Int J Comput Vis (2016) 117:131–141
DOI 10.1007/s11263-015-0855-4

Local Variation as a Statistical Hypothesis Test

Michael Baltaxe[1] · Peter Meer[2] · Michael Lindenbaum[3]

Received: 7 June 2014 / Accepted: 1 September 2015 / Published online: 16 September 2015

Abstract The goal of image oversegmentation is to divide an image into several pieces, each of which should ideally be part of an object. One of the simplest and yet most effective oversegmentation algorithms is known as *local variation* (LV) Felzenszwalb and Huttenlocher in Efficient graph-based image segmentation. IJCV 59(2):167–181 (2004). In this work, we study this algorithm and show that algorithms similar to LV can be devised by applying different statistical models and decisions, thus providing further theoretical justification and a well-founded explanation for the unexpected high performance of the LV approach. Some of these algorithms are based on statistics of natural images and on a hypothesis testing decision; we denote these algorithms *probabilistic local variation* (pLV). The best pLV algorithm, which relies on censored estimation, presents state-of-the-art results while keeping the same computational complexity of the LV algorithm.

Keywords Image segmentation · Image oversegmentation · Superpixels · Grouping

1 Introduction

Image segmentation is the procedure of partitioning an input image into several meaningful pieces or segments, each of which should be semantically complete (i.e., an item or structure by itself). Oversegmentation is a less demanding type of segmentation. The aim is to group several pixels in an image into a single unit called a superpixel (Ren and Malik 2003) so that it is fully contained within an object; it represents a fragment of a conceptually meaningful structure.

Oversegmentation is an attractive way to compact an image into a more succinct representation. Thus, it could be used as a preprocessing step to improve the performance of algorithms that deal with higher level computer vision tasks. For example, superpixels have been used for discovering the support of objects (Rosenfeld and Weinshall 2011), extracting 3D geometry (Hoiem et al. 2007), multiclass object segmentation (Gould et al. 2008), scene labeling (Farabet et al. 2013), objectness measurement in image windows (Alexe et al. 2012), scene classification (Juneja et al. 2013), floor plan reconstruction (Cabral and Furukawa 2014), object description (Delaitre et al. 2012), and egocentric video summarization (Lee and Grauman 2015).

One may ask whether specialized oversegmentation processes are needed at all, given that several excellent segmentation methods were recently proposed. Working in the high recall regime, these algorithms could yield excellent oversegmentation accuracy. Unfortunately, however, they are complex and therefore relatively slow. The segmentation approach described in Arbeláez et al. (2011), for example, combines the gPb edge detector and the oriented watershed

Communicated by Pedro F. Felzenszwalb.

This work was conducted while M. Baltaxe was with the Computer Science Department, Technion - Israel Institute of Technology.

✉ Michael Baltaxe
michael.baltaxe@orbotech.com

Peter Meer
meer@cronos.rutgers.edu

Michael Lindenbaum
mic@cs.technion.ac.il

[1] Orbotech Ltd., 8110101 Yavne, Israel

[2] Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854, USA

[3] Computer Science Department, Technion - Israel Institute of Technology, 32000 Haifa, Israel

transform to give very high precision and recall, but requires over 240 s per frame [as reported in Dollár and Zitnick (2015)]. The approach of Ren and Shakhnarovich (2013) uses a large number of classifiers and needs 30 s. Moreover, while the recently introduced edge detector (Dollár and Zitnick 2015) is both accurate and very fast (0.08 s per frame), it does not provide close edges and consistent segmentation. Using this edge detection for hierarchical multiscale segmentation (Arbeláez et al. 2014), indeed achieves state-of-the-art results in terms of precision and recall but requires processing time of 15 s per image. Thus, the time required for these methods is often too high for a preprocessing stage, and then specialized oversegmentation algorithms, running in one second or less, are preferred.

One of the many popular approaches to oversegmentation is the mean shift algorithm, which regards the image values as a set of random samples and finds the peaks of the associated PDF, thereby dividing the data into corresponding clusters (Comaniciu and Meer 2002). The watershed method is a morphological approach which interprets the gradient magnitude image as a topographical map. It finds the catchment basins and defines them as segments (Meyer 1994). Turbopixels is a level set approach, which evolves a set of curves so that they attach to edges in the image and eventually define the borders of superpixels (Levinshtein et al. 2009). The SLIC superpixel algorithm is based on k-means restricted search, and takes into account both spatial and color proximity (Achanta et al. 2012). The entropy rate superpixels (ERS) (Liu et al. 2011) approach partitions the image by optimizing an objective function that includes the entropy of a random walk on a graph representing the image, and a term balancing the segments' sizes. A thorough description of common oversegmentation methods can be found in Achanta et al. (2012).

The local variation (LV) algorithm by Felzenszwalb and Huttenlocher (2004) is a widely used, fast and accurate oversegmentation method. It uses a graph representation of the image to iteratively perform greedy merge decisions by evaluating the evidence for an edge between two segments. Although all the decisions are greedy and local, the authors showed that, in a sense, the final segmentation has desirable global properties. The decision criterion is partially heuristic and yet the algorithm provides accurate results. Moreover, it is efficient, with complexity $O(n \log n)$.

Throughout this paper we use the recall and undersegmentation error as measures of quality for oversegmentation, just as is done in Levinshtein et al. (2009), Achanta et al. (2012), and Liu et al. (2011). The recall (Martin et al. 2004) is the fraction of ground truth boundary pixels that are matched by the boundaries defined by the algorithm. The undersegmentation error (Levinshtein et al. 2009) quantifies the area of regions added to segments due to incorrect merges. Figure 1 presents a comparison of the aforementioned algorithms, together with probabilistic local variation (pLV), the method introduced in this paper.[1]

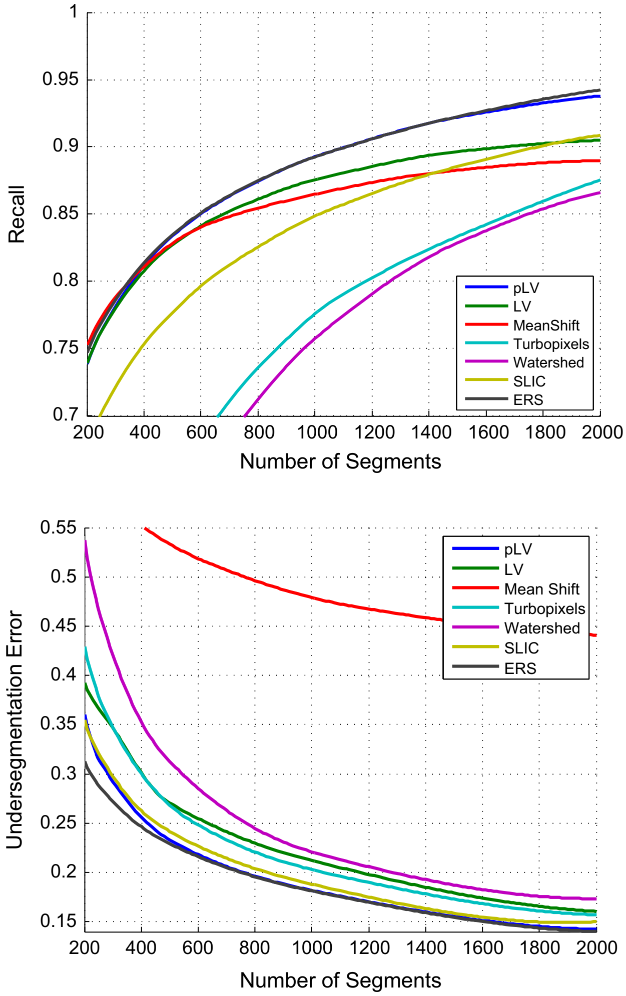

Fig. 1 Recall (*top*) and undersegmentation error (*bottom*) for probabilistic local variation and other common algorithms

Understanding the remarkably good performance of the greedy LV algorithm was the motivation for our work, which makes three contributions:

1. We briefly analyze the properties of the LV algorithm and show that the combination of all its ingredients is essential for high performance.
2. We examine several probabilistic models and show that LV-like algorithms could be derived by statistical consideration. One variation involves the estimation of the maximum sample drawn from a uniform distribution. The second uses natural image statistics and a hypothesis testing decision. This gives additional viewpoints, further justifies the original version of the algorithm, and

[1] Code available at http://cis.cs.technion.ac.il/index.php/projects/probabilistic-local-variation.

explains the excellent performance obtained from such a simple, fast method.

3. Following these models, we provide a new algorithm, which is similar to LV but relies on statistical arguments. This algorithm meets the best performance among the methods in the literature, at least for oversegmentation to many segments, but is faster.

The structure of the paper is as follows. We present the LV algorithm in Sect. 2. We study this algorithm empirically in Sect. 3. In Sect. 4 we introduce the first LV-like algorithm (based on maximum estimation), which does not perform as well as the original LV, but motivates the alternative and more successful algorithm of Sect. 5 (based on natural image statistics). Section 6 presents our experimental results. A brief discussion of single-linkage algorithms (which include the LV and pLV algorithms) as compared to non-single-linkage, hierarchical algorithms is given in Sect. 7. Finally, Sect. 8 concludes.

2 Local Variation Algorithm

The LV algorithm is a single-linkage, graph based, hierarchical clustering process. Let $G = (V, E)$ be a graph created from the input image, with vertices on the pixels and edges between each pair of neighboring pixels. Define a weight function on the edges, $w : E \rightarrow \Re$, representing pixel dissimilarity (for example, the RGB color distance between pixels). Likewise, let a component (segment) C_i be a set of connected pixels. The components change throughout the segmentation process, and, initially, the set of components $\{C_i\}$ is the set of pixels.

Following Felzenszwalb and Huttenlocher (2004), define the internal dissimilarity of component C_i, denoted by $Int(C_i)$, and a threshold function called the minimum internal difference and denoted $MInt(C_i, C_j)$, as:

$$Int(C_i) = \max_{e \in MST(Ci)} w(e) \tag{1}$$

$$MInt(C_i, C_j) = \min_{x \in \{i,j\}} \left(Int(C_x) + T(C_x)\right), \tag{2}$$

where $MST(C_i)$ is a minimum spanning tree of C_i, and $T(C_i) = K/|C_i|$ is a component dependent function, in which K is a user controlled parameter and $|C_i|$ denotes the number of vertices in component C_i.

The LV algorithm for image oversegmentation is presented in Algorithm 1. Intuitively, we can see that two components are merged only if the lightest edge that connects them is lighter than the heaviest edge in the MST of the components plus a margin. Since the edges are sorted in step 1, the edges causing merges are exactly those that would be selected by Kruskal's MST algorithm (Kruskal 1956). The parameter K controls the number of segments in the output segmentation: increasing K implies that more edges satisfy the merge condition and more merges are performed.

Algorithm 1 Local Variation Algorithm

Input: Weighted graph $G = (V, E)$ with weights $w(e)$, $e \in E$, defined by an image.
Output: Set of components $C_1, \ldots, C_n$ defining segments

```
1: Sort $E$ by non-decreasing edge weight $(e_1, e_2, \ldots, e_m)$
2: Initialize segmentation $S^0$ with each vertex being a component
3: for all $q = 1, \ldots, m$ do
4:     $e_q = (v_i, v_j) \leftarrow$ edge with the $q$th lightest weight
5:     $C_i^{q-1} \leftarrow$ component of $S^{q-1}$ containing $v_i$
6:     $C_j^{q-1} \leftarrow$ component of $S^{q-1}$ containing $v_j$
7:     if $\left(w(e_q) \leq MInt\left(C_i^{q-1}, C_j^{q-1}\right)\right) \wedge (C_i^{q-1} \neq C_j^{q-1})$ then
8:         $S^q = S^{q-1} \cup \left\{C_i^{q-1} \cup C_j^{q-1}\right\} \setminus \left\{C_i^{q-1}, C_j^{q-1}\right\}$
9:     else
10:        $S^q = S^{q-1}$
11:    end if
12: end for
13: Postprocessing: Merge all small segments to the neighbor with closest color.
```

Oversegmentation algorithms usually include a post-processing stage where small segments are removed (line 13 in Algorithm 1). We consider a segment as small when its size is 10 % of the average expected segment size, and merge it to its neighbor with the smallest color difference.

Compared to other oversegmentation algorithms, LV is among the best in terms of recall and running time. It is thus often the method of choice even though its undersegmentation error is not as small as that of some other algorithms; see Achanta et al. (2012) and Fig. 1.

The high accuracy obtained by the greedy LV algorithm is impressive. It is mainly due to the adaptive threshold $Int(C_i) + T(C_i)$, which depends on two components: the distribution of weights within the segments and their size. The particular combination of these two components meets the criterion used to decide whether a segmentation is too fine/coarse in Felzenszwalb and Huttenlocher (2004), but is not theoretically supported otherwise. In this paper we analyze the LV algorithm empirically and propose two statistical interpretations that lead, eventually, to LV-like algorithms, that follow a statistical decision procedure. One of these new algorithms, denoted as probabilistic local variation (or more specifically, pLV-ML-Cen), maintains LV's desirable properties, and improves its recall and undersegmentation error.

3 Empirical Study of the LV Algorithm

To analyze the importance of each of the two aforementioned components in LV, we test reduced versions of the algorithm by systematically removing each component:

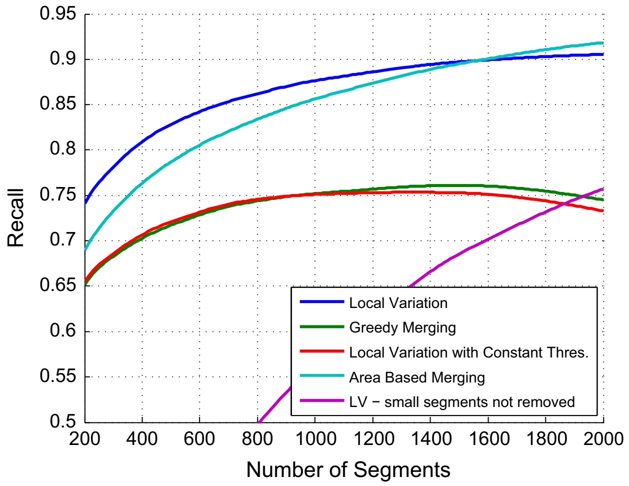

Fig. 2 The LV method against its reduced versions. No reduced version could achieve the performance of the original algorithm

1. *Greedy Merging:* By setting $MInt(C_i, C_j) = \infty$ the algorithm depends neither on the distribution nor on the segment sizes. This implies that the segments are merged greedily in non-decreasing edge weight.
2. *LV with a Constant Threshold:* By setting $T(C_i) = K$, where K is a constant, makes the decision size independent (but distribution dependent). Performance is substantially reduced and very large segments are created.
3. *Area Based Merging:* The condition in line 7 of Algorithm 1 is replaced by $\min_{x\in\{i,j\}} \left|C_x^{q-1}\right| < K$, where K is a constant. This condition depends only on the segment size but not on the distribution, yielding superpixels with roughly the same area.
4. *LV without Removing Small Segments:* A post-processing step in LV removes small components (line 13 in Algorithm 1). Without this step, a lot of very small, meaningless segments remain, implying that many erroneous merges are performed to obtain a prespecified number of segments.

Figure 2 compares the recall of LV and its reduced versions. All reduced versions yield lower recall than the original algorithm. Thus we conclude that both the distribution and size dependent terms are crucial to LV's performance.

4 Interpreting LV as Maximum Estimation

4.1 Estimating the Maximum Sample Drawn From a Uniform Distribution

The threshold used in LV has several possible statistical interpretations. In this section we consider one interpretation, which suggests that LV's decision rule is similar to maximum value estimation.

Consider a set of samples drawn from a uniform distribution specified by an unknown interval $[min_U, max_U]$. We want to estimate the parameters of the uniform distribution, min_U, max_U, from the samples.[2] For the special case where $min_U = 0$, let m be the sample maximum and S be the set size. Then, the minimum variance unbiased estimator for the maximum value is given by (Larsen and Marx 2012)

$$\hat{max}_U = m + \frac{m}{S}. \tag{3}$$

4.2 Interpreting LV as Maximum Estimation

The estimate (3) seems similar to the threshold expression used in the local variation algorithm [Algorithm 1 and Felzenszwalb and Huttenlocher (2004)]. Both expressions contain two terms. The first is a distribution related term, m, which is the maximal observed value. The second term in both cases is size dependent. The two expressions differ in that, in expresion (3), the size dependent term depends also on the maximal observed value, m. Thus, we hypothesize that an LV-like process would be obtained by considering the weight values in each segment, estimating their maximum under the uniform distribution assumption, and testing whether a new weight falls below the estimated maximum and therefore belongs to the same distribution. If the weight satisfies this test for the two segments, then merging them is justified.

Thus the algorithm is exactly like Algorithm 1 (LV), except that $MInt(C_i, C_j)$ is replaced by:

$$MInt(C_i, C_j) = \min_{x\in\{i,j\}} \left(Int(C_x) + \frac{Int(C_x)}{|C_x|} + c \right). \tag{4}$$

In theory the constant c should be 0. However, due to image quantization, the maximum observed value in a small segment is often 0, which prevents any further merges. Note also that the difference between two continuous grey levels may be almost 1, and yet, the difference between their quantized values is zero. To allow for these errors, we set $c = 1$. Otherwise, this LV-like process (4), denoted as LV-MaxEst, is parameterless.

Figure 5 presents an example. Note that the overall segmentation still does not seem natural: the segments on the bird look natural, but those in the sky do not.

[2] This is the continuous version of the problem known in the statistical theory literature as the German tank problem (Ruggles and Brodie 1947), because its solution was used by the Allies in WW2 to estimate the number of tanks produced by the Germans from the serial numbers of captured tanks.

4.3 A Controllable Segmentation Based on Maximum Estimation

Being parameterless, the decision rule based on the maximum estimation does not allow us to control the oversegmentation level. A simple, controllable extension would be to change $MInt(C_i, C_j)$ to:

$$MInt(C_i, C_j) = \min_{x \in \{i,j\}} \left(Int(C_x) + \frac{k \cdot Int(C_x)}{|C_x|} + c \right), \tag{5}$$

where the constant k controls the number of superpixels. We denote this method LV-MaxEst-c (where the suffix "c" stands for "controlled").

Interpreting LV as a maximum estimation problem sheds some light on the algorithm; however, some of the assumptions used in this section are not accurate and the decision rule in Eq. (5) actually does not perform as well as the original LV algorithm (see Sect. 6). This motivates the alternative model presented next.

5 Interpreting LV as Hypothesis Testing

The statistical interpretation in Sect. 4 is closely related to the original LV formulation but the assumption of a uniform distribution for edge weights seems unjustified. In this section, we present an alternative statistical model associated with natural image statistics.

5.1 Natural Image Statistics

Natural image statistics have been intensively explored over the last two decades. Statistical models consider specific image descriptors such as wavelet coefficients (De Bonet and Viola 1997) or intensity difference between adjacent pixels (Grenander and Srivastava 2001) and characterize them statistically. A common way to model the behavior of these image descriptors is by means of the generalized Gaussian distribution (Mallat 1989),

$$P(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(x/\alpha)^{\beta}}, \tag{6}$$

where, typically, α varies depending on the descriptor used and β falls in the range $[0.5, 0.8]$ (Srivastava et al. 2003). This model has been successfully used for, e.g., image denoising (Moulin and Liu 2006) and image segmentation (Heiler and Schnörr 2005).

We are interested in the weights of the graph edges, which are either the absolute differences between LUV color vectors or simply intensity differences. The intensity differences are closely related to some wavelet coefficient and to gradient strengths, both of which were modeled with the generalized Gaussian distribution (Mallat 1989; Huang and Mumford 1999).

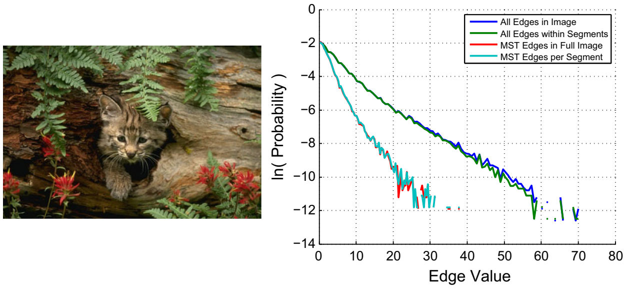

Fig. 3 A natural image and semilog plot with the edge weight histogram of several edge sets. See text for details

The population we consider is somewhat different, however. We are interested only in weights that are part of the MST and are inside segments (and not between them). We checked the validity of the exponential model, a particular case of the generalized Gaussian distribution with $\beta = 1$, on several images and found that the exponential assumption is reasonable; see Fig. 3, which shows one image example and 4 plots of the LUV edge weight statistics:

1. Weights of all edges in the image.
2. Weights of all edges within image segments, as marked by a human [as given in BSDS300 (Martin et al. 2001)].
3. Weights of edges in an MST of the full image.
4. Weights of edges in an MST of each segment, as marked by a human.

As expected, the distributions of the edge weights in the MSTs are biased towards lower weights compared to the distributions obtained from the edges in the complete image. On the other hand, the edge weight distributions within the segments and throughout the entire image are similar. This is also expected because the number of edges crossing segments is relatively small. Clearly, all distributions are close to an exponential law (observed as a straight line in the semilog plot). In this work, we shall therefore assume that the model underlying the distribution of edge weights is the exponential distribution:

$$P(x) = \lambda e^{-\lambda x}. \tag{7}$$

5.2 Merge Decisions from Hypothesis Testing

Consider the local merging context where we need to decide whether two segments S_i, S_j merge along edge e.

We propose to make this decision by testing the hypothesis that the weight $w(e)$ of the edge e belongs to the distribution $P_i(x) = \lambda_i e^{-\lambda_i x}$ of the weights in each one of the segments. If this hypothesis is rejected for at least one of the segments, the segments are not merged. Otherwise the segments are merged. We refer to this method as *probabilistic local variation (pLV)*.

To test the hypothesis that $w(e)$ belongs to S_i, we consider the probability:

$$P_i(x > w(e)) = \int_{w(e)}^{\infty} \lambda_i e^{-\lambda_i x} dx = e^{-\lambda_i w(e)}. \tag{8}$$

The hypothesis is rejected with a level of significance δ whenever $P_i(x > w(e)) < \delta$.

Thus, the probabilistic local variation approach uses the following alternative rule for deciding whether two segments should be merged:

1. Let e^* be the edge with the minimum weight connecting two segments, S_i, S_j.
2. For each segment $S_a \in \{S_i, S_j\}$, fit an exponential distribution, $P_a(x)$, to the weights in it.
3. For each segment $S_a \in \{S_i, S_j\}$, test the hypothesis that e^* belongs to the corresponding distribution using the hypothesis test, $P_a(x > w(e^*)) < \delta$.
4. If the hypothesis is rejected in at least one of the tests, do not merge. Otherwise merge.

In the rest of this section we describe several ways to estimate the parameters λ_i and the implied distribution.

A straightforward estimate for λ would be the maximum likelihood (ML) estimator. Given a population of n samples, $\{x_1, x_2, \ldots, x_n\}$, drawn i.i.d. from the exponential distribution (7), then the maximum likelihood estimator for the parameter, $\hat{\lambda}_{ML}$, is

$$\hat{\lambda}_{ML} = \frac{1}{\bar{x}} = \frac{n}{\sum_{i=1}^{n} x_i}. \tag{9}$$

When using this estimator, we will denote the probabilistic local variation as pLV-ML.

The ML estimator is, however, noise prone and highly unstable when the sample size is small, as is the case when the merged segments are small. One way to make a robust decision is by using confidence intervals (CI) for $\hat{\lambda}_{ML}$. The symmetric $100(1-\alpha)$ % CI for $\hat{\lambda}_{ML}$, for a population of n samples drawn from the exponential distribution, is given by:

$$\left(\frac{\chi^2_{1-\alpha/2,2n}}{2n} \hat{\lambda}_{ML}, \frac{\chi^2_{\alpha/2,2n}}{2n} \hat{\lambda}_{ML} \right), \tag{10}$$

where $\chi^2_{p,\nu}$ is a value specifying the tail of weight p in a χ^2 distribution with ν degrees of freedom (see section 7.6 of Ross (2009) for details regarding parameter estimation for the exponential distribution). $\chi^2_{p,\nu}$ increases approximately linearly with $\nu = 2n$, and for large n, converges to $\nu = 2n$, implying that the confidence interval decreases with the segment size, as intuitively expected.

The question of which value should be chosen within the CI remains. The effect of choosing a particular λ value is not straightforward because changes in λ influence other parameters. Suppose we are interested in a final segmentation with N superpixels. Choosing λ, say, at the lower limit of the confidence interval, must be compensated for by increasing δ. Otherwise, the threshold will be higher and too many merges will be performed. As explained in Sect. 5.3, the edge weights used for estimating λ are biased towards smaller values, which makes $\hat{\lambda}_{ML}$ biased to larger values. Therefore, we prefer the lower limit of the CI (10), and reject a merge whenever $P(x > w(e)) = e^{-w(e)\hat{\lambda}_{ML}\frac{\chi^2_{1-\alpha/2,2n}}{2n}} < \delta$, or equivalently, whenever

$$w(e) > \ln(1/\delta) \Big/ \left(\hat{\lambda}_{ML} \frac{\chi^2_{1-\alpha/2,2n}}{2n} \right). \tag{11}$$

Replacing $\hat{\lambda}_{ML}$ with the lower limit of the CI for λ (method denoted as pLV-ML-CI) indeed yields better recall, very similar to that of LV; see Fig. 6 (top). The role of the confidence parameter δ is analogous to that of K in the original LV algorithm: making δ larger results in more segments.

Note also that qualitatively, the algorithm behaves according to the LV principle and gives priority to the merging of smaller superpixels. This is not immediately clear, however, because, as discussed above, choosing λ at the lower limit of CI is compensated for by increasing δ. This compensation, however, is non-uniform. While the level of significance δ is uniform and applies to all merging decisions, smaller segments are associated with larger confidence intervals, which makes the lower limit of their confidence intervals lower on average. Therefore, the threshold on the edge weight (11) is higher for smaller superpixels, which gives them priority to merge.

While this model achieves nice segmentations and sheds some new light on the LV algorithm, we observe that the statistical assumption underlying the estimation of λ—that the measurements are obtained by i.i.d. sampling—seems inaccurate. First, the samples we have are sampled in nearby locations, which makes them correlated. Moreover, they are sampled with preference for lower weight edges. The first problem seems to be minor because the weights are derivative values, which are less correlated than the image intensities. The second problem, preferring smaller values, is considered in the next subsection.

5.3 Parameter Estimation Under Biased Sampling

Contrary to the assumption used in Sect. 5.2, the weights are sampled from the lightest to the heaviest and are not drawn i.i.d. Thus, at each step the population inside a segment is biased towards low values. In what follows we aim to correct this bias.

Suppose the segment has $m + 1$ pixels, and at some point in time our method of sampling provides us with n measurements, which are the lowest n elements in the MST of the segment, which contains m edges. Our task is to estimate the value of λ. The structure of this problem is an approximation to the one defined by *type II censoring*, in which m random variables are drawn i.i.d. but only the smallest $n < m$ values are observed (an approximation since the true segment may be split between several segments specified by the algorithm). This type of sampling is common in reliability estimation, where one tries to study the failure rate of some process/machine by analyzing only a subset composed of the samples which failed first (Epstein and Sobel 1953). Under these conditions, and following Epstein and Sobel, the maximum likelihood for λ can be derived as follows:

A sequence of size m is called partially n-ordered if the first n elements in it are non-decreasing (or non-increasing), and not larger (or not smaller) than any of the remaining $m-n$ elements. Consider a sequence, Y, of m samples drawn i.i.d. from some probability distribution. A partially n-ordered sequence, X, may be generated from it by choosing the n smallest elements from Y, and making them the first n elements of X, in non-decreasing order. The remaining $m - n$ elements in X are identical to the remaining $m-n$ elements in Y. Their order is the same order as in Y.

Note that many sequences Y may correspond to the same partially n-ordered X. The first n elements may come from $H = \frac{m!}{(m-n)!}$ different sets of locations, implying that the probability of getting a particular partially n-ordered sequence is H times the probability of getting the original sequence Y of i.i.d. drawn samples. Therefore, the joint probability density of observing the values $x_1, \ldots, x_n$ is

$$\begin{aligned} f(x_1,\ldots,x_n) &= HP(x_1,\ldots,x_n)P(x_{n+1} > x_n,\\ &\quad \ldots, x_m > x_n)\\ &= H\cdot P(x_1)\cdots P(x_n)\cdot P(x_{n+1} > x_n)\\ &\quad \cdots P(x_m > x_n)\\ &= H\cdot P(x_1)\cdots P(x_n)\cdot\left(\int_{x_n}^{\infty}\lambda e^{-\lambda x}dx\right)^{m-n}\\ &= H\cdot \lambda e^{-\lambda x_1}\cdots\lambda e^{-\lambda x_n}\cdot\left(e^{-\lambda x_n}\right)^{m-n}\\ &= H\lambda^n\cdot e^{-\lambda\sum_{i=1}^n x_i}\cdot e^{-\lambda(m-n)x_n}\\ &= H\lambda^n\cdot e^{-\lambda\left(\sum_{i=1}^n x_i+(m-n)x_n\right)}. \end{aligned} \quad (12)$$

The value of λ yielding the maximum likelihood for this function can be found by differentiating $\ln f(x_1, \ldots, x_n)$ with respect to λ and equating to zero, yielding

$$\hat{\lambda}_{ML-Cen} = \frac{n}{\sum_{i=1}^n x_i + (m-n)x_n}, \quad (13)$$

where the notation ML-Cen is for *censored maximum likelihood*.

Finally, consider a segment C with edge weights being $\{x_1, \ldots, x_n, \ldots, x_m\}$, and the merging decision when the edge under study is e. Then, combining Eqs. (11) and (13), the one-sided condition for merge rejection becomes:

$$\begin{aligned} w(e) &> HypThr(C)\\ &= \left.\frac{2\ln(1/\delta)\left(\sum_{i=1}^n x_i+(m-n)x_n\right)}{\chi^2_{1-\alpha/2,2n}}\right|_{\substack{x_i\in C\\ n=|C|}}, \end{aligned} \quad (14)$$

where m is a user specified parameter reflecting the expected size of the true segment. In the case that $m = n$, $\hat{\lambda}_{ML-Cen}$ converges to $\hat{\lambda}_{ML}$, which we use also for $n > m$. To gain further intuition about this threshold, recall that $\chi^2_{1-\alpha/2,2n}$ grows almost linearly with the segment size n.

In summary, our hypothesis testing algorithm for image oversegmentation (denoted pLV-ML-Cen) is exactly as Algorithm 1, but the minimum internal difference between two components is set to:

$$MInt(C_i, C_j) = \min_{x\in\{i,j\}} HypThr(C_x). \quad (15)$$

The value of λ for a given segment characterizes the distribution of its edges. The larger the variability within a segment, the smaller the value of λ; see Fig. 4. Figure 5 presents an example segmentation of the probabilistic local variation method with correction for biased sampling. Note that probabilistic local variation appears to have more segments than the original LV. This happens because the LV algorithm produces many elongated and hardly visible segments along the boundaries.

Probabilistic local variation with censoring is based on different principles and yet follows the same basic behavior as the LV algorithm. We consider the decision condition (14) and observe the following four properties:

1. As discussed above, the denominator $\chi^2_{1-\alpha/2,2n}$ grows with the size of the segment, n, while the numerator decreases with n. Thus, the threshold decreases with n and gives preference to merges of small segments.
2. The heaviest weight in the MST, x_n, appears in the numerator of $HypThr(C)$. Thus, heavier x_n leads to a higher threshold and to a more likely merge decision.

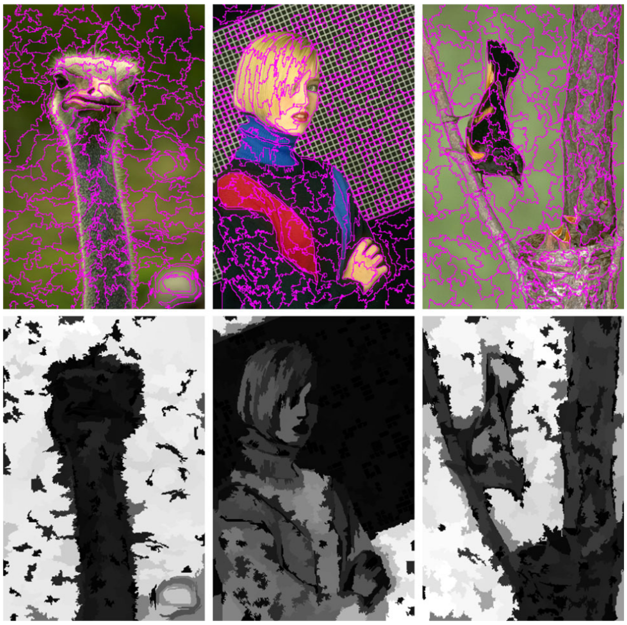

Fig. 4 *Bottom*: visualization of $\hat{\lambda}$ for each segment. The brighter the segment, the larger the value of $\hat{\lambda}$. Thus, white represents smooth segments and black represents textured ones. *Top*: corresponding oversegmentation

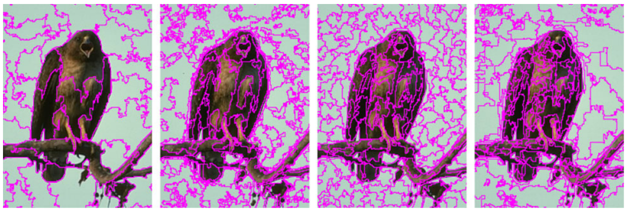

Fig. 5 Segmentations obtained by the methods described in the paper (*left* to *right*): LV-MaxEst, LV-MaxEst-c, pLV-ML-Cen and LV. The last 3 methods are tuned to have roughly the same number of segments

3. For small segments, where $n \ll m$, the importance of the heaviest edge, x_n, is amplified by a factor linear in $(m - n)/n$. For larger segments, though, the amplification factor is smaller, making the average weight of the segment edges more important.
4. It is straightforward to show that there is a predicate for which probabilistic LV leads to segmentation which is not too fine in the sense specified in Felzenszwalb and Huttenlocher (2004). Showing that it is not too coarse (Felzenszwalb and Huttenlocher 2004) seems harder because the sampling order is not tightly related to our threshold criterion.

5.4 Complexity of Hypothesis Testing

To calculate the threshold (14), we need to keep three values for each segment: the sum of its elements, the value of the last (heaviest) edge added to it, and its number of elements. All these values are updated in $O(1)$, and thus the complexity of the hypothesis testing method is exactly the same as that of the LV algorithm, namely $O(n \log n)$.

6 Experimental Results

6.1 Testing Probabilistic Local Variation

Following common practice, we use the boundary recall and the undersegmentation error as quantitative performance measures. The recall (Martin et al. 2004) is the fraction of ground truth boundary pixels that are matched by the boundaries defined by the algorithm. The undersegmentation error (Levinshtein et al. 2009) measures the area of incorrect merges of true segments (or parts of them); see Martin et al. (2004), Levinshtein et al. (2009) for implementation details. All experiments were performed on BSDS300 (test) (Martin et al. 2001).

The first probabilistic version, LV-MaxEst, is based on a uniform distribution model and estimates its interval. It is parameterless and gives a single segmentation (see Fig. 5). It is difficult to compare its average recall to that of other methods, because the segmentation of different images results in different numbers of segments (and different recall). Therefore, we use the generalized method, LV-MaxEst-c, for comparison; see Sect. 4.3. The recall curve is shown in Fig. 6 (top). It is clearly inferior to that of LV but in a sense it behaves similarly and is better than several of the other methods in the literature.

The other probabilistic version relies on the exponential distribution model, which is specified by a single parameter λ. It is important to estimate this parameter carefully. The maximum likelihood approach, pLV-ML, overestimates λ, yielding poor recall; see Fig. 6 (top). Estimating the confidence interval (95 %) and using its lower limit as in pLV-ML-CI (Sect. 5.2) results in the same recall as LV. Just like LV, the method depends on a single parameter δ, determined by the number of segments selected. Observing that the data is biased led to using censored estimation. The resulting algorithm (pLV-ML-Cen) gives a more accurate estimate of λ and yields excellent, state-of-the-art, results.

This algorithm depends on two parameters: m, the expected size of the segment, and δ, the level of significance of the decision [see Eq. (14)]. The number of segments is controlled by any combination of them. For a prespecified number of segments S (e.g., as needed for comparing different methods), we use the average segment size $m = ImageSize/S$, but found that setting m to any large value (e.g., 200) works equally for $S \in [200, 2000]$; δ is then tuned so that S segments are obtained. This option was used for generating Fig. 6. Alternatively, m, δ can be set by empirically

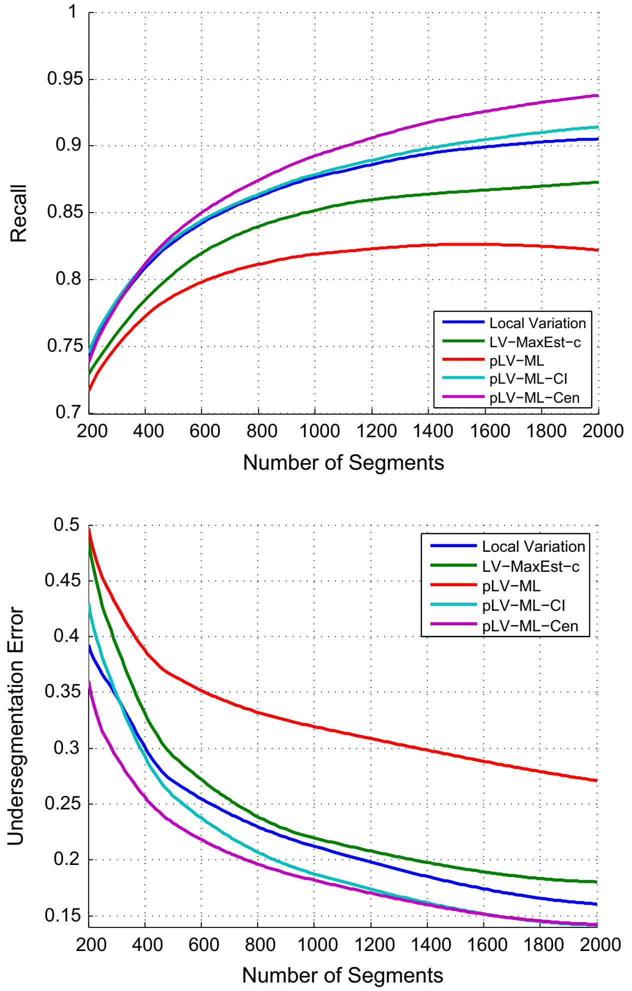

Fig. 6 Recall (*top*) and undersegmentation error (*bottom*) of the LV method against its probabilistic versions proposed in this paper. The proposed hypothesis test (with censored estimation) performs best for medium to large segment counts

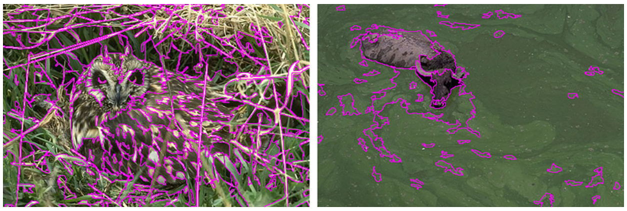

Fig. 7 Two images segmented with the probabilistic local variation algorithm (censored version) with parameters $m = 350, \delta = 0.05$. The number of segments depends on the characteristics of the image, yielding 268 (*left*) and 57 (*right*) segments respectively

maximizing the average performance, which depends on the image set. Then, using these parameters for a specific image provides adaptive segmentation, yielding more segments on "busy" images and fewer segments on smooth ones. See an example in Fig. 7.

Table 1 Multiclass segmentation accuracy when using the following methods for the oversegmentation stage: pLV, LV, mean shift (MS), turbopixels (TP), watershed (WS), SLIC and ERS

pLV	LV	MS	TP	WS	SLIC	ERS
77.0 %	74.6 %	70.3 %	76.0 %	74.7 %	76.9 %	78.0 %

In terms of recall, probabilistic local variation improves LV and matches ERS to achieve the best results for large numbers of segments. The undersegmentation error is just behind that of ERS, best for this measure; see Fig. 1. The running time of our method is exactly the same as LV (0.3 sec on a Pentium 4GB machine), which is almost as fast as the fastest method (SLIC) and much faster than the only method that achieves the same recall (ERS, 2.5 sec).

6.2 Multi-Class Segmentation Performance

Extracting superpixels is not an end in itself, and therefore testing oversegmentation in the context of common tasks is important. Following the evaluation framework presented in Achanta et al. (2012), we examine the quality of pLV (specifically pLV-ML-Cen) in the context of the higher level task of multi-class segmentation. We perform our experiments on the MSRC 21-class database (Shotton et al. 2009) and use the segmentation method of Gould et al. (2008), which proceeds as follows: the input image is divided into superpixels and a set of features is calculated on each superpixel, the calculated features are fed to a set of classifiers previously trained (one for each class), and the labeling is selected by minimizing an energy potential on a conditional random field. Table 1 presents the segmentation accuracy obtained when selecting different oversegmentation methods. Note that the accuracy achieved when using pLV is higher than when using the original LV algorithm. Furthermore, the accuracy obtained with pLV is improved only by the slower ERS algorithm.

7 Discussion

7.1 Single-Linkage Clustering

The LV algorithm, as well as its probabilistic versions considered here, are hierarchical algorithms, which, at every step, examine and possibly merge the two most similar segments.

Being single-linkage algorithms, they specify the dissimilarity between two segments, X and Y, as the minimal distance between their two closest elements: $D(X, Y) = \min_{x \in X, y \in Y} d(x, y)$, where $d(x, y)$ is some underlying dissimilarity function between elements. Recall that in the graph based notation the dissimilarities between elements are represented by the corresponding edge weights and therefore

$D(X, Y)$ is the minimal weight of an edge from X to Y. In the LV and pLV case, the decision whether to merge two segments is made by testing, independently for each segment, whether the weight $D(X, Y)$ belongs to the distribution of weights estimated for the segment.

This paper focuses on the LV algorithm (Felzenszwalb and Huttenlocher 2004) and on the implied single-edge based merging decision. Alternatively, the decision as to whether two segments should be merged can be made by testing whether all their weights may be explained by a single, common distribution. For one-dimensional distributions characterizing a segment, classical tests such as the Kolmogorov-Smirnov test (Ross 2009) may be used; see, e.g., Pauwels and Frederix (2000). See also Peng et al. (2011) for another use of one-dimensional test for segmentation. Usually, however, more effective, multidimensional distributions describing, say, the segment's texture or color, are preferred. Formal tests for multidimensional distributions are problematic (see, however, Glazer et al. (2012)). Usually the distribution is approximated as a mixture of visual words (textons) and the χ^2 distance between the histograms serves as a measure of dissimilarity between segments. A more reliable dissimilarity measure can be obtained by augmenting this distance with edge information (Martin et al. 2004).

In the context of hierarchical segmentation, the distribution comparison technique may be used in two different ways. One approach carries out the merging process according to a fixed order determined before the process begins, as done in single-linkage processes. This option is inconsistent with the hierarchical approach because the most similar pairs of segments (according to the merging criterion) are not tested before the others. The other option, to recalculate the order dynamically after every merge, is computationally expensive.

An effective combination studied in the literature is to divide the hierarchical merging process into stages. At the beginning of a stage, every segment is specified as an element, and the dissimilarity between the elements is calculated according to an arbitrary dissimilarity measure, which may depend on the distributions. Then all the merges in this stage proceed according to a single-linkage algorithm. This approach has shown to be a good trade-off between runtime and accuracy, as presented in Kim et al. (2011), Ren and Shakhnarovich (2013), Baltaxe (2014).

8 Conclusion

The local variation algorithm of Felzenszwalb and Huttenlocher (2004) is a simple yet amazingly effective oversegmentation method. In this paper we analyzed the LV algorithm using statistical and empirical methods and showed that the algorithm and its performance may be explained by statistical principles.

We proposed an oversegmentation algorithm, denoted probabilistic local variation, that is based on hypothesis testing and on the statistical properties of natural images.

We found that probabilistic local variation is highly accurate, outperforms almost all other oversegmentation methods, and runs much faster than the one equally accurate oversegmentation algorithm (ERS). This is remarkable because it follows from a statistical interpretation of a 10-year old method (LV), which is, by the way, still one of the best competitors.

Acknowledgments The authors would like to thank the US-Israel Binational Science Foundation, the Israeli Ministry of Science, Technology and Space, and the E. and J. Bishop Research Fund.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Susstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, *34*(11), 2274–2282.

Alexe, B., Deselaers, T., & Ferrari, V. (2012). Measuring the objectness of image windows. *PAMI*, *34*(11), 2189–2202.

Arbelaéz, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour detection and hierarchical image segmentation. *PAMI*, *33*(5), 898–916.

Arbelaéz, P., Pont-Tuset, J., Barron, J., Marqués, F., & Malik, J. (2014). Multiscale combinatorial grouping. In CVPR (pp. 328–335).

Baltaxe, M. (2014). Probabilistic local variation segmentation. Master's thesis, Technion.

Cabral, R., & Furukawa, Y. (2014). Piecewise planar and compact floorplan reconstruction from images. In CVPR (pp. 628–635).

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *PAMI*, *24*(5), 603–619.

De Bonet, J. S., & Viola, P. (1997). A non-parametric multi-scale statistical model for natural images. In NIPS (pp. 773–779).

Delaitre, V., Fouhey, D. F., Laptev, I., Sivic, J., Gupta, A., & Efros, A. A. (2012). Scene semantics from long-term observation of people. In ECCV (pp. 284–298).

Dollár, P., & Zitnick, C. L. (2015). Fast edge detection using structured forests. *PAMI*, *37*(8), 1558–1570.

Epstein, B., & Sobel, M. (1953). Life testing. *Journal of the American Statistical Association*, *48*(263), 486–502.

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2013). Learning hierarchical features for scene labeling. *PAMI*, *35*(8), 1915–1929.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *IJCV*, *59*(2), 167–181.

Glazer, A., Lindenbaum, M., & Markovitch, S. (2012). Learning high-density regions for a generalized kolmogorov-smirnov test in high-dimensional data. In NIPS (pp. 737–745).

Gould, S., Rodgers, J., Cohen, D., Elidan, G., & Koller, D. (2008). Multi-class segmentation with relative location prior. *IJCV*, *80*(3), 300–316.

Grenander, U., & Srivastava, A. (2001). Probability models for clutter in natural images. *PAMI*, *23*(4), 424–429.

Heiler, M., & Schnörr, C. (2005). Natural image statistics for natural image segmentation. *IJCV*, *63*(1), 5–19.

Hoiem, D., Efros, A. A., & Hebert, M. (2007). Recovering surface layout from an image. *IJCV*, *75*(1), 151–172.

Huang, J., & Mumford, D. (1999). Statistics of natural images and models. In CVPR (pp. 541–547).

Juneja, M., Vedaldi, A., Jawahar, C., & Zisserman, A. (2013). Blocks that shout: Distinctive parts for scene classification. In CVPR (pp. 923–930).

Kim, S., Nowozin, S., Kohli, P., & Yoo, C. D. (2011). Higher-order correlation clustering for image segmentation. In NIPS (pp. 1530–1538).

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, *7*, 48–50.

Larsen, R. J., & Marx, M. L. (2012). *An introduction to mathematical statistics and its applications* (5th ed.). New Jersey: Prentice Hall.

Lee, Y., & Grauman, K. (2015). Predicting important objects for egocentric video summarization. *IJCV*, *114*(1), 38–55.

Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., & Siddiqi, K. (2009). Turbopixels: Fast superpixels using geometric flows. *PAMI*, *31*(12), 2290–2297.

Liu, M.Y., Tuzel, O., Ramalingam, S., & Chellappa, R. (2011). Entropy rate superpixel segmentation. In CVPR (pp. 2097–2104).

Mallat, S. G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *PAMI*, *11*(7), 674–693.

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

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

Meyer, F. (1994). Topographic distance and watershed lines. *Signal Process*, *38*, 113–125.

Moulin, P., & Liu, J. (2006). Analysis of multiresolution image denoising schemes using generalized gaussian and complexity priors. *IEEE Transactions on Information Theory*, *45*(3), 909–919.

Pauwels, E. J., & Frederix, G. (2000). Image segmentation by nonparametric clustering based on the kolmogorov-smirnov distance. In ECCV (pp. 85–99).

Peng, B., Zhang, L., & Zhang, D. (2011). Automatic image segmentation by dynamic region merging. *IEEE Transactions on Image Processing*, *20*(12), 3592–3605.

Ren, X., & Malik, J. (2003). Learning a classification model for segmentation. In ICCV (pp. 10–17).

Ren, Z., & Shakhnarovich, G. (2013). Image segmentation by cascaded region agglomeration. In CVPR (pp. 2011–2018).

Rosenfeld & Weinshall, D. (2011). Extracting foreground masks towards object recognition. In ICCV (pp. 1371–1378).

Ross, S. M. (2009). *Probability and statistics for engineers and scientists. Wiley series in probability and mathematical statistics* (4th ed.). New Jersey: Academic Press.

Ruggles, R., & Brodie, H. (1947). An empirical approach to economic intelligence in world war II. *Journal of the American Statistical Association*, *42*(237), 72–91.

Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2009). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, *81*(1), 2–23.

Srivastava, A., Lee, A. B., Simoncelli, E. P., & Zhu, S. C. (2003). On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, *18*(1), 17–33.