Fast and Reliable Color Region Merging inspired by Decision Tree Pruning

Richard Nock
Université Antilles-Guyane
Dépt Scientifique Inter-Facultaire
Campus de Schoelcher, B.P. 7209
97275 Schoelcher, France
rnock@martinique.univ-ag.fr

Abstract

In this paper, we exploit some previous theoretical results about decision tree pruning to derive a color segmentation algorithm which avoids some of the common drawbacks of region merging techniques. The algorithm has both statistical and computational advantages over known approaches. It authorizes the processing of 512×512 images in less than a second on conventional PC computers. Experiments are reported on thirty-five images of various origins, illustrating the quality of the segmentations obtained.

1 Introduction

It is established since the Gestalt movement in psychology [14] that perceptual grouping plays a fundamental role in human perception. Even though this observation is rooted in the early part of the XX^{th} century, the adaptation and automation of the segmentation (and more generally, grouping) task with computers has remained so far a tantalizing and central problem for image processing. This is all the more important for computers as grouping authorizes the reduction of the size of data, and, by the way, can reduce dramatically the space and time complexities for some post-processing tasks, in a field where data are easily available and in huge quantities.

Roughly speaking, the problem can be presented as the transformation of the collection of pixels of an image into a meaningful arrangement of regions and objects [8]. But, how can we identify objects ? The Gestalt movement identified some factors leading to the perception of objects : symmetry, similarity, parallelism, etc. [4]. Though these rules are conceptually satisfying to explain the phenomenon at the human level, their lack of details makes it hard to see them as directly implementable algorithms [4], even if some of them can be extended to rigorous algorithms [15]. So far, image segmentation techniques have tackled the problem by studying mathematical properties of the image seen [16], or more rarely by a direct emphasis on algorithmic and computational issues [3].

There are four large categories of approaches to image segmentation [16], one of which is of direct interest to us : region growing and merging techniques. In this group of algorithms, regions are sets of pixels with homogeneous properties and they are iteratively grown by combining smaller regions or pixels, pixels being taken as elementary regions. Region growing techniques usually work with a statistical test to decide the merging of regions [11, 16].

In the field of image segmentation (or more generally, image processing), Machine Learning (ML) techniques might appear tailor-made for high-level tasks, such as in [9]. One might wonder whether applying such techniques or their adaptations to image processing based on low-level cues (typically, image segmentation) might be well worth the try, in a field where the approaches are already numerous, of many kinds (local filtering, balloons, snakes, MDL/Bayesian, region growing, etc. [16]), and where heavy mathematical approaches have obtained significant results (see *e.g.* the results of [16]). We think some clues coming from image processing indicate that ML-derived techniques might be of interest already for low-level processing tasks. First, vision is widely accepted as an *inference* problem, *i.e.* the search of what caused the observed data [4]. Second, as argued by [12], low-level image segmentation should not aim at producing a correct segmentation, but rather a good *approximation* given the post-processing stages of the image. One more reason can give evidence for the possible interest in using ML-related techniques. ML and Computational Learning Theory (COLT) works such as [7] have brought a number of useful results, or found a way to use a number of results, in statistics or probability, which remove many undesirable distribution assumptions on the data. The most commonly used theoretical founding model for ML/COLT is the PAC model of Valiant [13], which removes any such distribution assump-

tions. In image segmentation, there have been a number of probabilistic approaches to the problem [1, 16] or [4] (Chp. 18). Most of them rely on heavy, penalizing distribution assumptions on the data (such as normality), and finally some recent prominent work in image segmentation has made emphasis on more conventional, non-probabilistic approaches, such as discriminant analysis-type [12, 2].
Our aim in this paper is clearly *not* to propose a ML algorithm to segment images, but rather to exploit some of the common points between image segmentation and decision-tree pruning, and then adapt the theoretical ML/COLT ideas and tools of [7] to our setting, to derive a new segmentation algorithm. This paper consists therefore of an original (and novel) adaptation of previous ML/COLT work to low-level image processing, a rather seldom result in our context. The next section goes in depth in the common points between decision-tree pruning and image segmentation. It is followed by a section presenting a model of image generation. Then, two sections detail respectively the statistical and algorithmic contents of our approach. The last section discusses experiments on numerous images.

2 From pruning to merging

A decision tree is a classifier making recursive partitions of a representation space, according to some classes. Consider fig. 1, showing a 2D representation space restricted to the integer couples of the domain $[0, x_4] \times [0, y_3]$ for some $x_4, y_3 > 0$. Suppose that there are g classes (not shown). The point is that a probability distribution is given to domain$\times$classes. Each couple (observation, class) is called an *example*, and the problem is to fit as best as possible the domain of all examples, using only a potentially small subset of this domain, sampled according to the distribution. Decision trees are very efficient ways to tackle the problem using a very simple formalism [7]. The quality of a decision tree is evaluated by its error probability over the whole domain, also called structural risk, which we can only estimate using the error frequency over the examples seen, also called empirical risk [7]. The classification of an observation is made as follows: each internal node (and the root) of the tree is labeled by a test on a variable of the observations; each leaf is labeled by a class (not shown in fig. 1). The classification process starts from the root of the tree. If the observation satisfies the current test, then it follows the right path, otherwise it follows the left path, until it reaches a leaf, and therefore is given the corresponding class. In fig. 1, an observation for which $x = x_2$ and $y = y_2$ would fall in region A, as shown by the dotted path. The most popular decision tree learning algorithms (CART, C4.5 [7]) proceed by growing a very large decision tree to fit as best as possible the examples seen, and then pruning it to get (hopefully) a good fit of the whole domain. Pruning a

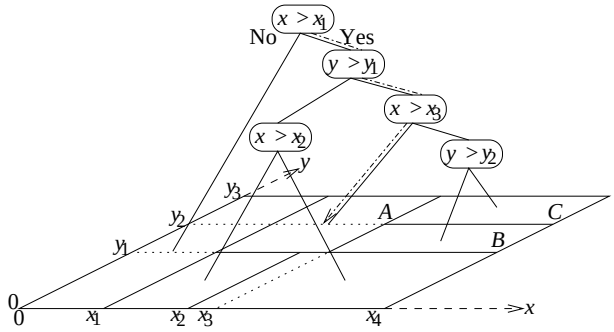

Figure 1. A domain and its recursive partition by a decision-tree (see text for details).

tree consists in removing iteratively an internal node and its subtree, thereby replacing the node by a class label. Pruning the node whose test is "$x > x_3$" in image 1 would boil down to merge regions A, B and C.
If we consider that the color values of pixels in an image are the result of a theoretical value (the one of the region they belong to), combined with a random variability factor (such as noise), then the task of region merging can be loosely reformulated as the recognition of the regions having the same theoretical values in an observed image. This problem is quite similar to decision tree pruning (consider that the image has $x_4 y_3$ pixels in fig. 1); more importantly, the tools used in both problems can be the same: the concentration of random variables to give confidence bounds, either for the theoretical color values of groups of pixels, or for the structural risks in decision-tree pruning.
However, our task appears to be more difficult than pruning, since we generate much more potential configurations. Consider *e.g.* fig. 1: the tree has only 7 possible prunings (including the original tree), whereas there are 42 possible segmentations of the domain with the 6 initial regions.
Image processing has potentially two characteristics which make it a good candidate for the adaptation of particular ML techniques. First, the quality of the pruning strongly depends on the available quantity of data, and small datasets typically lead to uncontrollable over-pruning. In image segmentation, the material is available in huge quantities. Second, image segmentation is a field in which fast algorithms are obviously crucial with the advent of real-time (video) image processing, but practice shows that "fast" is a field-dependent subjective notion: for example, [1] consider a "very fast" algorithm extracting a few dozens of regions in a 512$\times$512 image in less than 10 seconds, but on an high-end UltraSPARC workstation; [12] give execution times on 100$\times$120 images of about 2 minutes on conventional machines, and there are many other examples. We show in that paper that using our ML-derived tools can bring highly

competitive (or better) results, in reduced times.

3 Notations and models

The notation $|.|$ stands for cardinal. The observed image, I, contains $|I|$ pixels, each containing **R**ed-**G**reen-**B**lue (**RGB**) values, each of the three belonging to the set $\{1, 2, ..., g\}$. We have deliberately chosen not to use complex formulations of the colors, such as the $L*u*v*$ space [1]. I is an observation of a perfect scene I^* we do not know of, in which pixels are perfectly represented by a family of *distributions*, from which each of the observed color-level is sampled. In I^*, the optimal (or true) regions represent theoretical objects sharing a common homogeneity property:

- all pixels of a given true region have identical expectations for each **RGB** color-level,
- the expectations of adjacent regions are different for at least one **RGB** color level.

I is obtained from I^* by sampling each theoretical pixel for observed **RGB** values. Fig. 2 presents an example of a color-level for one pixel in I^* and how to generate the corresponding observed color-level of the pixel in I. In each pixel of I^*, each color-level is replaced by a set of Q independent random variables (r.v.) taking positive values on domains bounded by g/Q, such that any possible sum of outcomes of these Q r.v. with non-zero probability belongs to $\{1, 2, ..., g\}$.
The sampling of each pixel and its color levels are supposed independent from each other. Our model of image generation does not unfortunately prevent us to depend on this usual assumption (also widely used in ML/COLT). Our experiments shall demonstrate that it does not seem to have an impact on the results of the segmentation. It is important to note that this is the only assumption we make on I^*. In particular, we do not make any further assumption on the nature of each distribution (such as normality, homoscedasticity), which can be therefore different from one another in each pixel of a region, as long as the sum of their expectations is constant for each color inside a true region.
Our goal is then straightforward: find observed regions in I approximating as best as possible the true regions in I^*.
Q is certainly the less intuitive parameter of our model. We have chosen to introduce it for our model and our analyses to be as general as possible (standard analyses would fix $Q = 1$). Practically speaking, Q represents the major advantage to be a trade-off parameter, adjustable to obtain a compromise between the power of the model and the quality of the observed results. Indeed, if Q is small (say, $Q = 1$), then scarcely nothing can be estimated reliably for small regions in I, but our model is the most general possible. On the other hand, if Q is too large, then our model becomes restricted ($Q = g$ brings back a conventional Binomial law), but our estimations are the most reliable. Between, the user can choose a convenient value to keep a powerful enough model while ensuring reliable estimations on I.

4 The merging test and its properties

Our main result is based on the concentration of random variables, a tool widely used in ML/COLT literature [7]. Our model necessitates a recent result due to [10]:

Theorem 1 *(The independent bounded difference inequality, [10]) Let $\boldsymbol{X} = (X_1, X_2, ..., X_n)$ be a family of independent r.v. with X_k taking values in a set A_k for each k. Suppose that the real-valued function f defined on $\prod_k A_k$ satisfies $|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \leq c_k$ whenever vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ differ only in the k-th coordinate. Let μ be the expected value of the r.v. $f(\boldsymbol{X})$. Then for any $t \geq 0$,*

$$\mathbf{Pr}(f(\boldsymbol{X}) - \mu \geq t) \quad \leq \quad e^{-\frac{2t^2}{\sum_k (c_k)^2}} \tag{1}$$

In our case, we now prove that with high probability, the observed average $\overline{R}_a$ of any region R in I shall not deviate too much from its theoretical expectation $\mathbf{E}_a(R)$ for a single color-level a. By theoretical expectation, we mean the average of the sum over each of its pixels of the expectations of its Q distributions for color a in I^*.

Theorem 2 *Let R be a region in I. Let $\mathcal{R}_l$ be the set of regions having l pixels in I. Fix an $a \in \{\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B}\}$. $\forall \delta' > 0$, the probability that there exists a region in $\mathcal{R}_{|R|}$ such that*

$$\left|\overline{R}_a - \boldsymbol{E}_a(R)\right| \quad \geq \quad g\sqrt{\frac{1}{2Q|R|}\left(\log\frac{2|\mathcal{R}_{|R|}|}{\delta'}\right)} \tag{2}$$

is no more than δ'.

(Proof omitted due to the lack of space). Of course, the probability of occurrence of the event for *some* of the **R**, **G**, or **B** levels, is no more than $3\delta'$. What is interesting in theorem 2 is that region R is not required to belong to a true region of I^*. Indeed, if the region merging algorithm has merged together subsets of true regions $O_1, O_2, ..., O_k$ in R, then the bound of theorem 2 still holds. The use of theorem 2 will be straightforward. The probability that there exists a region in I (regardless of its size) for which the average of some of its **RGB** color deviates from its expectation by more than the right-hand-side of inequality 2 is no more than $3|I|\delta'$, since the region can have size 1, 2, ..., or $|I|$. Then, if we fix

$$\delta \quad = \quad 3|I|\delta' \tag{3}$$

and

$$b(R) \quad = \quad g\sqrt{\frac{1}{2Q|R|}\left(\log\frac{2}{\delta'} + \log|\mathcal{R}_{|R|}|\right)} \tag{4}$$

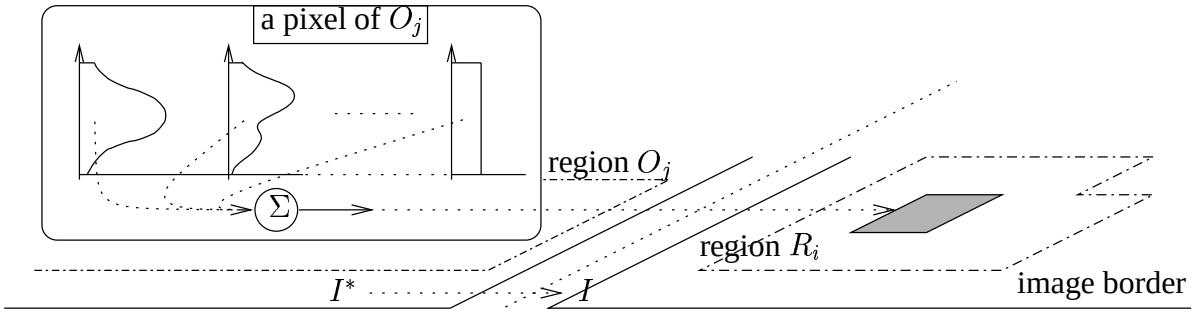

Figure 2. Generation of a single color-level for one pixel from I^* to I.

then we know that, with high probability ($> 1-\delta$), any possible region R will have a bounded variation of $\overline{R}_a$ around $\mathbf{E}(R)_a$ for *all* values $a \in \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$. Because of the triangle inequality, we know that any two regions R and R' having the same expectations ($\mathbf{E}_a(R') = \mathbf{E}_a(R) = q_a, \forall a \in \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$) shall observe a quantity $|\overline{R'}_a - \overline{R}_a|$ which will be also concentrated around zero, up to radius $b(R) + b(R')$ actually, since $|\overline{R'}_a - \overline{R}_a| \leq |\overline{R'}_a - q_a| + |\overline{R}_a - q_a|$ (triangle inequality). If $|\overline{R'}_a - \overline{R}_a| \leq b(R) + b(R')$ for all $a \in \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$, then we can suppose that R and R' belong to the same true region in I^*, and merge them. This gives the merging predicate $\mathcal{P}(R, R')$ of our region merging algorithm, returning whether R and R' can be merged or not:

$$\mathcal{P}(R,R') = \begin{cases} \texttt{true} & \text{iff} \quad \forall a \in \{\mathbf{R},\mathbf{G},\mathbf{B}\}, \\ & \quad |\overline{R'}_a - \overline{R}_a| \leq b(R) + b(R') \\ \texttt{false} & \text{otherwise} \end{cases}$$

We also need to compute the value $|\mathcal{R}_{|R|}|$. Because of the fact that exact values are difficult to compute, we have used a practical bound $|\mathcal{R}_l| \approx \left((l+2)^{\min\{g,l\}}\right)/(l+1)$ (proof omitted) which follows as closely as possible the true value, being smaller for the small values of l (thus, for small regions, where typically $l < g$), and larger for the large values of l. The risk in computing an upperbound for $|\mathcal{R}_l|$ was to get a too large upperbound, thereby leading to a too large risk of over-merging (we shall see it in the next section).

5 The algorithm and its properties

Suppose that the image I contains r_I rows and c_I columns. This represents $N = 2r_Ic_I - r_I - c_I$ couples of adjacent pixels (in 4-connexity). Name S_I as the set of these couples. Denote $f(.)$ the function which takes a couple of pixels (p, p'), and returns the maximum of the three color differences (**R**, **G** and **B**) in absolute value between p and p'. Denote as `quicksort`(S_I,f) the output of the quicksort algorithm for set S_I, in increasing order of function $f(.)$. For any pixel p of I, we denote as $R(p)$ the current region to which p belongs in I. The algorithm is called `PSIS`, for Probabilistic Sorted Image Segmentation.

Algorithm 1: `PSIS`(I)

Input: an image I

S'_I=`quicksort`(S_I,f);
for $i = 0$ ***to*** $N-1$ **do**
  /*(p_i, p'_i) is the i^{th} couple of S'_I*/
  if $R(p_i) \neq R(p'_i)$ ***and*** $\mathcal{P}(R(p_i), R(p'_i))$=`true` **then**
    `Union`($R(p_i)$,$R(p'_i)$);

One might wonder why we have chosen to order the couples of pixels in `PSIS`. Indeed, such a requirement is *a priori* not clear from our theory, and it brings an $\mathcal{O}(|I| \log |I|)$ complexity (sometimes smaller, [2]) instead of the (almost) linear complexity reachable without this stage. There are basically three kind of errors our algorithm can suffer with respect to the optimal segmentation. First, under-merging represents the case where one or more regions obtained are strict subparts of true regions. Second, over-merging represents the case where some regions obtained strictly contain more than one true region. Third, there is the "hybrid" (and most probable) case where some regions obtained contain more than one strict subpart of true regions. Ordering the pixels test is a way to limit this third kind of error, since we approximate the following property:

(P) All merging tests inside true regions are made before any merging test between true regions.

Obviously, **(P)** is unreachable in practice; however, from a purely theoretical point of view, **(P)** has a key effect when used with our statistical material: `PSIS` only suffers over-merging with probability $> 1 - \delta$ (eq. 3). Indeed, we know that for any regions R, R' coming from the same true region in I^*, we have $\forall a \in \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}, |\overline{R'}_a - \overline{R}_a| \leq |\overline{R'}_a - q_a| + |\overline{R}_a - q_a|$ where $q_a = \mathbf{E}_a(R') = \mathbf{E}_a(R)$. With probability $> 1-\delta$, we also know (theorem 2 and eq. 3) that $|\overline{R'}_a - \overline{R}_a| \leq b(R) + b(R')$ for any regions R, R' coming from the same true region in I^*. Since this is our merging test, and since **(P)** holds, it follows that the segmentation obtained is an under-segmentation of I^*.
Therefore, theoretically speaking, **(P)** allows to eliminate

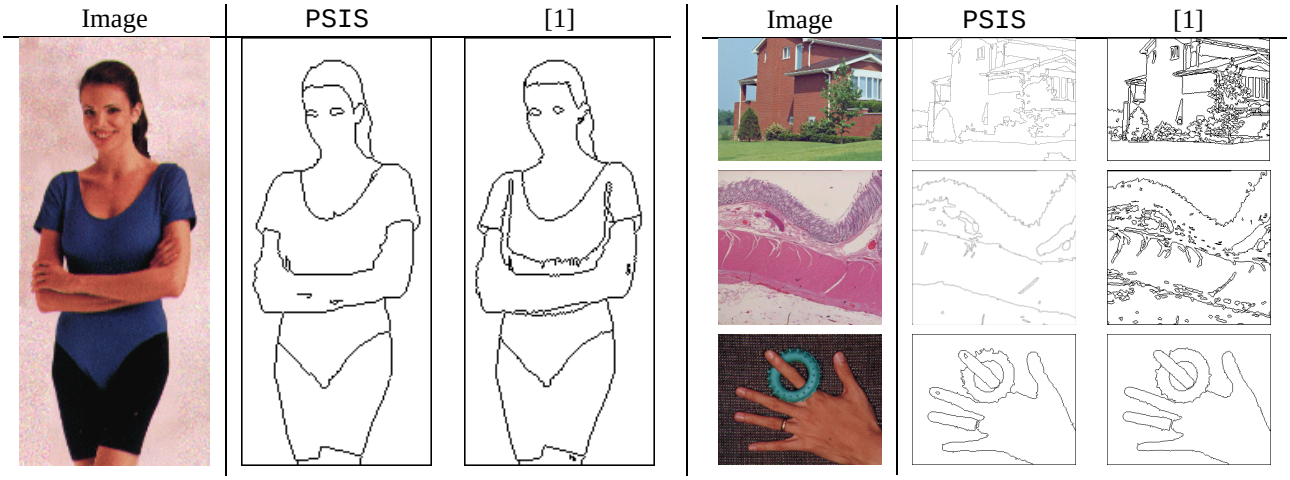

Figure 3. Results obtained by PSIS and [1] on four images. Region are white with black borders.

under-merging and the hybrid error case with high probability; this motivated us in fast approximations of it. In fact, over-merging can somewhat be controlled in practice for regions with size large enough (typically $\geq g$) as long as the bound for $|\mathcal{R}_l|$ is not too large (see Eq. 4). This is the reason for our choice of the upperbound for $|\mathcal{R}_l|$.

6 Experimental results

Due to the lack of space, we report experiments only on thirty-five images among all on which PSIS was tested (figures 3, 4 and 5). While looking at the results, the reader may keep in mind that:

> the values of the parameters of PSIS are the same for all images: $\delta' = 1/(3|I|^2)$ (theorem 2) and $Q = 32$. Furthermore, the images were used as they are, *i.e.* without any preprocessing.

Therefore, the results of PSIS do not stem from any domain- or image-dependent preprocessing or parameter tuning. PSIS was implemented in C. Fig. 3 displays some results to be compared with [1, 16]. While PSIS obtained record times for processing these images, the results are highly competitive with the other approaches. Consider the woman image. Her bust is better segmented than [1]. When looking at [16]'s result for this image, our result appears to be better: after over a hundred iterations, [16]'s technique of region competition does not manage to find the woman's eyes. [16] argue that that the eyes are too small and assimilated to noise. This is clearly not a mistake PSIS has made. The hand image also displays good results, all the better if we consider that the model of image segmentation of PSIS does not explicitly integrate textures. In fig. 4, the left table demonstrates that PSIS obtains again nice results on texture segmentation (tennis and rock images), all the more interesting if we compare them to the approaches of [5, 6], tailor-made for texture segmentation. The right image shows for some images a particular region isolated by PSIS. Remark from lena and squirrel that PSIS is able to isolate regions with high variability (*e.g.* the grass), and obtains results even better than [16] on the squirrel image: their segmentation, although tailor-made for textured images, obtains a segmentation of the grass with many holes, a popular drawback of region-merging techniques [16], see also the result of [2] in fig. 5 (bottom row, region #2) for the grass. Note also the nice segmentation of the truck compared to [2] (fig. 5, bottom row, region #4). The small images in fig. 5 (a-x) display the ability of PSIS to handle reasonable gradients due to lightning and slanted surfaces (images b, f, i, j, k, r, s, u, w, x). This corrects another drawback of many segmentation techniques, relying on the assumption that the image is piecewise constant [2]. Note that in image x, PSIS obtains exactly two regions. The bowl region (x,#2) approximates nicely the true object, in a better way than [2], who find significantly more regions, again with many holes compared to PSIS. In image w, which contains gradients and light effects, PSIS finds exactly three regions, approximating almost perfectly the three parts of the image: the background, the pot, and its lid. The gradients and light effects of image j are much more difficult to handle. In that case, PSIS found four regions (all shown), two of which (#2 and #3) are good approximations of two parts of the object (exterior/interior). More generally, many regions found by PSIS in these small images approximate conceptually distinct parts of the objects: see (a,#4), (b,#3), (c,#4), (e,#3), (f,#3), (n,#4), (s,#3), (t,#3), and many others.

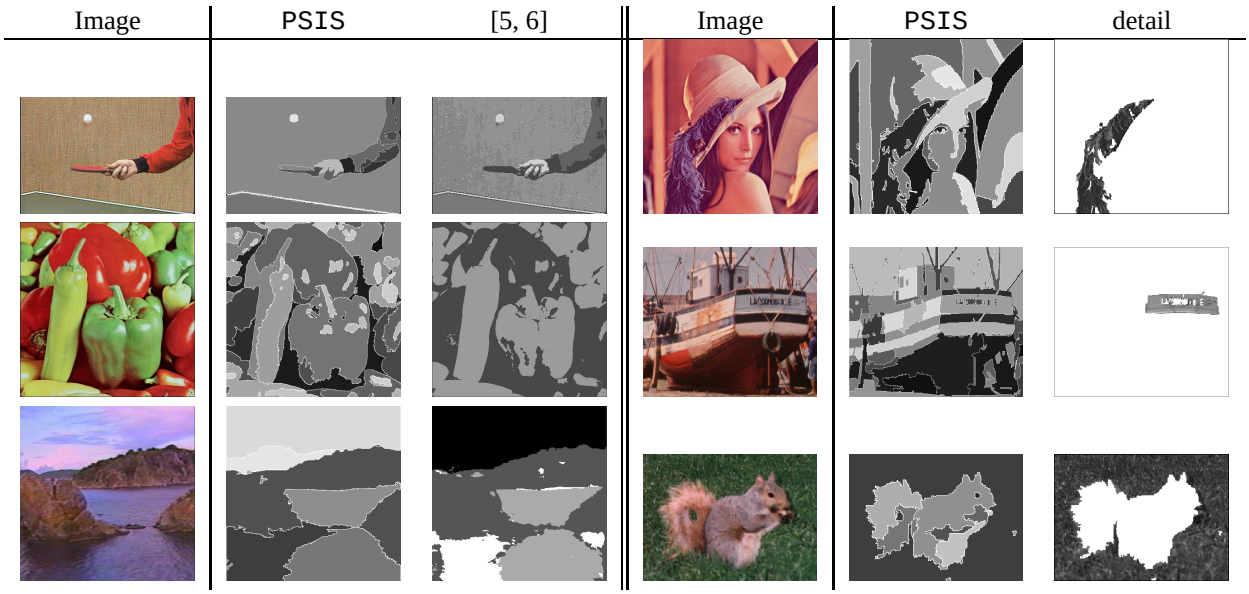

Figure 4. Results obtained by PSIS and [5, 6] on various images. The "detail" column is a region of interest in PSIS's results. PSIS's segmentations are grey-leveled averaged with white borders. Conventions for the results of [5, 6] are various but intuitive.

References

[1] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *Proceedings of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 750–755, 1997.

[2] P. F. Felzenszwalb and D. P. Huttenlocher. Image segmentation using local variations. In *Proceedings of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 98–104, 1998.

[3] C. Fiorio and J. Gustedt. Two linear time Union-Find strategies for image processing. *Theoretical Computer Science*, 154:165–181, 1996.

[4] D. Forsyth and J. Ponce. *Computer Vision - A Modern Approach*. Prentice Hall, 2001. Forthcoming.

[5] Z. Kato. *Modélisations Markoviennes Multirésolutions en vision par ordinateur. Application à la segmentation d'images spot.* PhD thesis, Université de Nice-Sophia Antipolis, 1994.

[6] Z. Kato, T. C. Pong, and J. C. M. Lee. Motion compensated color video classification using markov random fields. In *Proceedings of the 3 rd Asian Conference on Computer Vision*, 1998.

[7] M. J. Kearns and Y. Mansour. A Fast, Bottom-up Decision Tree Pruning algorithm with Near-Optimal generalization. In *Proceedings of the 15 th International Conference on Machine Learning*, pages 269–277, 1998.

[8] T. Leung and J. Malik. Contour continuity in region based image segmentation. In *Proceedings of the 5 th Europ. Conf. on Computer Vision*, pages 544–559, 1998.

[9] O. Maron and A.-L. Ratan. Multiple instance learning for natural scene classification. In *Proceedings of the 15 th International Conference on Machine Learning*, pages 341–349, 1998.

[10] C. McDiarmid. Concentration. In M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 1–54. Springer Verlag, 1998.

[11] T.-Y. Philips, A. Rosenfeld, and A. C. Sher. O(log n) bimodality analysis. *Pattern Recognition*, pages 741–746, 1989.

[12] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[13] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, pages 1134–1142, 1984.

[14] M. Wertheimer. Laws of organization in perceptual forms. In W. B. Ellis, editor, *A Sourcebook of Gestalt Psychology*, pages 71–88. Harcourt, Brace and Company, 1938.

[15] S.-C. Zhu. Embedding gestalt laws in markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:1170–1187, 1999.

[16] S.-C. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:884–900, 1996.

Figure 5. Some images, and some of the most significant regions obtained by PSIS. The bottom row shows the result of [2] on the street image. The conventions are the same for all region images (everything that is not the region is white), except for region #1 on the street results, surrounded by black due to the brightness of the road.