International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 07 | July - 2021 ISSN: 2582-3930

A Review on applying MDL based estimation to Gaussian mixture model for interactive Image segmentation

Trupti Jagtap , Dr. Archana Lomte

Computer Engineering, Bhivarabai Sawant Institute of Technology and Research

Pune, India

Abstract

Segmentation of objects from still images has many practical applications. In the past decade, combinatorial graph cut algorithms have been successfully applied to get fairly accurate object segmentation, along with considerable reduction in the amount of user interaction required. In particular, the Grabcut algorithm has been found to provide satisfactory results for a wide variety of images. This work is an extension to the Grabcut algorithm. The Grabcut algorithm uses Gaussian mixture models to _t the color data. The number of Gaussian components used in mixture model is however _xed. We apply an unsupervised algorithm for estimating the number of Gaussian components to be used for the models. The results obtained show that segmentation accuracy is increased by estimating the Gaussian components required, prior to applying the Grabcut algorithm.

Keywords: Interactive image segmentation, Gaussian mixture models, Minimum description length, Expectation maximization, Mincut/maxow algorithm

I. INTRODUCTION

Image segmentation is a fundamental step in many areas of computer vision including object recognition, video surveillance, face recognition, fingerprint recognition, iris recognition, medical analysis etc. It provides additional information about the contents of an image by identifying edges and regions of similar colour and texture. Although a first step in high level computer vision tasks, there are many challenges to an ideal image segmentation. Segmentation subdivides an object into its constituent regions or objects. The level of detail to which the subdivision is carried on depends on the problem being solved. That is the segmentation should stop when regions or objects of interest have been detected. For example, if an image consists of a tree, the segmentation algorithm may either stop after detecting the entire tree or further divide the tree into trunk and leaves.

II. LITERATURE SURVEY

A] Intelligent scissors

Intelligent scissors is one of the earlier approaches used for interactive image segmentation. Segmentation using intelligent scissors or the live-wire tool requires the user to enter seed points through mouse around the object to be segmented. The intelligent scissors selects the boundary as an optimal path between the current mouse position and previously entered seed point.

Let p and q be two neighbouring pixels in the image. l (p; q) is the cost on the link directed from p to q. We compute the cost function as weighted sum of image features: Laplacian Zero-Crossing cz, Gradient Direction cd, Gradient Magnitude cg, Inside-Pixel Value ci and Outside-Pixel Value co,Edge Pixel Value cp.

$$l(p;q) = wz \cdot cz(q) + wg \cdot cg(q) + wd \cdot cd(p;q) + wp \cdot cp(q) + wi \cdot ci(q) + wo \cdot co(q) \quad (2.1)$$

Laplacian zero crossing cz provides for edge localization around the object. It creates a binary cost feature. If the pixel is on the zero crossing, then Laplacian component cost from all links to that pixel is zero. Also, from a pair of neighbouring pixels which have opposite signs for their Laplacians, the pixel which is closer to zero is treated as having zero-crossing at that pixel. The gradient magnitude feature cg distinguishes between strong and weak edges. A smoothness constraint is added to the boundary by gradient direction cd. It assigns a high cost for sharp changes in gradient direction at neighbouring pixels, while the cost is low if the direction of the gradient at the two pixels is like each other.

Continuous training is performed while the boundary detection is performed i.e. the algorithm learns the characteristics of already detected boundary and uses them to make decision about current boundary. This allows the algorithm to select edges which are similar to the already sampled edges rather than just selecting the strong edges. Training features are updated interactively as the object boundary is being defined. The training considers pixels from the most recently defined object boundary, which allows the algorithm to adapt to gradual changes in edge characteristics. The image features cp, ci and co are used for training. Edge pixel values cp are simply the scaled source image pixel values directly beneath the portion of the object boundary used for training. The inside pixel value ci for pixel p is sampled a distance k from p in the gradient direction and the outside pixel value is sampled at an equal distance in the opposite direction. Following values are generally set for the weight coe_cients wz = 0:3, wg = 0:3, wd = 0:1, wp = 0:1, wi = 0:1, and wo = 0:1.

We assign weights to the edges and compute an optimal path from each pixel. This creates an optimal spanning tree. A variant of Dijkstra's algorithm is used for the purpose. The limitation of this approach is that multiple minimal paths may exists between the current cursor position and the previous seed point which increases the amount of user interaction required to get a satisfactory result.

B] Graph-cut for image segmentation

Y.Boykov and M-P Jolly proposed an interactive technique for segmentation of N-dimensional image. The user specifies a set of object and background pixels which form the hard constraints on the segmentation problem i.e. a segmentation is valid only if it correctly classifies the seed points as per user input. The soft constraints are specified such that both the region and boundary properties of the image are considered. These soft constraints define a cost function. The goal is to find the global minimum of the cost function that satisfies the hard constraints. To achieve this, we define the graph structure for the image in a manner that minimum graph cut corresponds to the optimal solution.

Let P be set of pixels. Let N be the neighbourhood system. N consists of

all unordered pairs {p,q} of neighbouring elements in P. Let (A1;A2;A3:::AjPj) be a binary vector. Each Ap can either be “obj" or “bkg", specifying that the pixel p belongs to object or background respectively. The cost function is then defined as

$$E(A) = \lambda R(A) + B(A) \tag{2.2}$$

where

$$R(A) = \sum_{p \epsilon P} R_p(A_p) \tag{2.3}$$

$$B(A) = \sum_{(p,q) \epsilon N} B_{\{p,q\}} \cdot \delta(A_p, A_q) \tag{2.4}$$

and

$$\delta(A_p, A_q) = \begin{cases} 1, & \text{if} A_p \neq A_q. \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

R(A) is the regional term and B (A) is the boundary term in the cost function E (A). represents the relative importance of the regional term and the boundary term. Rp (Ap) is the penalty of assigning the pixel p to Ap where Ap can either be “obj" or “bkg" as mentioned before. Boundary term is the summation of the the edge weights between those pixels p; q such that p and q belong to different classes. In order to calculate the regional term the object and background seeds are used. Let O and B denote the set of object and background pixels respectively.

Two histograms are created, one each for object and background, from the seeds entered by the user. These histograms are used to calculate the object and background intensity distributions Pr(I/O) and Pr(I/B).

The regional penalties are then set to the negative log-likelihoods of the probabilities.

$$R_p(\text{“}obj\text{”}) = -\ln \Pr(I_p/O) \tag{2.6}$$

$$R_p(\text{“}bkg\text{”}) = -\ln \Pr(I_p/B) \tag{2.7}$$

Bp;q represents the penalty for discontinuity between neighbouring pixels. Bp;q must be large when the pixels are similar to each other and close to zero when the pixels are dissimilar. The penalty also decreases with increase in distance between the pixels. Bp;q is thus given by,

$$B_{p,q} \propto exp\left(\frac{-(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)} \tag{2.8}$$

This equation sets a high value for Bp;q if Ip - Iq < σ while the value is small When Ip - Iq > σ , where σ is the expected value of the intensity difference between neighbouring pixels over the entire image.

To segment the image, graph G(V;E) is created. The set V of vertices includes two types of nodes. Every pixel p belonging to P is a node in graph. In addition, two more nodes, object terminal S and background terminal T are created. Therefore

$$V = P \cup (S \cup T)$$

Graph consists of edges of two types known as t-links and n-links. The edges between neighbouring pixels of the image are known as n-links. The edges between each pixel and the two terminals S and T are known as t-links. Denoting the n-links by the neighbourhood set N and the t-links for pixel p by {p; S} ; {p; T}, the set E of edges is

$$E = N \underset{p \epsilon P}{\cup} \{p, S\} \cup \{p, T\}$$

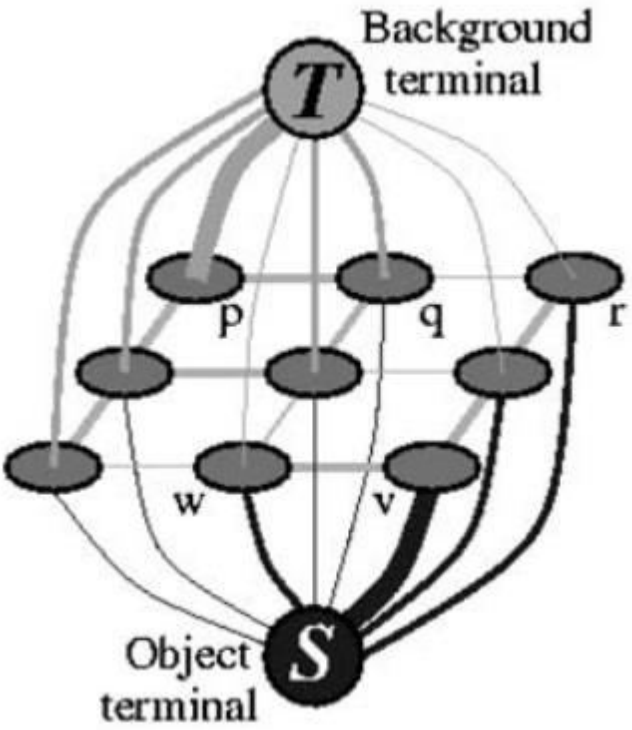

Figure 2.1: Graph structure of the image. Figure taken from [12]

Figure 2.1 shows such a graph.
Weights are assigned to the edges in E according to the Table 2.1

Edge	Weight	Pixel description
$\{p,q\}$	$B_{p,q}$	$\{p,q\} \in N$
$\{p,S\}$	$\lambda.R_p\,(\text{“}bkg\text{”})$	$p \in P, p \notin O \cup B$
	K	$p \in O$
	0	$p \in B$
$\{p,T\}$	$\lambda.R_p\,(\text{“}obj\text{”})$	$p \in P, p \notin O \cup B$
	0	$p \in O$
	K	$p \in B$

where

$$K = 1 + \underset{p \in P}{max} \sum_{q:\{p,q\} \in N} B_{p,q}$$

Table 2.1: Assignment of weights to the edges of the graph

C] Grabcut algorithm

Rother et al. [28] extended the image segmentation technique described in [12] for color images. Their technique reduced the user interaction to drawing a rectangle around the object to be segmented. Further

additions include iterative estimation and incomplete labelling, using the Gaussian Mixture Models(GMMs) for color data modelling.
The cost function, regional term and the boundary term are the same as in equation 2.2, 2.3 and 2.4 respectively. Calculations for coefficients Bp;q and Rp (Ap) are updated as described below.
The modified equation for the boundary coefficients Bp;q considering color pixels is,

$$B_{p,q} = \gamma \cdot exp\left(\frac{-(z_p - z_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist\,(p,q)} \qquad (2.9)$$

where zp and zq are the colours of pixels p and q respectively. The constant was set to 50, which was found to be a versatile setting for a wide variety of images.

The regional coefficients Rp (Ap) are estimated using GMMs for object and background regions. A trimap is considered for the pixels . The value of the trimap for each pixel can either be TrimapObject, TrimapBackground or TrimapUnknown. User creates an initial trimap by drawing a rectangle around the object. Pixels inside the rectangle are marked as TrimapUnknown. Pixels outside of rectangle are marked as TrimapBackground.
TrimapObject is initially an empty set. The TrimapUnknown pixels are then used to learn the initial object GMM while the TripmapBackground pixels are used to learn background GMM. The number of components in each GMM is set to 5.

Each pixel in the TrimapUnknown is assigned to the most likely Gaussian component in the object GMM. Similarly, each pixel in the TrimapBackground is assigned to the most likely Gaussian component in background GMM.

The GMMs are discarded and new GMMs are learned from the pixel assignments to Gaussian components done in the previous step. These new GMMs are used to calculate the regional coefficients Rp (Ap). Rp (“obj") and Rp (“bkg") are the likelihoods that the pixel p belongs to the background and object GMM respectively.

$$R_p\,(A_p) = -\ln \sum_{i=1}^{K} \left[\pi\,(A_p, i) \cdot \frac{1}{det\Sigma\,(A_p, i)} \times \exp\left(\frac{1}{2}\left[z_p - \mu\,(A_p, i)\right]^T \Sigma^{-1}\left[z_p - \mu\,(A_p, i)\right]\right)\right] \qquad (2.10)$$

where μ(Ap; i), Σ(Ap; i) and π (Ap; i) are the mean, covariance matrix and weight of component i of object GMM if Ap = “obj" and background GMM if Ap = “bkg"

After calculating the boundary and regional coefficients, weights are assigned to the edges of the graph as per Table 2.1. TrimapObject and TripmapBackground represent the set O and B in the table. This is followed by the mincut/maxow algorithm which separates the object and the background pixels.

After the initial segmentation result is displayed, the user can mark certain pixels as object pixels or background pixels, which get added to the set TrimapObject and TrimapBackground respectively, and re estimate the segmentation.

D] MDL based estimation for GMMs

Bouman et.al. developed an algorithm for modelling Gaussian mixtures

based on the Minimum Description Length(MDL) criteria proposed by Rissanen.
The algorithm estimates number of Gaussian components required to best fi the data along with the component parameters. We use this algorithm with a slightly modified MDL criteria to account for mixtures known as the Mixture MDL criteria. The EM algorithm described in section 2.4 gives the maximum likelihood (ML) estimation of observed data involving latent variables. In order to use EM for mixture models, the number of components of the mixture should be known beforehand. However, for applications like image segmentation the number of components is not known. It is observed that the likelihood obtained by the EM algorithm increases with the increase in number of components. This is because increase in the number of components results in better fitting of the data by the mixture model. However, choosing too many components, leads to the problem of over-fitting of data. The resulting mixture model is not very useful for classification purpose.

In order to overcome this problem, a penalty term is added to the likelihood which penalises higher order models. The penalty term acts as a trade-o_ between the likelihood of the data and the model complexity. The algorithm adds the penalty term based on the MDL principle suggested by Rissanen. Rissanen used the encoding length as a criteria for estimating the model accuracy. According to the MDL principle, the best model is the one that requires minimum number of bits to encode the data and parameters of
the model.

A MDL estimate depends on the particular coding technique used. However, Rissanen developed an approximate estimate based on some assumptions
which gives the MDL criteria. The steps og the algorithm can be summarized as follows:
1. Initialize GMM with maximum number of Gaussians, Kmax
2. Initialize θ 1 with below equations

$$\pi_k = \frac{1}{K_0}, \text{ where } K_0 \text{ is the initial number of components}$$

$$\mu_k = x_n, \text{ where } n = \frac{(k-1)(N-1)}{(K_0-1)} + 1$$

$$\Sigma_k = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^t$$

3. Apply the iterative EM algorithm to minimize the MDL criteria for current K
4. Store the parameter set θ along with the MDL value obtained.
5. If the number of Gaussians is greater than 1, reduce the number of
Gaussians by applying below equation and go to step 3.

$$(l, m) = \arg\min_{(l,m)} d(l, m)$$

6. Select the value K and corresponding parameters θ which give minimum value of MDL over all the values of K.

III CONCLUSION

Images from the Berkley image segmentation database [2] and the bounding boxes provided by the authors of Grabcut algorithm [3] were used for evaluation of the segmentation algorithm. F-measure is used as a metric for comparison of results. Thus, we see that the performance of the Grabcut algorithm can be improved by applying MDL based estimation to Gaussian mixture models. For certain images however, the number of Gaussian components is not correctly estimated. This leads to decrease in segmentation quality. Different methods for GMM estimation can be applied and tested to make the method applicable to wider variety of images.

REFERENCES

[1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In CVPR, pages 1125–1134, 2017.
[2] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In CVPR, pages 577–585, 2018.
[3] Guotian Yang, Minglei Han, Yingnan Wang, Zicheng Wang, Geng Wei, "YCbCr Color Space Based Image Segmentation Algorithm with Feedback in the Underground Cable Tunnel", Chinese Automation Congress (CAC) 2019, pp. 692-696, 2019.
[4] Jun Ma, Haoting Liu, Shaohua Yang, Baojun Duan, Ming Yan, Wei Long, "Image Segmentation and Feature Analyses of Imitated Intense X-ray Spot Radiation Source", Control Automation and Robotics (ICCAR) 2019 5th International Conference on, pp. 266-270, 2019.
[5] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In CVPR, pages 616–625, 2018.
[6] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. Nonparametric higher-order learning for interactive segmentation. In CVPR, pages 3201–3208, 2010.
[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, pages 1097–1105, 2012.
[8] http://research.microsoft.com/en-us/um/cambridge/projects/
visionimagevideoediting/segmentation/grabcut.htm.
[9] Christian Hennig. Methods for merging gaussian mixture components.
Advances in data analysis and classi_cation, 4(1):3{34, 2010.