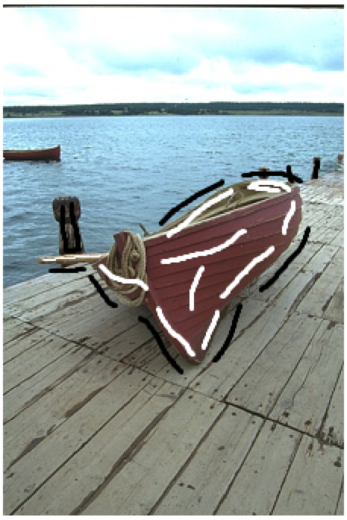
原图在image文件夹中，scribble在scribble文件夹中，α以及前景背景在result文件夹中。

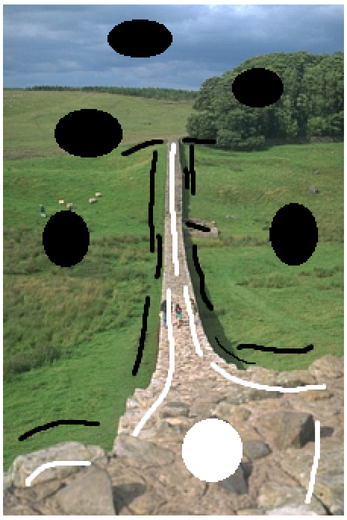
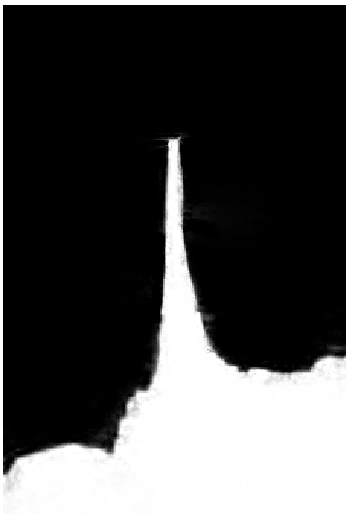
所有图片大小在200*300左右，运行时间大约为20s

原图	**scribble**	α
		
foreground(αF)	background($(1-\alpha)B$)	
		

原图	**scribble**	α
		
foreground(αF)	background($(1-\alpha)B$)	
		

原图	**scribble**	α
		
foreground(αF)	background($(1-\alpha)B$)	
		

原图	**scribble**	α
		
foreground(αF)	background($(1-\alpha)B$)	
		

原图	scribble	α
		
foreground(αF)	background($(1-\alpha)B$)	
		

运行方法

运行方法在`run.sh`中

```
python main.py --image_path ./image/dandelion.png --scribble_path
./scribble/dandelion.png
python main.py --image_path ./image/ship.png --scribble_path ./scribble/ship.png
python main.py --image_path ./image/woman.png --scribble_path
./scribble/woman.png
python main.py --image_path ./image/boy.png --scribble_path ./scribble/boy.png
python main.py --image_path ./image/road.png --scribble_path ./scribble/road.png
```

参考资料

[1]A. Levin, D. Lischinski and Y. Weiss, "A Closed-Form Solution to Natural Image Matting," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228-242, Feb. 2008, doi: 10.1109/TPAMI.2007.1177.

[2]github高星实现https://github.com/MarcoForte/closed-form-matting，参考解F和B的部分

[3]知乎[Closed Form Matting 算法抠图以及优化（1）——目标函数 - 知乎 (zhihu.com)]()，参考公式推导以及计算拉普拉斯矩阵

[4]知乎Closed Form Solution to Natural Image Matting公式推导 - 知乎 (zhihu.com)，参考解拉格朗日方程的推导