HAL
open science

Simultaneous Segmentation and Filtering via Reduced Graph Cuts

Nicolas Lermé, François Malgouyres

To cite this version:

Nicolas Lermé, François Malgouyres. Simultaneous Segmentation and Filtering via Reduced Graph Cuts. ACIVS, 2012, Brno, Czech Republic. pp.201 - 212, 10.1007/978-3-642-33140-4_18 . hal-01486861

HAL Id: hal-01486861

https://hal.science/hal-01486861v1

Submitted on 10 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simultaneous Segmentation and Filtering via Reduced Graph Cuts

N. Lermé
LAGA UMR CNRS 7539
LIPN UMR CNRS 7030
Université Paris 13
nicolas.lerme@lipn.fr

F. Malgouyres
IMT UMR CNRS 5219
Université Paul Sabatier
fmalgouy@math.univ-toulouse.fr

Abstract. Recently, optimization with graph cuts became very attractive but generally remains limited to small-scale problems due to the large memory requirement of graphs, even when restricted to binary variables. Unlike previous heuristics which generally fail to fully capture details, [8] proposes another band-based method for reducing these graphs in image segmentation. This method provides small graphs while preserving thin structures but do not offer low memory usage when the amount of regularization is large. This is typically the case when images are corrupted by an impulsive noise. In this paper, we overcome this situation by embedding a new parameter in this method to both further reducing graphs and filtering the segmentation. This parameter avoids any post-processing steps, appears to be generally less sensitive to noise variations and offers a good robustness against noise. We also provide an empirical way to automatically tune this parameter and illustrate its behavior for segmenting grayscale and color images.

Keywords: graph cuts, reduction, image segmentation, filtering.

1 Motivation and scope

Graph cuts have become increasingly popular due to their ability to efficiently compute the Maximum A Posteriori of Markov Random Fields (MRF). This popularity is notably driven by the introduction of a fast maximum-flow (max-flow) algorithm [3] making near real-time performance possible for solving numerous labeling problems such as denoising, image segmentation, stereovision, etc.

In parallel, technological advances in image acquisition have both increased the amount and the diversity of data to process. As an illustration, in the satellite SPOT-5 launched by Arianespace in 2002, the embedded high resolution sensors can capture multispectral and panchromatic images of about 1GB.

Processing this type of data amounts to solve large scale optimization problems. In the image segmentation context, almost all graph cuts-based methods are impractical to solve such problems due to the memory requirements for storing the underlying graphs. To overcome this situation, some amount of work has been done in this direction and a number of heuristics [9,10,12,5] and exact methods [7,4,13] have been proposed in recent years.

To our best knowledge, this problem seems to be first addressed in [9] where the underlying graph is built upon a pre-segmentation. Although this approach greatly reduce the computational burden of graph cuts, the results strongly depend on the algorithm used for computing the pre-segmentation. Also, better results are obtained when over-segmentation occurs, losing in this way the main benefit of such a reduction.

Others have also reported band-based methods [10,12,5]. A low-resolution of the image is first segmented and the solution is propagated to the finer level by only building the graph in a narrow band surrounding the interpolated foreground/background interface at that resolution. While such an approach drastically reduce time and memory consumption, it is limited to segment roundish objects. This problem is notably reduced in [12] but still present for low-contrasted details. In [5], finer bands are obtained using an uncertainty measure associated to each pixel.

Exact methods have been also investigated [7,4,13]. In [7], binary energy functions are minimized for the shape fitting problem with graph cuts in a narrow band while ensuring the optimality on the solution. One makes a band evolve around the object to delineate by expanding it when the minimum-cut (min-cut) touches its boundary. This process is iterated until the band no longer evolves. Although the algorithm generally converges in few iterations toward the optimal solution, an initialization is required and no bound on the band size is given.

A parallel max-flow algorithm yielding a near-linear speedup with the number of processors is described in [4]. Nevertheless, the algorithm is relatively sensitive to the available amount of physical memory and remains less efficient on small graphs.

The approach used in [13] is different: instead of reducing the graphs, the problem is decomposed into optimizable sub-problems, solved independently and updated according to the results of the adjacent problems. This process is iterated until convergence and optimality is guaranteed by Lagrangian decomposition.

Finally, another band-based method called Reduced Graph Cuts (RGC) was proposed for reducing graphs in binary image segmentation [8]. The graph is progressively built by only adding nodes which locally satisfy a condition. In the manner of [12,5], the graph nodes are typically located in a narrow band surrounding the object edges to segment. Empirically, the authors show in [8] that the solutions obtained with and without reduction are identical and the time for reducing the graph is even compensated by the time for computing the min-cut in the reduced graph when the regularization is of moderate level.

The rest of this paper is organized as follows. We first briefly review the graph cuts framework in Section 2 as well as the band-based strategy of [8] in Section 3 for reducing graphs. Afterwards, a new parameter is introduced in Section 4 for further reducing the graphs and removing small undesired segments in the segmentation due to noise. The sensitivity of this parameter as well as its robustness against noise are also evaluated through experiments for segmenting grayscale and color images.

2 Preliminaries

Consider a multi-channels image $I : \mathcal{P} \subset \mathbb{Z}^d \rightarrow [0,1]^c$ $(c > 0)$ as a function, mapping each pixel $p \in \mathcal{P}$ to a vector $I_p \in [0,1]^c$ [1]. We define a binary segmentation as an application u affecting to each pixel $p \in \mathcal{P}$ either 0 (background) or 1 (object) and we write $u \in \{0,1\}^{\mathcal{P}}$. A popular strategy to segment I is to minimize a MRF of the form [2]:

$$E(u) = \beta \sum_{p \in \mathcal{P}} E_p(u_p) + \sum_{(p,q) \in \mathcal{N}} E_{p,q}(u_p, u_q), \tag{1}$$

among $u \in \{0,1\}^{\mathcal{P}}$ and for a fixed parameter $\beta \in \mathbb{R}^+$. The neighborhood system $\mathcal{N} \subset (\mathcal{P} \times \mathcal{P})$ is a subset of all pixel pairs. In the sequel, "connectivity 0" will denote 4 and 6 neighbors in 2D and 3D images while "connectivity 1" will denote 8 and 26 neighbors for the same images. In (1), the data term $E_p(.)$ is defined as the negative log-likelihood of a label being assigned to pixel p and is computed from its color and the appearance models of the object and background seeds [2]:

$$\begin{cases} E_p(1) = -log\ \mathbb{P}(I_p | p \in \mathcal{O}) \\ E_p(0) = -log\ \mathbb{P}(I_p | p \in \mathcal{B}) \end{cases} \tag{2}$$

For any pixels pair $(p,q) \in \mathcal{N}$, the corresponding smoothness term in (1) is defined as a contrast-sensitive Ising model:

$$E_{p,q}(u_p, u_q) = \begin{cases} 0 & \text{if } u_p = u_q, \\ \frac{1}{\|p-q\|_2} exp\Big(- \frac{\|I_p - I_q\|_2^2}{2\sigma^2} \Big) & \text{otherwise,} \end{cases} \tag{3}$$

where $\|.\|_2$ is the Euclidean norm (either in $\mathbb{R}^d$ or $\mathbb{R}^c$) and σ is a free parameter generally related to noise acquisition. As an illustration, when pixels p and q belong to a uniform area, we have $\|I_p - I_q\|_2 < \sigma$. This implies a large cost in the exponential and discourages any cut between pixels p and q. Conversely, when these pixels are on both sides of a contour, we have $\|I_p - I_q\|_2 > \sigma$. This encourages any cut between pixels p and q due to low value of the exponential.

When the smoothness terms are submodular [6], the minimizer of (1) can be efficiently obtained by computing a min-cut in a weighted digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of nodes $\mathcal{V} = \mathcal{P} \cup \{s,t\}$, a set of edges $\mathcal{E} \subset (\mathcal{V} \times \mathcal{V})$ and edge capacities $c : (\mathcal{V} \times \mathcal{V}) \rightarrow \mathbb{R}^+$. The terminal nodes s and t are called the source and the sink, respectively. The set of edges $\mathcal{E}$ is split into two disjoint sets $\mathcal{E}_n$ and $\mathcal{E}_t$ denoting respectively n-links (edges linking two nodes of $\mathcal{P}$) and t-links (edges linking a node of $\mathcal{P}$ to the terminal s or t). Once the min-cut is computed in $\mathcal{G}$, we set $u_p = 1$ if a node p is connected to the source s and $u_p = 0$ if p is connected to the sink t.

[1] Usually, $\mathcal{P}$ corresponds to a rectangle.

[2] In this setting, the distributions are estimated using a Gaussian Mixtures Model. The number of Gaussians is automatically computed using a statistical criterion [1].

3 Reduction

As explained before, the memory consumption of graph cuts for segmenting high-resolution data can be very large. As an illustration, the max-flow algorithm of [3] v3.01 used in the experiments of Section 4, allocates $25\sharp\mathcal{P} + 16\sharp\mathcal{E}_n$ bytes [3]. For a fixed amount of RAM, one clearly see that the maximum image size quickly decreases as the dimensionality d of $\mathcal{P}$ increases. As shown in [8], most of the nodes in the graph are however useless during the max-flow computation since they are not traversed by any flow. Ideally, one would like to extract the smallest possible graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ from $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ while keeping the max-flow value f'^* in $\mathcal{G}'$ identical or very close to the max-flow value f^* in $\mathcal{G}$. This corresponds to an ideal optimization problem which we will not try to solve since the method for determining $\mathcal{G}$ also needs to be (very) fast.

Let us first introduce some terminology before reviewing the method of [8] for building $\mathcal{G}'$. For the sake of clarity, the same notations are used as in [8]. In accordance with the construction given in [6], we consider (without loss of generality) that a node is connected to at most one terminal

$$(s,p) \in \mathcal{E}_t \Rightarrow (p,t) \notin \mathcal{E}_t, \qquad \forall p \in \mathcal{P}. \tag{4}$$

We also summarize t-links capacities by

$$c(p) = c(s,p) - c(p,t), \qquad \forall p \in \mathcal{P}. \tag{5}$$

For any $B \subset \mathbb{Z}^d$ [4] and a node $p \in \mathcal{P}$, we denote by B_p the set translation of B at p

$$B_p = \{q + p \mid q \in (B \cap \mathcal{P})\}. \tag{6}$$

For $Z \subset \mathcal{P}$ and $B \subset \mathbb{Z}^d$, we denote by Z_B the dilation of Z by B as

$$Z_B = \{p + q \mid q \in (B \cap \mathcal{P}), p \in Z\} = \bigcup_{p \in Z} B_p. \tag{7}$$

From here, the idea developed in [8] for building $\mathcal{G}'$ is to remove from the nodes of $\mathcal{G}$ any $Z \subset \mathcal{P}$ where all nodes are linked to s (resp. to t) and such that all the flow that might get in (resp. out of) the region Z_B does so by traversing its boundary and can be absorbed (resp. provided) by the band $Z_B \setminus Z$. Building such Z is done by testing each individual pixel $p \in \mathcal{P}$. In the manner of [12,5], the remaining nodes are therefore located in a narrow band surrounding the object edges to segment. In practice, the authors of [8] use a more conservative test by testing each node $p \in Z$ in a square window B of size $(2r+1)^d$ centered in p:

$$\begin{cases} \text{either} & \Big(\forall q \in B_p,\ c(q) \geq +\delta_r\Big), \\ \text{or} & \Big(\forall q \in B_p,\ c(q) \leq -\delta_r\Big). \end{cases} \tag{8}$$

[3] The operator '$\sharp$' stands for the cardinality of a set.

[4] In practice, B is a square centered at the origin.

where $\delta_r = \frac{P(B)}{(2r+1)^d - 1}$ and $P(B)$ is defined as

$$\begin{aligned} P(B) = \max(&\sharp\{(p,q) \ : \ p \in B, q \notin B \text{ and } (p,q) \in \mathcal{N}\}, \\ &\sharp\{(q,p) \ : \ p \in B, q \notin B \text{ and } (q,p) \in \mathcal{N}\}). \end{aligned} \tag{9}$$

In words, for any node $p \in Z$ satisfying the first (resp. second) condition of (8), all its neighbors $q \in B_q$ are only linked to s (resp. t) and the flow that might get in (resp. out) through t-links in $B_p \setminus \{p\}$ suffices to saturate the n-links going out (resp. in) B_p. Thus, p becomes useless and need not to be added to $\mathcal{G}'$. An algorithm of complexity $O(1)$ (i.e. independent of r) is also mentioned in [8] for computing (8). Additionally, the extra memory storage required by this algorithm is a table of dimensionality $(d-1)$ and is therefore negligible over the image and the graph size. A key point of [8] is that the pixel error between segmentations obtained with and without reduction remains extremely low, hence preserving thin structures which are ubiquitous in some applications.

The experiments presented in [8] confirm the intuitive dependence between the size of the reduced graph $\mathcal{G}'$ and the model parameters. Indeed, when minimizing (1) by graph cuts, the t-links capacities are all multiplied by β. It is therefore straightforward to observe that the test (8) is harder to satisfy as β decreases. In such a situation, we need a larger window radius for decreasing δ_r in order to reduce the size of the reduced graph $\mathcal{G}'$. This results in wider bands around the object contours. Conversely, when β is large, we can afford a large δ_r and therefore a small window radius to decrease the size of the reduced graph $\mathcal{G}'$. An ideal situation therefore consists of large area of nodes connected to the same terminals separated by rough borders. As opposite, a less favorable situation occurs when these area consist of nodes connected to different terminals. This is typically the case when dealing with noisy images. In the next section, we embed a new parameter in (8) to both further reducing $\mathcal{G}'$ and filtering the segmentation while keeping β large.

4 Simultaneous segmentation and filtering

A naive approach to filter the segmentation is to apply morphological operators (e.g. opening or closing). Nonetheless, when the amont of noise is large, such an approach fail during the reduction since a lot of nodes would be added to the reduced graph. Another approach would consist in denoising the image first and then applying [8]. This would lead to unsatisfactory results in the case for instance of echographic images since contours of objects would be over-smoothed.

Another way to filter the segmentation is to relax (8) is to allow some nodes in B_p to fail complying the test. The proportion of nodes satisfying (8) can be controlled by a parameter $\eta \in [0,1]$. As η decreases, the test (8) becomes easier to satisfy since a larger proportion of nodes can be connected to opposite terminals. Embedding η in (8) leads to

$$\begin{cases} \text{either} & \Big(\sharp\{q \in B_p \mid c(q) \geq +\delta_r\} \geq \eta \sharp B_p\Big), \\ \text{or} & \Big(\sharp\{q \in B_p \mid c(q) \leq -\delta_r\} \geq \eta \sharp B_p\Big). \end{cases} \tag{10}$$

4.1 Further reducing graphs

The parameter η can be used for decreasing the memory consumption of graph cuts. The Figure 3 illustrates how far the test (8) can be relaxed for further reducing graphs while getting nearly the same segmentation. In this experiment, the segmentation as well as the reduced graph are shown for segmenting a 2D noisy image. Since the test (10) is easier to satisfy as η decreases, the reduced graph $\mathcal{G}'$ becomes thicker around the object contours.

4.2 Automatic tuning of η

Lower bound For a fixed window radius, notice first that the value of η must be large enough to not increase the number of components in the reduced graph $\mathcal{G}'$ (see Figure 2). Indeed, below some value (denoted by η_{min}), the reduced graph $\mathcal{G}'$ is split into multiple pieces in areas with high-curvature and the min-cut is no longer ensured of being fully embedded into $\mathcal{G}'$. This implies that some voxels could be wrongly labeled in the segmentation.

The Figure 1 illustrates a situation where η_{min} can be easily computed with an image consisting of two highly-contrasted areas. Using (10) with a square window of radius r and $\eta = 1$, the reduced graph $\mathcal{G}'$ corresponds to a thin band of size $2r$. An easy under-estimation of η_{min} is obtained by imposing that η_{min} permits to segment these two contrasted areas. In order to do so, we want the test (10) to be false for any pixel p at the boundary between these areas. For such a pixel, we have (assuming e.g. that $c(p) \geq +\delta_r$)

$$\sharp\{q \in B_p \mid c(q) \geq +\delta_r\} = (r+1)(2r+1)^{d-1}. \tag{11}$$

As a consequence, if

$$\eta \leq \frac{(r+1)(2r+1)^{d-1}}{(2r+1)^d}, \tag{12}$$

the node p does not belong to the reduced graph $\mathcal{G}'$. Since we want to avoid the situation, we must therefore have

$$\begin{aligned} \eta &> \tfrac{(r+1)(2r+1)^{d-1}}{(2r+1)^d} \\ &= 1 - \tfrac{r}{2r+1} = \eta_{min}. \end{aligned} \tag{13}$$

Remark that (13) does not depend on the dimensionality d of $\mathcal{P}$. By observing (13), it is straightforward to see that, as the window radius r tends to infinity, the proportion of nodes allowed to be connected to opposite terminals tends to $\frac{1}{2}$. In practice, we also observed that (13) is less accurate in connectivity 0 than in connectivity 1 (see Figure 2).

Upper bound For a fixed window radius r and a positive amount of noise ξ, one can observe in Figure 3 that there exists a value of the parameter η for which most of the nodes in noisy regions are removed from the graph $\mathcal{G}$, leading to a diminution of the size of the reduced graph $\mathcal{G}'$.

The purpose of this paragraph is to identify, from a statistical point of view, a reliable value of the parameter η for which all nodes of $\mathcal{P}$ are very likely to be removed from $\mathcal{G}$. For a fixed amount of noise ξ in the image I, we therefore want to find an upper bound on η by finding the maximum value of η in such a way that we control the proportion of nodes corresponding to noisy pixels in homogeneous areas.

Consider a noisy constant image I with a noise generated by a Bernoulli distribution of parameter $\xi \in]0,1[$, corresponding to the amount of noise in I [5]. The two cases where $\xi = 0$ and $\xi = 1$ are trivial and are not considered in our analysis. Assume now that the graph $\mathcal{G}$ is defined as in Section 3 where the nodes corresponding to noise-free pixels are connected to the sink t with a capacity $c(q) \leq -\delta_r$ and the nodes corresponding to noisy pixels have a capacity $c(q) > -\delta_r$.

First, let X be a discrete random variable counting degraded pixels in a square window B of size $n = (2r+1)^d$ in the image I. Then, the probability that at least k pixels are corrupted in B is given by

$$\mathbb{P}(X > k) = \sum_{i=k+1}^{n} \binom{n}{i} \xi^i (1-\xi)^{n-i}, \tag{14}$$

where $\binom{n}{i} = \frac{n!}{i!(n-i)!}$. For a fixed window radius r, it is straightforward to see that (14) is decreasing in k and tends to ξ^n if we impose that $\xi \in]0,1[$. According to the test (10) and the hypothesis on $\mathcal{G}$: a node $p \in \mathcal{P}$ can be removed from $\mathcal{G}$ if and only if

$$\sharp\{q \in B_p \mid c(q) \leq -\delta_r\} = \sharp\{q \in B_p \mid q \text{ is noise-free}\} \geq \eta n \tag{15}$$

Moreover, we assumed

$$\sharp\{q \in B_p \mid q \text{ is noise-free}\} \sim (n - X). \tag{16}$$

Therefore, we have

$$\mathbb{P}(p \text{ is not removed}) = \mathbb{P}((n-X) < \eta n) = \mathbb{P}(X > (1-\eta)n). \tag{17}$$

Fixing a proportion $\varepsilon \simeq 0$ of wrongly constructed nodes, we choose

$$\eta^+ = \max\{\eta \in [0,1] \mid \mathbb{P}(X > (1-\eta)n) \geq \varepsilon\}, \tag{18}$$

Considering the lower bound η_{min} defined in (13), we set

$$\eta_{max} = \max\{\eta_{min}, \eta^+\}. \tag{19}$$

Combining the definitions of the lower and upper bounds (see (13) and (19)), it now becomes easy to get an estimation of the parameter η^* for a fixed window radius by setting

$$\eta^* = \Big(\frac{\eta_{min} + \eta_{max}}{2}\Big). \tag{20}$$

[5] Histogram-based techniques can be for instance used to estimate ξ.

Let us now analyze the joint behavior of the lower and the upper bounds. When the amount of noise ξ is fixed, one can easily observe that the gap

$$\Delta\eta = (\eta_{max} - \eta_{min}), \tag{21}$$

grows as the window radius r increases. Indeed, we have previously seen that the lower bound η_{min} tends to $\frac{1}{2}$ as the window radius r increases (see (13)). The previous observation is also due to the fact that the upper bound η_{max} grows as the window radius r increases.

Similarly, when the window radius r is fixed, remark that (21) decreases when the amount of noise ξ increases. This situation is consistent because η_{min} remains the same but η_{max} tends to $\frac{1}{2}$ since it is more likely that the number of degraded pixels increase in the window B. Increasing the window radius r can compensate the augmentation of the amount of noise only up to $\xi = 0.5$. Finally, we empirically found that setting $\varepsilon = 0.05$ gives best estimate of η_{max} in (18).

4.3 Filtering

The parameter η can also serves to filter the segmentation. This behavior is illustrated in Figure 4 for segmenting a 3D noisy image acquired from a confocal microscope. White spots correspond to cell nuclei in a mouse cerebellum. Observe how far the filtering acts for small values of η: small regions in the reduced graph $\mathcal{G}'$ as well as in the segmentation are progressively removed as η decreases.

The robustness (see Figure 6 and 5) and sensitivity to noise of the parameter η are now (see Figure 7) are now analyzed. Let us describe the experimental setup. The experiment consists in segmenting four grayscale and five 2D color images in connectivity 1 with an increasing noise level ranging from 4 to 48%. For each image, we compute a reference segmentation on the noise-free image by placing the seeds by hand. We set $\beta = +\infty$ and automatically estimate the σ parameter as explained in [11]. Then, for each impulsive noise level, we select the segmentation maximizing the Dice Similarity Coefficient (DSC) between the reference image and all segmentations obtained through a fixed range of window radii and η values. We choose window radii r from 1 to 12 and eight linearly spaced values of η from η_{min} (w.r.t. r) to 1. Then, each segmentation is computed using the same seeds as those used for the computing the reference segmentation. Again, the σ parameter is automatically estimated as in [11].

As shown in Figure 6, for an impulsive noise level up to 45%, the parameter η appears to be reasonably robust with a DSC always greater than 94% for all images, except for the image "rice". However, such high and stable noise robustness can only be reached by increasing the amount of seeds (see Figure 6). The reason why the algorithm behaves poorly on the "rice" image is the following. As said earlier, r must be large enough when ξ increases for removing a maximum number of segments due to noise. This implies wider bands in $\mathcal{G}'$ around the object contours. However, the object contours further oscillate as ξ increases. Another reason is due to the proximity of the objects to segment. As an illustration, consider two circles over a uniform background, separated by

a distance $d_0 > 0$. We clearly see that the test (10) becomes more and more difficult to satisfy when the window radius r increases. When $(2r+1) \geq d_0$, the reduced graphs of both circles fuse into one component. This is typically the case for the image "rice" because it consists of small assembled rice grains near from each other. Finally, the Figure 7 also illustrates that the parameter η is not very sensitive to the variations of r and η. The DSC does not vary much with respect to noise, except for the image "rice". This exception can be explained for the same reasons as before.

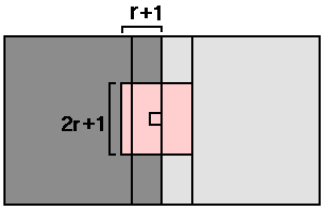
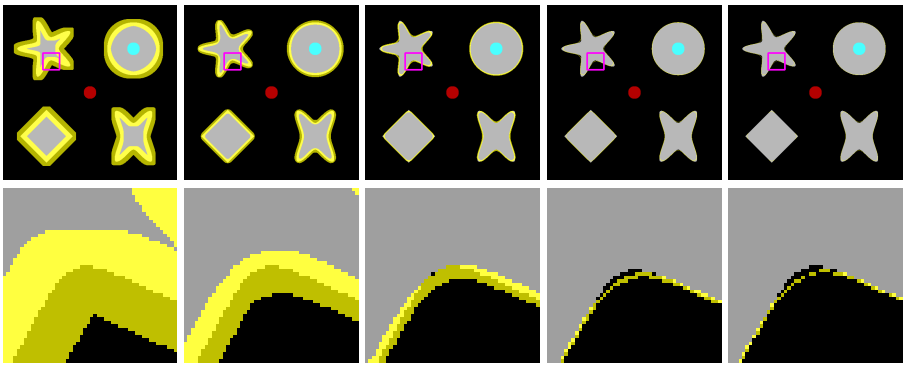


Fig. 1: Toy example for computing the lower bound η_{min}.

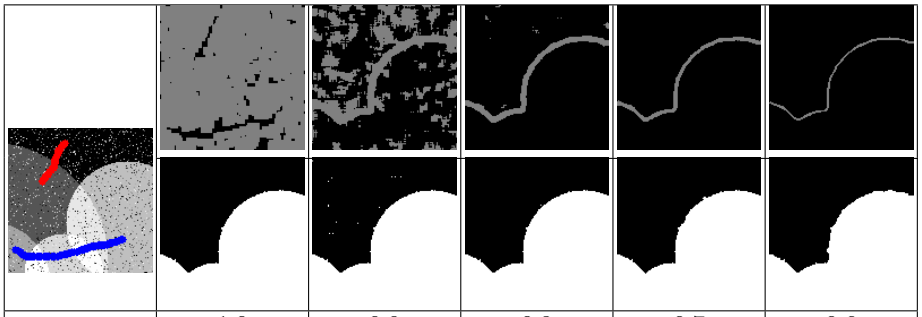


$\eta = 1.0$ $\eta = 0.8$ $\eta = 0.6$ $\eta = 0.53$ $\eta = 0.52$

Fig. 2: Illustration of the lower bound η_{min} for segmenting a 2D synthetic image. In this experiment, $\eta_{min} \simeq 0.523$ and we set $r = 10$ using connectivity 1. On all images, the nodes of $\mathcal{G}'$ are superimposed in yellow to the image by transparency. The bottom row correspond to a close-up of the box in purple color. Observe how the reduced graph $\mathcal{G}'$ splits into multiple pieces as soon as $\eta \leq \eta_{min}$.

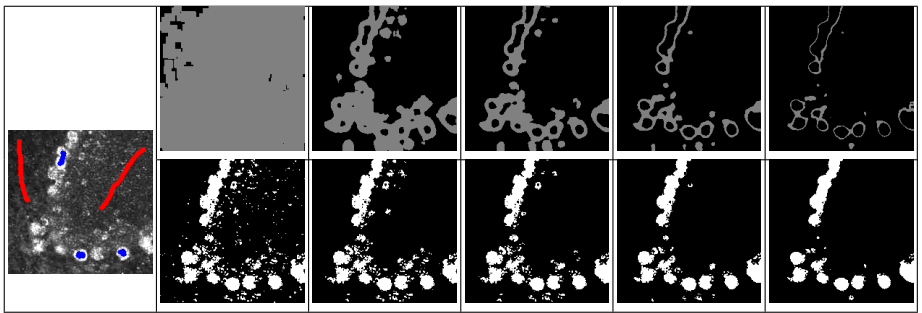
5 Conclusion

In this paper, we have presented a new parameter to embed in [8] for simultaneously reduce graphs and filter segmentations with the same computational complexity. We have also described an original manner to automatically tune it with the window radius parameter. In the proposed experiments, this new parameter generally appears to be less sensitive to noise but only for a limited amount, typically less than 50%. To overcome this situation, we could for instance inspect larger neighborhoods to further discriminate signal from noise.



η	1.0	0.9	0.8	0.7	0.6
$100 \times \sharp\mathcal{V}'/\sharp\mathcal{V}$	93.28%	30.99%	5.74%	3.65%	2.00%

Fig. 3: Memory gain when segmenting a 2D synthetic image corrupted by 10% of impulsive noise (left). Top row shows the nodes of the reduced graph in light gray while bottom row shows the corresponding segmentation. In this experiment, we set $r = 3$ and use connectivity 1.



η	1.0	0.9	0.8	0.7	0.6
$100 \times \sharp\mathcal{V}'/\sharp\mathcal{V}$	55.70%	37.15%	18.26%	12.65%	8.87%

Fig. 4: Simultaneous segmentation and filtering of a 3D noisy image (left). In this picture, the white spots correspond to cell nuclei in a mouse cerebellum. Top row shows the nodes of the reduced graph in light gray while bottom row shows the corresponding segmentation. In this experiment, we set $r = 5$ and use connectivity 1.

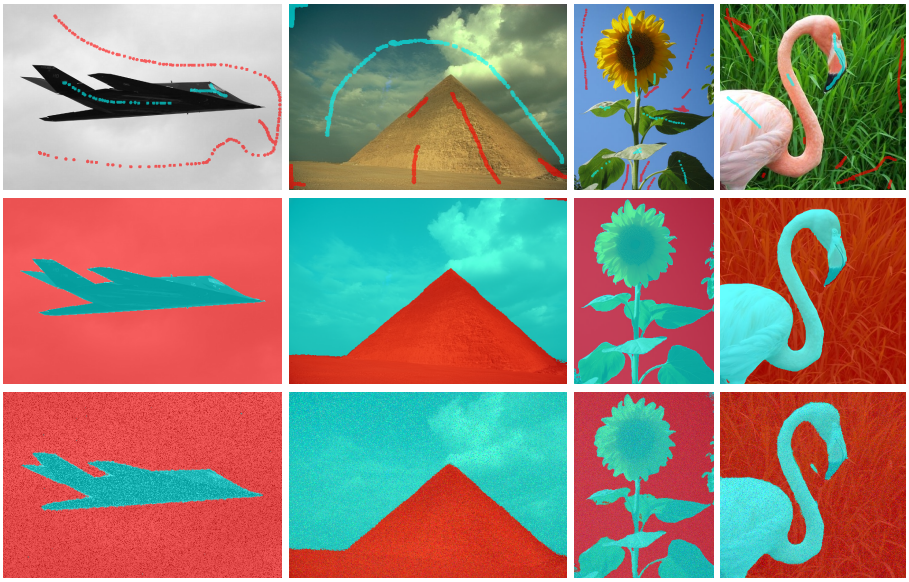


Fig. 5: Qualitative analysis of the robustness to noise for segmenting the images "f117" (left-most column), "pyramid" (left column), "sunflower" (right column) and "flamingo" (right-most column) in with a fixed impulsive noise level of 36%. Seeds and model parameters are the same than those used in Figure 6 (top row).

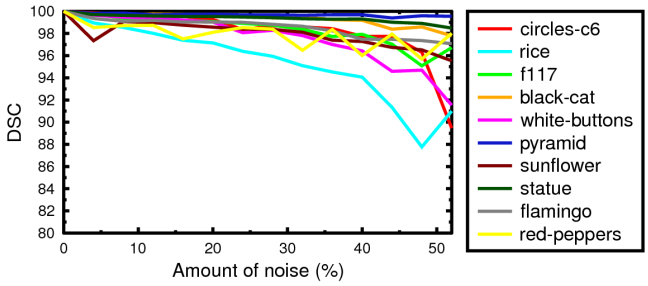


Fig. 6: Quantitative analysis of the robustness to noise for segmenting in connectivity 1 four 2D grayscale images (top-most curves in the list) and five 2D color images with an impulsive noise level ranging from 4 to 48%.

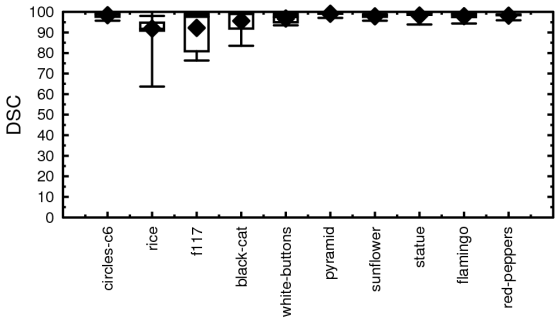


Fig. 7: Sensitivity of η for segmenting the images in Figure 6 with an impulsive noise level of 36%. The seeds and model parameters are the same than those used in Figure 6.

References

1. C. A. Bouman. Cluster: An unsupervised algorithm for modeling Gaussian mixtures. Available from http://www.ece.purdue.edu/~bouman, April 1997.
2. Y. Boykov and M-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, volume 1, pages 105–112, 2001.
3. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on PAMI*, 26(9):1124–1137, 2004.
4. A. Delong and Y. Boykov. A scalable graph-cut algorithm for N-D grids. In *CVPR*, pages 1–8, 2008.
5. P. Kohli, V. Lempitsky, and C. Rother. Uncertainty driven multi-scale energy optimization. In *DAGM*, pages 242–251, 2010.
6. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on PAMI*, 26(2):147–159, 2004.
7. V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *CVPR*, pages 1–8, 2007.
8. N. Lermé, F. Malgouyres, and L. Létocart. Reducing graphs in graph cut segmentation. In *ICIP*, pages 3045–3048, 2010.
9. Y. Li, J. Sun, CK. Tang, and HY. Shum. Lazy Snapping. *ACM Transactions on Graphics*, 23(3):303–308, 2004.
10. H. Lombaert, Y.Y. Sun, L. Grady, and C.Y. Xu. A multilevel banded graph cuts method for fast image segmentation. In *ICCV*, volume 1, pages 259–265, 2005.
11. C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, pages 309–314, 2004.
12. A.K. Sinop and L. Grady. Accurate banded graph cut segmentation of thin structures using laplacian pyramids. In *MICCAI*, volume 2, pages 896–903, 2006.
13. P. Strandmark and F. Kahl. Parallel and distributed graph cuts by dual decomposition. In *CVPR*, pages 2085–2092, 2010.