OBJECT-RESPECTING COLOR IMAGE SEGMENTATION

Hongdong Li, Chunhua Shen

RSISE, Australian National University.
VISTA,National ICT Australia.
Email: firstname.lastname@anu.edu.au

ABSTRACT

The problem of foreground/background segmentation is of great importance in image processing and computer vision. We present a novel Linear-Programming (LP)-based algorithm for color image segmentation. This algorithm segments an image into a conceptually-meaningful foreground region (usually corresponding to the object of interest) and background regions. From a few user specified strokes we learn two Gaussian Mixture models corresponding to the foreground and background region respectively. The algorithm performs well even when the object region consists of several different colors and textures. Due to the global optimality of LP, our algorithm is free from the drawback of getting into local minima.

1. INTRODUCTION

Image segmentation is one of the central problems in image analysis and computer vision [3].

This paper addresses the problem of *object-respecting color image segmentation*, which means that: instead of segmenting an image into color homogenous regions, we segment an image into a foreground region corresponding to an object of interest, and background regions.

In some sense, our algorithm ia able to perform a high-level image analysis task, because the user-specified high-level notion of object of interest has been incorporated into to the low-level image segmentation task.

We implement the above idea via a simple Linear Programming (LP) approach. The image/object segmentation task has been naturally formulated as linear optimization, under a set of linear constraints. Since Linear Programming is a well-known mathematical technique, and many efficient algorithms are available for solving it, our color image segmentation method enjoys many of the advantages of LP.

Most remarkably, unlike many conventional image segmentation methods (such as Mean-shift or Belief Propagation) , our method is free from the risk of getting trapped into local minima. Experiments on various textured images have obtained successful results.

NICTA is funded through the Australian Government's Backing Australia's Ability Initiative, in part through the ARC.

2. RELATED WORK

Colors play a significant role in human visual system. We use color information as the key cue for image segmentation. There are several classical algorithms that are popularly adopted for solving the color image segmentation problem. One simple idea is to perform a simple k-means clustering on all the pixel values in a proper color space (e.g., RGB space or CIELab space, etc). An improved version of the clustering idea is through Gaussian Mixture Model estimation. The EM algorithm is commonly used for estimating mixture parameters in the GMM.

The normalized-cut method is an important image segmentation algorithm [2]. It performs well, has sound theoretic foundations and a simple implementation, hence has received much attention.

Yet another commonly adopted algorithm is the Mean-Shift method [1]. Mean-shift proves to be very efficient in detecting multiple modes existing in a color feature space, each mode corresponding to a cluster of color pixels.

The above norm-cut and mean-shift algorithms have common drawbacks. They all ignore the *local coherency* among neighboring pixels, which is believed to be crucial in image analysis. These algorithms belong to the so-called *global* approach, which means that they operate directly on a bag of orderless color feature vectors. None of them takes into account the local consistency issue, hence they often yield erroneous (e.g., over-segmentation) results.

To exploit such local information, sophisticated algorithms using delicate graph structures have been used for image segmentation. For example, Graph-cut and Belief-propagation based on MRF (Markov Random Field) image model have all been applied to the problem of image segmentation, and very successful results have been obtained [7][5][4].

These two algorithms (Graph-cut and Belief Propagation) represent the state of the art methods for image segmentation. However, both algorithms involve complicate non-convex optimization procedures. For example, in the Graph-cut, *ad hoc* local swap operations are used, while in the BP iterative local message passing is needed. The graph-cut seems to be able to converge to a solution very close to the true optimum.

However, its implementation is not easy.

In this paper, we provide a new approach for color segmentation. Our method takes into account not only global clustering information, but also local pixel-wise interaction information.

The resulting formulation is a simple Linear Program, i.e., minimize a linear cost-function under a set of linear constraints. Moreover, the user can easily incorporate constraints, so long as the constraints are linearly representable.

Since the LP is a mature mathematical technique and widely adopted by researchers in both academia and industry, it is expected our LP-based segmentation algorithm will find a wider audience.

3. GAUSS MIXTURE MODEL AND MRF FIELD

3.1. Global term: GMM

Since our goal is to segment an image into a conceptually-meaningful region (corresponding to the object of interest) and some remaining background regions, we need models to describe both the object and the background. To this end, we make use of the Gaussian Mixture Model (GMM) of their color distributions.

We denote the Gauss modes (i.e., components) of the object region by $f_k(X)$ ('f' stands for 'foreground'), where X is a color value $\in \mathbb{R}^3$ (e.g., (R, G, B)), f_k is the k-th Gauss mode, and $k = 1, \cdots, K$. K is the number of modes, usually specified by the user (in our experiments we chose K to be 7). By the GMM model the probability p_f of pixel i taking color value X is represent as: $p_f(X_i) = \sum_{k=1}^{K} \alpha_k g(X_i, \mu_k^f, \Sigma_{f,k})$, where $\sum_k \alpha_k = 1.0, \alpha_k \geqslant 0$.

Similarly, we can define the background GMM model as a combination of the background Gauss modes denoted by b_k ('b' stands for 'background'). Hence, the GMM probability p_b of a pixel belonging to the background region is (the number of background modes is K too):
$p_b(X_i) = \sum_{k=1}^{K} \beta_k g(X_i, \mu_k^b, \Sigma_{b,k})$,where $\sum_k \beta_k = 1.0, \beta_k \geqslant 0$.

To obtain the GMM model for a certain object, one can either use a supervised learning process, or via user intervention. For example in the latter way, the user achieves this by drawing some strokes on the image to be segmented, specifying the foreground and background regions. Fig-1 illustrates the process. Such an interactive way has been adopted by many methods such as the Grab-cut algorithm[5] and some image matting algorithms [4]. Some of our object segmentation results are shown in fig-2.

A potential difficulty with using only a few strokes is that the estimated GMM models may not faithfully represent the true color distributions of the whole image regions. To overcome this, in our formulation to be described below we do not directly use the GMM probability functions. Instead, Mahalanobis distances between a given color and each of the Gauss modes are used. This approach increases the robustness of the segmentation, because it allows for accounting for visual occlusions and illumination changes. Details are given below.

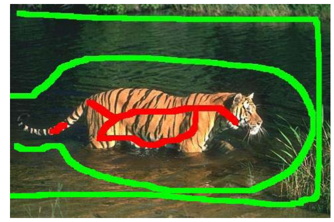

Fig. 1. Interactively build the GMM models for foreground and background from some user specified strokes.

We define the *foreground distance* of pixel i (denoted by $d_f(X_i)$) as the minimum Mahalanobis distance from the pixel value to each of the foreground Gauss modes:

$$d_f(X_i) = \min_k d(X_i, f_k) = \min_k \sqrt{(X - \mu_k^f)^{\mathrm{T}} \Sigma_{f,k}^{-1} (X - \mu_k^f)}.$$

Similarly define the background distance of X_i as:

$$d_b(X_i) = \min_k d(X_i, b_k) = \min_k \sqrt{(X - \mu_k^b)^{\mathrm{T}} \Sigma_{b,k}^{-1} (X - \mu_k^b)}.$$

Combining these two distances, we can further define a *difference of distance* as:

$$\delta(X_i) = d_b(X_i) - d_f(X_i). \tag{1}$$

Note that while both $d_f(X_i)$ and $d_b(X_i)$ are non-negative scalars, this $\delta(X_i)$ can be either non-positive or non-negative.

3.2. Local term: MRF

Local spatial coherency among neighboring pixels is a very important cue for image segmentation. To model such local interactions we use a MRF model with 4-neighboring connections.

Specifically, we encourage two neighboring pixels to have the same label, unless the color difference between them are sufficiently large. For segmentation purposes, the label of pixel i is $z_i = 1$ if it is a foreground pixel, and $z_i = 0$ otherwise. To reflect the above analysis, we express the label-consistency at pixel i as a weighted L_1 norm:

$$\gamma(i) = \sum_{j \in N(i)} (w_{ij} |z_i - z_j|),$$

where $N(i)$ is the neighbor set of i. The weights w_{ij} are defined by:

$$w_{ij} = \exp(-(\frac{X_i - X_j}{\sigma})^2) \tag{2}$$

where σ is a user-specified parameter. This formula is akin to the weight formula used in the normalized cut algorithm [2].

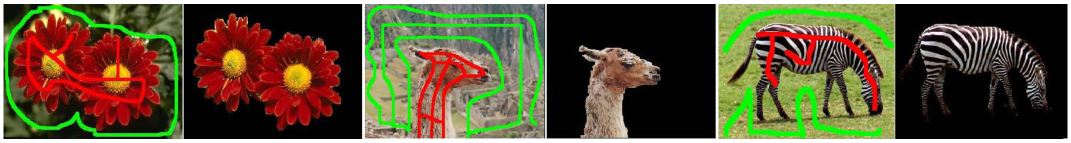

Fig. 2. Some of our object segmentation results.

3.3. 0-1 Integer Programming

We now formulate the color segmentation problem as an *energy minimization* problem. Particularly, it is a 0-1 integer programming problem because the variables to be minimized are 0-1 labels of pixels (1=foreground and 0=background).

Our cost function consists of two terms: a global term and a local term. The global energy is formed by summing up the difference of distance of all the pixels:

$$E_{global} = \sum_i z_i \delta_i. \tag{3}$$

The local energy tern is the sum of all local label-consistencies:

$$E_{local} = \sum_i \gamma_i = \sum_i \sum_{j \in N(i)} w_{ij} |z_i - z_j|. \tag{4}$$

Now the problem can be rephrased as finding the best 0-1 labels $z_i, i = 1, 2, ..., M \times N$ that minimize the energy function. M and N being the number of columns and the number of rows of the input image, respectively.

$$\min_{z_i} E = E_{global} + E_{local} =$$
$$\sum_i z_i \delta_i + \sum_i \sum_{j \in N(i)} w_{ij} |z_i - z_j|$$
$$\text{such that,}$$
$$\forall x_k \in L, x_k = b_k,$$
$$\forall i, z_i \in \{0, 1\}, \tag{5}$$

where L is the subset of user-specified pixels (i.e., the strokes), and b_k are the corresponding pixel labels.

Solving the above 0-1 Integer Programming problem exactly is an extremely hard problem (indeed, it has been proven to be NP-hard).

Rather than seeking the exact global optimal solution, if we settle for an approximate sub-optimal solution, then there exist many efficient algorithms for finding those approximate optimal solutions. Linear Programming Relaxation is one of the efficient algorithms, and is adopted in the paper.

4. THE MAIN RESULT: LP RELAXATION

From the definition it is clear that the weights in the local energy are all positive. Therefore we can effectively minimize it by minimizing an upper bound. Denote the upper bound at pixel i by y_i.

Now, we *relax* the 0-1 constraints to bound constraints. Then the above 0-1 Integer Programming problem is relaxed to a Linear programming problem:

$$\min_{z_i, y_i} \sum_i z_i \delta_i + \sum_i \sum_{j \in N(i)} w_{ij} y_i$$
$$\text{such that,}$$
$$\forall i, \forall j \in N(i), |z_i - z_j| \leqslant y_i, y_i \geqslant 0, 1 \geqslant z_i \geqslant 0,$$
$$\forall x_k \in L, x_k = b_k. \tag{6}$$

As a result, we have a standard LP. Being a well-studied technique, the LP has many highly-efficient algorithms such as the Interior point method. There are also many commercial and public implementation available for solving LP problems.

5. ALGORITHM OUTLINE

Our algorithm for color image segmentation proceeds as follows.

1. Input a RGB color image. Convert the color values into a proper color space. In this paper we use the CIELab space, as it is most close to human's visual color perception.
2. The user draws some sample strokes specifying the foreground object and backgrounds. Estimate the GMM models corresponding to the foreground and the background region respectively. Compute the δ_i for each pixel i (ref to sec-3.1).
3. Compute the weights w_{ij} for each neighboring pixel-pair. Here we use a 4-neighbor system.
4. Establish the LP Problem, according to eq.(6). Compute this LP problem using an LP solver.
5. Output labels as $\hat{z}_i = round(z_i)$, where the $\hat{z}_i$ is z_i rounds to 0 or 1. End.

5.1. Remarks

- Unlike many other optimization-based image segmentation methods such as Graph-cut or Belief propagation, using the proposed LP form we are guaranteed to find the true global optimum (only up to a constant factor due to the relaxation). There is no need for an initial guess, and no risk of local minima.
- Since polynomial-time algorithms exist and widely adopted for solving LP problems, our algorithm is also computationally efficient.
- Due to the LP formulation, we can easily incorporate prior knowledge about the scene into the segmentation process. For example, by letting $z_i = z_j$ we can specify that two pixels i and j have the same label.

6. EXPERIMENTAL RESULTS

This section gives some experimental results. We have implemented the above algorithm in Matlab. The LP solver used is Mosek's `Linprog`. It is an instance of the Interior Point Method. We test our algorithm on a moderate P-4 3Ghz 1GB machine. The image sizes are all normalized to about 120×120. The solver takes on-average 12 iterations to converge, and costs about 3 seconds.

Given an input image, we draw some strokes on the image to indicating the user-intended object region and background region. Some examples are shown in fig-1 and fig-3. Figure-4 and figure-5 show our segmentation results. [1] Clearly, the user's notion of foreground (object) and background has been well captured, and the segmentation results are very satisfactory.

Video object cutout. The proposed algorithm can be easily extended to video object cutout. The user is only required to specify some scribbles in the key frame(s). The segmentation result of the key frame is then propagated into other frames without further human interaction. Some sample resulting frames are given in fig-6.

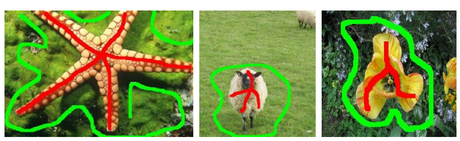

Fig. 3. Some sample images with user-specified strokes.

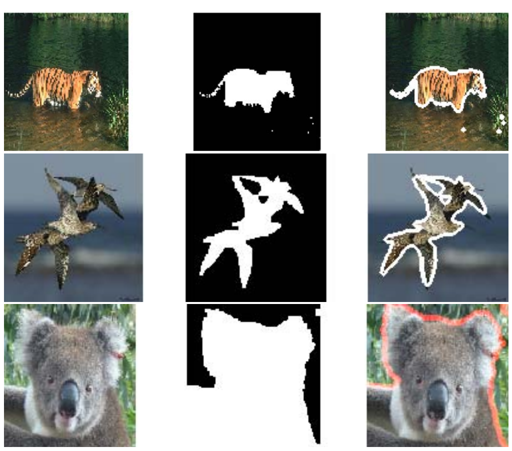

Fig. 4. Our segmentation results: L:input; R:segmentation.

[1]The test images are obtained from the Berkeley Segmentation Data-set and Benchmark: "www.cs.berkeley.edu/projects/vision/grouping/segbench", whom is much acknowledged.

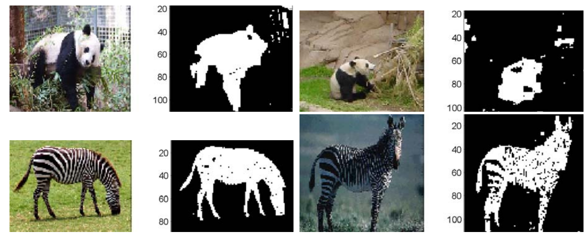

Fig. 5. More segmentation results: L:input; R:segmentation.

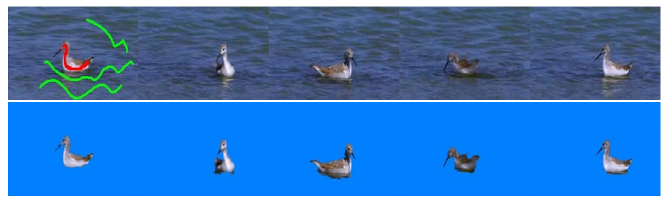

Fig. 6. Results of video-cutout.

7. CONCLUSION

This paper has addressed the important problem of segmenting still image into some conceptually-meaningful regions. An simple yet effective LP approach is described. This approach is easier to understand and implement than state-of-the art algorithms such as that based on Graph-cut and belief propagation. The computation is efficient because of the existence of polynomial-time LP solvers. We are planning to test our algorithm on other applications such as *background subtraction* in video surveillance.

8. REFERENCES

[1] B. Georgescu, I. Shimshoni, P. Meer, Mean Shift Based Clustering in High Dimensions: A Texture Classification Example, ICCV, 2003.

[2] J. Shi, J. Malik, Normalized cuts and image segmentation, PAMI, 2000.

[3] D. Martin, J. Malik et al, A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics, ICCV, 2001.

[4] Wang, J. and Cohen, M.,An iterative optimization approach for unified image segmentation and matting. In Proc. ICCV, 2005.

[5] Rother, C., Kolmogorov, V., and Blake, A. GrabCut: interactive foreground extraction using iterated graph cuts, SIGGRAPH, 2004.

[6] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. Proc. ECCV, 2004.

[7] Boykov, Y.,and Jolly, M., Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In Proc. ICCV,2001.