Project Report : Grabcut

Zeyuan Shang
2011012032
Tsinghua University

January 15, 2015

1 Introduction

The problem of efficient, interactive foreground/background segmentation in still images is of great practical importance in image editing. The resulting foreground object is an alpha-matte which reflects the proportion of foreground and background.

In this project, we experimentally examined approach proposed in [1], which could extract a foreground object.

2 Aproach

2.1 Image segmentation by graph cut

An energy function E is defined so that its minimum should correspond to a good segmentation. This is captured by a Gibbs energy of the form:

$$E(\alpha, \theta, z) = U(\alpha, \theta, z) + V(\alpha, z)$$

The data term U evaluates the fit of the opacity distribution α to the data z, given the histogram model θ, and is defined to be:

$$U(\alpha, \theta, z) = \sum_{\pi} - \log h(z_n; \alpha_n)$$

The smoothness term can be written as:

$$V(\alpha, z) = \gamma \sum_{(m,n) \in C} dis(m,n)^{-1} [\alpha_n \neq] e^{-\beta(z_n - z_m)^2}$$

The constant β is chosen to be:

$$\beta = \frac{1}{2(z_n - z_m)^2}$$

2.2 Gaussian Mixture Model

As we know, the Gaussian Mixture Model is as follows:

$$D(x) = \sum_{i=1}^{K}(\pi_i g_i(x;\mu_i,\Sigma_i)), \sum_{i=1} K(\pi_i = 1), 0 <= \pi_i <= 1$$

$$g(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

2.3 Colour Data Modelling

From 2.1 and 2.2, we could get:

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

2.4 Segmentation by Iterative Energy Minimization

2.4.1 Initialize

- User initialises trimap T by supplying only T_B. The foreground is set to $T_F = /0$; $T_U = T_B$, complement of the background.
- Initialise $\alpha_n = 0$ for $n \in T_B$ and $\alpha_n = 1$ for $n \in T_U$
- Background and foreground GMMs initialised from sets $\alpha_n = 0$ and $\alpha_n = 1$ respectively

2.4.2 Iterative minimisation

- Assign GMM components to pixels: for each n in T_U,

$$k_n := arg \min_{k_n} D_n(\alpha_n, k_n, \theta, z_n)$$

.

- Learn GMM parameters from data z:

$$\theta := arg \min_{\theta} U(\alpha, k, \theta, z)$$

.

- Estimate segmentation: use min cut to solve:

$$\min_{\alpha_n : n \in T_U} \min_{k} E(\alpha, k, \theta, z)$$

.

- Repeat from step 1, until convergence.

3 Result

Besides the output of the Grabcut, I also made some comparsion with the GrabCut in OpenCV, and my implementation is about $10 - 20\%$ faster than OpenCV on my computer. Please see figures below to check the results.

Iteration	My Time(ms)	OpenCV Time(ms)
1	4004	4426
2	4855	5710
5	8189	10260
10	15442	20084

Table 1: Comparsion with OpenCV

Reference

[1] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH*, 2004.

(a) input

(b) output

Figure 1: Statue