Segmentation by Grouping Junctions *

Hiroshi Ishikawa   Davi Geiger

Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
New York, NY 100 12
{ ishikawa, geiger} @cs.nyu.edu

Abstract

We propose a methodfor segmenting gray-value images. By segmentation, we mean a map from the set of pixels to a small set of levels such that each connected component of the set of pixels with the same level forms a relatively large and "meaningful" region. The method finds a set of levels with associated gray values by first finding junctions in the image and then seeking a minimum set of threshold values that preserves the junctions. Then it finds a segmentation map that maps each pixel to the level with the closest gray value to the pixel data, within a smoothness constraint. For a convex smoothing penalty, we show the global optimal solution for an energy function that fits the data can be obtained in a polynomial time, by a novel use of the maximum-flow algorithm. Our approach is in contrast to a view in computer vision where segmentation is driven by intensity gradient, usually not yielding closed boundaries.

1. Introduction

Image segmentation is a prototypical problem in computer vision, where one needs to organize the image and separate figure from ground.

This problem incorporates and goes beyond edge detection, since the output of the system must be regions delineated by closed contours.

1.1. Background

Of course, the large clustering community [13] have discussed various distinct approaches to this problem, e.g., the merge and split techniques, the K-mean approach, and others. Usually, they are not described as solutions to clear optimization criteria.

A class of approaches to this problem is an extension of the edge-detection view of images [1, 4, 8, 9, 1.5, 16], where image contrast (intensity gradient information) with some grouping process yields the final image boundaries. These techniques do not always yield closed boundary contours as the output, and they are never guaranteed to be optimized in polynomial time on the size of the image. Gradient approaches have other difficulties. While it is true that usually large image gradient are perceived as region boundaries, small intensity changes can also yield 'region boundaries (illusory contours being an extreme example).

Region approaches are our main interest. Most of the work, however, is ad-hoc when predefining the number of regions or predefining values for the regions. Layers approaches have shown promises in motion segmentation (Weiss [21]), and they yield segmentation directly (Darrell and Pentland [6]). Their usual problems are to estimate the number of layers and the values associated with the layers, where ad-hoc methods or prohibitive optimization computations (e.g., reducing to EM algorithms) are employed.

Recently, Shi and Malik [20], in a related approach to the one by Sarkar and Boyer [19], have proposed an interesting method that uses graph partitioning techniques to find what they call the normalized cut. Because their approach is again computationally prohibitive to solve exactly, they use a generalized eingenvalue approximation technique. The maximization of total dissimilarity between different groups of pixels and similarity within groups is interesting and we also consider it in a different form.

The optimization step in our approach is in the spirit of graph partition, but we map our optimization problem (grouping criteria) to a minimum cut problem on a directed graph. An advantage of our approach is that the globally

*This work was supported by NSF CAREER award and AFOSR Grant No. F49620-96-1-0159.

O-8186-8497-6/98 $10.00 © 1998 IEEE

optimal solution is obtained in a polynomial time by the use of the maximum-flow algorithm. Moreover, we propose a new way of estimating the number of layers and their values based on junction information.

1.2. Our approach

A segmentation is a classification of pixels into a small set of labels, which we call levels in this paper, because here each of them is associated with a gray value. Suppose we assign to each pixel the level with the nearest gray value. This is likely to yield an unsatisfactory result if the number of levels and their associated gray values are chosen arbitrarily. Thus, we wish to determine the optimal set of levels and gray values. We do this by detecting junctions in the image and choosing gray values that are needed to preserve the junctions in the resulting segmentation. Given a set of levels, we then assign one of the levels to each pixel. Though simply assigning the nearest level to each pixel may work in the noise-free case, in general we would also need some smoothing effect for the resulting segmentation to be useful. So we minimize a certain energy functional that balances the fitness to the data and the smoothness of the assignment map. The optimal solution is obtained by mapping the model to a maximum-flow problem in a directed graph, and solving it in a polynomial time.

Intuitively, we are proposing the use of the maximum-flow algorithm as a mechanism that group distinct regions of detected junctions into larger regions.

1.3. Related work

There are a number of recent work on application of network-flow algorithms to computer vision. For binary images, Greig, Porteous, and Seheult [10, 1 1] provided an efficient and optimal solution. Recent work [2, 7, 12, 18] extended this result to more than two levels in different ways. Since the problem is in general NP-hard, it is (at least) difficult to find an efficient exact solution to all of them. Approximate solution is one way: Boykov, Veksle, and Zabih [2] used an approximate multiway-cut algorithm to solve it approximately for a specific type of smoothing function, while Ferrari, Frigessi, and de Sá [7] used color coding. Limiting the applicable class of problems and exactly solving them is another way, including our approach: Roy and Cox [18] used maximum-flow algorithm for N-camera stereo correspondence problem, though they did not expressly mention the limitation. As a use of maximum-flow algorithm, our approach is most similar to [18]. Also, [12], which used directed graph maximum-flow for binocular symmetric stereo, used varied capacities (weights) for discontinuity penalties according to the context information, as [2] later but independently included in their method.

2. Formulation of the problem

We assume the input g to be an image corrupted by noise. We can typically raster scan an image and so g is represented by a vector in an N^2-dimensional vector space (for a square image of size N x N.) Then, $g_k(k=1,\ldots,N^2)$ represents the gray value at pixel ***k.*** Here we assume an 8-bit gray-scale image.

2.1. Junctions and levels

We segment an image by assigning to each pixel a level it belongs to. Each level has an associated gray value, and we wish to assign a level with the nearest gray value to each pixel, within a smoothness constraint.

We first determine the number of levels and their associated gray values. We take junctions (e.g., corners, T-junctions) as strong indicators of a region boundary. T-junctions and corners often arise from overlapping surfaces. which we wish to obtain as distinct segments in the image. Though sometimes they can arise from a mark on a surface, even in that case it is often appropriate to divide the mark from the rest as a distinct region. Let us assume that we have a junction detector (we have m'odified a junction detection model presented in Parida, Geiger and Hummel [17].) Each detected junction is a partition of a small disk in the image into ***K*** pie-shaped regions ***(K =*** 2 for corners, 3 for T- or Y-junctions, 4 for X-junctions, etc.), each with an assigned gray value. (See Figure l left.) The detector has several parameters we can use to filter junctions, including the contrast between partitions. We set the threshold for the contrast relatively high, so that only high-contrastjunctions are detected.

Now, we want these junctions to survive our segmenting process, since we are supposing these junctions indicate region boundaries. Therefore, the set ***F*** of gray values should satisfy the following condition:

> ***For each pair (e, e') of gray values assigned to neighboring regions in a junction, the nearest values in F to e and e' are always** different.*

We look for the minimum set of threshold values that separates any two neighboring regions in detected junctions by gray values. Suppose a junction has four regions a, b, c, and d with gray values e_a, e_b, e_c, and e_d in this order, and that the relations $e_a < e_b, e_c < e_b, e_c < e_d$, and $e_d < e_a$ hold. We call(e_a, e_b), (e_c, e_b), (e_c, e_d), and (e_d, e_a) the intervals

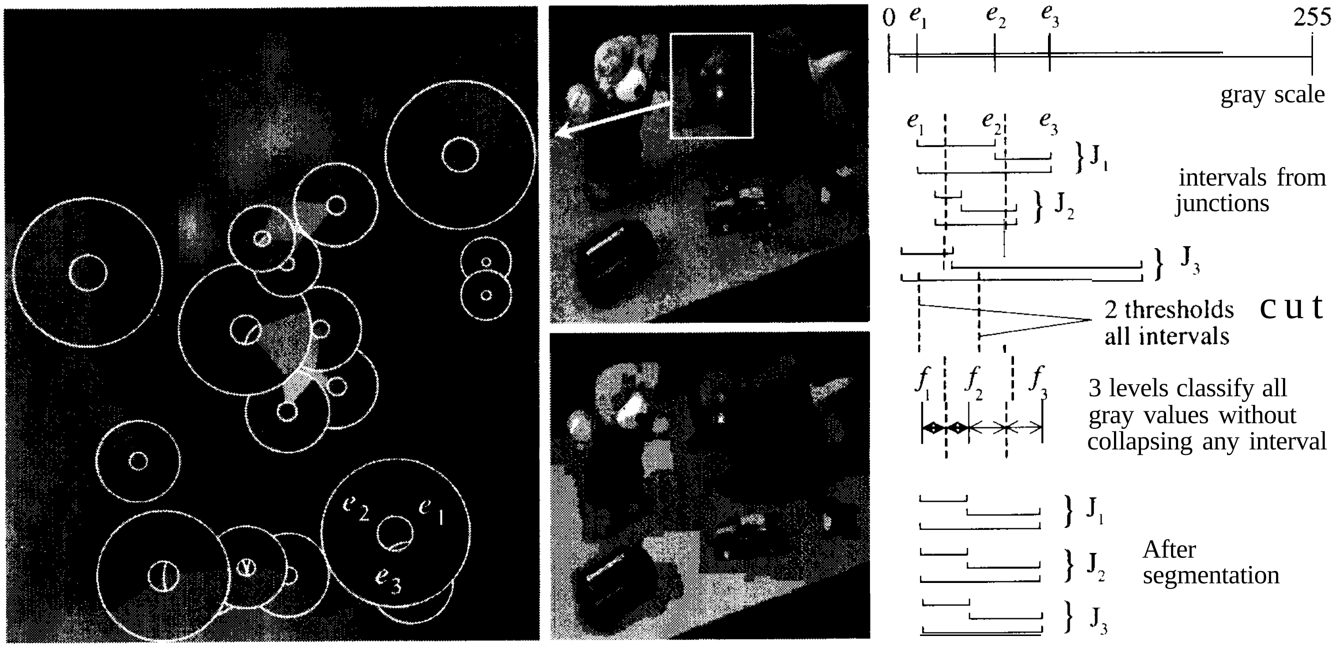

Figure 1. The segmentation process. First, junctions are detected, whose gray values are used to determine gray values for the levels. The map from pixels to levels are computed using the maximum-flow algorithm.

associated with the junction. Note that if we define thresholds so that each interval contains at least one threshold, all four gray values are divided by these thresholds. (See Figure 1 right.)

Let I be the set of all intervals that are associated with the detected junctions. We define T to be the smallest among the sets of gray values satisfying “for any $\iota \in$ I, there exists $t \in T$ such that $t \in \iota$”. Given such a set T, *we* define the set of gray values $F = \{f_1, f_2, \ldots, f_L\}$ for the levels to be the smallest set whose Voronoi diagram includes T. Then for each interval $(e,e') \in$ I, the nearest values in F to e and e’ are always different. We assume that the gray values are ordered in a natural manner:

$$f_a < f_{a+1}, \quad a = 1,...,L-1. \qquad (1)$$

2.2. The map

The next problem is to assign one of the possible levels to each pixel k, i.e., to find an assignment of a level $u(k) \in \{1,\ldots,L\}$ to each pixel k. We expect each grey value data g_k at pixel k to be close to the assigned grey value $f_{a(k)}$. This suggests a cost

$$\text{Error}(a,k) = \sum_{k=1}^{N^2} G(f_{a(k)} - g_k) ,$$

where G(x) is some error measure for which a square function is usually employed. However, our approach can handle any function G(x) as efficiently.

We also impose a smoothness constraint on the assignment function, where nearby pixels are encouraged to share the same level a. We consider the cost

$$\text{Smoothness}(a,k) = \sum_{k=1}^{N^2} \sum_{j\in N_k} F(a(k) - u(j)),$$

where N_k represents the neighborhood of pixel k. Typically, a four near neighbors is chosen, setting $N_k = (k-1, k+1, k-N, k+N)$. $F(x)$ is a smoothness function and we will show in Appendix that $F(x)$ must be a convex function for our method to guarantee an optima1 solution with polynomial time in N. We point out that the smoothing penalty function does not depend on the specific grey values $f_{a(k)}$ but rather on the level assignments $u(k)$. It is important to stress that the penalty is given to different *assignments* rather than a change in gray value $f_{a(k)}$. It is still the case, for increasing monotonic $F(x)$, that the further away the levels the larger is the penalty, since the levels are ordered as in (1).

This smoothness function will encourage regions to grow (not to have too many small regions) and account for noise errors.