GrabCut论文复现及分析

Key Issues

1.What is GrabCut?

Graph Cut + GMM + Iterative energy

2.What matters in GrabCut?

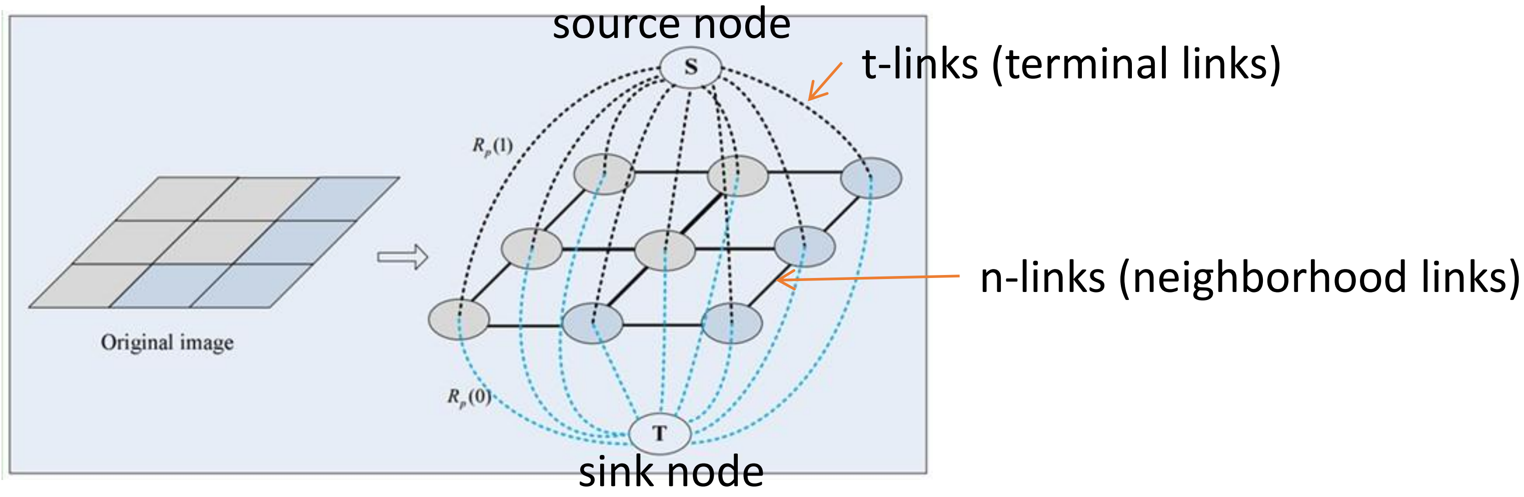
γ and K

3. GMM vs. Colorful Histogram?

User Interaction and Apply Restrictions

1.1 Graph cut

1. s-t graph



2.maxflow

最大流最小割定理：在一个网络流中，从源点到汇点的最大的流量，等于它的最小割中每一条边的容量之和

[1] BOYKOV, Y., AND JOLLY, M.-P, "Interactive Graph Cutsfor Optimal Boundary & Region Segmentation of Objects in N-D Images, " In Proc. IEEE Int. Conf. on Computer Vision, CD–ROM, 2001.

1.2 Energy function

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

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

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

区域能量项（全局分析）

边界平滑项（局部分析）

目标：将图像分为前景和背景两个不相交的部分，运用图像分割方法来实现。
建图：两类顶点 + 两类带权边（分别由区域能量项和边界平滑项计算得到）
分割：min-cut计算权值和最小的边的集合——对应能量最小化

1.3 Gaussian Mixture Model

1. d维高斯分布

$$\mathrm{p}(x|\{\mu,\Sigma\})=\frac{1}{\sqrt{(2\pi)^d|\Sigma|}}\exp[(x-\mu)^T\Sigma^{-1}(x-\mu)]$$

```cpp
//3维高斯分布：计算color像素属于第k个高斯分量的概率
double GMM::Possibility(int k, const Vec3d color) const
{
    double res = 0;
    if(Pi[k] > 0){
        Vec3d diff = color;
        double* m = mean + 3 * k;
        // X - \mu
        diff[0] -= m[0]; diff[1] -= m[1]; diff[2] -= m[2];
        /* (X - \mu)^T * cov^{-1} * (X - \mu)
            = [diff{0}, diff{1}, diff{2}] *  [covInv[k][0][0], covInv[k][0][1], covInv[k][0][2]] * [diff{0}]
                                             [covInv[k][1][0], covInv[k][1][1], covInv[k][1][2]]   |diff{1}|
                                             [covInv[k][2][0], covInv[k][2][1], covInv[k][2][2]]   [diff{2}]
        */
        double mult = (diff[0] * covInv[k][0][0] + diff[1] * covInv[k][1][0] + diff[2] * covInv[k][2][0]) * diff[0]
            + (diff[0] * covInv[k][0][1] + diff[1] * covInv[k][1][1] + diff[2] * covInv[k][2][1]) * diff[1]
            + (diff[0] * covInv[k][0][2] + diff[1] * covInv[k][1][2] + diff[2] * covInv[k][2][2]) * diff[2];
        res = 1.0f / sqrt(covDet[k]) * exp(-0.5f * mult);
    }
    return res;
}
```

2. 混合模型（加权和）

$$\mathrm{P}(x|\{\mu,\Sigma\})=\sum_{k=1}^{K}\pi_k\cdot\mathrm{p}(x|\{\mu_k,\Sigma_k\})$$

3. Energy function

$$\mathrm{U}(\alpha,\theta,z)=\sum_n-\log h(z_n,\alpha_n)$$

$$\mathrm{U}(\alpha,k,\theta,z)=\sum_n\mathrm{D}(\alpha_n,k_n,\theta,z_n)$$

$$\mathrm{D}(\alpha_n,k_n,\theta,z_n)=-\log(\pi(\alpha_n,k_n)\cdot\mathrm{p}(z_n|\alpha_n,k_n,\theta))$$

1.4 GrabCut流程

Initialisation

- User initialises trimap T by supplying only T_B. The foreground is set to $T_F = \emptyset$; $T_U = \overline{T}_B$, 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.

Iterative minimisation

1. *Assign GMM components to pixels:* for each n in T_U,
$$k_n := \arg\min_{k_n} D_n(\alpha_n, k_n, \underline{\theta}, z_n).$$
2. *Learn GMM parameters* from data $\mathbf{z}$:
$$\underline{\theta} := \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$
3. *Estimate segmentation:* use min cut to solve:
$$\min_{\{\alpha_n :\, n \in T_U\}} \min_{\mathbf{k}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}).$$
4. Repeat from step 1, until convergence.
5. Apply border matting (section 4).

User editing

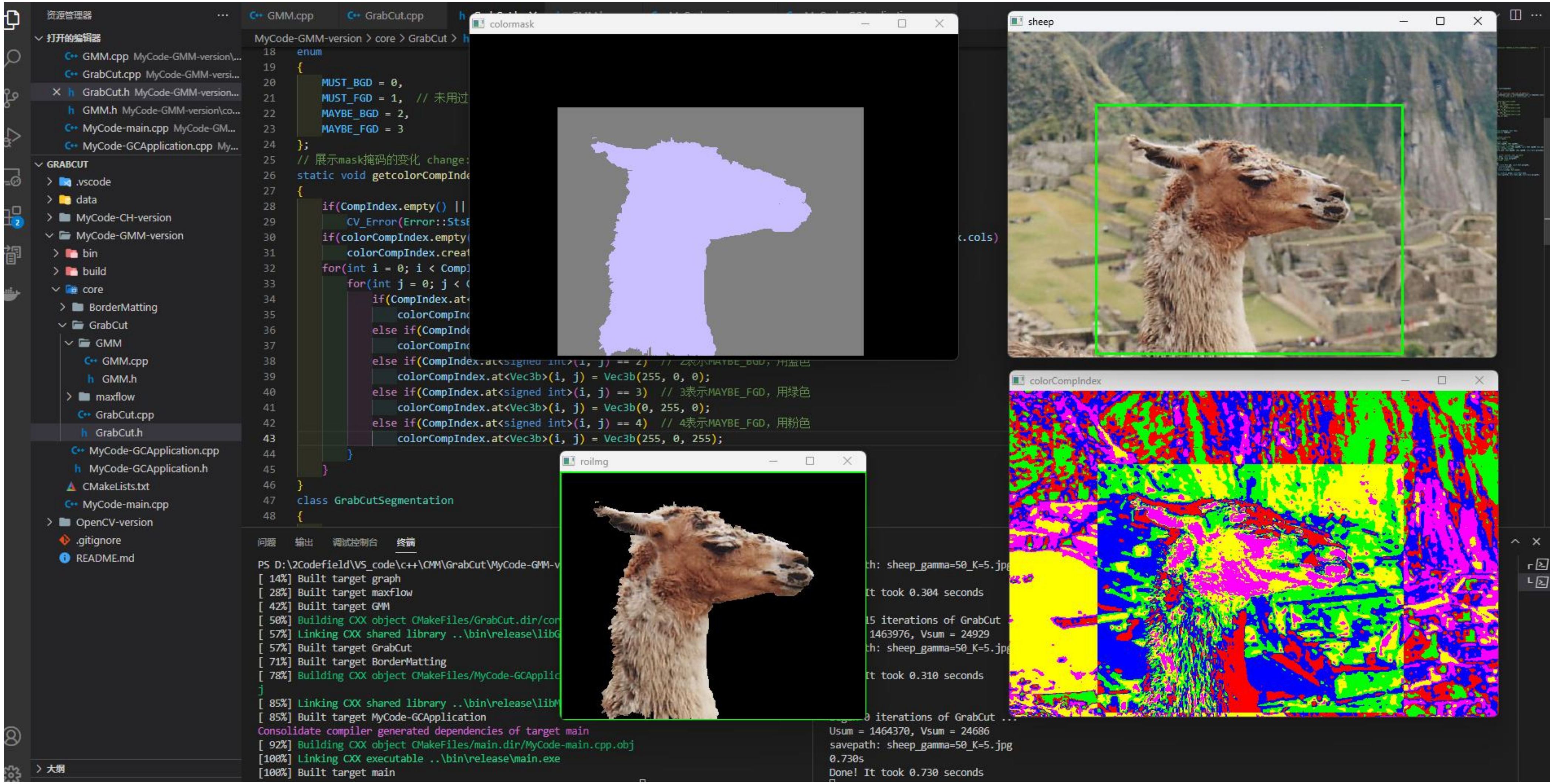
- *Edit:* fix some pixels either to $\alpha_n = 0$ (background brush) or $\alpha_n = 1$ (foreground brush); update trimap T accordingly. Perform step 3 above, just once.
- *Refine operation:* [optional] perform entire iterative minimisation algorithm.

```cpp
//GrabCut 主函数
void GrabCutSegmentation::GrabCut(InputArray arrayimg, InputOutputArray arraymask, Rect rect,
                                  InputOutputArray _bgdModel, InputOutputArray _fgdModel,
                                  int iterCount, int mode)
{
    Mat img = arrayimg.getMat();
    Mat& mask = arraymask.getMatRef();
    Mat& bgdModel = _bgdModel.getMatRef();
    Mat& fgdModel = _fgdModel.getMatRef();
    GMM backgroundGMM(bgdModel), foregroundGMM(fgdModel);  // 构建背景和前景的GMM模型
    if(mode == GC_WITH_RECT){
        // 初始化mask
        initMaskWithRect(mask, img.size(), rect);
        // 初始化GMM模型
        initGMMs(img, mask, backgroundGMM, foregroundGMM);
    }
    if(iterCount <= 0) return;

    // 计算Beta的值
    const double beta = CalcBeta(img);
    // 计算平滑项(边界能量项V)
    Mat leftWeight, upleftWeight, upWeight, uprightWeight;
    CalcSmoothness(img, beta, gamma, leftWeight, upleftWeight, upWeight, uprightWeight);
    // 存储每个像素属于哪个高斯模型
    Mat ComponentIndex(img.size(), CV_32SC1);
    const double lambda = 8 * gamma + 1;
    for(int i = 0; i < iterCount; i++){
        int vCount = img.cols*img.rows;
        int eCount = 2 * (4 * vCount - 3 * img.cols - 3 * img.rows + 2);  // 无向图=双向图
        Graph<double, double, double> graph(vCount, eCount);  // 建图
        AssignGMMComponents(img, mask, backgroundGMM, foregroundGMM, ComponentIndex);
        LearnGMMParameters(img, mask, backgroundGMM, foregroundGMM, ComponentIndex);
        getGraph(img, mask, backgroundGMM, foregroundGMM, leftWeight, upleftWeight, upWeight, uprightWeight, lambda, graph);
        EstimateSegmentation(graph, mask);
        // CalcEneryFunction(graph, mask, leftWeight, upleftWeight, upWeight, uprightWeight);
    }
}
```

[2] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut"– Interactive foreground extraction using iterated graph cuts," ACM TOG, vol. 23, no. 3, pp. 309–314, 2004.

Show



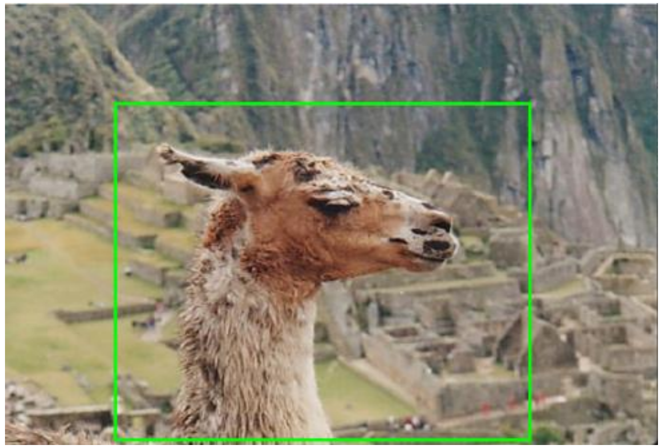
2.1 γ and K

控制变量分析：相同图片，相同位置的交互框，控制迭代2次

分析变量1：　$\mathrm{E} = \mathrm{U} + \gamma \cdot \mathrm{V}$

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

分析变量2：　K



2.1 γ & K

	K=2	K=5	K=10	K=20
$\gamma = 12$				
$\gamma = 50$				
$\gamma = 200$				

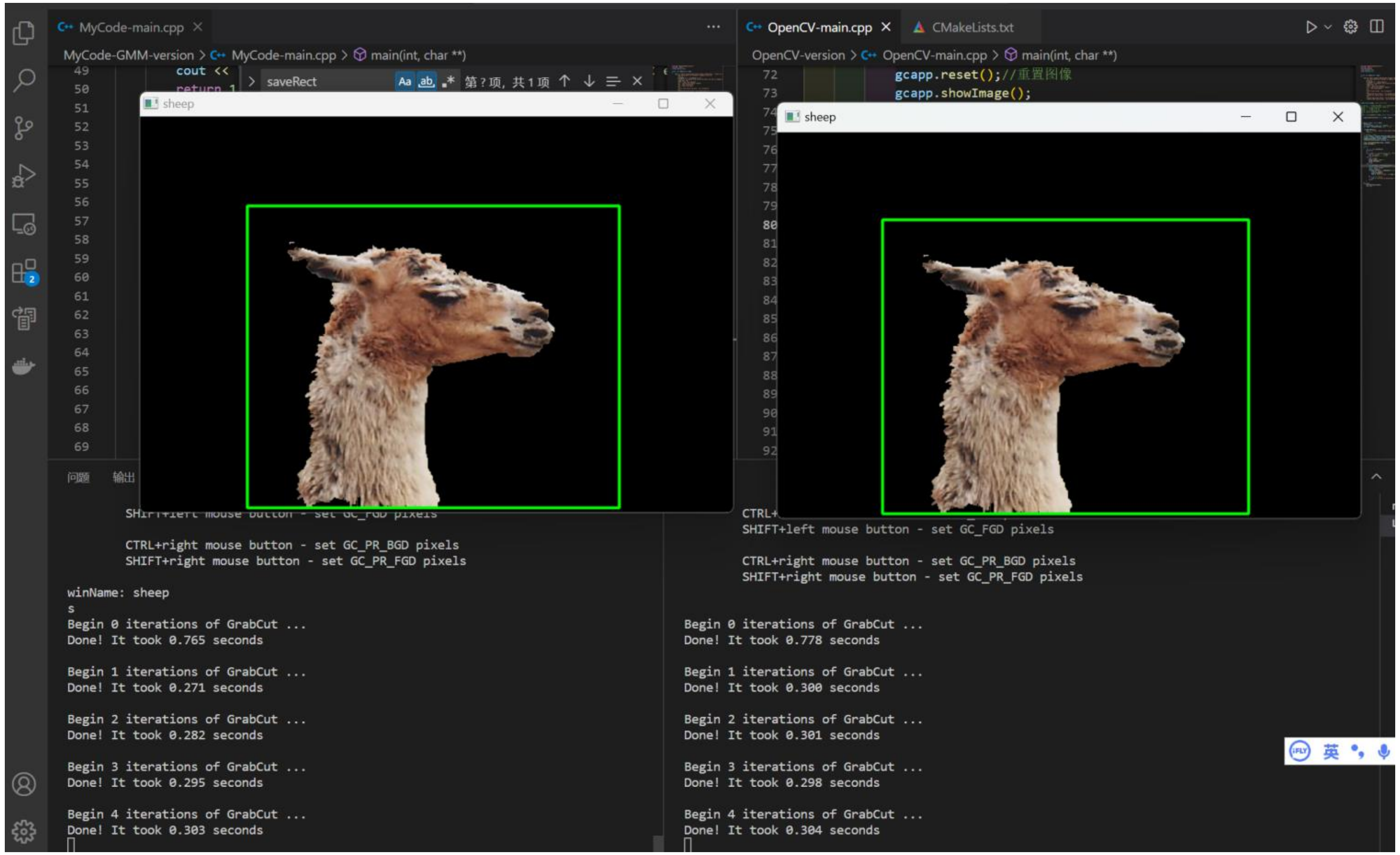
2.2 Analysis

1. **γ 作为平衡区域能量项和边界平滑项的权重，在分割效果上至关重要**

2. **K作为GMM划分特征的“超参数”，在小范围变动时影响不大。**

2.3 Time-consuming

	自己实现	调用opencv包
第1次迭代	0.765s	0.778s
第2次迭代	0.271s	0.300s
第3次迭代	0.282s	0.301s
第4次迭代	0.295s	0.298s



3.1 GMM vs. Colorful Hist

彩色直方图加速：量化颜色，平滑处理，保留颜色

1. 对于目标物体对比程度&显著性

2. 对于目标物体占比

[3] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 569–582, 2015.

Reference

[1] BOYKOV, Y., AND JOLLY, M.-P, "Interactive Graph Cutsfor Optimal Boundary & Region Segmentation of Objects in N-D Images, " In Proc. IEEE Int. Conf. on Computer Vision, CD–ROM, 2001.

[2] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut"– Interactive foreground extraction using iterated graph cuts," ACM TOG, vol. 23, no. 3, pp. 309–314, 2004.

[3] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 569–582, 2015.

Thanks