Improving Image Segmentation for Boosting Image Annotation with Irregular Pyramids

Annette Morales-González[1], Edel García-Reyes[1], and Luis Enrique Sucar[2]

[1] Advanced Technologies Application Center. 7a # 21812 b/ 218 and 222, Rpto. Siboney, Playa, P.C. 12200, La Habana, Cuba
{amorales,egarcia}@cenatav.co.cu
[2] Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, Mexico
esucar@ccc.inaoep.mx

Abstract. Image Segmentation and Automatic Image Annotation are two research fields usually addressed independently. Treating these problems simultaneously and taking advantage of each other's information may improve their individual results. In this work our ultimate goal is image annotation, which we perform using the hierarchical structure of irregular pyramids. We propose a new criterion to create new segmentation levels in the pyramid using low-level cues and semantic information coming from the annotation step. Later, we use the improved segmentation to obtain better annotation results in an iterative way across the hierarchy. We perform experiments in a subset of the Corel dataset, showing the relevance of combining both processes to improve the results of the final annotation.

Keywords: image annotation, image segmentation, irregular pyramids.

1 Introduction

Automatic image segmentation and annotation are two prominent fields in Computer Vision, that are usually addressed individually, disregarding the benefits they can provide to each other. Image segmentation based only on low-level cues (without prior knowledge of the object being segmented) is insufficient to delineate objects due to the semantic gap. Image segmentation presupposes an abstraction process of low-level features and when it is not guided by a semantic interpretation of the segments, the resulting partition is dependant on the defined mathematical equivalence relation. Also, Automatic Image Annotation (AIA) will not provide good results if the underlying segmentation is not correct (i.e. instances of different classes are merged together in a region, boundaries of objects are lost).

In the literature, some works have addressed these two problems together. In [1] the author proposes an object recognition scheme that involves a hierarchy (tree) of class-specific object parts (fragments). He combines a recognition process with a top-down segmentation, where the latter process takes advantage of the classification information, but not the other way around. In [2] they propose

J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.): CIARP 2013, Part I, LNCS 8258, pp. 399–406, 2013.

to use a hierarchy of segmentations to guide a selective search for improving object classification results, but the segmentation is based only on low level cues. The same happens in [3] and [4], where a hierarchy of segmentations is used for object detection and image annotation respectively, but semantic information is not used in the segmentation process. In [5] they combine four segmentation algorithms to obtain an enhanced partition, and they refine classification in this partition by using classification information from the initial partitions. Yet, there is no contribution of the semantic information to improve the segmentation. In [6] they perform detection and segmentation simultaneously, allowing cross information between these processes, but they need ground truth segmentations at training stage and the proposal is intended for detecting/segmenting specific objects in the images.

In this work we use an idea similar to the one presented in [4] to perform AIA using irregular pyramids [7], but we propose an iterative process where the segmentation hierarchy is rebuilt and improved using the classification information obtained from the annotation process in each level of the pyramid. Our ultimate goal is to improve the results of image annotation. We show in the experiments performed on the CorelA dataset how much the synergy between segmentation and annotation can contribute to this task, improving almost in 5% the reported accuracy in this collection. Our contributions are (1) the introduction of a new criterion to create new levels in the irregular pyramid, combining semantic and low-level information and (2) the proposal of an iterative process where segmentation is improved using the annotation results of the previous level, and each new segmentation level is annotated taking the advantages of a better partition.

2 Introduction to Irregular Pyramids

A Region Adjacency Graph (RAG) that represents an image is a graph $G = (V, E)$, whose vertices (V) represent regions, and the edges (E) represent adjacency relations between them. An irregular pyramid [7] is composed by a set of successively reduced RAGs, (being the base level the high resolution input image). When we build an irregular pyramid [8] from an image, each level represents a partition of the pixel set into cells, i.e. connected subsets of pixels. On the base level (level 0) of the pyramid, the cells represent single pixels and the neighborhood of the cells is defined by the 4-connectivity of pixels. A cell on level k (parent) is a union of neighboring cells on level $k-1$ (children). Each graph is built from the graph below by selecting a set of surviving vertices and mapping each non surviving vertex to a surviving one. Each surviving vertex represents all the non surviving vertices mapped to it and becomes their father [7]. At any level these parent-child relations may be iterated down to the base level and the set of descendants of one vertex in the base level is named its receptive field (RF). Within the irregular pyramid framework the reduction process is performed by a set of edge contractions. The edge contraction collapses two adjacent vertices into one vertex and removes the edge. This set is called a Contraction Kernel (CK) [7][8]. The contraction of the graph reduces the number of vertices while maintaining the connections to other vertices.

3 Proposed Approach

3.1 Automatic Image Annotation Using Hierarchical Random Fields

In order to annotate regions in an image, we use the method proposed by [4]. In this work they use a base classifier to classify image regions based only on low-level features of these regions. After this first classification step, a Markov Random Field (MRF) is constructed for every level of the irregular pyramid. In addition to using the spatial Markovian neighborhood (defined by all the vertices adjacent to one vertex), they proposed to include a hierarchical Markovian neighborhood (composed by the father, in level $k+1$ of the pyramid, and children, in level $k-1$ of a vertex). This hierarchical MRF structure is used to improve the initial annotation by using contextual information from the adjacent regions and hierarchical information from regions in adjacent levels.

First, this annotation process is performed bottom-up. All the MRFs are solved starting from the lowest level using only information from the children regions in level $k-1$. After the top level is reached, the annotation is reconsidered again, and all the MRFs are computed once more in a top-down process, now with information from father and children regions in adjacent levels.

3.2 Improving Segmentation Based on Annotation Results

The approach presented in [4] is limited in terms of annotation accuracy because of the underlying image segmentation. In the irregular pyramid implementation employed in [4], the only criterion for deciding whether two regions must be joined for the next level is based on the similarity between the average color of each region. The average color of the regions is a feature that becomes less and less discriminative as regions grow bigger. We believe that the combination of low-level cues and semantic information resulting from the annotation step can improve the image segmentation and ultimately, the final annotation results.

For this task we are proposing to modify the criterion employed to create the Contraction Kernels (CK) by using the classification information at each level and the edge information extracted from each image. We propose to compute a value $V_{contract}$ that will label every edge at every level of the pyramid and will combine a semantic measure V_S and a low-level measure V_B.

For computing the semantic value $V_S(i,j)$ between vertices v_i and v_j, we use the information of the classes obtained and the prior probability given by the base classifier. For each vertex v_i, after the classification step (and correction using the MRF) we have the following information:

- A class C_i^{MRF} assigned to vertex v_i after the MRF was solved.
- The prior probability that the base classifier obtained for this class in this vertex $P(C_i^{MRF})$.
- A list of all the n classes $[C_{i,1}^{BC}, C_{i,2}^{BC}, \ldots, C_{i,n}^{BC}]$, ordered by the prior probability of each class for representing vertex v_i, obtained with the base classifier

(BC). In this way, we can notice that class $C_{i,1}^{BC}$ was the one assigned finally to vertex v_i by the BC.
– A list of all the prior probabilities $[P(C_{i,1}^{BC}), P(C_{i,2}^{BC}), ..., P(C_{i,n}^{BC})]$ obtained with the BC to v_i for each class.

The first thing to do is to check whether the classes annotated for v_i and v_j are the same. If this is the case, the value of $V_S(i,j)$ is the sum of the probabilities given by the base classifier for these classes. If the classes are different, there is a chance that the base classifier made a misclassification, therefore, we check the confidence of the class assigned to these vertices. The confidence of the classification is a logical value (true/false) given by Equation 1.

$$Confidence(C_i^{MRF}) = [(P(C_{i,1}^{BC}) - P(C_{i,2}^{BC})) > \delta] \tag{1}$$

We consider that there is confidence in the classification of vertex v_i with class C_i^{MRF} if the difference between the two highest probabilities assigned by the base classifier for this vertex, is bigger than a threshold δ. If there is confidence in the classification of both vertices, the value of $V_S(i,j)$ will be -1, indicating that semantically, these two vertices should not be joined. But if the classification for one of the vertices has no confidence, we check whether the first or second class assigned to it with higher probabilities are the same of the other vertex class, and if this happens, we sum up the probabilities for those classes to obtain $V_S(i,j)$ (depicted in Equation 2 with the name of $MisclassValue(i,j)$).

$$MisclassValue(i,j) = \begin{cases} P(C_i^{MRF}) + P(C_{j,1}^{BC}) & \text{if } C_i^{MRF} = C_{j,1}^{BC} \\ P(C_i^{MRF}) + P(C_{j,2}^{BC}) & \text{if } C_i^{MRF} = C_{j,2}^{BC} \\ -1 & \text{otherwise} \end{cases} \tag{2}$$

The process for computing $V_S(i,j)$ can be summarized in Equation 3:

$$V_S(i,j) = \begin{cases} P(C_i^{MRF}) + P(C_j^{MRF}) & \text{if } C_i^{MRF} = C_j^{MRF} \\ MisclassValue(i,j) & \text{if } Confidence(C_i^{MRF}) = 0 \\ MisclassValue(j,i) & \text{if } Confidence(C_j^{MRF}) = 0 \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

Based on the above explanation, it can be noticed that $V_S(i,j)$ intuitively represents the likelihood for two adjacent regions of being of the same class, and therefore, the likelihood for joining them given this information.

On the other hand, the value $V_B(i,j)$ represents intuitively the likelihood of joining vertices v_i and v_j taking into account the boundary information when they are two separate regions and when they are combined into a single region. For this, we use the Canny edge detector [9] to find the edges for each image, and we use the resulting edge mask to evaluate the convenience of joining two

adjacent regions. We will call the set of edge pixels in the Canny mask B_{Canny}. The set of edge pixels corresponding to the boundary of the receptive field (RF) of vertex v_i is B_i and the set of edge pixels resulting from joining the RFs of v_i and v_j is $B_{i\cup j}$. We compute in the first place how many pixels of $B_{i\cup j}$ match the edge pixels in B_{Canny} (Equation 4), and then we find the intersection between the edge pixels in B_{Canny} and the union of B_i and B_j (Equation 5).

$$B_1(i,j) = |B_{i\cup j} \cap B_{Canny}| \quad (4) \qquad B_2(i,j) = |(B_i \cup B_j) \cap B_{Canny}| \quad (5)$$

We propose to compute $V_B(i,j)$ as shown in Equation 6. In this case, we can notice that if $B_2(i,j) > B_1(i,j)$, there is a boundary between the regions of v_i and v_j that is present in the Canny edge mask, and that would be removed if these two regions were joined. This is not desirable, since this is a boundary that we would like to preserve, therefore in this case the value of $V_B(i,j)$ is -1, invalidating the contraction of these two vertices. Otherwise, the value of $V_B(i,j)$ is the relation between $B_{i\cup j}$ and the intersection of $B_{i\cup j}$ with B_{Canny}, i.e, intuitively how many edge pixels of the joint regions representing v_i and v_j match the Canny mask, with respect to the total edge pixels of the union. If all the edge pixels from the union of the two regions are present in the Canny mask, the value $V_B(i,j)$ will be 1.

$$V_B(i,j) = \begin{cases} -1 & \text{if } B_2(i,j) > B_1(i,j) \\ \frac{B_1(i,j)}{|B_{i\cup j}|} & \text{otherwise} \end{cases} \quad (6)$$

Once we have $V_S(i,j)$ and $V_B(i,j)$, we can compute $V_{contract}(i,j)$ as expressed in Equation 7, using a weight (α) to balance the importance of each type of information.

$$V_{contract}(i,j) = \begin{cases} 0 & \text{if } V_S(i,j) = -1 \quad \text{or} \\ & V_B(i,j) = -1 \\ \alpha V_S(i,j) + (1-\alpha)V_B(i,j) & \text{otherwise} \end{cases} \quad (7)$$

Once we compute $V_{contract}$ for every edge in the graph, in order to create the new CKs, each surviving vertex will use this information to select which of its adjacent vertices is more likely to be joined with it. The biggest value of $V_{contract}$ corresponds with the edge with best conditions to be contracted, given the semantic and boundary information employed. If $V_{contract}$ is 0, that edge will never be contracted, either because the vertices it connects have different semantic classes or because there is a boundary between the underlying regions that must be preserved. The combination process is illustrated in Figure 1.

Using this contraction criterion, a new level will be created. This is part of an iterative process where a level is first annotated with a base classifier, then this classification is refined by solving the associated MRF and finally the new CKs are found using the classification and boundary information, giving birth to a new level of segmentation. All the process will be repeated until we reach a level where no more contractions are allowed.

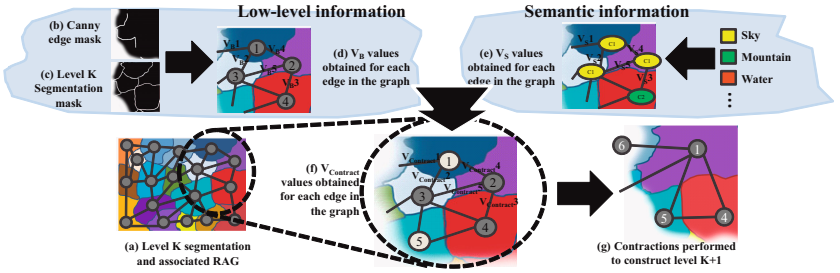

Fig. 1. Combination of the low-level and the semantic information for building a new segmentation level. In (f), white vertices are the surviving ones, and they use the $V_{Contract}$ value to determine which non-surviving vertex will be merged to it.

4 Experiments

We ran experiments on a subset of the Corel image collection (CorelA) developed by [10]. This dataset contains 205 natural scene images split into two subsets with 137 images for training and 68 images for testing. All images have been segmented and manually annotated with 22 classes.

We compute irregular pyramids for all the images, with an average of 20 levels per image. We consider that image over-segmentation is sufficient to perform efficient classification of small objects or object parts, therefore we start our process from level 10 of the original pyramids. In [4] and [11] they use KNN as base classifier We chose for our proposal to use a more sophisticated base classifier, in this case, Random Forests [12]. In order to train the base classifier, we used the ground truth annotations of this dataset, i.e. a group of regions per image, each one manually annotated with a class label. To perform the base classification on the test set, we use the training information to classify all the regions at level 10 of the test image pyramid, then we compute the MRF associated to this level and find the new CKs to construct a whole new level 11. This process will be repeated until reaching level 20 for all pyramids. We measured the annotation accuracy at pixel level for every segmentation level of each test image, with respect to the ground truth labels. Following the idea of [13], we used as visual features for each vertex (region) of each graph, the quantization of the RGB values in 16 bins per channel, yielding a 48-dimensional color histogram, and a local binary pattern (LBP) histogram to characterize texture in the region. The value of δ was set to 0.3 empirically.

In Table 1 we can see a comparison among the annotation results obtained using the base classifier Random Forest (RF) alone, the results from the HMRF-Pyr algorithm [4], which keeps the original pyramid levels of segmentation throughout the annotation process, and our proposal HMRF-PyrSeg, which uses the same annotation method of HMR-Pyr, but improves the segmentation by creating new levels. In [4] they use KNN as base classifier, with 32% of annotation accuracy, and after the HMRF-Pyr method is applied, they improved the results up to 44.6%. Nevertheless, as shown in Table 1, when we use a better base

Table 1. Results obtained in the CorelA subset for each level of the pyramid

	Pyramid levels										
Algorithm	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
RF (base classifier)	37.3%	38.5%	39.7%	40.7%	41.7%	42.6%	43.1%	43.1%	42.7%	42.2%	41.6%
HMRF-Pyr	39.7%	40.4%	41.2%	42.0%	42.7%	43.3%	43.7%	44.1%	44.4%	44.6%	44.8%
HMRF-PyrSeg	**48.7%**	**48.7%**	**48.9%**	**49.3%**	**49.7%**	**49.9%**	**50.0%**	**50.2%**	**50.3%**	**50.4%**	**50.4%**

classifier for HMRF-Pyr, the relative improvement decreases (being 3.2% the highest improvement). Since the base classifier does a better job, there is less room for refining the annotation given the underlaying image representation. We chose to modify the image segmentation by creating new levels that would take into account the annotation results from the level below, and therefore, it is possible to obtain much better annotation results with respect to the base classifier.

In Table 2 we can see a comparison of our approach with other methods that were tested on this dataset, in terms of overall accuracy. To illustrate the improvement of segmentation, in Figure 2 we show the best segmented levels for one sample image using HMRF-Pyr and HMRF-PyrSeg. With these results we can notice the relevance of having a better underlying segmentation during the process of image annotation and how these two processes can be combined to take advantage of each other's feedback for improving their results.

Table 2. Comparison with other methods in the CorelA subset. Second row shows the accuracy of each algorithm.

Algorithm	gML1o [10]	MRFs AREK [11]	HMRF-Pyr [4]	HMRF-PyrSeg
Overall accuracy	36.2%	45.6%	44.6%	**50.4%**

Original Image **Ground Truth** **HMRF-Pyr** **HMRF-PyrSeg**

Fig. 2. Example segmentation result using HMRF-Pyr and HMRF-PyrSeg. Colors represent different classes. (Best seen in color)

5 Conclusions

In this paper we proposed an approach that combines image annotation and segmentation in an iterative and hierarchical way. The segmentation step is improved using semantic information coming from a previous annotation and the subsequent annotation takes advantages of a better partition. As experimental results showed, this synergy can boost the final results of image annotation.

In a future work we plan to make experiments showing the improvements in image segmentation, as an alternative goal of this combination.

References

1. Ullman, S.: Object recognition and segmentation by a fragment-based hierarchy. Trends in Cognitive Sciences 11(2), 58–64 (2007)
2. van de Sande, K.E.A., Uijlings, J.R.R., Gevers, T., Smeulders, A.W.M.: Segmentation as selective search for object recognition. In: Proceedings of ICCV 2011, pp. 1879–1886. IEEE Computer Society (2011)
3. Akcay, H.G., Aksoy, S.: Automated detection of objects using multiple hierarchical segmentations. In: IGARSS, pp. 1468–1471. IEEE (2007)
4. Morales-González, A., García-Reyes, E., Sucar, L.E.: Hierarchical markov random fields with irregular pyramids for improving image annotation. In: Pavón, J., Duque-Méndez, N.D., Fuentes-Fernández, R. (eds.) IBERAMIA 2012. LNCS, vol. 7637, pp. 521–530. Springer, Heidelberg (2012)
5. Vieux, R., Benois-Pineau, J., Domenger, J.P., Braquelaire, A.: Segmentation-based multi-class semantic object detection. Multimedia Tools Appl. 60(2), 305–326 (2012)
6. Torrent, A., Lladó, X., Freixenet, J., Torralba, A.: A boosting approach for the simultaneous detection and segmentation of generic objects. Pattern Recogn. Lett. 34(13), 1490–1498 (2013)
7. Brun, L., Kropatsch, W.: Contains and inside relationships within combinatorial pyramids. Pattern Recogn. 39(4), 515–526 (2006)
8. Haxhimusa, Y., Kropatsch, W.G.: Hierarchy of partitions with dual graph contraction. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 338–345. Springer, Heidelberg (2003)
9. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8(6), 679–698 (1986)
10. Carbonetto, P.: Unsupervised statistical models for general object recognition. Tech. Rep., The Faculty of Graduate Studies, Department of Computer Science, The University of British Columbia, West Mall Vancouver, BC Canada (2003)
11. Hernández-Gracidas, C., Sucar, L.E.: Markov random fields and spatial information to improve automatic image annotation. In: Mery, D., Rueda, L. (eds.) PSIVT 2007. LNCS, vol. 4872, pp. 879–892. Springer, Heidelberg (2007)
12. Breiman, L.: Random forests. Mach. Learn. 45(1), 5–32 (2001)
13. Morales-González, A., García-Reyes, E.B.: Simple object recognition based on spatial relations and visual features represented using irregular pyramids. Multimedia Tools Appl. 63(3), 875–897 (2013)