Improving Color Image Segmentation using the Gradient Network Method

R. Bertoldi[1], D. D. Abdala[2], A. v. Wangenheim[1], X. Jiang[2], M. M. Richter[3], L. Priese[4], F. Schmitt[4]

[1]Image Proc. and Computer Graphics Lab – LAPIX, Federal University of Santa Catarina, Florianópolis, Brazil {fogo, awangenh}@inf.ufsc.br

[2]CS Department, University of Münster, Münster, Germany {abdalad,xjiang}@math.uni-muenster.de

[3]CS Department. University of Calgary, Alberta, Canada mrichter@cpsc.ucalgary.ca

[4]CS Department, University of Koblenz, Koblenz, Germany {priese,fschmitt@uni-koblenz.de

Abstract

The objective of this paper is to evaluate a new combined approach intended for reliable color image segmentation, in particular images presenting color structures with strong but continuous color or luminosity changes, such as commonly found in outdoors scenes. The approach combines the Gradient Network Method with common region-growing approaches used as pre-segmentation steps. The GNM is an efficient post-segmentation procedure based on graph analysis of global color and luminosity gradients altogether with a segmentation algorithm to produce a fast and reliable segmentation result. The approach was automatically evaluated using a close/open world approach. Different region-growing segmentation methods, namely CSC, Watershed, EDISON and Mumford & Shah with and without the GNM post-processing were compared against ground truth images using Rand and Bipartite Graph Matching indexes. These results were also confronted with other well established segmentation methods (RHSEG, JSEG and Blobworld). We further conclude that the CSC-GNM approach shows a good compromise in dealing with outdoors images.

Keywords: color image segmentation, region-growing, outdoor scenes, Gradient Network Method

1 Introduction

Natural color scenes, such as outdoors images composed by many colored objects that are acquired under uncontrolled conditions can show complex illumination patterns across one same object in the picture. Examples are variations in lightness and specular effects. State-of-the-art region-growing segmentation methods [1, 2, 3] present two main features that limit their applicability for dealing efficiently with natural scenes:

(a) *A static region similarity concept*, where pixels or textures within a region are expected to be homogeneous. Typical natural scenes, however, show strong continuous variations of color, presenting a different, dynamic order that is not taken into account by such algorithms. They divide a sky region with different intensities of blue into segments or will represent an irregularly illuminated surface as a set of different regions. When the parameters of such algorithms are stressed in order to try to accomplish a correct segmentation of a large object showing a long continuous gradient of color, typically with a gradual but large color variation, a region leakage into other objects in the image is likely to occur. Then the algorithm will become unstable and even inapplicable;

(b) *Increase in complexity to present more stable results*, that usually demands complex computations to detect segment-correlation clues, or that are built upon additional texture information. This slows down considerably the processing time without being much more stable when extreme color variations are present.

A number of segmentation approaches already have tried to deal with one or both problems: [1] proposes a hierarchical model to improve the segmentation of color scenes, and [2, 4, 5] address the problem by the additional analysis of object textures. Some authors have tried to overcome the problem using different methods of chromatic and luminescence evaluation [6, 7]. Finally, in [8, 9, 10] physical illumination models were proposed as basis for the formulation of segmentation algorithms.

The Gradient Network Method - GNM [11], intends to address the problem of luminance and reflection variations by searching for structured gradients along the various neighboring segments. It was devised to work together with a fast segmentation algorithm, which provides a pre-segmentation as a starting point. The general approach is to rely on fast segmentation algorithms such as CSC [12] or any other producing as output an over-segmentation with edge preservation and subsequently apply GNM, who will iteratively merge segments logically connected through organized gradient patterns improving the final segmentation.

2 Objectives

In this paper we are interested in evaluating the criteria to specify an optimal pre-segmentation algorithm to work with the GNM post-segmentation procedure and to compare the achieved results with other approaches. For this purpose we devised the following procedure, wich included an evaluation using a close/open world approach:

(a) First, different first step well-known segmentation methods to be used as pre-segmentation procedures were chosen: CSC [1], EDISON [13], Watershed [14] and Mumford & Shah [3].

(b) Full segmentations performed with each of these algorithms were compared against ground-truth images using Rand [15] and Bipartite Graph Matching (BGM) [16] indexes. For each segmentation algorithm we selected a wide range of segmentation parameters, selected the segmentation considered to be the best one for every pair of image set and segmentation algorithm using visual inspection and generated Rand and BGM scores for the complete set of ground-truths for each image

(c) These same segmentation algorithms where also selected to generate over-segmented images to be used in combination with the GNM post-processing algorithm. For each algorithm we selected a set of segmentation parameters that produced oversegmented images where no segment leakage with respect to any ground truth was allowed. Each of these results was used as an input for the GNM algorithm, which was also run with a set of different parameters. The resulting segmentations after post-processing with the GNM also underwent a selection of the segmentation considered to be the best one based on visual inspection . This result was also compared against the ground-truth images using Rand and BGM indexes.

(d) We compared these results to three other well-established segmentation methods: RHSEG [17], JSEG [2] and Blobworld [18] also using the ground-truth images and the Rand and BGM indexes.

The structure of this paper is as follows: In section 3 we review briefly the GNM main concepts. Section 4 describes the experiment and further discusses the objective evaluation of segmentation results compared against ground-truths. Section 5 reviews the evaluation indexes used, discussing the main characteristics and drawbacks of each of them. Finally, section 6 shows and discusses the results of our close/open world evaluation. All data and results are accessible in detail under http://www.lapix.ufsc.br/gnm.

3 The Gradient Network Method

The Gradient Network Method previously described in [11] and closely related to [19] was developed to deal with segmentation problems where objects in the scene will be represented by several different but similar and gradually varying color shades, as they often are found in outdoors scenes. The GNM looks for a higher degree of organization in the structure of the scene through search and identification of continuous and smooth color gradients.

For the GNM to be able to process an image and identify these characteristics, a pre-segmentation of the image must be performed. The goal is to obtain groups of segments with a high degree of similarity represented in a simple way, avoiding possible problems with local noise induced by high granularity (e.g. at pixel level). The clusters in eq. I are sets of pairs of coordinates in the Ω space of R^n values (in the case of our tests, inside the R^2 image spaces) and color ranges in the R^m, similar according to the segmentation approach used to realized the rough verification of the image and with dimensionality *m* determined by the color space selected (GNM currently uses HSI color space, so R^3 is its color range dimensionality).

$$C = \{(x_1, r_2),...,(x_i, r_j) \mid x_1,...,x_i \in \Omega \wedge \Omega \subseteq R^n \wedge r_1,...r_j \in R^m\} \quad \text{I}$$

Every cluster $C = \{c_1, \ldots, c_n\}$ is a discrete clustering meaning that each pixel is unequivocally assigned to a specific cluster, such that $X_k \cap X_l = 0$ and $U^n_{i=1} X_i = \Omega$ for the every $X_k = \{ x_1,...,x_p \subset C_1 \}$ and $X_l = \{ x_1,...,x_q \subset C_2\}$, $k \neq l$. Every cluster will also receive an unique label in a way that for any $L(c_k)$ and $L(c_l)$, $k \neq l$, are always different.

Given the clusters produced by the first segmentation step, GNM creates a graph G(V, E) to structure this data, providing a simple way to describe the topology of these clusters of similarity. The graph is constructed from these clusters where vertices $V=\{v_1,...v_n\}$ represent the roughly similar groups of pixels (see equation eq. II) and edges $E=\{e_1,...,e_m\}$ represent the existence of a neighborhood relationship between the given vertices. Each edge also is labeled by the existing gradient between the two vertices he connects, as described in equation eq. III.

$$V = \{v_i \in V \mid v_i = c_j \wedge c_j \in C\} \quad \text{II}$$

$$E = \{e_i \in E \mid e_i = (v_k, v_m, \nabla) \wedge v_k, v_m \in V \wedge \nabla \in R^+\} \quad \text{III}$$

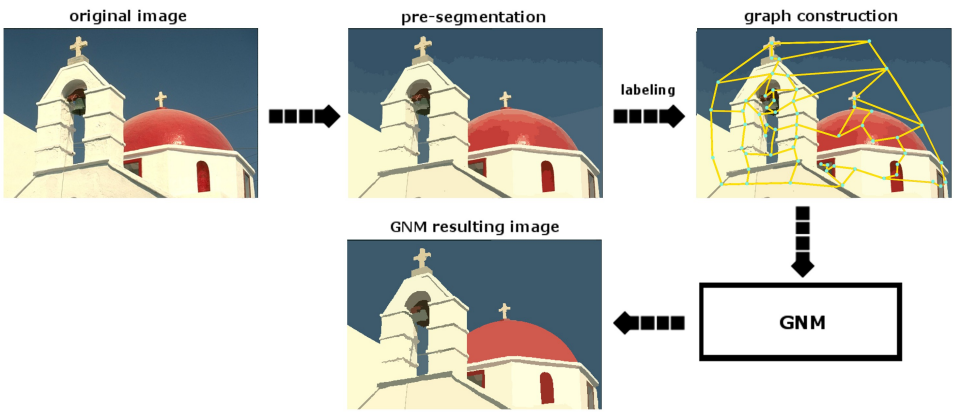

Figure 1. The GNM workflow, showing the steps to partition an image into similar objects using the GNM algorithm.

The objective of the algorithm is to minimize ||G|| (vertices cardinality) by a merging procedure that searches subgraphs $S_i \subseteq G$ which presents $\min(\sum_{k}^{N} \nabla_k) < \Phi$ where N is the edges' cardinality. GNM concentrates on regions of high similarity, specifically in the aspect of low color variation. So, after creating the graph G(V, E), the next step is checking all the neighborhood relations if they comply to the similarity measure and provide continuous and smooth color gradients.

The evaluation of the continuity of the gradients along the paths found in the graph is performed by a function f_p that takes into account the perception variations, a concept similar to that found in [20]. This allows a better evaluation of the similarity among regions in different conditions of illumination. This is important because most long continuous gradients are the result of the presence of lighting effects in the scene of an image. With this additional feature, the algorithm becomes more robust in presence of such characteristics. Therefore, even when the neighborhood contains darker or more illuminated regions, it will search for the best possible gradient path in the graph [11]. An improvement of the implementation used in our tests over the algorithm described in [11] is that now all $e \in E$ will be ordered crescently by the resulting values from the perception function and then evaluated by the chosen similarity measure. Regions found acceptably similar will be grouped in meta-regions.

To generate the resulting subgraph, every vertex in the graph G(V, E) is correlated to a meta-region. Meta-regions M are logical containers that will be used to represent the union among different vertices (eq. IV), as they are found necessary along the evaluation of the edges using the f_p function. Initially, each meta-region $m \in M$ will have a one-to-one relationship with a vertex $v \in V$.

$$M = \{m_i \in M \mid \bigcup (v_j \in V) \wedge \forall j \neq k, v_j \neq v_k\} \qquad \text{IV}$$

The resulting meta-regions at the end of the process will be the partitioned image output produced by the GNM segmentation. The total complexity for all these steps is $O(n(1+n) + m(1+\log m))$, where n is the number of vertices and m the number of edges.

Another feature was added to the GNM implementation used here, improving its segmentation results: After the segmentation is performed, a post-processing step is executed to remove

regions that are considered insignificant to be represented as partitions of the image. The process of running through this set eliminating unnecessary regions has a complexity of $O(k^2)$, where k is the number of meta-regions that should be checked by this step (the set can be built along the normal process of the algorithm). Every meta-region in this set is iterated, looking if this meta-region should be eliminated. The criterion for this elimination is if the area of a region is considered to be too small. This region is merged with the neighbor meta-region which is less affected by the union with it, considering the meta-region average color and area.

It is important to note, though it cannot be accounted in the GNM complexity, that the chosen algorithm for the pre-segmentation has an effect on the total time of processing. It can also affect the image segmentation results, depending on how properly the pre-segmentation performs the role that GNM expects from it. Therefore, a proper technique must be selected according to the application necessity, in a way that can ally quality and velocity performance as desired.

4 Experiment

In order to be able to objectively achieve the task we have proposed ourselves of choosing the optimal pre-segmentation algorithm to be used in conjunction with the GNM approach and then, to chose the optimal segmentation parameters for each algorithm, we accepted that ground-truths, or hand-made segmentations, representing the judgment of an human observer, should play the role of golden standards. The process of evaluation of image segmentation results was investigated by a number of researchers [21, 22, 23].

We adopted the Berkeley ground-truth image dataset [24]. This dataset is well known in the image processing community and each sample image contains various ground-truths generated by different subjects. The evaluation schema adopted in this paper works over a subset of 16 images selected from this dataset which present strong gradients in the form of continuous color variations in the same object.

In order to objectively evaluate the quality results of GNM we have envisioned an evaluation procedure. The first step was the generation of a series of segmentation results for CSC, Mumford & Shah (MS), Watershed (WS), RHSEG, JSEG, Blobworld and EDISON algorithms added of the tests with GNM combined with pre-segmentations generated with CSC, Mumford & Shah, EDISON and Watershed. Our purpose is to compare the GNM results with those obtained by several state-of-art algorithms, looking to validate our algorithm as a technique as capable as those selected for our tests.

Regarding the tests performed with the GNM, all pre-segmentations are obtained from a set of different parameter ranges. Though the parameters are specific to each pre-segmentation, the goal is only one: achieve an over-segmented image avoiding leakages and preserving smooth gradients to suit and allow a better segmentation by GNM. The parameter used for CSC was *threshold* = 30; Mumford & Shah images were generated with *lambda* = 600; EDISON was generated with a *SS* = 3 and *SR* = 8; Watershed images used *conductance* = 2.0, *number of interactions* = 10 and *threshold* = 0.01. Over each of the pre-segmented images produced by these algorithms was performed a series of GNM segmentations. Two parameters only were varied along all these tests, *t_{cp}* and *t_{rp}*, both in the [0.01, 0.07] range, with an offset of 0.005. All other GNM parameters were kept with the standard values described in [11].

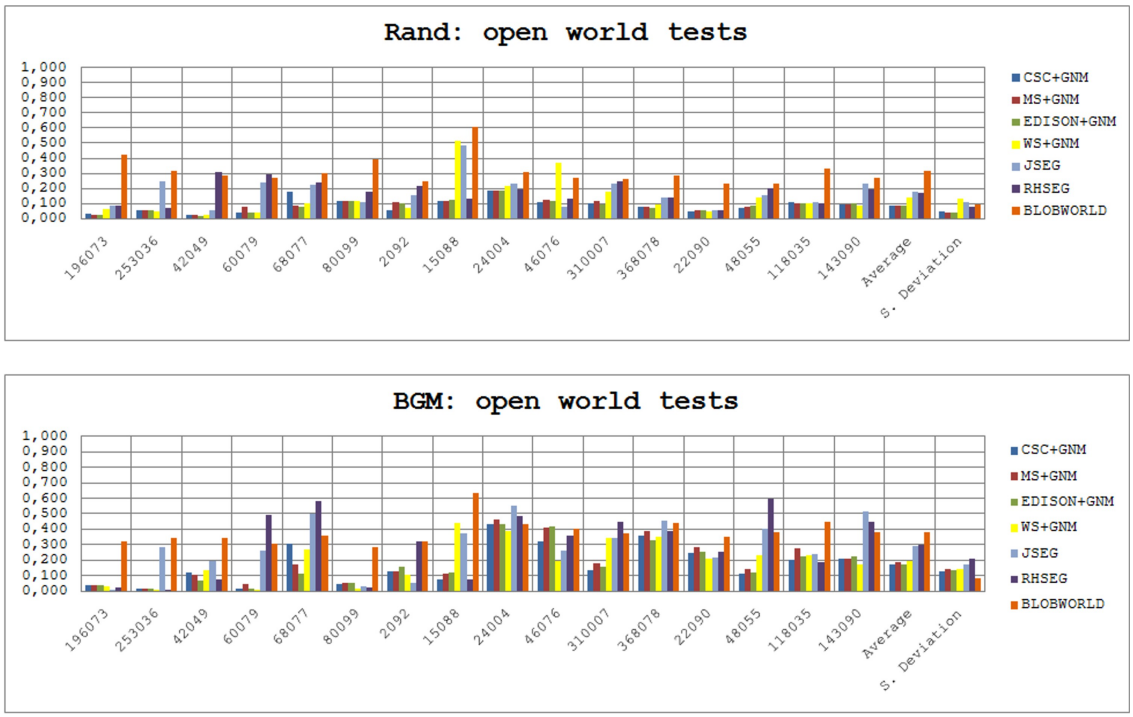

Figure 2. Graphs showing the average values obtained for all selected image sets with every tested algorithm. The two columns most to the right represent, respectively, average and standard deviation values for the algorithms.

A series of segmentation results was also produced using those segmentation techniques alone and used to be evaluated against the combined GNM approach. The parameter ranges and increment steps used for these segmentation methods were the following: 1) CSC: $20 \leq threshold \leq 100$, $step_{threshold} = 10$; 2) Mumford & Shah: $1000 \leq lambda \leq 15000$, $step_{lambda} = 500$; 3) EDISON:,$3 \leq SS \leq 30$, $step_{SS} = 1$, $SR = 8$; 4) Watershed: $0 \leq threshold \leq 0.5$, $step_{threshold} = 0.05$. Blobworld and JSEG are unsupervised techniques and do not require parameters. RHSEG, however, is supervisioned but, as it is intrinsically a hierarchical segmentation, was not necessary to be tested over a range of values. All RHSEG segmentation results were created with the similarity "entropy" (segmentation parameter number 9 in the `params` file), with a *factor of convergence* equal to 1.75 and with a 0.1 importance to *spectral clustering*.

After all the image segmentations were produced by these techniques with the parameters listed, we adopted visual inspection to select the segmentation considered the best one for every pair of image set and segmentation algorithm. The selected results were those that seemed to better partition the images into relevant regions. This human-based selection is supposed to be a rough filtering of the segmentations and the group of image segmentations selected will be next evaluated by distance measures able to quantify the quality of image segmentations. The explanation about how this evaluation and which measures were used follows in the next chapter.

5 Evaluation

To better outline the quality of the results obtained with the different pre-segmentation + GNM combinations, the image segmentations were objectively evaluated with distance measures properly developed for this task. The goal of these measures is to compare an image segmentation to a given ground-truth image and quantify the quality of the object identification.

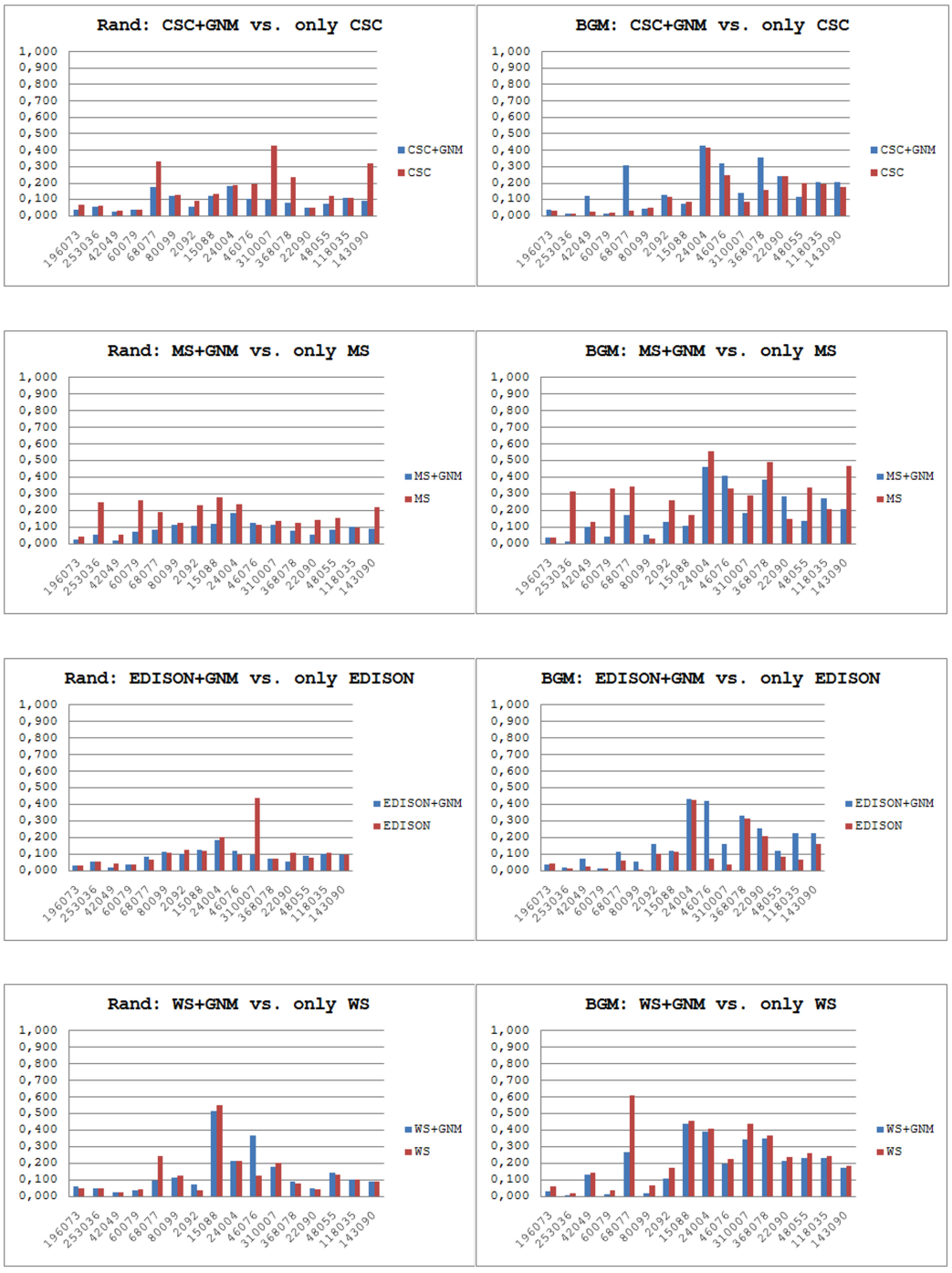

Figure 3. Comparison between the Rand and BGM average values for the GNM combined with several segmentations against these same selected segmentations used on their own:

The aim is to achieve the closest segmentation to the expected regions defined by the ground-truth. There are several approaches to these distance measures. According to [25], there are distances that evaluate through *counting of pairs* and *set matching*. In our tests, we used one measure of each kind: Rand [15] and Bipartite graph matching (BGM) [16], respectively a pairs-counting and a set-matching measure.

The ground-truth images provided by the Berkeley image dataset were tested with both distance measures as the quality standard for the GNM segmentations. Both Rand and BGM produce values between 0 and 1, 0 meaning a perfect match between segmentations and 1 meaning no

relation at all between them. The fact that both measures are in the same range facilitates comparing algorithms with different approaches and features. Details about these two evaluation measures can be found in the appendix A.

The GNM-combined segmentation results obtained a better average score with the Rand measure (see Figure 2.a) except for the WS+GNM, which showed comparable values. The mean Rand score for CSC+GMM was 0.087, MS+GNM was 0.089, EDISON+GNM was 0.085 and WS+GNM was 0.137. The next best result after the GNM results for Rand was EDISON with an average of 0.111, while the worst average was Blobworld's, 0.313. The GNM tests also averaged lesser values of standard deviation, all around 0.04, except for the WS+GNM that had a standard deviation value of 0.132. The smallest standard deviation after GNM was obtained with Mumford & Shah, being of 0.074.

The mean BGM scores for the GNM–combined segmentation results were not the best results, but were similar to the best BGM score (see Figure 2.b). The best BGM score was obtained with EDISON, that averaged 0.108, while the worst one was Blobworld with 0.383. The GNM results got the best scores after EDISON and CSC, which scored 0.129. BGM scores obtained with the GNM tests were 0.171 for CSC+GNM and EDISON+GNM, 0.187 for MS+GNM and 0.196 for WS+GNM. The standard deviation for all GNM segmentations were similar, around the 0.130 to 0.140 range. The smallest standard deviation was 0.082 for Blobworld, which had the worst average score. The next best standard deviation value were 0.111 and 0.117 for, respectively, CSC and EDISON. Considering both these segmentations got also the best average scores, in terms of the BGM evaluation, these are very stable and efficient segmentations.

Comparing the GNM–combined segmentation results with CSC, Mumford & Shah, EDISON and WS when run alone (Figure 3), GNM shows a sensible improvement in Rand scores for all techniques, except for WS, where the run-alone score of 0.130 was slightly better than the one obtained in combination with GNM. Still regarding Rand scores, MS and CSC showed the highest improvement (respectively, 0.165 versus 0.089 and 0.157 versus 0.087). The BGM scores showed a split, with CSC (0.129 versus 0.171) and EDISON (0.108 versus 0.171) showing better results when run alone while MS (0.277 versus 0.187) and WS (0.245 versus 0.196) were improved by a combination with GNM. As before, GNM shows a better performance when evaluated in terms of Rand conditions than BGM, though the BGM results are still better than several of the other tested segmentations.

Considering the approaches underlying each of these measures, GNM shows better results when evaluated in terms of region-pairing with the tested ground-truth images than when measured in terms of graph nodes matching. The BGM evaluation index rewards more aggressive approaches, even if the results are somewhat under-segmented, showing segment leakages. Rand scores better values for the segmentations that preserve details better. GNM proved to be a reliable technique by both measures, especially by Rand, and for both measures the results presented some of the best values in terms of deviation, also showing that it is a stable technique. The other algorithm that is very reliable by both measures was EDISON. Blobworld presented the smallest level of similarity measured. It was expected, given its inherent characteristics, where the focus is directed to capturing objects and providing a delimitation for them, and not the identification of illumination and slow progressive gradients.

The average execution time of the implementation of our GNM implementation was around 0.6 seconds. Some of the other segmentations tested averaged the following times: CSC average

time was around 0.085 seconds; MS tests averaged 5 seconds; JSEG average was around 12 seconds; and EDISON average was a little more than 20 seconds. All tests were performed on the same computer (AMD Athlon 64, 2.2 GHz and 512MB RAM memory) and all images selected from Berkely dataset have the same number of pixels. This comparison is simple and

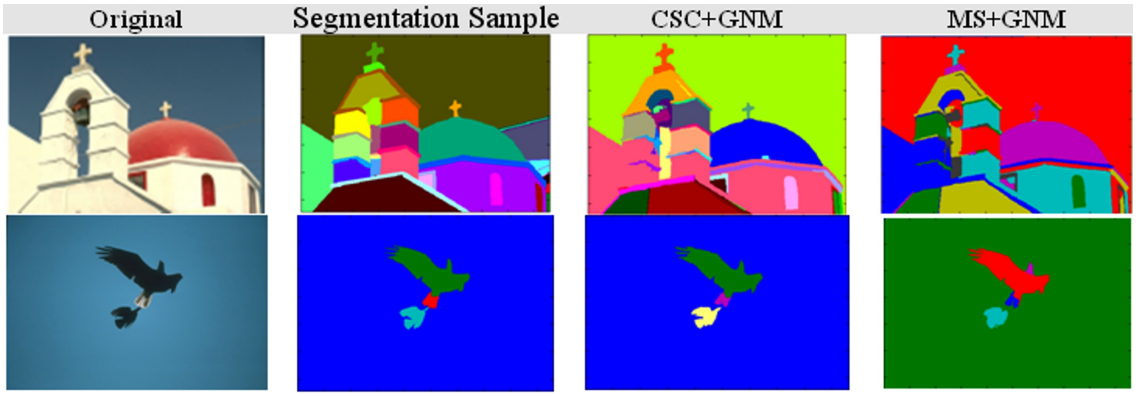

Figure 4. Examples of the image processing results obtained by the GNM, when combined with either CSC or Mumford-Shah. The first row corresponds to image number 118035 and the second row corresponds to image 135069 from the Berkeley image dataset. The segmentation sample image is also provided by the Berkeley dataset. More image sets, results comparing several techniques and higher resolution images can be found in www.lapix.ufsc.br/gnm. We also refer to [11] to view some image comparisons.

can't be viewed as definite benchmark, but indicate that the GNM can be a reasonably fast solution, as fast as CSC, for example, while producing also segmentation results of quality comparable to the most robust state-of-art techniques.

As the tests we realized resulted in more data than we would be able to properly display in this paper, we have made all the data tests available in a website. All the resulting data from the evaluation for all image sets is accessible at http://www.lapix.ufsc.br/gnm. There the graphs here displayed are also present, along individual graphs for every image set, tables showing all scores obtained and the images tested and the parameters used to obtain them.

6 Results and Discussion

We have shown empirically and through well known validation measures that the quality of the segmentations generated by our two-step approach is very promising and comparable to segmentations generated by state-of-the-art segmentation methods that were available for comparison when this paper was being written. The Gradient Network Method is a segmentation post-processing method that is independent of the region-growing method applied to generate the super-segmented input image, as shown by the comparison between the results produced by the four different segmentation methods used as the pre-processing step.

It is important to note that in any evaluation measure that intends to point the “best” possible result in such a complex field, as actually is color image segmentation, the values provided have to be taken with care. These measures may not provide an absolute way to determine which is actually the best segmentation, but they provide an interesting objective resource where you can compare and evaluate different segmentations under the optics of the specific approach the measure takes. Both selected measures, Rand and BGM, apply very distinct approaches to what is a better segmentation and, as it is tough to define how correct a segmentation is, it is hard to

argue which one proves to be more correct. What these measures actually provide are ways of finding if under expected conditions, reflected by the evaluation approach, a segmentation is able to perform at a desired level of quality. Observing the results presented here, we demonstrated that GNM, under the conditions of both selected measures, can obtain segmentation of at least similar quality as several state-of-art algorithms.

The processing time, however, is extremely shorter when a rapid approach like the CSC, which was originally developed for real-time color segmentation, is used, while also showing good results in terms of the validation scores. This shows that the processing step with the Gradient Network Method allows us to rely on very fast pre-segmentation methods that reduce the total processing time while producing end-segmentations of good quality. Further improvements could still be achieved in terms of efficiency with the use of a graphics processing unit for performing the necessary computations of the involved algorithms. This kind of technology, referred as General-Purpose Computing on Graphics Processing Units (GPGPU), would fulfill the requirements for real time applications. This could make the combination of CSC and GNM a feasible solution that deal with outdoor scenes, as robotics or traffic monitoring applications.

7. Acknowledgments

Daniel D. Abdala thanks the CNPq-Brazil for granting him a Ph.D scholarship under the process number 290101/2006-9.

8. References

[1] V. Rehrmann and L. Priese. Fast and robust segmentation of natural color scenes. In ACCV, pages 598–606, 1998.

[2] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color texture regions in images and video. In Transactions on Pattern Analysis and Machine Inteligence, volume 23, pages 800–810. IEEE, 2001.

[3] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. In Commun. Pure Applied Mathematics, volume 42, pages 577–684, 1989.

[4] A. Dupuis and P. Vasseur. Image segmentation by cue selection and integration. Image and Video Computing, 24(10):1053–1064, 2006.

[5] Z. Kato and T. C. ong. A markov random field image segmentation model for color textured images. Image and Vision Computing, 24(10):1103–1114, 2006.

[6] R. D. Dony and S. B. Wesolkowski. Edge detection on color image using rgb vector angles. In Proceedings of the 1999 IEEE Canadian Conference on Electrical and Comuter Engineeringg, volume 9, pages 687–692, Alberta-Canada, May 1999.

[7] M. K. Schneider, P. W. Fieguth, W. C. Karl, and A. S. Willsky. Multiscale methods for the segmentation and reconstruction of signal and images. In Transactions on Image Processing, volume 9, pages 456–468. IEEE, 2000.

[8] Gudrun Klinker, Steven A. Shafer, and Takeo Kanade. A physical approach to color image understanding. International Journal of Computer Vision, 4:7–38, 1990.

[9] P.W.M. Tsang and W.H. Tsang. Edge detection on object color. In IEEE International Conference on Image Processing, volume C, pages 1049–1052, 1996.

[10] G. Healey. Segmenting images using normalized color. IEEE Transactions on Systems, Man and Cybernetics, SMC-22(1):64–73, January-February 1992.

[11] A. Wangenheim, R. Bertoldi, D. D. Abdala, and M. M. Richter. Color image segmentation guided by a color gradient network. Pattern Recognition Letters, 28(13):1795–1803, October 2007.

[12] L. Priese and P. Sturm. Introduction to the color structure code and its implementation, March 2003.

[13] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):603–619, 2002.

[14] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. In Transactions on Pattern Analysis and Machine Inteligence, volume 9, pages 735–744. IEEE, 1991.

[15] W. M. Rand. Objective criteria for the evaluation of clustering methods. Journal of American Statistical Association, 66(846-850), 1971.

[16] Xiaoyi Jiang, Cyril Marti, Christophe Irniger, and Horst Bunke. Distance measures for image segmentation evaluation. EURASIP Journal on Applied Signal Processing, pages 1–10, 2006.

[17] J.C. Tilton. D-dimensional formulation and implementation of recursive hierarchical segmentation. Disclosure of Invention and New Technology: NASA Case No. GSC 15199-1, May 2006.

[18] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: image segmentation using expectation-maximization and its application to image querying. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(8):1026–1038, August 2002.

[19] A. Tremeau and P. Colantoni. Regions adjacency graph applied to color image segmentation. In Transactions on Image Processing, volume 9, pages 735–744. IEEE, 2000.

[20] K. Huang, Q. Wang, and Z. Wu. Natural color image enhancement and evaluation algorithm based on human visual system. Computer Vision and Image Understanding, 103(1):52–63, 2006.

[21] Ranjith Unnikrishnan, Caroline Pantofaru, and Martial Herbert. Toward objective evaluation of image segmentation algorithms. In Transactions on Pattern Analysis and Machine Intelligence, volume 29, pages 929–943. IEEE, 2007.

[22] Nivedita Sahasrabundhe, John E. West, Raghu Machiraju, and Mark Janus. Structured spatial domain image and comparison metrics. In Proceedings of the Conference on VIsualization, pages 97–104, 1999.

[23] Vito Di Gesu and Valery Starovoitov. Distance-based functions for image comparison. Pattern Recognition Letters, 20(2):207–214, 1999.

[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings of 8th International Conference on Computer Vision, volume 2, pages 416–423, 2001.

[25] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang. Color image segmentation: advances and prospects. Pattern Recognition, 34:2259–2281, 2001.

Appendix

A Distance Measures

A.1 Rand Index

The *Rand index* first reported in [15] and reviewed in [16] is a similarity measure specially developed to evaluate the quality of clustering algorithms by comparison with other clustering results or with a golden standard (in our case, ground-truths). To compare two clustering results $C_1=\{c1_1, c1_2, ..., c1_N\}$ and C2= $\{c2_1, c2_2,..., c2_M\}$ over the same image $P = \{p_1, p_2, ..., p_K\}$ where each element of C1 or C2 is a subset of P and $c1_j = \{p1_j, p2_j, ..., pL_j\}$, the following quantities are calculated:

- N_{11} - the number of pixels in the same cluster in both C1 and C2.
- N_{00} - the number of pixels in different clusters both in C1 and C2.

The rand index is so defined by eq. A.I

$$R(C1, C2) = 1 - \frac{N_{11} + N_{00}}{\frac{n(n-1)}{2}} \qquad \text{A.I}$$

To compute the quantities N_{11} and N_{00} one must iterate over the entire image for each pixel in order to evaluate the conditions defined above given an $O(n^4)$ algorithm. A clever approach is to use the method described in [16] where a matching matrix is used to summarize the occurrences of pixels in the respective classes. The matching matrix is constructed allocating each cluster from the clustering C1 to a row and each cluster from clustering C2 to a column. The matrix cells are then defined as the intersection of the clusters specifying each row and column. If the matching matrix has kxl size each cell can be defined as $m_{ij} = |c_i \cap c_j|$, $c_i \in C1$, $c_j \in C2$.

The quantities N_{11} and N_{00} can be computed in terms of the matching matrix as follows:

$$N_{11} = \frac{1}{2}\left(\sum_{i=1}^{k} \sum_{j=1}^{l} m_{ij}^2 - n \right) \qquad \text{A.II}$$

$$N_{00} = \frac{1}{2}\left(n^2 - \sum_{i=1}^{k} n_i^2 - \sum_{j=1}^{l} n_j^2 + \sum_{i=1}^{k} \sum_{j=1}^{l} m_{ij}^2 \right) \qquad \text{A.III}$$

where *n* is the cardinality of P and n_i and n_j are the cardinality of the clusters $c1_i$ and $c2_j$.

A.2 Bipartite Graph Matching

The *BGM index* [16] compute an one-to-one correlation between clusters at the same time trying to maximize the relationship. It considers each cluster of the C1 and C2 clustering as vertices of a bipartite graph. Edges are added between each vertex of the two partitions and they are valued as $|c1_i \cap c2_j|$, a value that can be directly extract from the matching matrix. Then the maximum-weight bipartite graph is defined as the subgraph $\{(c1_{i1}\},c2_{j1}), ..., (c1_{ir},c2_{jr})\}$ where

only the edges from $c1_i$ to $c2_j$ with maximum weight are present. After all max-valued edges were found the overall graph weight is calculated by sum of all remaining edge weights.

$$BGM(C1, C2) = 1 - \frac{w}{n} \qquad \text{A.VI}$$