From Color Groups to Scene Interpretation

N. Jouandeau $^{\#1}$, P. Bonnin $^{\#2}$, V. Hugel $^{\#3}$, P. Blazevic $^{\#4}$

$^{\#1}$*LIASD, Paris8 University*
2 rue de la liberté, 93526, Saint-Denis Cedex, FRANCE
$^{\#2-4}$*LISV, Versailles University*
10-12 Av de l'Europe, 78140, Velizy, FRANCE
$^{\#1}$n@ai.univ-paris8.fr
$^{\#2-4}$(bonnin, hugel, blazevic)@lisv.uvsq.fr

Abstract— **In this paper we detail an algorithm that deals with objects and shape recognition based on color groups identification. The algorithm can dynamically adjust the unit set of pixels used to define color groups. Each color group is selected from the current scene. The algorithm is optimized to run faster than classical ones. Experiments were conducted on soccer images captured from real situations of the standard platform league (SPL). They also show the influence of camera parameters like contrast and gain. They lead to the conclusion that a set of rules could help to aggregate color groups and to identify scenes according to a classical predefined color look-up table (CLUT).**

I. Introduction

Region image segmentation is a very important topic in computer vision. A wide range of computational vision applications and vision problems could in principle make good use of segmented images, were such segmentations reliably and efficiently computable. A similar algorithm has already been applied with success [1]. We address the robustness of the algorithm over subsampling and its interpretation into the scene.

Section 2 describes related work. Section 3 explains the algorithm, its subsampling policy and its results. Section 4 presents the scene interpretation.

II. Related Works

Region image segmentation is needed for many applications, either using classical RGB images (for example object detection for humanoid robot context [2]), or using particular images, for example infrared thermography for nondestructive testing [3] or range image for mobile robot navigation [4].

In addition, intermediate-level vision problems such as stereo and motion estimation require an appropriate region of support for matching operations. Higher-level problems such as recognition and image indexing can also make use of segmentation results for matching.

Region segmentation was studied in the 70s (Pavlidis split and merge well known algorithm [5], and survey of Zucker [6]). But, the problem seems not to be solved yet considering work in these recent years. They are a few new proposed methods, but many well-known methods have been improved: watershed (introduced by S.Beucher [7], improved in [8], [9], graph based [10], seed algorithms [11], optimization of a multiple criteria function [12], and Voronoi based [13].

Another way to improve well-known methods is to study complementary algorithms, and to combine them in an hybrid method, which produces better results, according to the criteria of evaluation. Milgrams and Ohlender algorithms are combined in [14]. Closer to our work, because dedicated to robotic applications, [15], [16] present an evaluation of two popular segmentation algorithms, the mean shift-based segmentation algorithm and a graph-based segmentation scheme. Then, they also consider a hybrid method which combines the other two methods. Three characteristics are taken into account :

- Correctness: the ability to produce segmentations that match human intuition, that is, segmentations that correctly identify structures in the image at neither too fine nor too coarse level of detail.
- Stability with respect to parameter choice: the ability to produce segmentations of consistent correctness for a range of parameter choices.
- Stability with respect to image choice: the ability to produce segmentations of consistent correctness using the same parameter choice on a wide range of different images.

In [1], we proposed a hybrid method that combines 3x3 pyramidal adaptations of Horowicz and Pavlidis algorithm [5], with A. Rosenfeld, JL Pfalz blob detection. As detailed in [1], the criteria of our Clear Box Evaluation is the speed up factor, considering a minimun quality of the segmentation results that allows the robot to achieve its task: even though all pixels are not correctly labeled, attributes of regions such as surface, and gravity center must be correct.

Our motivation is to find a segmentation method that is robust to changes of lighting conditions as well for the RoboCup challenges in the standard league (Nao Humand robot), as for other applications in outdoor scenes. First, we defined a procedure of camera parameters tunning that minimizes lighting conditions' sensitivity. Second, we expect that the combination of pyramidal gathering, blob detection and our region interpretation makes the best of each parts. In contrast to previous works [17], the neighborhood is not computed only on a region level.

In our last participations at robocup (from Paris in 1998 to Osaka in 2005, in Legged League with AIBO robot), in

Proceedings of the 4th Workshop on Humanoid Soccer Robots
A workshop of the 2009 IEEE-RAS Intl. Conf. On Humanoid Robots (Humanoids 2009),
Paris(France), 2009, December 7-10
ISBN 978-88-95872-03-2
pp. 38-44

order to meet real time constraints, a pixel color classification followed by a blob detection were implemented. But, the pixel color classification requires many parameters such as threshold values that depend very much on lighting conditions. The region color classification requires the same parameters, but it is less dependent on these conditions.

III. Static Region Segmentation Algorithm

Our algorithm realizes the region segmentation in two steps : the first one defines `R1`, `R2` and `R3`, sets of 3x3, 9x9 and 27x27 regions (lines 6 to 11). The second one defines `R′`, the set of regions with same color attributes (lines 13 to 29).

The threshold mentioned in line 9 of the algorithm defines a minimal number of regions with the same color space attributes. Each region has 8 surrounding regions, that can be selected if its distance to the central region is below a fixed value (in YUV color space, we use $(dy + 1.2 \times du + 1.2 \times dv)/3.4 < value$). If more than 5 surrounding regions are selected, then the central region and selected surrounding regions are regrouped.

```
void regionSegmentation (image)
 1 R = {R0, R1, R2, R3}
 2 R0 = R1 = R2 = R3 = {}
 3 S = {3x3, 9x9, 27x27}
 4 for each pixel p in image
 5   add p in R0
 6 for i = 0 ; i < 3; i++
 7   for each e in R[i]
 8     compute S[i] region centered at e
 9     if under threshold
10       add e to R[i+1]
11       remove e from R[i]
12 R′ = {}
13 for i = 3 ; i > 0; i--
14   for each e in R[i]
15     blob_regroup = false
16     for each r′ in R′
17       regroup e with connected r′
18       if regrouped
19         remove e from R[i]
20         add e to r′
21         blob_regroup = true
22     if blob_regroup == false
23       add e to new_region
24       regroup e with connected R[i] elt
25       if regrouped
26         for each connected R[i] elt c
27           remove c from R[i]
28           add c to new_region
29       add new_region to R′
```

In the second part (lines 13 to 29), bigger regions are selected first and regrouped in `R′`. If a region is not regrouped with any others, then it makes a new region (line 29).

Table I compares the number of regions for 6 classical robocup views, where the first two are simple scenes, the next two are less simple and the last two are most complicated. Fig. 1 shows a part of the central circle and the penalty point. Fig. 2 shows the same scene type but with blur. Such a case can occur if motion is not well synchronized with the video grabbing process. Fig. 3 shows a simple static view, including the ball and a part of the goal. Fig. 4 shows the field partially occulted by another moving Nao. In this case, the blur due to other robots moving cannot be avoided with motion synchronization. Fig. 5 shows a Nao with the ball. Fig. 6 shows a scene with public watching the game. The 3x3 column shows that the number of initial 3x3 regions are closely the same except for complicated scenes, involving details smaller than a 3x3 region like in Fig. 6. Starting from this, the number of regions decreases in the next two columns, which means that some of these regions become 9x9 and 27x27 regions. For Fig. 6, results are similar for 9x9 and 27x27 regions. Although only 151 regions are regrouped (which means 4% of the 9x9 regions), the last 27x27 region synthesis is really important for scene interpretation. The next three columns show the blob-regroup steps. The scene complexity is expressed with the final number of regions showed in the r3x3 column. The number of regions for blurred images is slightly greater than for non-blurred images.

TABLE I
RESULTS WITH CLASSICAL ALGORITHM

	3x3	9x9	27x27	r27x27	r9x9	r3x3
Fig. 1	8360	1756	1210	1127	875	55
Fig. 2	8401	1760	1168	1093	896	253
Fig. 3	7939	2865	2541	2494	2140	352
Fig. 4	8066	2674	2259	2198	1933	526
Fig. 5	7498	2841	2518	2469	2181	663
Fig. 6	6643	3657	3506	3487	3270	1427

TABLE II
RESULTS WITH SUBSAMPLING POLICY

	3x3	9x9	27x27	r27x27	r9x9	r3x3
Fig. 1	2094	477	356	340	264	7
Fig. 2	2090	485	343	326	278	67
Fig. 3	1972	835	783	778	676	101
Fig. 4	1995	801	727	719	635	192
Fig. 5	1836	797	743	739	660	206
Fig. 6	1597	931	892	888	848	366

To adapt our algorithm to the Nao's processing capacity, we applied a simple subsampling policy. Therefore we modified the 4th line of the algorithm by taking one out of two pixels, and one out of two lines.

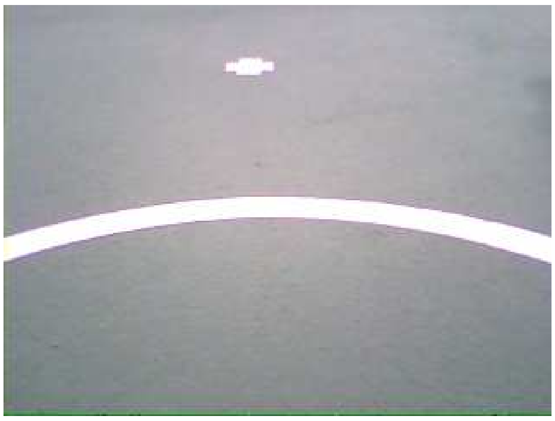

Fig. 1. Classical view facing the line

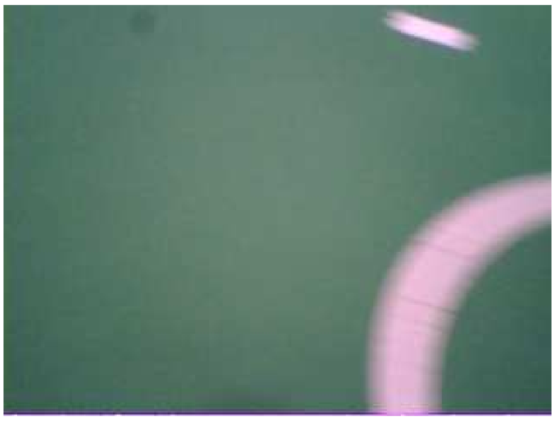

Fig. 2. Classical view facing the line

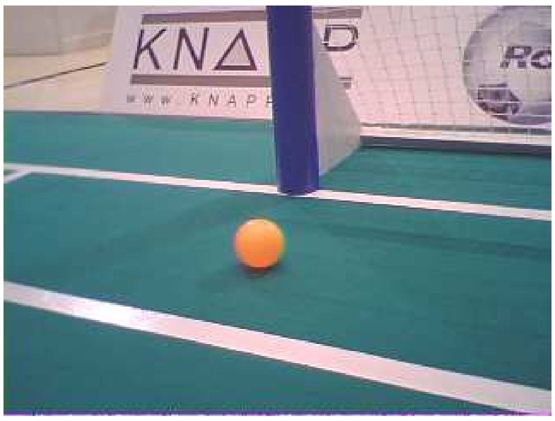

Fig. 3. Classical view facing goal and ball

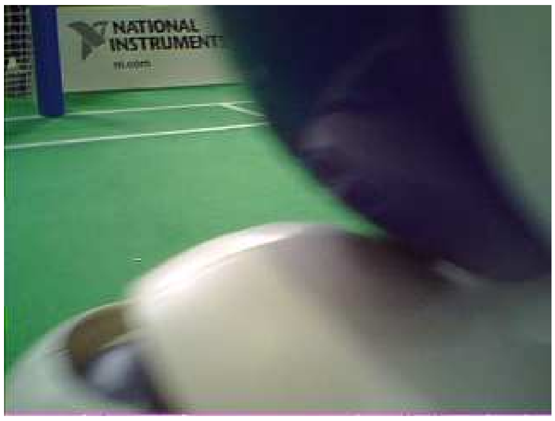

Fig. 4. Classical view facing moving Nao

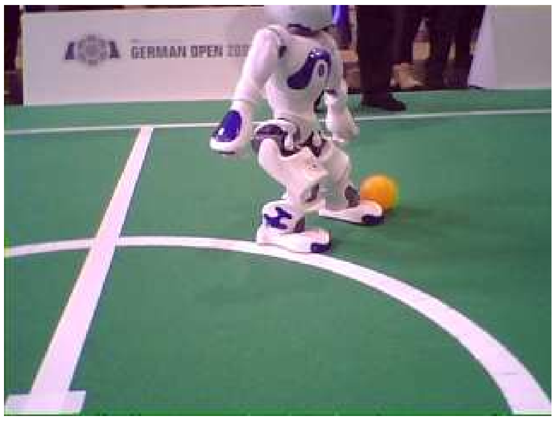

Fig. 5. Classical view facing Nao with the ball

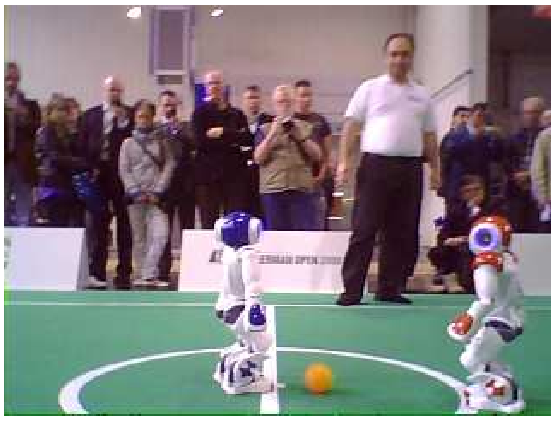

Fig. 6. Classical view facing soccer field and public

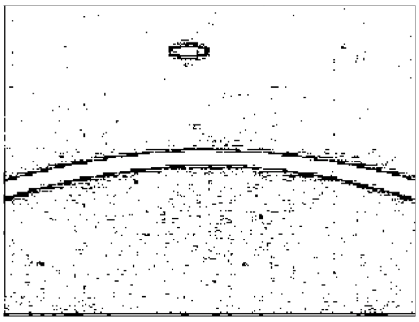
Fig. 7. Previous Fig. 1 view with Alg. 1

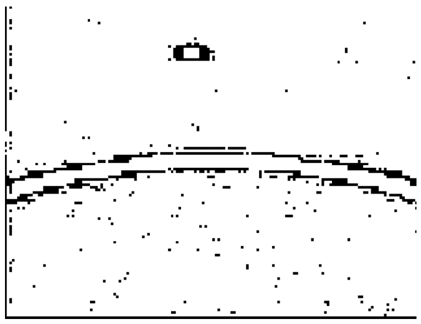
Fig. 8. Previous Fig. 1 view with Alg. 1 and a subsampling policy

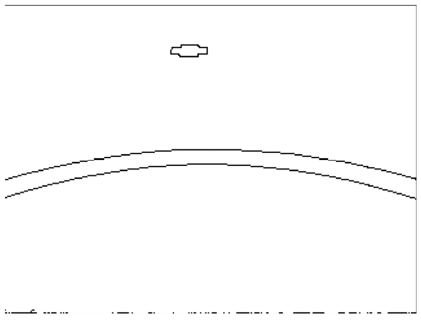
Fig. 9. Previous Fig. 1 view through Canny

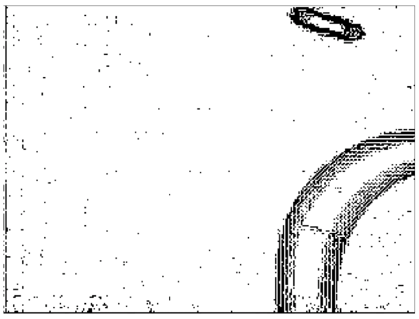
Fig. 10. Previous Fig. 2 view with Alg. 1

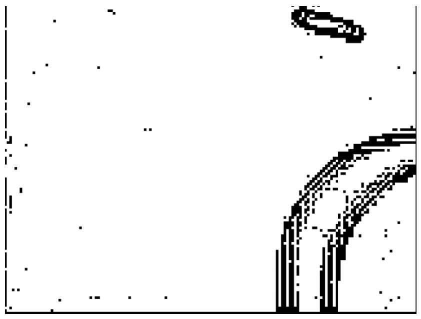
Fig. 11. Previous Fig. 2 view with Alg. 1 and a subsampling policy

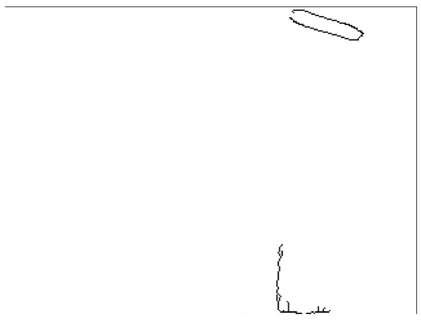
Fig. 12. Previous Fig. 2 view through Canny

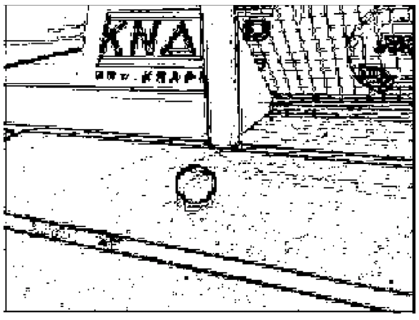
Fig. 13. Previous Fig. 3 view with Alg. 1

Fig. 14.

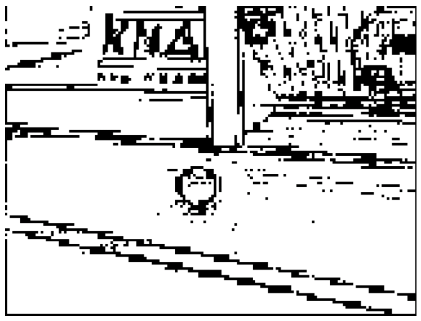
Fig. 15. Previous Fig. 3 view with Alg. 1 and a subsampling policy

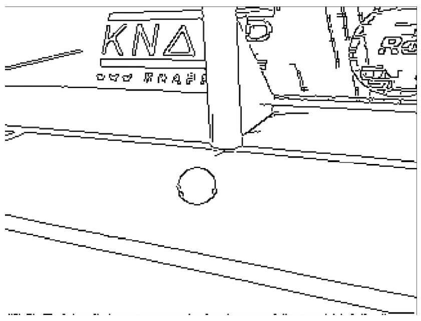
Fig. 16. Previous Fig. 3 view through Canny

Proceedings of the 4th Workshop on Humanoid Soccer Robots
A workshop of the 2009 IEEE-RAS Intl. Conf. On Humanoid Robots (Humanoids 2009),
Paris(France), 2009, December 7-10
ISBN 978-88-95872-03-2
pp. 38-44

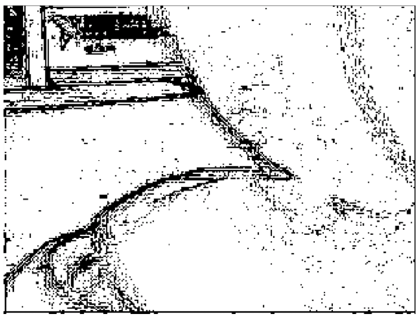
Fig. 17. Previous Fig. 4 view with Alg. 1

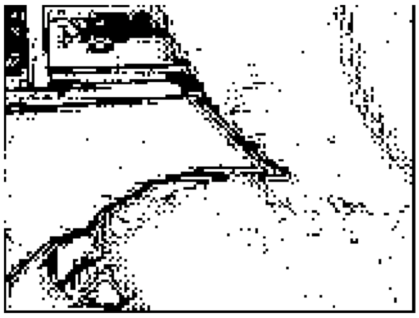
Fig. 18. Previous Fig. 4 view with Alg. 1 and a subsampling policy

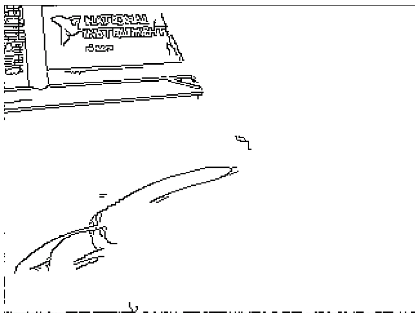

Fig. 19. Previous Fig. 4 view through Canny

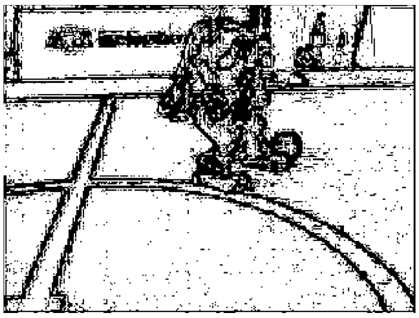
Fig. 20. Previous Fig. 5 view with Alg. 1

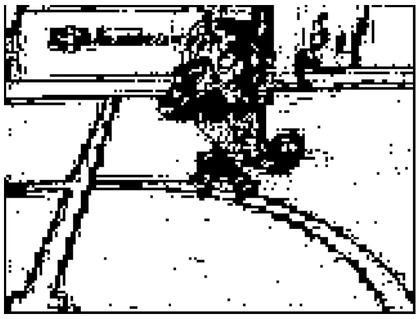
Fig. 21. Previous Fig. 5 view with Alg. 1 and a subsampling policy

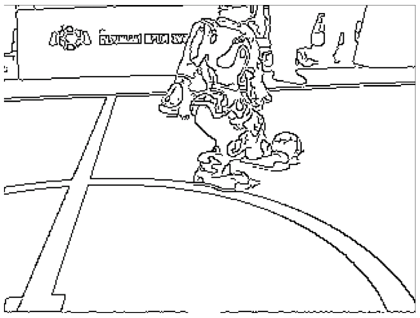
Fig. 22. Previous Fig. 5 view through Canny

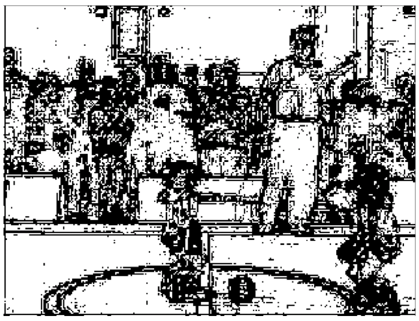
Fig. 23. Previous Fig. 6 view with Alg. 1

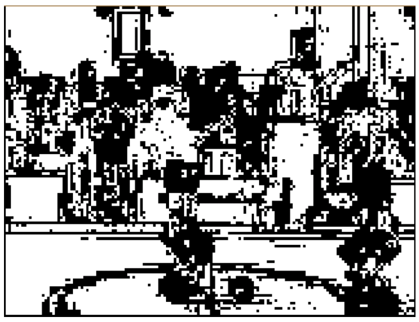
Fig. 24. Previous Fig. 6 view with Alg. 1 and a subsampling policy

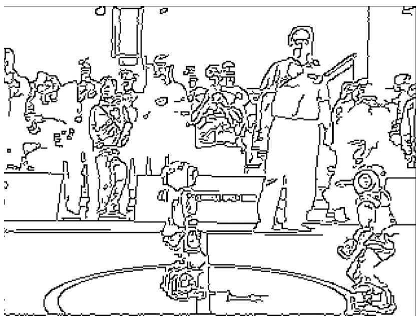
Fig. 25. Previous Fig. 6 view through Canny

Table II shows the results with such policy with 320x240 intial image size (subsampling policy used with Alg. 1 equals to standard sampling for a 160x120 image). 3x3 regions are 4 times smaller than in Table I, proportionally to its initial number of pixels. Over regrouping regions, this ratio varies from 3 times and greater. Blurred images remain slightly complicated. Ratios between each step are closely the same. After region segmentation, some pixels are out of any region. These pixels are located in edges and in regions smaller than 3x3. Figures 7 to 25 show such results and compare them to Canny edge detector openCV implementation [18]. Table III gives the times (in 10^{-3}sec. computed with Nao cpu-unit) for both region segmentation policies. Our algorithm with subsampling is faster than Canny for complex scenes. Canny produces less edge points but fails to grab blurred parts (see Fig. 12 and 19).

As openCV Canny implementation expects a grayscale input image, each YUV input image has to be converted in gray-level. As Alg. 1 deals simultaneously with 3 planes (YUV), a similar edge detection based on such Canny edge detector would take at least 3 times more than this one.

TABLE III
COMPARISON WITH CANNY EDGE DETECTOR

	Alg. 1		Alg. 1 with subsampling		Canny	
Fig. 1	Fig. 7	19.90	Fig. 8	12.84	Fig. 9	19.82
Fig. 2	Fig. 10	23.66	Fig. 11	12.23	Fig. 12	19.42
Fig. 3	Fig. 14	48.56	Fig. 15	25.20	Fig. 16	19.23
Fig. 4	Fig. 17	36.88	Fig. 18	20.11	Fig. 19	16.22
Fig. 5	Fig. 20	37.25	Fig. 21	17.88	Fig. 22	22.95
Fig. 6	Fig. 23	29.70	Fig. 24	11.84	Fig. 25	25.73

IV. SEGMENTED REGIONS INTERPRETATION

In order to interpret all areas defined by region segmentation, we use a Color Lookup Table to simplify the input regions into regions of interest as follows:

```
void regionInterpretation(R′)
 1 for each r in R′
 2   r.supposed_id = null
 3   for each pixel p of r
 4     for each Lookup entry e
 5        if p has valid e
 6          r.e identity ++
 7   r.supposed_id = max(r.e identities)
 8   if r.supposed_id == null
 9     remove r from R′
10   if r.supposed_id > threshold
11     r.true_id = r.suppose_id
12     add r to R′′
13     remove r from R′
14 for each r1 in R′′
15   for each r2 of R′ surrounding r1
16    if r2.supposed_id == r1.true_id
17       and r2.supposed_id > threshold
18      r2.true_id = r1.true_id
```

This process is divided in a two step process. The first defines truly assumed regions interpretation (lines 1 to 13). The second defines the interpretation of adjacent regions (lines 14 to 18). `R′` regions are converted to `R′′` regions, that are clearly associated to Color Lookup Entries. In our CLUT, one pixel can be associated with multiple object identity. Therefore, the identification is validated if a real maximal identity can be found with respect to others (line 11). During this first process, each identity is associated with an accumulation. Regions without identity are removed from `R′`. For the second process, this accumulation must reach a minimal threshold to allow identity propagation (line 17). At the end, we regroup regions with same true identity and others (without true identity) are discarded. Therefore, lonely pixels are discarded due to their region identity and lonely regions are discarded if they are not truly identified.

Table IV gives averages and standard deviations for number of regions over a classical half-time. The classical region segmentation produces more regions than the subsampled segmentation. The interpretation of classical sampling produces less regions than the interpretation of subsampled segmentation.

TABLE IV
INTERPRETATION SEGMENTATION RESULTS

	classical sampling		subsampling	
step	average	std deviation	average	std deviation
3x3	7929	288	1982	75
9x9	2153	562	657	154
27x27	1684	667	559	179
r27x27	1624	680	548	182
r9x9	1395	660	484	184
r3x3	348	259	117	88
r”	6	4	12	6

Fig. 26 shows region segmentation of Fig. 3. Fig. 27 shows ball region segmentation. Fig. 28 shows region interpretation of Fig. 26. With subsampling policy, these figures are respectively Fig. 29, 30 and 31. It shows that the number of regions can grow up for smaller objects (like the ball). Finally, edges between these regions are lost. We think that it would be interesting to recover them during the region interpretation process.

V. CONCLUSION

We have presented two algorithms that deal with region segmentation and interpretation. We have addressed their robustness over subsampling to decrease time processing. The speedup over the subsampling policy is approximately twice. The interpretation process automatically selects regions by proximity of truly identify regions and permits to aggregate color groups to identify scenes according to a classical predefined color look-up table.

REFERENCES

[1] A. D. Cabrol, P. Bonnin, T. Costis, V. Hugel, and P. Blazevic, “A new video rate region color segmentation and classification for sony legged robocup application,” in *Lecture Notes in Computer Science, RoboCup 2005: Robot Soccer World Cup IX., Springer-Verlag,* 2005.

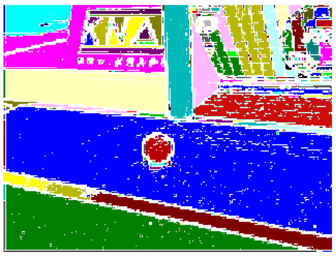

Fig. 26. Region segmentation of previous Fig. 3

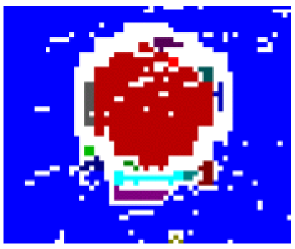

Fig. 27. Focus on the ball of previous Fig. 26

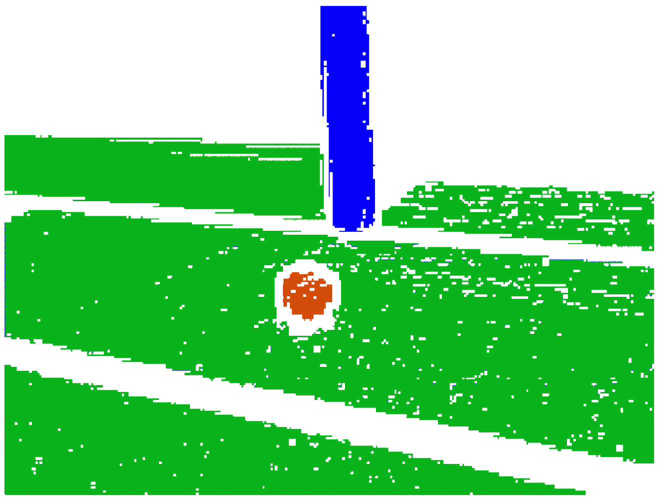

Fig. 28. Region interpretation of previous Fig. 26

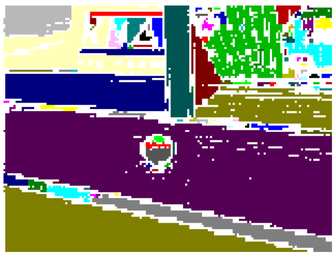

Fig. 29. Region segmentation of previous Fig. 3 with subsampling policy

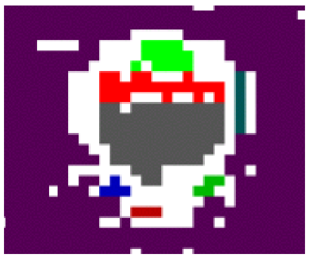

Fig. 30. Focus on the ball of previous Fig. 29

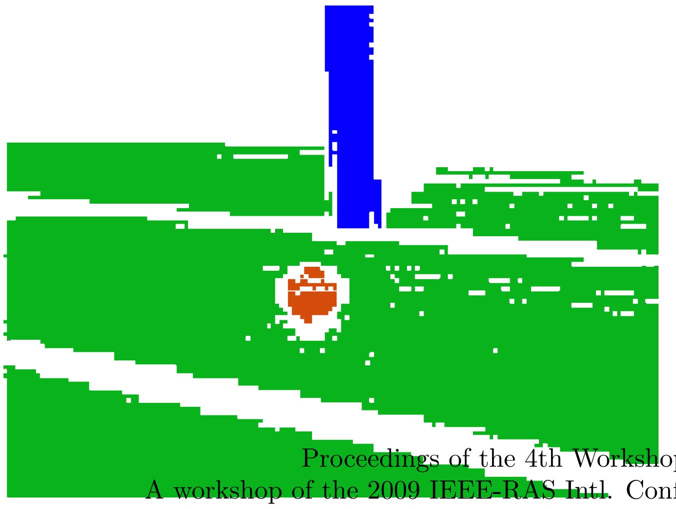

Fig. 31. Region interpretation of previous Fig. 29

[2] A. Arsenio, “Object recognition from multiple percepts,” in *IEEE-RAS/RSJ International Conference on Humanoid Robots*, 2004.

[3] B. Rani, N. Nandhitha, and N. Manoharan, “Euclidean distance based color image segmentation algorithm for dimensional characterization of lack of penetration from weld thermographs for on-line weld monitoring in gtaw,” in *17th World Conference on Nondestructive Testing, , Shanghai, China*, 25-28 Oct 2008.

[4] S. Gächter, “Results on range image segmentation for service robots,” in *Laboratoire de Systèmes Autonomes (LSA), Ecole Polytechnique Fédérale de Lausanne (EPFL), EFPL-LSA-2005-01*, 2005.

[5] S. Horowitz and T. Pavlidis, “Picture segmentation by a directed split and merge procedure,” in *Computer Methods in Images Analysis*, 1977, pp. 101–11.

[6] S. Zucker, “Region growing: Childhood and adolescence,” *CGIP*, vol. 5, no. 3, pp. 382–399, September 1976.

[7] S. Beucher, “Segmentation dimages et morphologie mathématique (in french),” Ph.D. dissertation, ENSMP - CMM Centre de Morphologie Mathématique, ENSMP., 1990.

[8] S. Lefevre, “Segmentation par ligne de partage des eaux avec marqueurs spatiaux et spectraux,” in *Colloque GRETSI sur le Traitement du Signal et des Images*, september 2009.

[9] M. Khiyal, A. Khan, and A. Bibi, “Modified watershed algorithm for segmentation of 2d images,” in *Issues in Informing Science and Information Technology*, Volume 6 2009.

[10] P. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[11] B. Mičušík and A. Hanbury, “Automatic image segmentation by positioning a seed,” in *Computer Vision ECCV 2006*, 2006, pp. 468–480.

[12] B. Mičušík and T. Pajdla, “Multi-label image segmentation via max-sum solver,” in *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on*, 2007, pp. 1–6.

[13] P. Andres Arbelaez and L. Cohen, “Segmentation d’images couleur par partitions de voronoï,” in *GRETSI, Saint Martin d’H/‘eres, France*, 2004.

[14] C. Thomas, B. Hoeltzener, and B. Zavidovique, “When milgram met ohlander,” in *Congrès COGNITIVA 90*, Novembre 1990, Madrid, pp 791-796.

[15] C. Pantofaru and M. Hebert, “A comparison of image segmentation algorithms,” Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-40, September 2005.

[16] R. Unnikrishnan, C. Pantofaru, and M. Hebert, “Toward objective evaluation of image segmentation algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 929–944, June 2007.

[17] T. Röfer, “Region-based segmentation with ambiguous color classes and 2-d motion compensation,” in *RoboCup 2007: Robot Soccer World Cup XI, No. 5001, pp. 369376, Lecture Notes in Artificial Intelligence. Springer*.

[18] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.