Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration

Sharon Alpert Meirav Galun Ronen Basri Achi Brandt
Department of Computer Science and Applied Mathematics
The Weizmann Institute of Science
Rehovot, 76100, Israel
{sharon.alpert,meirav.galun,ronen.basri,achi.brandt@weizmann.ac.il}

Abstract

We present a parameter free approach that utilizes multiple cues for image segmentation. Beginning with an image, we execute a sequence of bottom-up aggregation steps in which pixels are gradually merged to produce larger and larger regions. In each step we consider pairs of adjacent regions and provide a probability measure to assess whether or not they should be included in the same segment. Our probabilistic formulation takes into account intensity and texture distributions in a local area around each region. It further incorporates priors based on the geometry of the regions. Finally, posteriors based on intensity and texture cues are combined using a mixture of experts formulation. This probabilistic approach is integrated into a graph coarsening scheme providing a complete hierarchical segmentation of the image. The algorithm complexity is linear in the number of the image pixels and it requires almost no user-tuned parameters. We test our method on a variety of gray scale images and compare our results to several existing segmentation algorithms.

1. Introduction

Segmentation algorithms aim at partitioning an image into regions of coherent properties as a means for separating objects from their backgrounds. As objects may be separable by any of a variety of cues, be it intensity, color, texture, or boundary continuity, many recent algorithms (e.g. [17, 16, 15]) have been designed to utilize and combine multiple cues. Typically in such algorithms, each cue is handled by a separate module whose job is to assess the coherence of nearby pixels or regions according to that cue, and a segmentation decision is obtained by incorporating these similarities into a combined measure. Careful design of these modules along with the use of appropriate optimization methods has led to notable successes, but the challenge of reliably segmenting objects in a variety of natural images still lies ahead.

The utilization of multiple cues aggravates an old problem. In many multi-cue segmentation algorithms each module comes with its own set of parameters, and those join an additional set of parameters intended to control the relative influence of each module. These parameters may depend non-trivially on the particular statistics of the input image, or even the statistics of different regions in the same image. While existing methods may be robust to changes in some of those parameters, segmentation results in many cases may depend critically on the proper assignments of parameter values. The common practice is to leave those parameters to be set by the user, but in effect most users leave the parameters in their default values. Allowing these parameters to automatically adapt to an image (or even locally to image portions) can greatly simplify the use of segmentation algorithms and potentially allow them to consistently provide better results. Indeed, recent algorithms attempt to achieve parameter-free segmentation either by relying on a training set that includes a variety of manually segmented images (e.g., [11]) or by estimating a global set of parameters based on stability criteria [15].

In this paper we explore a different approach which relies primarily on local information available within the image to be segmented. We present a parameter free probabilistic approach to segmentation. Beginning with an image, we execute a sequence of steps in which pixels are gradually merged to produce larger and larger regions. In each step we consider pairs of adjacent regions and provide a probability measure to assess whether or not they should be in-

Research was supported in part by the European Community grant IST-2002-506766 Aim@Shape, by the US-Israel Binational Science Foundation grant number 2002/254, by the A.M.N. Fund for the promotion of science, culture and arts in Israel, and by the Israel Institute of Technology. The vision group at the Weizmann Institute is supported in part by the Moross Foundation.

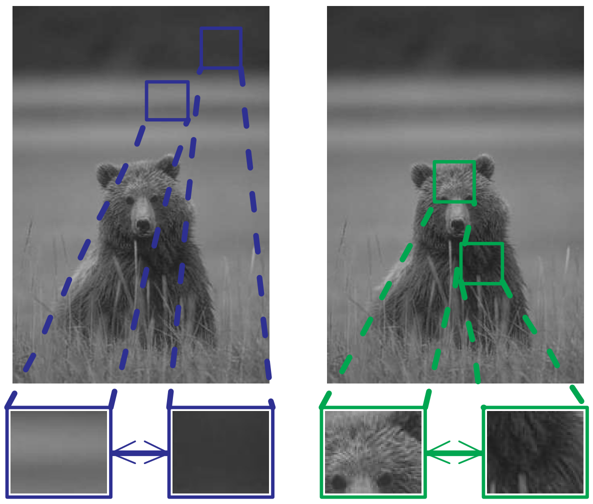
Figure 1. The importance of adaptive, local cue integration. Left: two patches that can be distinguished by intensity (the patches have uniform textures). Right: two patches with similar texture that should be merged despite their different intensities (due to lighting).

cluded in the same segment. We illustrate this method by constructing modules to handle intensity contrast and texture differences, and use an adaptively controlled "mixture of experts"-like approach to integrate the different cues and reach unified segmentation decisions. To demonstrate the importance of adaptive, local cue integration consider the example in Figure 1, which shows two pairs of regions. The left pair can be distinguished by intensity cues, whereas the right pair of patches, which have similar texture, should be merged despite their different intensities.

Our approach is designed to work with bottom-up merge strategies for segmentation. A large number of methods approach segmentation using bottom-up merge strategies, beginning with the classic agglomerative clustering algorithm [4] to watershed [19, 12] and region growing (including methods that use probabilistic approaches [14, 13]) to more recent algebraic multigrid inspired aggregation [16]. Merge algorithms generate a hierarchy of segments, allowing subsequent algorithms to choose between possible segmentation hypotheses. For implementation we adapt the coarsening strategy introduced in [16], as it enables incorporating at every level of the hierarchy measurements appropriate to the scale at that level. We further test our parameter-free approach on a database with manually segmented images and compare our results to several existing algorithms.

The paper is divided as follows. Section 2 introduces our probabilistic framework. Section 3 describes how we incorporate our probabilistic framework into a graph coarsening procedure. Finally, Section 4 provides experimental evaluation of our method.

2. Probabilistic framework

We consider a bottom-up aggregation approach to image segmentation. In this approach beginning with an image, we execute a sequence of steps in which pixels are gradually merged to produce larger and larger regions. In this section we focus on one step of such a procedure, in which a division of the image into a set of regions $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$ is given, along with a set of observations, $\vec{\mathcal{H}}_i \in \mathbb{R}^d$ for each region R_i ($i = 1 \ldots n$). Our objective is to further merge these regions to produce larger regions of coherent properties.

To achieve this goal we consider pairs of adjacent regions, R_i and R_j, and provide a measure to assess whether or not they should be merged into a single segment. We define a binary random variable $\mathbf{s}_{ij}$ that assumes the values $\mathbf{s}_{ij}^+$ if R_i and R_j belong to the same segment and $\mathbf{s}_{ij}^-$ if they do not. We then wish to estimate the probability $P(\mathbf{s}_{ij}^+|\vec{\mathcal{H}}_i, \vec{\mathcal{H}}_j)$ which we will use to determine whether or not to merge the two regions based on their respective properties.

Since segmentation decisions may be affected by several cues, we need a method to integrate the different cues. Here we consider both intensity and texture cues and integrate them using the "mixture of experts"-like model, as follows.

$$\begin{aligned} P(\mathbf{s}_{ij}^+|\vec{\mathcal{H}}_i, \vec{\mathcal{H}}_j) &= \\ &\sum_k P(\mathbf{s}_{ij}^+, c_k|\vec{\mathcal{H}}_i, \vec{\mathcal{H}}_j) = \\ &\sum_k P(\mathbf{s}_{ij}^+|\vec{\mathcal{H}}_i, \vec{\mathcal{H}}_j, c_k) P(c_k|\vec{\mathcal{H}}_i, \vec{\mathcal{H}}_j). \end{aligned} \tag{1}$$

This equation implies that the probability of a merge is determined separately for each cue c_k, and the term $P(c_k|\vec{\mathcal{H}}_i, \vec{\mathcal{H}}_j)$ enables us to adjust the influence of each cue dynamically according to the characteristics of the regions.

To evaluate the probability of a merge for each cue we apply Bayes' formula:

$$P(\mathbf{s}_{ij}^+|\vec{\mathcal{H}}_i, \vec{\mathcal{H}}_j, c_k) = \frac{L_{ij}^+ P(\mathbf{s}_{ij}^+|c_k)}{L_{ij}^+ P(\mathbf{s}_{ij}^+|c_k) + L_{ij}^- P(\mathbf{s}_{ij}^-|c_k)} \tag{2}$$

where $L_{ij}^\pm \triangleq p(\vec{\mathcal{H}}_i, \vec{\mathcal{H}}_j|\mathbf{s}_{ij}^\pm, c_k)$ denote the likelihood densities given $\mathbf{s}_{ij}^\pm$ respectively. These likelihoods are determined locally according to properties of surrounding regions. We further use a prior that is independent of cue, $P(\mathbf{s}_{ij}|c_k) = P(\mathbf{s}_{ij})$, and determine this prior based on the geometry of the two regions, i.e., their relative length of common boundaries.

In the remainder of this section we elaborate on how we model the likelihood densities, the cue arbitration, and prior probabilities.

2.1. likelihood densities

Below we describe how we derive the likelihood densities for each of our cues, intensity and texture. Both likelihoods are determined from the image by local properties of surrounding regions. Roughly speaking, the underlying principle in our choice of likelihoods is that in principle we consider it likely that a region would merge with its most similar neighbor, while we consider it unlikely that a region would merge with all of its neighbors. We further define these likelihoods to be symmetric and take scale considerations into account.

2.1.1 Intensity likelihood density

For two neighboring regions R_i and R_j, denote their average intensities by $\overline{I_i} \in \vec{\mathcal{H}}_i$ and $\overline{I_j} \in \vec{\mathcal{H}}_j$, we model both likelihoods $L_{ij}^{\pm}$ for the case of intensity in (2) as zero mean Gaussian density functions of their average intensity difference $\Delta_{ij} = \overline{I_i} - \overline{I_j}$, i.e.,

$$L_{ij}^{\pm} = p(\Delta_{ij}|\mathbf{s}_{ij}^{\pm}) = \mathcal{N}(0, \sigma_{ij}^{\pm}), \tag{3}$$

where the standard deviations $\sigma_{ij}^{\pm}$ are given as sums of two terms:

$$\sigma_{ij}^{\pm} = \sigma_{local}^{\pm} + \sigma_{scale}. \tag{4}$$

To determine σ_{local}^{+}, we consider for region i its neighbor whose average intensity is most similar (and likewise for region j). Denote the minimal external difference by $\Delta_i^+ = \min_k |\Delta_{ik}|$, where k denotes the regions immediate neighbors, then

$$\sigma_{local}^{+} = \min(\Delta_i^+, \Delta_j^+). \tag{5}$$

To determine σ_{local}^{-}, we take into account for region i, and similarly for region j, the average intensity difference over all of its neighbors, Δ_i^-, i.e.,

$$\Delta_i^- = \frac{\sum_k(\tau_{ik}\Delta_{ik})}{\sum_k(\tau_{ik})}, \tag{6}$$

where τ_{ik} denotes the length of the common boundaries between R_i and each of its neighbors R_k (see Section 3.2). Then we define

$$\sigma_{local}^{-} = \frac{\Delta_i^- + \Delta_j^-}{2}. \tag{7}$$

We further increase the standard deviation of each of the likelihoods by σ_{scale}. Suppose the image contains additive zero mean Gaussian noise with known standard deviation σ_{noise}. As we consider larger regions the effect of the noise on the average intensity of the regions shrinks. In particular, for a region R_i containing Ω_i pixels the standard deviation of the noise added to the average intensity is approximately

$$\sigma_{noise}^{R_i} = \frac{\sigma_{noise}}{\sqrt{\Omega_i}}. \tag{8}$$

Hence we choose

$$\sigma_{scale} = \frac{\sigma_{noise}}{\min(\sqrt{\Omega_i}, \sqrt{\Omega_j})}. \tag{9}$$

σ_{noise} can be estimated in a number of ways ([7]), e.g., by taking the minimal standard deviation across random image patches. Throughout our experiments, however, we used a constant value.

2.1.2 Texture likelihood densities

To account for texture we apply to each region R_i a bank of edge filters and store their total absolute responses in a histogram $\mathbf{h}_i \in \vec{\mathcal{H}}_i$ containing $\nu = |\mathbf{h}|$ bins (the filters we use are specified in Section 3.2). To measure the difference between two histograms $\mathbf{h}_i$ and $\mathbf{h}_j$ we use a measure similar to the χ^2 difference test [8]:

$$D_{ij} = \sum_k \left(\frac{\mathbf{h}_i(k) - \mathbf{h}_j(k)}{\mathbf{h}_i(k) + \mathbf{h}_j(k)}\right)^2. \tag{10}$$

Assuming that each response is distributed normally $\mathbf{h}_i(k) \sim \mathcal{N}(\mu_k, \sigma_k)$ we construct two new χ_ν^2 variables (ν denotes the number of degrees of freedom), which are expressed as products of the form $\alpha^+ D_{ij}$ and $\alpha^- D_{ij}$ as follows. We use again the concept that two regions with similar texture are more likely to be in the same segment. Recall, that the χ_ν^2 distribution receives its maximum at $\nu - 2$. Let $D_i^+ = \min_k D_{ik}$ we model L_{ij} in (2) by

$$L_{ij}^{\pm} = p(D_{ij}|\mathbf{s}_{ij}^{\pm}) = \chi^2(D_{ij}\alpha^{\pm}), \tag{11}$$

where $\alpha^+ = \frac{\nu-2}{min(D_i^+, D_j^+)}$ guaranties that the closest region in terms of texture will receive the highest likelihood. Similarly, we set α^- to reflect the difference in texture relative to the entire neighborhood. We therefore compute the average texture difference in the neighborhood, weighted by the length of the common boundaries between the regions

$$D_i^- = \frac{\sum_k(\tau_{ik}D_{ik})}{\sum_k(\tau_{ik})}, \tag{12}$$

and set $\alpha^- = \frac{\nu-2}{\frac{1}{2}(D_i^- + D_j^-)}$.

2.2. Prior

We determine the prior $P(\mathbf{s}_{ij}^{\pm})$ according to the geometry of the regions. Roughly speaking, a-priori we consider neighboring regions with long common boundaries more likely to belong to the same segment than regions with short common boundaries. Hence, we define the prior as:

$$P(\mathbf{s}_{ij}^{+}) = \frac{\tau_{ij}}{\min(\sum_k \tau_{ik}, \sum_k \tau_{jk})}. \tag{13}$$

2.3. Cue integration

As we mentioned in the beginning of Section 2 we integrate segmentation decisions from different cues using a local "mixture of experts"-like model. This model allows us to control the influence of each cue and adapt it to the information contained in each region. Thus, for example, when we compare two textured regions we can discount the effect of intensity and by this overcome brightness variations due to lighting.

To determine the relative influence of every cue we need to estimate $P(c_k|\vec{\mathcal{H}}_i, \vec{\mathcal{H}}_j)$. To that end we want to evaluate for each region whether or not it is characterized by texture. For each region R_i we calculate a 256-bin histogram of local gradients magnitudes G^i inside the region. Since, textured regions are often characterized by significant edge responses in different orientations and scales [9], we expect the gradients magnitude histogram of a non-textured region to be fairly sparse. To measure sparseness we first normalize the histogram ($\sum_k G^i_k = 1$) and apply to each region the measure [6]:

$$S_i = \frac{1}{\sqrt{n}-1}\left(\sqrt{n} - \frac{\|G^i\|_1}{\|G^i\|_2}\right), \tag{14}$$

where n denotes the number of bins in G^i and $\|G^i\|_p$ denotes the ℓ_p norm of G^i. Note that we exclude from this calculation pixels which lie along the boundary of a region since they may reflect boundary gradients rather than texture gradients. Finally, we combine these measures by

$$p(c_2|\vec{\mathcal{H}}_i, \vec{\mathcal{H}}_j) = \min(P(c_2|\vec{\mathcal{H}}_i), P(c_2|\vec{\mathcal{H}}_j)), \tag{15}$$

with c_2 denotes the intensity cue. We further model the individual probabilities using the logistic function:

$$p(c_2|\vec{\mathcal{H}}_i) = \frac{1}{(1 - e^{-(aS_i+b)})}. \tag{16}$$

To estimate the constant parameters a, b we used 950 random patches form the Brodatz data set [2] and a similar number of non-textured patches selected manually from random images as a training set. A sample from this set is shown in Figure 2. Then, a maximum likelihood estimation (MLE) regression was used to estimate a, b, and these parameters were used throughout all our experiments.

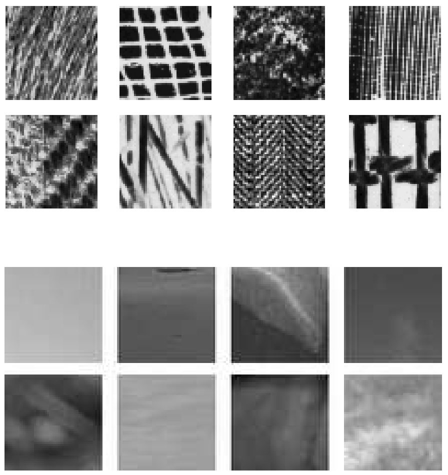
Figure 2. Samples from the training set used to determine the logistic function (16). Top: texture samples. Bottom: intensity samples

3. Algorithm

Our probabilistic framework is designed to work with any merge algorithm for segmentation. Here we use the merge strategy suggested for the Segmentation by Weighted Aggregation (SWA) algorithm [16, 5], which employs a hierarchy construction procedure inspired by Algebraic Multigrid (AMG) solutions for differential equations [1]. The SWA algorithm begins with a weighted graph representing image pixels, and in a sequence of steps creates a hierarchy of smaller ("coarse") graphs with soft relations between nodes at subsequent levels. The edge weights in the new graphs are determined by inheritance from previous levels and are modified based on regional properties. These properties are computed recursively as the merge process proceeds. Below we use the coarsening strategy of the SWA algorithm and modify it to incorporate our probabilistic framework. In particular, we use as edge weights the posterior probabilities defined in Section 2. We produce the coarser graphs using the coarsening strategy of SWA, but replace inheritance of weights by computing new posteriors. Overall, we achieve a method that is as efficient as the SWA algorithm, but relies on different, probabilistic measures to determine segmentation and requires almost no user tuned parameters.

3.1. Graph coarsening

Given an image we begin by constructing a 4-connected graph $G^{[0]} = (V^{[0]}, E^{[0]})$, in which every pixel is represented by a node and neighboring pixels are connected by an edge. Using the formulation described in Section 2, we associate a weight p_{ij} with each edge e_{ij},

$$p_{ij} = P(\mathbf{s}^+_{ij}|\vec{\mathcal{H}}_i, \vec{\mathcal{H}}_j), \tag{17}$$

utilizing a uniform prior at this first stage.

We then execute repeatedly the following steps in order to progressively construct smaller graphs, $G^{[1]}, G^{[2]}, ...$, each contains about half the number of nodes in the preceding graph:

Coarse node selection: Given a graph $G^{[s-1]} = (V^{[s-1]}, E^{[s-1]})$ we begin the construction of $G^{[s]}$ by selecting a set of seed nodes $C \subset V^{[s-1]}$, which will constitute the subsequent level. Let us denote the unselected nodes by $F = V^{[s-1]} - C$. Then, the selection of the seeds is guided by the principle that each F-node should be "strongly coupled" to nodes in C, i.e., for each node $i \in F$ we require that

$$\frac{\sum_{j \in C} p_{ij}}{\sum_{j \in V^{[s-1]}} p_{ij}} > \psi, \tag{18}$$

where ψ is a parameter (usually, $\psi = 0.2$). The construction of C is done using a sequential scan of the nodes in $V^{[s-1]}$, adding to C every node that does not satisfy (18) with respect to the nodes already in C. The scanning order may be determined according to a certain desired property of the regions, e.g., by decreasing size of the nodes, influencing C to contain larger regions.

Once C is selected we construct $V^{[s]}$ to include copies of the nodes in C. To simplify notations we assume without loss of generality that the nodes $1, 2, ..., |C| \in V^{[s-1]}$ compose C, while the rest are in F. This allows us to assign the same index to nodes in $V^{[s]}$.

Inter-level interpolation: We determine the inter-level interpolation weights as follows. For each node $i \in F$ we denote by $N_i = \{j \in C \mid p_{ij} > 0\}$ its "coarse neighborhood." We define a matrix $T^{[s-1][s]}$ of size $|V^{[s-1]}| \times |C|$ by:

$$t_{ij} = \begin{cases} p_{ij} / \sum_{k \in N_i} p_{ik} & \text{for } i \in F, j \in N_i \\ 1 & \text{for } i \in C, j = i \\ 0 & otherwise. \end{cases} \tag{19}$$

Computing regional properties: For each coarse node $i \in V^{[s]}$ we compute intensity and texture properties by averaging over the properties of its descendants. These are stored in a feature vector $\vec{\mathcal{H}}_i^{[s]}$. We further elaborate on the computation of regional properties in Section 3.2.

Coarse graph probabilities: Finally, the edge weights of the coarse graph are determined. Unlike the SWA, we do not inherit those weights from the previous level. Instead we compute new posteriors for the nodes of the coarse graph. For every pair of neighboring nodes, $i, j \in V^{[s]}$ we assign the weight

$$p_{ij}^{[s]} = P(\mathbf{s}_{ij}^{+} | \vec{\mathcal{H}}_i^{[s]}, \vec{\mathcal{H}}_j^{[s]}). \tag{20}$$

These posteriors are determined, as is described in Section 2, using the newly computed regional properties.

3.2. Features

In order to determine the edge weights at every level we need to compute posterior probabilities as in Section 2. The computation of these posteriors uses the average intensity and histogram of filter responses computed for every region, as well as the length of boundaries between every two neighboring regions. The merge strategy described above enables us to compute these properties efficiently for every node, by averaging the same properties computed for its descendants. The properties we are using can be divided into two kinds: unary features, computed for a single region, e.g., the average intensity or histogram of filter responses, and binary features, e.g., the length of the common boundary between two regions. Below we describe how we compute these properties during the coarsening process.

3.2.1 Unary features

Our intensity and texture features can be obtained by summation of the corresponding feature values over all pixels in a region. For every node k at scale s we can compute such a feature by taking a weighted sum of the feature values of its descendants. Specifically, for a pixel i we denote its feature value by q_i. Denote by $T_{ik}^{[s]}$ the extent to which pixel i belongs to the region k at scale s, $T_{ik}^{[s]}$ can be determined from the matrix product $T^{[s]} = \prod_{m=0}^{s-1} T^{[m][m+1]}$. We further denote by $\bar{Q}^{[s]}$ the weighted average of q_i for all pixels i which belong to region k i.e.,

$$\bar{Q}_k^{[s]} = \frac{\sum_i t_{ik}^{[s]} q_i}{\sum_i t_{ik}^{[s]}}. \tag{21}$$

Then, $\bar{Q}_k^{[s]}$ can be computed using the following recursive formula:

$$\bar{Q}_k^{[s]} = \frac{\sum_j t_{jk} \Omega_j^{[s-1]} \bar{Q}_j^{[s-1]}}{\sum_j t_{jk} \Omega_j^{[s-1]}}, \tag{22}$$

where $\Omega_j^{[s-1]}$ denote the size of aggregate j at scale $s-1$, which is computed recursively in a similar way, and t_{jk} is the element jk in the matrix $T^{[s-1][s]}$.

We use this recursive formulation to compute the following features:

Average intensity: Starting with the intensity value I_i at each pixel i at scale 0, the quantity $\bar{I}_k^{[s]}$ provides the average intensity in a region k at scale s.

Texture: For each pixel, we measure short Sobel-like filter responses, following [5], in four orientations $0, \frac{\pi}{2}, \frac{\pi}{4}, \frac{3\pi}{4}$ and accumulate them recursively to obtain a 4-bin histogram for each region at each scale. Since filter responses at points near the boundaries of a segment may respond strongly to the boundaries, rather than to the texture at the region we employ a top-down cleaning process to eliminate these responses from the histogram.

3.2.2 Binary features

To determine the prior probability $P(\mathbf{s}_{ij}^{\pm})$ we need to compute for every pair of neighboring regions the length of their common boundaries. Beginning at the level of pixels, we initialize the common boundaries τ_{ij} of two neighboring pixels to 1 (we use 4-connected pixels) and 0 otherwise. Then, for every neighboring regions k and l at scale s we compute the length of their common boundaries using the formula:

$$\tau_{k,l}^{[s]} = \sum_{ij} \tau_{ij}^{[s-2]}, \tag{23}$$

where the indices i and j sum respectively over all the *maximal decedents* of k and l of level $s-2$; i.e. i and j are aggregates of level $s-2$ that respectively belong to k and l with largest interpolation weights relative to all other nodes of scale s. Again, this property can be accumulated recursively from one level to the next.

4. Experiments

Evaluating the results produced by segmentation algorithms is challenging, as it is difficult to come up with canonical test sets providing ground truth segmentations. This is partly because manual delineation of segments in everyday complex images can be laborious. Furthermore, people often tend to incorporate into their segmentations semantic considerations which are beyond the scope of data driven segmentation algorithms. For this reason many existing algorithms show only few segmentation results. An important attempt to produce an extensive evaluation database for segmentation was recently done at Berkeley [10]. This database however has its own limitations, as can be noticed by the differences between subjects. In many cases images are under-segmented, and semantic considerations seem to dominate the annotated segmentations.

To evaluate our method and compare it to recent algorithms we have compiled a database containing 100 gray level images along with ground truth segmentations. The database was designed to contain a variety of images with objects that differ from their surroundings by either intensity, texture, or other low level cues. To avoid potential ambiguities we only selected images that clearly depict one object in the foreground. To obtain ground truth segmentation we asked about 50 subjects to manually segment the images into two classes, foreground and background, with each image segmented by three different human subjects. We further declared a pixel as foreground if it was marked as foreground by at least two subjects. A sample from the database is shown in Figure 3. The complete database and the segmentation results are available in the supplementary material.

We evaluated segmentation results by assessing its consistency with the ground truth segmentation and its amount of fragmentation. For consistency we used the *F-measure* [18]. Denote by P and R the precision and recall values of a particular segmentation than the F-measure is

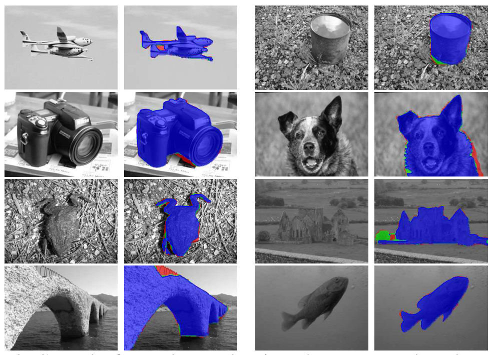
Figure 3. Sample from the evaluation dataset. Each color represents a different amount of votes given by the human subject according to the following key: blue=3, green=2 red=1.

defined as

$$F = \frac{2RP}{P+R}. \tag{24}$$

The amount of fragmentation is given simply by the number of segments needed to cover the foreground object.

We applied our segmentation algorithm to all 100 images in the database and compared our results with several state of the art algorithms including:

1. Segmentation by weighted aggregation (SWA)[16]. We tested two variants, one which uses the full range of features described in [5] (denoted by SWA V1) and a second variant which relies only on features similar to the ones used by our method, i.e., intensity contrast and filter responses (denoted by SWA V2) (WINDOWS implementation at www.cs.weizmann.ac.il/∼vision/SWA/).

2. Normalized cuts segmentation including intervening Contours [8] (Matlab implementation at www.cis.upenn.edu/∼jshi/).

3. Mean-Shift [3]. This method uses intensity cues only (EDISON implementation at www.caip.rutgers.edu).

For our method only a single parameter, σ_{noise} needed to be specified. We set this parameter to a fixed value for all images ($\sigma_{noise} = 18$). The other algorithms were run with several sets of parameters. The normalized cuts algorithm was run with the requested number of segments between $2-10$. For the Mean-Shift and SWA we tested roughly 40 different sets of paraments. In each case we selected for the final score the set of parameters that gave the best performance for the entire database.

We performed two tests. In the first test we selected in each run the segment who fits the best the foreground, according to the F-measure score. The results are given in Table 1. Our method outperforms other methods, demonstrating the highest averaged F-measure score. The next

Algorithm	F-measure Score
Our Method	0.86 ± 0.012
SWA V1	0.83 ± 0.016
SWA V2	0.76 ± 0.018
N-Cuts	0.72 ± 0.018
MeanShift	0.57 ± 0.023

Table 1. One segment coverage test results

Algorithm	Averaged F-measure Score	Average number of fragments
Our Method	0.87 ± 0.017	2.66 ± 0.30
SWA V1	0.89 ± 0.013	3.92 ± 0.35
SWA V2	0.86 ± 0.012	3.71 ± 0.33
N-Cuts	0.84 ± 0.013	3.12 ± 0.17
MeanShift	0.88 ± 0.011	12.08 ± 0.96

Table 2. Fragment coverage test results

best score is achieved by the SWA algorithm utilizing its full set of features. Note that the performance of the mean shift algorithm suffers since this implementation does not handle texture. In the second test, we have permitted a few segments to cover the foreground by combining segments that largely overlap with the foreground object. Then on each union, we have measured the F-measure score and the number of segments comprising it. The results are given in Table 2. The averaged F-measure of the different algorithms is fairly similar. Yet, our method needs the least number of fragments to cover the foreground. Figure 4 shows a collection of test images along with segmentation results.

5. Summary

We have presented a parameter-free approach to image segmentation. Our approach uses a bottom-up aggregation procedure in which regions are merged based on probabilistic considerations. The framework utilizes adaptive parametric distributions whose parameters are estimated locally using image information. Segmentation relies on an integration of intensity and texture cues, with priors determined by the geometry of the regions. We further applied the method to a large database with manually segmented images and compared its performance to several recent algorithms obtaining favorable results.

References

[1] A. Brandt. Algebraic multigrid theory: The symmetric case. *Applied Mathematics and Computation*, 19(1-4):23–56, 1986.

[2] P. Brodatz. *"Textures: A Photographic Album for Artists and Designers"*. Dover Publications, New York, NY ,USA, 1966.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *TPAMI*, 24(5):603–619, 2002.

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.

[5] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. *ICCV*, pages 716–723, 2003.

[6] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[7] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang. Noise estimation from a single image. *CVPR (1)*, pages 901–908, 2006.

[8] J. Malik, S. Belongie, T. K. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[9] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A*, 7(5), 1990.

[10] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *ICCV (2)*, pages 416–423, July 2001.

[11] D. R. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *TPAMI*, 26(5):530–549, 2004.

[12] H. T. Nguyen and Q. Ji. Improved watershed segmentation using water diffusion and local shape priors. *CVPR (1)*, pages 985–992, 2006.

[13] D. K. Panjwani and G. Healey. Markov random field models for unsupervised segmentation of textured color images. *TPAMI*, 17(10):939–954, 1995.

[14] T. Pavlidis and Y.-T. Liow. Integrating region growing and edge detection. *TPAMI*, 12(3):225–233, 1990.

[15] A. Rabinovich, S. Belongie, T. Lange, and J. M. Buhmann. Model order selection and cue combination for image segmentation. *CVPR (1)*, pages 1130–1137, 2006.

[16] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442(7104):810–813, June 2006.

[17] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888–905, 2000.

[18] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

[19] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *TPAMI*, 13(6):583–598, 1991.

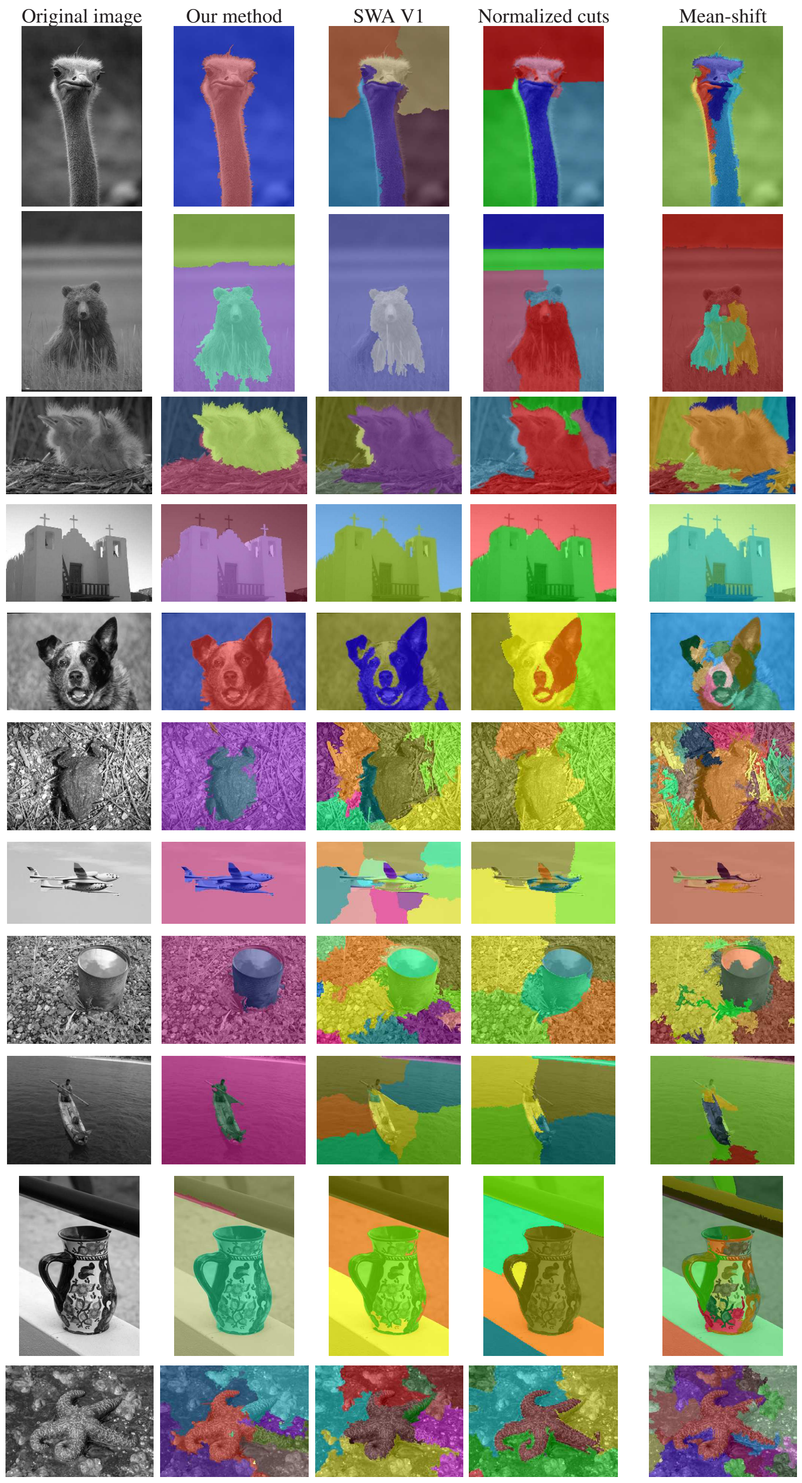

Figure 4. Results of applying our method compared to other state of the art segmentation algorithms. The top four images are taken from the Berkeley segmentation database [10] and the rest from our evaluation database.