

USING RELATIVE SPATIAL RELATIONSHIPS TO IMPROVE INDIVIDUAL REGION RECOGNITION

C. Millet^{†‡}, I. Bloch[‡], P. Hède[†], P.-A. Moëllic[†]

[†]CEA/LIST/LIC2M, 18 Route du Panorama, 92265 Fontenay aux Roses, France

[‡]GET-ENST - Dept TSI - CNRS UMR 5141 LTCI - Paris, France

milletc@zoe.cea.fr, phone: (+33) 1 46 54 96 46, fax: (+33) 1 46 54 75 80

Keywords: spatial relationships, object recognition, knowledge inference.

Abstract

As well as words in text processing, image regions are polysemic and need some disambiguation. If the set of representations of two different objects are close or intersecting, a region that is in the intersection will be recognized as being possibly both objects. We propose here a way to disambiguate regions using some knowledge on relative spatial positions between these regions. Given a segmented image with a list of possible objects for each region, the objective is to find the best set of objects that fits the knowledge. A consistency function is constructed that attributes a score to a spatial arrangement of objects in the image. The proposed algorithm is demonstrated on an example where we try to recognize backgrounds (sky, water, snow, trees, grass, sand, ground, buildings) in images. An evaluation over a database of 10000 images shows that we can reduce the number of false positive while keeping almost the same recognition rate.

1 Introduction

Recognizing regions as individual entities is an old and still very active issue in computer vision. Many learning methods have been developed for this purpose, but most of them focus on single regions. However, an image contains many regions which are in relationships with each others, and such information is not often exploited. The results of individual region recognition should be consistent in the whole image. For example, detecting a grass region above a sky region does not make sense even if each object is individually recognized with a good probability. Relative spatial relationships can easily be used here to tell there is a contradiction, and to try to solve it.

Furthermore, individual region recognition will never reach perfection, because in image processing, regions are prone to ambiguity at least as much as words in text processing. Two regions can have exactly the same texture, color and form, but represent completely different objects depending on their context in the image.

In order to solve that issue, Carbonetto et al [4] proposed to learn the co-occurrences of objects using a database annotated at image level. A Markov random field is trained that takes the neighbors into account when classifying a region. In this approach, the presence of an airplane can be used for example to make the difference between sky and water.

In this paper, we propose a different image region disambiguation based on the knowledge of how regions should be spatially arranged. In a different field, a similar approach has been proposed in [11] to improve the recognition of musical scores: structural information such as relationships between symbols and musical rules is used to choose the best hypothesis among three for each detected symbol.

In our approach, the image is first segmented into regions, and each region is analyzed individually using a Support Vector Machine which returns several hypotheses with associated probabilities. The object recognition algorithm used for our background recognition example is described in Section 2. We then compute relative spatial relationships between regions (Section 3). Then, the different hypotheses for all the regions of the image are compared. The final recognition is achieved by maximizing the hypotheses probabilities under the constraint of generating a spatially consistent description of the image. The maximum of the consistency function proposed in Section 4 meets these criteria.

Such a reasoning can be used in any spatially structured scene (satellite imaging, medical imaging,...). An example for background recognition in photograph images is given in Section 5 where backgrounds are of eight types: sky, water, snow, trees, grass, sand, ground and buildings.

2 Backgrounds recognition

Recognizing backgrounds using low-level features has first been proposed in 1997 by N.W.Campbell et al. [3]. Subsequent work improved results by considering about ten backgrounds and testing on larger databases [12, 1, 7]. The approach is always similar: color and texture features (Color Histograms, Edge Direction Histograms, Wavelet,...) are computed, and a learning algorithm (Neural Network, Support Vector Ma-

chine,...) is trained to classify backgrounds. All published methods reported good results ranging from 84% and 99% depending on the background. These results are very good, but they can not reach 100% because of different backgrounds having the same color and texture. Furthermore, they have not been working on false positives (backgrounds that are detected for a region that does not represent a background) which should be minimized too. Our goal is to use the spatial informations to improve the recognition rate while reducing the false positives.

For the segmentation step, we applied the fast implementation of waterfall based on graphs developed by B. Marcotegui and S. Beucher [8], which is both fast and efficient for color images. We parameterized it so that we obtain at most twenty regions. Then, for each region, a 512-bins texture local edge patterns histogram [6] and a 64-bins color histogram (each R, G and B plane is quantified into 4 values) are used as features for learning. A binary Support Vector Machine (SVM) that returns probability values [5] is then learned for each class, taking the background we want to learn as the positive class, and the other backgrounds as negative samples. We want to recognize eight types of backgrounds (sky, water, snow, trees, grass, sand, ground, buildings), so this will result in the learning of eight binary SVMs.

As we aim at dealing with overlapping classes, binary SVMs are not really appropriate if all images are weighted equally. Let us consider the case where we have only two classes: *sky* and *water*. A sky-SVM (resp. water-SVM) is learned with *sky* (resp. *water*) as the positive class and *water* (resp. *sky*) as the negative class. If the positive and negative samples have the same weights, these two SVMs will give the same results: if a region is classified by the sky-SVM as *sky* with a probability of 80%, then it will be classified as *water* by the water-SVM with a probability of 20% (or *non sky* with a probability of 80%). What we want is that a region that belongs to the *sky* and *water* classes is given a good probability for both.

A solution consists in giving more weight to the positive class. Then, the same region can be learned as *sky* when trained the sky-SVM, whereas it will be recognized as *water* by the water-SVM. For more details on how to obtain probabilities with binary SVMs and on how to weight data, see [5]. For each region, we keep the hypotheses for which the returned probability is above 30%, and we add the hypothesis that the object is unknown with a probability of 30%.

The effect of weights on learning is outlined in Figure 1. We consider two overlapping classes 'x' and 'o' (for example, 'x' can be the *sky* class and 'o' the *water* class). When learned with equal weights, the frontier leant between the two classes does not depend on which class is the positive sample as the problem is symmetrical. However, if we apply a more important weight on the positive class when learning, object in the overlapping region will be learned as 'x' when 'x' is the positive class, and as 'o' when 'o' is the positive class. We used a weight of three which means that misclassifying a positive sample costs as much as misclassifying three negative samples.

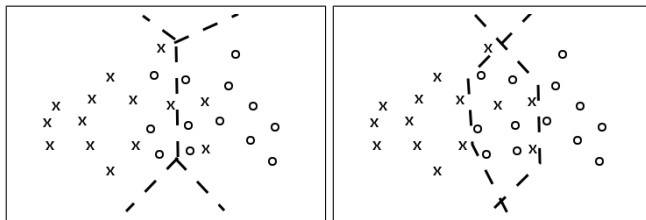


Figure 1: Example of the effect of weights for classification. Left image: Same weight for positive and negative samples. Right image: a weight of 3 is applied for the positive examples and a weight of 1 for the negative examples thus better highlighting the intrinsic ambiguity of the data in the overlapping area.

3 Relative spatial relationships

Relative spatial relationships have been studied mainly in the field of artificial intelligence. In image processing, it is sparsely applied. The main applications are model-based structure recognition in medical images, and linguistic description of images. Many techniques have been proposed in order to compute these relationships, among which angle histograms, force histograms and mathematical morphology methods. A review and comparison of relative spatial relationships computing methods for image processing can be found in [2].

We are computing four relationships: above, below, left of, and right of using the angle histogram method presented in [9]. An angle histogram is computed between two regions considering all possible pair of points. For each pair, the angle made between this segment and the horizontal axis (Figure 2) is added in the histogram.

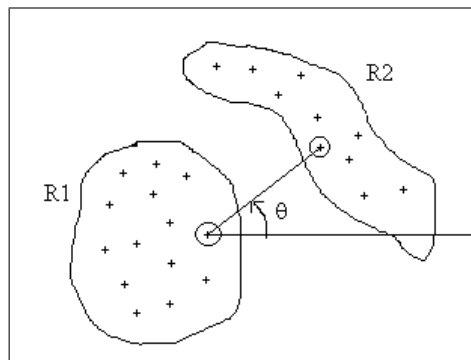


Figure 2: Spatial relation between two points of two regions.

Then, this histogram is normalized and multiplied by a fuzzy function which is a square cosine function centered in 0 radian (resp. $\pi/2$, π , and $3\pi/2$) to obtain the percentage with which the right of (resp. above, left of, and below) relationship is verified (See Figure 3). For example, if h is the angle histogram "R₂ is right of R₁" is verified with the confidence \mathcal{R}_{right} de-

defined by:

$$\mathfrak{R}_{right} = \sum_{\theta=-\pi/2}^{\pi/2} h(\theta) * \cos^2(\theta)$$

This gives the relation of region R_2 regarding R_1 : an angle of 0 radian means that R_2 is right of R_1 , and that R_1 is left of R_2 .

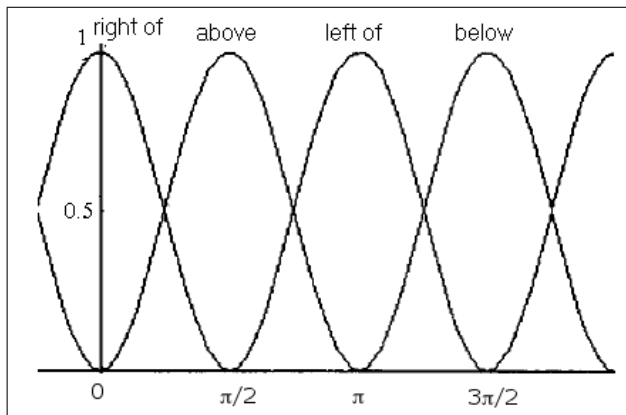


Figure 3: Square cosine functions used as a fuzzy set for the four directions.

In order to compute them faster, only 500 pixels are kept for large regions. When choosing these 500 pixels, we must be careful that they are representative of the shape of the region. We achieve this by sorting the pixels in a list in the reading order (the top left pixel is the first, the bottom right is the last), and picking one out of that list regularly. We checked that this extraction of significant pixels provides a very good approximation of the real relation obtained when preserving all pixels. This method presents the advantage of being fast to implement and compute.

4 Consistency function

The aim of the consistency function is to evaluate which hypotheses are best, taking into account the knowledge of how objects should be spatially arranged, the probabilities of object detection returned by the SVMs and the spatial relationships between the regions in the image. The following function has been used to measure the consistency of a given set of backgrounds with their spatial relationships:

Given N regions R_i in the image that may be backgrounds according to the SVMs, we compute a consistency formula for a each possible hypothesis. We note B_i a background attributed to the region R_i . An hypothesis can be for example: $B_1 = sky, B_2 = unknown, B_3 = water$, meaning that "region 1 is sky, region 2 is not a background, and region 3 is water". We propose using the following formula:

$$C(Image) = \sum_{i,j=1}^N \sum_{\substack{\mathfrak{R} \\ i \neq j}} C(R_i(B_i), R_j(B_j))$$

where

$$C(R_i(B_i), R_j(B_j)) = P(B_i) * P(B_j) * (R_i \mathfrak{R} R_j) * Eval(B_i, \mathfrak{R}, B_j)$$

- $P(B_i)$ is the probability of detection of the background B_i for the region R_i returned by the SVM.
- \mathfrak{R} is a spatial relationship between two backgrounds.
- $(R_i \mathfrak{R} R_j)$ is the degree of confidence for the relative spatial relationship \mathfrak{R} between the regions R_i and R_j . If for example region R_i is 80% above region R_j , 10% below and 10% right of it, then " $R_i \text{above} R_j = 0.8$."
- $Eval(B_i, \mathfrak{R}, B_j)$ is a knowledge based function. It returns a value representing if a relation (B_i, \mathfrak{R}, B_j) between two backgrounds is in agreement with the rules (+1), in contradiction with them (-1), or not represented by them (0).

These notations will be illustrated on an example in Section 5. Of course, this function can only be applied if there are two or more regions: with only one region, no relative spatial position can be computed.

Finding the Maximum of this function is achieved trying all the possibilities. We have typically no more than 4 backgrounds in the image, and 3 hypotheses for each, which gives 81 combinations, so it is reasonable. If we had more combinations, we could use algorithms such as simulated annealing or other optimization methods.

In order to analyze this function, let us imagine an image with three backgrounds. Several cases are possible:

1. Each couple of backgrounds is compatible ($Eval(B_i, \mathfrak{R}, B_j) > 0$). Then, the contribution of each couple is positive, the global score is also positive, and labeling a region as unknown will lessen that score, so the best score is obtained when keeping all backgrounds.
2. Two backgrounds B_1 and B_2 are incompatible ($Eval(B_1, \mathfrak{R}, B_2) < 0$), but are compatible with B_3 . Then, the score depends mostly on the consistency of each couple of backgrounds. If $C(R_1(B_1), R_2(B_2)) + C(R_1(B_1), R_3(B_3)) > 0$ and $C(R_1(B_1), R_2(B_2)) + C(R_2(B_2), R_3(B_3)) > 0$ then all three backgrounds are kept, else, the background B_1 or B_2 whose consistency with B_3 is the smallest is labeled as unknown. When all three backgrounds are kept, the final scene description remains inconsistent, and we would need more knowledge (for example a fourth background) to solve it.
3. A background B_1 is inconsistent with the two others B_2 and B_3 , but B_2 and B_3 are consistent. Then, the contribution of B_1 is negative in both couples, giving the unknown label will increase the score. The combination (B_2, B_3) is better as it gives a positive score. B_1 alone gives a score of zero.

4. All three couples of backgrounds are inconsistent. Then, each couple has a negative contribution, and the global score is always negative. The best score is 0: we keep the background whose probability of detection is the highest, the two others are labeled unknown.

In general, if all couples of regions are inconsistent, then a best score of 0 will be obtained for all combinations where we keep only one region, or no region so that we have multiple global maxima. In this case, we keep the region whose individual recognition rate is the higher.

If we can find at least 2 regions which are not contradictory, it ensures that the best score is above 0, and that the best combination will contain several regions.

When comparing the results of detection before and after applying spatial reasonings, we typically have three kinds of modification for a given region: the label can be kept, the label can be changed into another background, or the region can be considered as not being a background.

5 An example

We now apply this consistency function on an example where we look for backgrounds in photograph images. Eight backgrounds have been considered: sky, water, snow, trees, grass, sand, ground and buildings. These backgrounds can be classified into three groups according to their relative position to the skyline. The first group (groupA) contains backgrounds that are always above the skyline, the second group (groupB) those which can cross the skyline, and the third group (groupC) those always below the skyline. The following groups are therefore defined:

$$\begin{cases} \text{groupA} = \{ \text{sky} \} \\ \text{groupB} = \{ \text{trees, buildings} \} \\ \text{groupC} = \{ \text{water, grass, snow, sand, ground} \} \end{cases}$$

We are not willing to detect skyline, but these groups allow us to build simple rules. The Eval function that explicits these rules is developed in Table 1.

Considering this table, we noticed that the above-right relationship causes some errors in the final detection. Consider for example an image containing a green region recognized as either trees or grass located top right of a water region. When computing the fuzzy relationships of these two regions, we get 50% right and 50% above. For the $(\text{trees}, \text{water})$ couple, only the above relationship is taken into account with the rule $(\text{groupB}, \text{above}, \text{groupC}) = 1$. The consistency is then:

$$C = P(\text{trees}) * P(\text{water}) * (0.5) * 1$$

whereas for the $(\text{grass}, \text{water})$ couple, both relations are taken into account via the rule $(\text{groupC}, \text{any}, \text{groupC}) = 1$, which gives the following:

$$C = P(\text{grass}) * P(\text{water}) * (0.5 + 0.5) * 1$$

| (A, \mathfrak{R}, B) | $Eval(A, \mathfrak{R}, B)$ |
|--|----------------------------|
| $(\text{groupA}, \text{above}, \text{groupB})$ | +1 |
| $(\text{groupA}, \text{below}, \text{groupB})$ | -1 |
| $(\text{groupA}, \text{above}, \text{groupC})$ | +1 |
| $(\text{groupA}, \text{below}, \text{groupC})$ | -1 |
| $(\text{groupB}, \text{above}, \text{groupA})$ | -1 |
| $(\text{groupB}, \text{below}, \text{groupA})$ | +1 |
| $(\text{groupB}, \text{above}, \text{groupC})$ | +1 |
| $(\text{groupB}, \text{below}, \text{groupC})$ | -1 |
| $(\text{groupC}, \text{above}, \text{groupA})$ | -1 |
| $(\text{groupC}, \text{below}, \text{groupA})$ | +1 |
| $(\text{groupC}, \text{above}, \text{groupB})$ | -1 |
| $(\text{groupC}, \text{below}, \text{groupB})$ | +1 |
| $(\text{groupA}, \text{any}, \text{groupA})$ | +1 |
| $(\text{groupB}, \text{any}, \text{groupB})$ | +1 |
| $(\text{groupC}, \text{any}, \text{groupC})$ | +1 |
| <i>otherwise</i> | 0 |

Table 1: Description of the Eval function. *any* relationships is one of the four "above, below, right of and left of".

So, the *grass* hypothesis is clearly unfairly more advantaged than the *trees*. To overcome this issue, the relationships are modified when considering two elements that are not in the same group: the above and below relation are stretched by the same factor so that their sum equals 100%. The two consistency functions shown above as example are then comparable.

Let us take an example where a wrong background is corrected. In the image in Figure 4, the sky (region 1) is detected has being snow (44%), sky (43%) or unknown (30%); the region 2 is buildings (36%) or unknown (30%); region 3 is not recognized because of an imprecise segmentation and region4 is recognized as ground (42%) or unknown (30%).

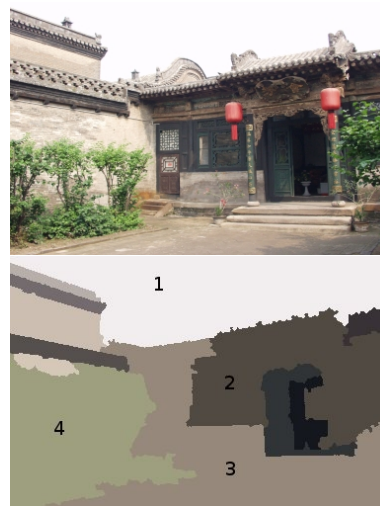


Figure 4: Image example and its segmentation. Four regions are detected as possible backgrounds

The scores returned by the consistency function for each hy-

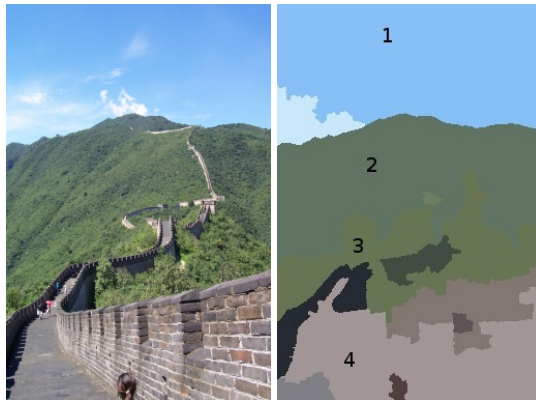
pothesis are given in Table 2.

| region 1 | region 2 | region 3 | score |
|----------|-----------|----------|-------|
| unknown | unknown | unknown | 0 |
| sky | unknown | unknown | 0 |
| snow | unknown | unknown | 0 |
| unknown | buildings | unknown | 0 |
| sky | buildings | unknown | 0.31 |
| snow | buildings | unknown | -0.32 |
| unknown | unknown | ground | 0 |
| sky | unknown | ground | 0.36 |
| snow | unknown | ground | 0.37 |
| unknown | buildings | ground | 0.28 |
| sky | buildings | ground | 0.96 |
| snow | buildings | ground | 0.34 |

Table 2: Scores for the consistency function applied on the image in Figure 4.

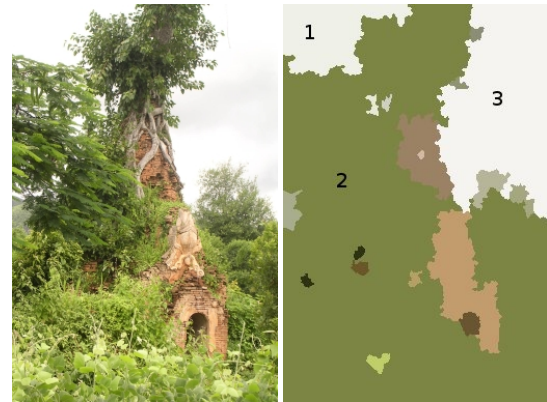
The individual regions detection gives $1=snow$, $2=buildings$, $4=ground$ as the best set, whereas the consistency function is maximum for $1=sky$, $2=buildings$, $4=ground$. The second best hypothesis is $1=snow$, $2=unknown$, $4=ground$ which is also consistent and has a better score than $1=snow$, $2=buildings$, $4=ground$ which is not.

More examples are given in Figures 5, 6, 7 and 8.



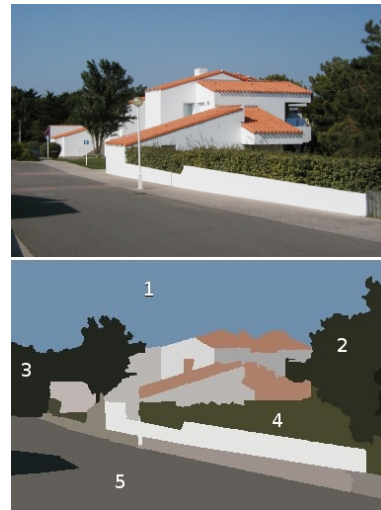
| Region | Without SR | With SR |
|--------|------------|-----------|
| 1 | sky | sky |
| 2 | grass | trees |
| 3 | trees | trees |
| 4 | buildings | buildings |

Figure 5: Comparison of background detection without Spatial Reasoning (SR) and with Spatial Reasoning. In this example, the incorrect grass label has been changed into trees because it was conflicting with the buildings and the trees detected below it



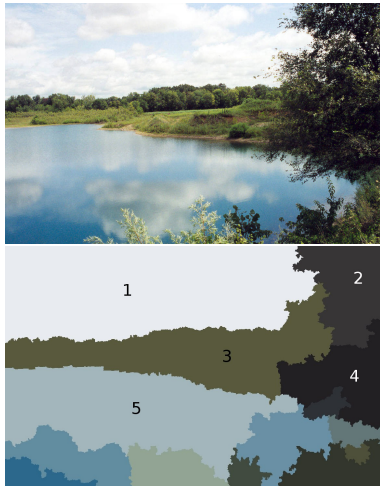
| Region | Without SR | With SR |
|--------|------------|---------|
| 1 | snow | sky |
| 2 | trees | trees |
| 3 | sky | sky |

Figure 6: The trees and the sky invalidate the snow hypothesis, and validate the sky alternative.



| Region | Without SR | With SR |
|--------|------------|---------|
| 1 | water | sky |
| 2 | trees | trees |
| 3 | grass | trees |
| 4 | trees | trees |
| 5 | sky | unknown |

Figure 7: The sky label is changed into unknown for the street. The presence of trees in regions 2 and 4 also allow to resolve the water/sky ambiguity for Region 1 and the grass/trees ambiguity for Region 3



| Region | Without SR | With SR |
|--------|------------|---------|
| 1 | sky | sky |
| 2 | tree | tree |
| 3 | tree | tree |
| 4 | tree | tree |
| 5 | sky | snow |

Figure 8: An example with an image from the Internet. The sky hypothesis has been discarded for the region 5. It has been replaced with snow (instead of water) because the reflected clouds have a texture and color closer to snow than to water. This is an example of the limitation of our algorithm which is unable to resolve the snow/water confusion.

6 Results

Our algorithm has been evaluated on a database of 10000 manually annotated images where 4076 come from the Corel database [13] and 5924 from the CLIC database’s kernel [10]. The background learning database contains about 300 images extracted from the Corel database, but not from the CLIC database.

The evaluation process is the following: each image is first segmented, then each region whose size is greater than 5% of the image is classified by each Support Vector Machine to get the list of candidate backgrounds with their probability. The combination that keeps the backgrounds with the higher probability without any spatial reasoning is the result ”before applying spatial relationships”. The combination that maximizes the consistency function described above is the result ”after applying spatial relationships”. Duplicate labels in an image after automatic classification are removed, for example, if a sky region is segmented into two regions, and both are correctly classified, then we just keep one. A label present in both the automatic classification and the manual annotation is a correct classification. A label found by the automatic classification that is not in the manual annotation is a false positive.

For example, if an image contains sky, trees and water, these

will be the three elements reported in the manual annotation. Then, we suppose that the sky is segmented into two regions recognized as sky, the trees are recognized and the water is recognized as snow. The automatic annotation is then *sky, sky, trees*. We eliminate duplicate labels: *sky, trees*. In this example, *sky* and *trees* are two correct classification, *water* is an undetected background (we are not interested in them in this article), and *snow* is a false positive.

On this 10000 images database, 4124 images were classified as containing at least two backgrounds by the background recognition classifier without applying any spatial reasoning. As the algorithm has no effect on images with one or no background, we tested it on these 4124 images. Because the algorithm can just change an existing background label into another background label, or into the unknown label, it tends to reduce the number of backgrounds in images. Nevertheless, we do not lose too much backgrounds, and it even preserves images with more than five backgrounds

it is not too destructive as we can see in Table 3.

| Nb | Number of images | |
|------|------------------|-------|
| | before | after |
| 0 | 0 | 0 |
| 1 | 0 | 116 |
| 2 | 1419 | 1518 |
| 3 | 1390 | 1336 |
| 4 | 803 | 723 |
| 5 | 371 | 314 |
| 6 | 104 | 88 |
| 7 | 27 | 20 |
| 8 | 9 | 8 |
| 9 | 1 | 1 |

Table 3: Number of images containing Nb backgrounds before and after applying spatial relationships analysis.

Table 4 shows the ”ratio of correct classification rate” and the ”ratio of false positive rate” obtained with the addition of spatial reasoning. The ”ratio of correct classification rate” (resp. false positive rate) is defined as the correct classification rate (resp. false positive rate) without spatial reasoning divided by the same rate obtained when spatial reasoning is applied. The ideal ratio for the correct classification would be 100% or more. A ratio of 100% is achieved if no correct background is eliminated. A ratio greater than 100% is obtained when an incorrect background is changed into a correct background. That is the case here for the *ground* background. We notice also that the ratio of false positive is below the ratio of correct classification for all backgrounds, which is encouraging.

One of the most common error is to confuse snow with sky. The algorithm corrects well this case as can be seen in Table 4: 54.1% of false positive snow detections are modified (changed to another background or removed) when applying the spatial reasoning. Other common mistakes are sky/water and grass/trees.

| Background | Ratio of correct classification | Ratio of false positive |
|---------------|---------------------------------|-------------------------|
| sky | 98.9% | 81.1% |
| water | 97.5% | 87.1% |
| trees | 98.9% | 91.0% |
| buildings | 98.3% | 94.2% |
| grass | 92.1% | 83.4% |
| snow | 80.0% | 45.9% |
| ground | 105.0% | 88.2% |
| sand | 88.9% | 85.7% |
| average mean | 94.9% | 81.7% |
| weighted mean | 98.1% | 86.8% |

Table 4: Ratio of correct classification and false positive when applying the spatial reasoning. The weighted mean takes into account the number of each background.

Within the 4611 images containing two or more backgrounds, we recognized 13410 backgrounds among which 760 (5.7%) have been modified by the consistency analysis. These modifications can be classified into 5 sets:

1. a label is modified from an incorrect background to a correct one (good)
2. a label is modified from a correct background into another correct background (not good, but not bad). This can happen for example in images containing both trees and grass. If the two corresponding regions have been merged by the segmentation, then, changing its label from trees to grass will not change the number of correctly identified backgrounds in the image
3. a label is modified from a correct background to an incorrect background (bad)
4. a label for a correct background is removed (bad)
5. a label for an incorrect background is removed (good)

The distribution of these modifications is reported in Table 5.

| Kind of modification | Number of images |
|----------------------|------------------|
| 1 | 14 (1.8%) |
| 2 | 2 (0.3%) |
| 3 | 41 (5.4%) |
| 4 | 70 (9.2%) |
| 5 | 633 (83.3%) |

Table 5: Number of images concerned by each modification

The prevalent modification made is the removal of a incorrect background (83.3%) which was the primary goal we wanted to achieve. Concerning the modification of a background into another background, 5.4% are worsening the classification, whereas only 1.8% are enhancing it. So, we would score better if we do not do these kinds of modification (kinds 1, 2 and 3),

just keeping the possibility of changing a background into the unknown label (kinds 4 and 5).

The main reason why a correct background is removed is because an incorrect background is detected somewhere else in the image, and has a high probability of detection with no other candidates. It is mostly due to non-background regions recognized as background regions. For example, tigers are often classified as trees or grass, elephants are often classified as buildings, and street as water or sky thus disturbing the spatial analysis. This can not be solved in the closed world of eight backgrounds, except if we give some images of animals as examples of non-background images, that is if we create a ninth class that contains everything that is not a background. We recently tried to add some images in this ninth class, and it strongly reduces the false positive rate. However, this is hazardous, and the images have to be selected carefully so that they do not overlap with too much with backgrounds which would result in reducing the detection rate.

Another alternative to creating this ninth class would be to change the learning method from binary classifiers to density estimator such as one-class SVM. We should also consider working on the weights applied for learning. Currently, the weight is fixed to 3 for the positive class, and 1 for the negative class. We plan to try to automatically choose this weight to handle unbalanced data, because each background does not have the same number of samples.

7 Conclusion

Results are promising, as the objective function is still simple but ensures that the final set of backgrounds found is consistent in most cases. An evaluation made on a 10000 images database shows that it can remove a lot of false detections, with the drawback of removing also some correct classifications. Future work aims at improving the learning method and dealing with more objects. To achieve this, we may introduce two spatial relationships: inside and surround. We are also planning to automatically learn the spatial relationships of objects.

References

- [1] Z. Aghbari, A. Makinouchi. "Semantic approach to image database classification and retrieval", *NII Journal*, **7**, (September 2003).
- [2] I. Bloch. "Fuzzy spatial relationships for image processing and interpretation: A review", *IVC*, **23**(2), pp. 89–110, (February 2004).
- [3] N. W. Campbell, W. P. J. Mackeown, B. T. Thomas, T. Troscianko. "Interpreting image databases by region classification", *Pattern Recognition (Special Edition on Image Databases)*, **30**(4), pp. 555–563, (April 1997).

- [4] P. Carbonetto, N. Freitas, K. Barnard. “A statistical model for general contextual object recognition”. In *ECCV 2004*, (May 2004).
- [5] C.-C. Chang, C.-J. Lin. *LIBSVM: a library for support vector machines*, (2001). Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [6] Y.-C. Cheng, S.-Y. Chen. “Image classification using color, texture and regions.”, *Image Vision Comput.*, **21**(9), pp. 759–776, (2003).
- [7] C. Cusano, G. Ciocca, R. Schettini. “Image annotation using svm”. In *Internet Imaging V. Edited by Santini, Simone; Schettini, Raimondo. Proceedings of the SPIE, Volume 5304, pp. 330-338 (2003).*, pages 330–338, (December 2003).
- [8] B. Marcotegui, S. Beucher. “Fast implementation of waterfall based on graphs”. Volume 30 of *Computational Imaging and Vision*, pages 177–186. Springer-Verlag, Dordrecht, (2005).
- [9] K. Miyajima, A. Ralescu. “Spatial organization in 2d segmented images: Representation and recognition of primitive spatial relations”, *FuzzySets*, **65**, pp. 225–236, (1994).
- [10] P.-A. Moëllic, P. Hède, G. Grefenstette, C. Millet. “Evaluating content based image retrieval techniques with the one million images clic testbed”. In *Proceedings of the Second World Enformatika Congress, WEC’05*, pages 171–174, Istanbul, Turkey, (February 2005).
- [11] F. Rossant, I. Bloch. “A fuzzy model for optical recognition of musical scores”, *Fuzzy sets and systems*, **141**, pp. 165–201, (2004).
- [12] C. Town, D. Sinclair. “Content based image retrieval using semantic visual categories”. Technical Report TR2000-14, AT&T Laboratories Cambridge, (2000).
- [13] J. Z. Wang, J. Li, G. Wiederhold. “Simplicity: Semantics-sensitive integrated matching for picture libraries”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(9), pp. 947–963, (2001).