Evolving Visual Object Recognition for Legged Robots

Juan Cristóbal Zagal, Javier Ruiz-del-Solar, Pablo Guerrero, and Rodrigo Palma

Department of Electrical Engineering, Universidad de Chile
{jzagal,jruizd,pguerrer,ropalma}@ing.uchile.cl

Abstract. Recognition of relevant game field objects, such as the ball and landmarks, is usually based upon the application of a set of decision rules over candidate image regions. Rule selection and parameters tuning are often arbitrarily done. We propose a method for evolving the selection of these rules as well as their parameters with basis on real game field images, and a supervised learning approach. The learning approach is implemented using genetic algorithms. Results of the application of our method are presented.

1 Introduction

Teams of the four legged league have generally reported good vision strategies for the recognition of objects on the game field [3,14], such as the ball, landmarks and goals. Approaches often rely on the sequential application of a set of recognition rules, such as making comparisons of sizes and distances between regions of connected pixels of certain colors. These rules are pre-engineered and their parameters are manually adjusted until they become useful at recognizing objects under different locations, illuminations, and poses into the image. After applying theory and models to the problem, the engineering task often falls into a trying and error optimization process. Some teams [14] have even reported their uncertainty with respect to the application of some rules.

We believe that this engineering process can be supported or even automated by the use of a supervised learning approach, for which a large set of real pre-classified images can be used as a training data. These real images are expected to cover as much as the interesting examples that one might imagine. A main advantage of this approach is that such system will gain its knowledge from its own experience rather than being product of an arbitrary design.

Machine learning methods like evolutionary computation provides optimization tools which are used in robotics for learning behaviors [5], such as the case of evolutionary robotics, and also for the adaptation of perceptual systems [6,10,11,12]. A main idea is that genetic algorithms can search for solutions on highly dimensional spaces contaminated with natural noise. Evolved systems are expected to be more robust to unseen data sets than those resulting from simulated evolution.

In this work we will explore a method for evolving the selection and tuning of a group of visual object recognition rules, intended for recognizing objects in the context of the RoboCup four legged league. The proposed evolutionary learning approach is based on the use of a large set of pre-classified real images. In this paper are presented results for the detection of ball, goals, and landmarks. We are currently work-

D. Polani et al. (Eds.): RoboCup 2003, LNAI 3020, pp. 181–191, 2004.

ing towards the final goal of this approach, which is to derive rules for the recognition of other robot players into the game field. Given the complexity of this problem, it has not yet received sufficient attention from the research community.

In section 2 the related work is presented. Section 3 describes our implemented vision module which is used for extracting candidate object regions, and section 4 describes our proposed object recognition approach. Section 5 describes our results, and finally section 6 presents the conclusions and projections of our work.

2 Related Work

Cliff et al [4,5] uses genetic algorithms for evolving neural-network based controllers for visually guided robots. They use a computer graphics based model for simulating the robot vision, their model considers the introduction of certain amount of noise for preventing it from being entirely deterministic. The resulting approach was computationally expensive and with poor image resolution.

The approach of using evolutionary computation for computer vision problems has been widely explored, for example Köppen et al [9] proposes framework for the automated generation of texture filters using both, genetic algorithms (GA), and genetic programming (GP).

The object recognition problem, addressed with evolutionary computation, has been first attacked for the character recognition task. Koza [7] shows an experiment using GP for the classification of just four characters on small bitmaps, this approach relies on using a computationally expensive attention marker method. He also proposes a system which uses Automatically Defined Functions (ADFs) which were successful at finding solutions [8], but required populations of extremely large size (8000 individuals). Andre [1] uses both GP and GA simultaneously, first a GA determines feature templates and then a GP is used for classifying character bitmaps.

Teller and Veloso [13] used genetic programming for their proposed Parallel Algorithm Discovery and Orchestration (PADO) system. This system performs object recognition on real gray scale images. Genetic programming is used to induce programs which operate on pixel values in the image and return a confidence value that the given image corresponds to the class which is intended to be recognized.

3 Our Vision Module

The software architecture of our UChile1 four legged team is divided in task oriented modules. One of them, the vision module, is in charge of recognizing relevant objects from the images captured with the robot cameras. This module in particular, is mainly inspired on the large experience showed by the UNSW and CMPack teams [3,14]. This module is decomposed into four processing sub-modules: color segmentation, run-length encoding, labeling of connected regions, and finally object recognition.

For the color segmentation sub-module we use a look-up table of 64 levels in each YUV dimension, the table is generated by taking a large number of color samples (about 5000) from images of the game field. Once all samples have been collected, a median filter is operated over the look-up table values having the effect of clearing the interfaces between clusters of different colors and filling empty elements inside clusters which were not assigned during data collection. This process is particularly useful for solving ambiguities between red and orange clusters for example. A main consideration is that we train not just our seven color classes, but also a class for the set of non-relevant colors.

The output of the labeling sub-module is a set of connected regions of certain color, or blobs. Each blob can be characterized with a set of descriptors such as the size in pixels, the integer color index which in this case might take the values {0,1,2,3,4,5,6}, a set of coordinates describing the bounding box, and the coordinates of its center of mass.

The task of the object recognition sub-module is to identify image regions which are related to the relevant game field objects. The recognition of objects is performed by evaluating the response of a set of rules. For example, the detection of a ball usually requires that the related blob has the color of the ball, and if this is not the case one might expect to reject this candidate blob. These rules operate over all the image blobs or over combinations of them, such as pairs of blobs.

4 Learning Visual Object Recognition

4.1 General Approach

We propose to evolve the visual object recognition sub-module by first collecting reference region descriptors of objects which are present on a large set of real images; this stage is performed by an expert user. Then candidate regions are defined as those automatically extracted with the vision system, or combinations of them, see Figure 3 (left). Then, under a supervised rule learning process, candidate regions are compared with corresponding reference regions on each image, and the overall degree of correspondence serves as fitness for a genetic algorithm which learns the system recognition rules. Clearly, the effectiveness of the recognition sub-module is directly related to the degree of correspondence between candidate regions and reference regions.

We have used in our experiments a set of 180 real images for reference accumulation; these images contain objects such as the ball, landmarks and goals, as well as non relevant objects on the surroundings of the game field. The images consider a broad range of viewpoints, rotations, non canonical poses, and even variations on the illumination conditions. Figure 1 shows examples of these images.

In order to generate a database containing object identifiers for each reference image blob, or the so called references, we have developed a software which allows an expert user to define image regions related to relevant objects in terms of their bounding rectangles, and linking them to their corresponding identifier by just pressing on the corresponding object button. Figure 2 shows a screenshot of this software tool.

During the learning process a genetic algorithm evolves a population of recognition rules intended for detecting a particular object. These rules operate over region descriptors which are automatically extracted from each image with our vision module. In case a region, or a combination of them, is regarded as an object, its degree of correspondence with the reference is calculated by means of a correspondence quality function. The overall degree of correspondence between detected regions and references is then used as fitness for each individual generated with the genetic algorithm.

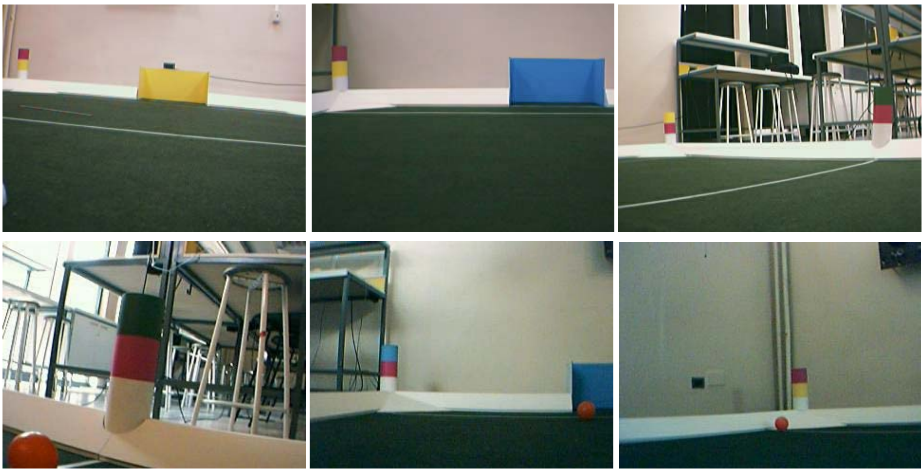


Fig. 1. Examples of images collected from the game field. Each image is subject to inspection from both an expert and our visual system under adaptation.

4.2 Fitness Function

Assigning a good fitness function is not trivial in this case. This measure should take its maximum when there is a perfect overlap between reference and candidate image regions, but it is not necessarily clear how to handle partial overlaps between them. Köppen et al [9] has proposed a quality function which well fits on our problem. It consist on measuring the area A of the reference region which does not overlaps the candidate, the area B of the overlapping region, and the area C of the candidate region which does not overlaps the reference, see Figure 3 (right). This results in the following three measures:

$r_1 = B / (A+B)$, the relative amount of correct overlapping pixels within the reference,
$r_2 = 1 - (C / (Q - A - B))$, the relative amount of correct empty pixels within the image, where Q is the total number of image pixels, and
$r_3 = B / (B+C)$, the relative amount of correct overlapping pixels within the candidate.

The intention is that genetic search increases all these measures, but we can identify some priorities among them. For example it is desired that the correspondence degree counts better for subsets of the reference, as well as for subsets of the reference which are supersets of other subsets of the reference. We also would like to refuse to assign good correspondence degrees to false positives, i.e. empty regions.

The following weighted correspondence degree, as proposed by Köppen, accounts for these requirements:

$$CD = 0.1r_1 + 0.5r_2 + 0.4r_3 \tag{1}$$

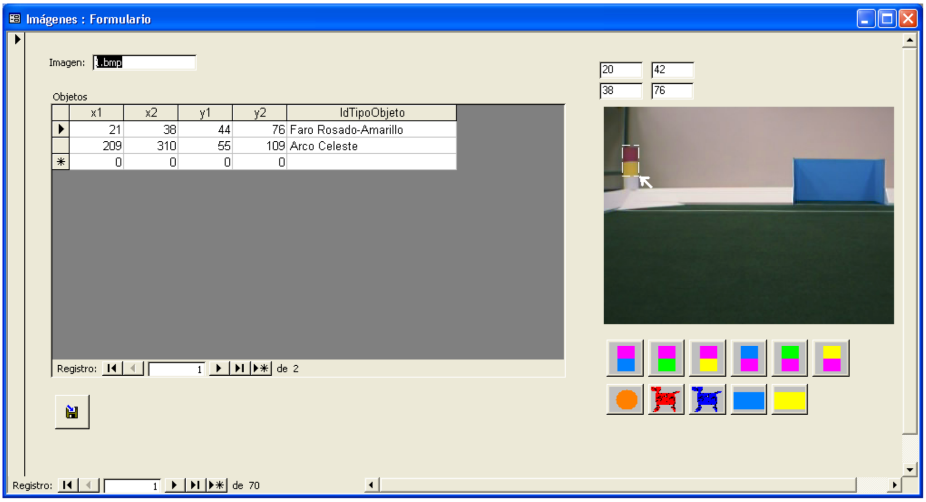


Fig. 2. A screenshot of the software developed for labeling image regions which are related to game objects. It can be seen how coordinates (left) defining rectangles, and object identifiers, are related to each object which is being selected. This task is performed by just pressing the corresponding object button (right).

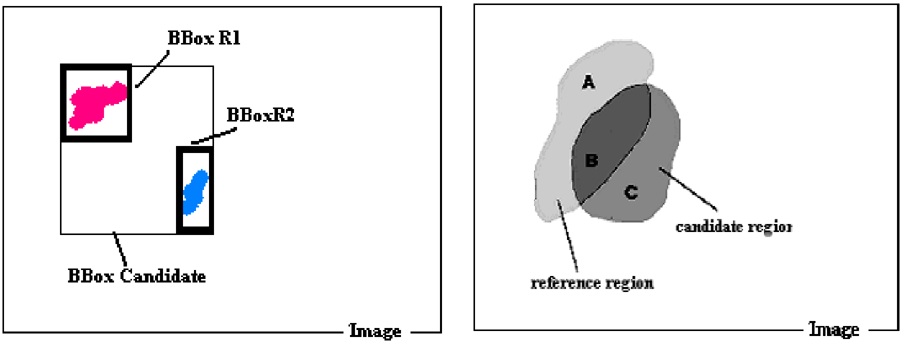


Fig. 3. Left: Illustration of how the candidate image region for the beacon detection is derived from two image regions. It can be seen how the resulting rectangle is defined in terms of the region bounding boxes. Right: An example of partial overlap between a reference region and a candidate region. The sub regions A, B and C are defined as the figure shows.

Using this measure, we compute the fitness for each individual as the sum of the correspondence degrees over the whole set of images, only when at least one candidate region is being selected:

$$fitness = \sum_i CD_i \tag{2}$$

As a consequence, genetic search evolves the population towards the higher weighted objective first, in this case r_2, rejecting false positives, and then towards r_3, allocating correct portions of the reference. Once candidates correspond to subsets of the reference, the fitness is increased by expanding them to cover the whole reference.

4.3 Genetic Rule Representation

For each candidate region, detected with the vision process, a set of N binary rules $R_i=f(\{p_{ij}\},\{Region\_Descriptors_k\})$ is evaluated. Each rule is described in terms of M parameters p_{ij}. The rules have as argument the set of candidate region descriptors. In general the binary rules have the following structure:

$$R_i = \begin{cases} 1 & if \quad p_{i1} \geq COND \geq p_{i2} \\ 0 & \quad if \quad not \end{cases} \tag{3}$$

Where *COND* might correspond to a value, as for example the size of a region, the quotient between regions sizes, and in general to the result of logical or arithmetic operations performed between the region descriptors. Each candidate region receives a score computed as the weighted sum of the rule outputs. The region having a maximum score is regarded as an object if and only if its score is greater than a certain threshold. This score is computed as follows:

$$Score = \sum_i w_i R_i \geq T \tag{4}$$

In our implementation, the weights $w_i \in [0,1]$, the thresholds $T \in [0, N]$ and all the M parameters $p_{ij} \in [0,1]$ are represented as 16 bit strings. Thus the chromosome which encodes a rule has length $16x(N+M+1)$, where M is the total number of rule parameters. In some cases, as it will be indicated, the parameters are re-scaled or discretized to a reduced set of values. These chromosomes will be evolved with a genetic algorithm. This algorithm uses fitness-proportionate selection with linear scaling, no elitism scheme, two-point crossover with a crossover probability Pc=0.75 and mutation with a mutation rate of Pm=0.015 per bit. The population size is 8 individuals evolved over a course of 100 to 150 generations.

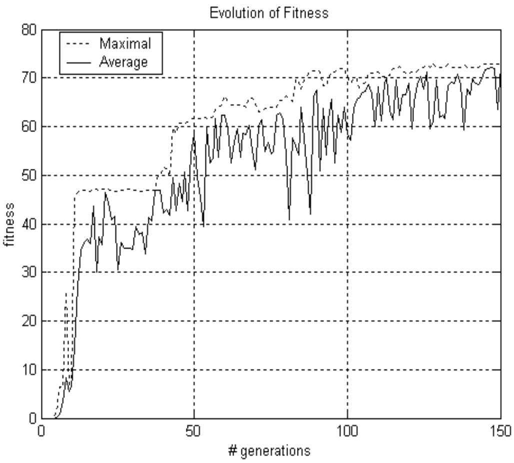
5 Results

5.1 Ball Recognition Experiment

We have chosen a group of six rules for the ball recognition experiment. The shapes of these rules, as well as the range of their parameters are indicated on Table 1. Figure 4 shows results of this experiment. We can first notice from these results, that the trivial color test implicit on rule R6, is satisfied, i.e. the system learns to choose the right color of the ball. The parameter P61 is discretized as the integer number 7 which correspond to exactly the index of the orange color. It is also recognized as the most important rule, since it has the maximum weight. The second rule on importance is R1 the low value of its parameter P11 indicates that the candidate region should have a minimum width of two pixels. The third rule in importance, R4 indicates that the bounding box of the region should be relatively square with a minimum quotient between width and height, or vise versa, of P41=0.5754. The fourth rule, R5 indicates that the quotient between the region size and the bounding box size should fall between P51=0.027 and P52=0.553, which is quite logical for our implementation. The lower bound accounts for those cases in which the ball correspond to a silhouette of segmented orange pixels, i.e. a region small in size but with a large bounding box. The upper bound serves for rejecting orange regions which are too close to a square. The fifth rule, R2 indicates that the region height should be at least a quarter of the image height. The last rule, R3 establishes that the region size should be at least one third of the image size. However, we should notice that the weight of R3 is quite low therefore it is not a relevant rule. The resulting threshold scaled by 6 (the number of rules) correspond to T=0.99, which can be compared to the maximum theoretical score of 3.69 (the sum of weights). This means that the threshold is set at the 26% of the maximum score.

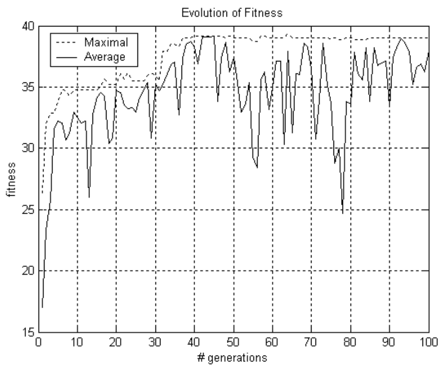
Table 1. The six rules for the ball recognition experiment, described in terms of their shape, and parameters range. The region descriptors correspond to **width_reg**, the region width; **height_reg**, the region height; **area_reg**, the region area; $\mathbf{h_{bb}}$, the height of the region bounding box; $\mathbf{w_{bb}}$, the width of the region bounding box; and **color_reg**, the color of the region.

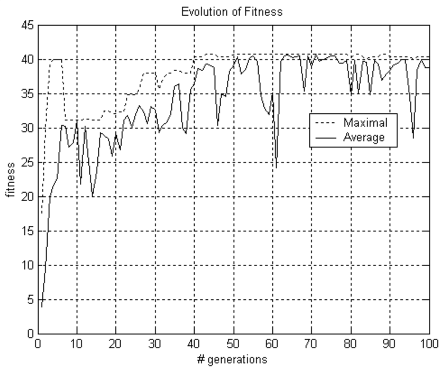
Rules for Ball Detection		
Rule	Activation Condition	Parameters Range
R_1	width_reg $> P_{11}$	integer [0,image_width]
R_2	height_reg $> P_{21}$	integer [0,image_height]
R_3	area_reg $> P_{31}$	integer [0,image_size]
R_4	$\min(w_{bb}/h_{bb}, h_{bb}/w_{bb}) > P_{41}$	double [0,1]
R_5	$P_{51} <$ area_reg/area_bbox $< P_{52}$	double [0,1]
R_6	color_reg $= P_{61}$	integer[1,num_colors=7]



Best Evolved Individual		
Pr.	**Value**	**Interpretation**
w1	0.952347	Ranking 2
w2	0.45491	Ranking 5
w3	0.07095	Ranking 6
w4	0.805679	Ranking 3
w5	0.459213	Ranking 4
w6	0.961151	Ranking 1
P11	0.00372314	Minimum width 2 pix.
P21	0.253232	1/4 image height
P31	0.308493	1/3 image size
P41	0.575424	Relatively square
P51	0.0270081	Counts for silhouettes
P52	0.552963	Rejects false balls
P61	0.98053	Orange color index.
T	0.166499	Equivalent to 0.99

Fig. 4. Resulting evolution of fitness (left), and the resulting weights, parameters, and threshold for the best evolved individual (right), for the ball recognition experiment.





Best Evolved Individual Light Blue Goal		
Pr.	**Value**	**Interpretation**
W1	0.802551	Ranking 1
W2	0.784454	Ranking 2
W3	0.348038	Ranking 5
W4	0.366501	Ranking 6
W5	0.606674	Ranking 3
W6	0.661896	Ranking 4
P11	0.0901642	Min width 9%
P21	0.0627899	Min height 6%
P31	0.647183	Min size 6%
P41	0.65889	Square-like ratio
P51	0.400696	Lower area bound
P52	0.864853	Upper area bound
P61	0.779999	L-blue. color indx.
T	0.0565796	Equivalent to 0.34

Best Evolved Individual Yellow Goal		
Pr.	**Value**	**Interpretation**
W1	0.999039	Ranking 1
W2	0.524292	Ranking 5
W3	0.859741	Ranking 2
W4	0.0393372	Ranking 6
W5	0.83844	Ranking 3
W6	0.743652	Ranking 4
P11	0.0493469	Min width 5%
P21	0.589996	Min height 6%
P31	0.503348	Min size 5%
P41	0.0301056	Square-like ratio
P51	0.173755	Lower area bound
P52	0.9658875	Upper area bound
P61	0.134277	Yellow color indx.
T	0.0552063	Equivalent to 0.33

Fig. 5. Results for the light blue goal experiment (left), and for the yellow goal experiment (right). The evolution of fitness (top), and resulting weights, parameters, and thresholds for best evolved individuals (bottom), are indicated.

5.2 Goal Recognition Experiment

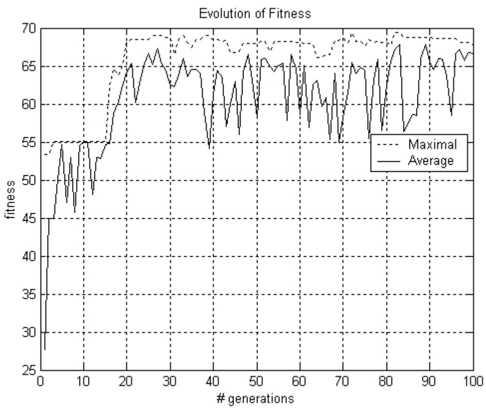
In general there are fewer rules reported for the goal detection than for the case of the ball. We will use the same group of rules which were used for the ball detection experiment, see Table 1. This group seems to be a good super set of relevant rules for our analysis. Figure 5 shows the results for the detection of the light blue and yellow goals. We have again that the color rule R6 obtains right parameters corresponding to light blue and yellow for each corresponding experiment. This rule is ranked in fourth place in both experiments. The most important rule is in both cases R1, which establishes a 9% and a 5% of the image width as lower bounds for candidate regions. The second place is for R2 in the light blue goal experiment and R3 in the yellow goal experiment. In both experiment it is established a minimum region height of 6% of the image height. Similarly, in both experiments, R3 establishes that the minimum region size should be 6% and 5% of the total image size. The third place is for R5 in both experiments, it establishes bounds for the quotient between the region size and its bounding box sizes, the resulting parameters are similar in both experiments. The fifth place is for R3 in the light blue goal experiment and for R2 in the yellow goal experiment. The less important rule as indicated in both experiments is R4, taking quite different parameters on each experiment. One interpretation of this is that genetic search concentrates on optimizing the parameter of relevant rules, leaving the irrelevant ones with arbitrary parameters. The thresholds are quite similar in both experiments; their corresponding scaled values are 0.33 and 0.34. The sum of weights is 3.569 for the light blue goal experiment, and 4 for the yellow goal experiment. Therefore the threshold is established at the 10% of the maximum score.

Table 2. The six rules for the ball recognition experiments, described in terms of their shape, and parameters range. The region descriptors correspond to **distX**, the distance between regions in the x axis; **distY**, the distance between regions in the y axis; **Sreg**, the region size; and **color_reg**, the color of the region.

Rules for Bacon Detection		
Rule	Activation Condition	Parameters Range
R_1	distX(reg1,reg2) $< P_{11}$	integer [0,image_width]
R_2	distY(reg1,reg2) $< P_{21}$	integer [0,image_height]
R_3	min(Sreg1/Sreg2,Sreg2/Sreg1) $> P_{31}$	double [0,1]
R_4	1/2 (Sreg1+Sreg2)/sqrt(dist(reg1,reg2)) $> P_{41}$	double [0,1]
R_5	color reg1 = P_{51} OR color reg2 = P_{51}	integer[1,num_colors=7]
R_6	Color_reg2 = P_{61} OR color reg2 = P_{61}	integer[1,num_colors=7]

5.3 Beacon Recognition Experiment

In this experiment candidate regions are generated as a combination of the bounding boxes of two image regions, see Figure 3 (left). In this case it is necessary not to just evaluate the rules over each region extracted from the image, but also to evaluate these rules over all the possible pairs of them, clearly the rules of this experiment have two region descriptors as input. The pair of regions which obtains the maximum score is selected if satisfies equation 4, and a candidate region is derived from them as indicated in Figure 3 (left), this region is used for calculating the correspondence degree as indicated in equation 1. For this experiment we have selected a group of six rules, presented in Table 2. We will explore the particular case of the category beacon of colors pink-light-blue and light-blue-pink without distinguishing between the vertical orders of the colors.



Best Evolved Individual		
Pr.	**Value**	**Interpretation**
W1	0.549803	Ranking 3
W2	0.40321	Ranking 5
W3	0.101231	Ranking 6
W4	0.538501	Ranking 4
W5	0.633870	Ranking 2
W6	0.875532	Ranking 1
P11	0.032041	Min distance in X 3%
P21	0.083218	Min distance in Y 8%
P31	0.0031345	Quotient sizes.
P41	0.832451	Relative distances
P51	0.896420	Color is pink
P61	0.7559361	Color is light blue.
T	0.138913	Equivalent to 0.83

Fig. 6. Resulting evolution of fitness (left), and the resulting weights, parameters, and thresholds for the best evolved individual (right), for the beacon recognition experiment.

It can be seen form these results that the two color rules R6 and R5 are regarded as the more important ones, their corresponding parameters fit exactly to the expected colors. The third rule on importance correspond to W1 which performs a minimal check for the horizontal distance between the blobs, the corresponding parameter establishes a threshold of 3% of the image width. The fourth rule in importance is R4 which checks for the distances between blobs with invariance to the scale of the objects, this rule was proposed in [14]. The fifth rule is R2 establishing a minimum vertical distance between regions of 8% of the image height. Finally R3 is regarded as the less important rule.

The maximum theoretical score is in this case 3.1 which means that the threshold was established at the 27% of the maximum score.

6 Conclusions and Projections

We have presented a method for automating and aiding the selection and tuning of visual object recognition rules in the domain of the RoboCup four legged league. The system shows to be consistent with the training data sets, and it allows the extraction of interesting parameters for different rules as well as the identification of the more relevant ones from a given set. It was particularly explored the case of ball, goal and landmark detection. We aim at extending this research, first we will explore rules for the detection of other robots into the game field, and then we will explore the application of a similar learning method for aiding the visual estimation of robot pose.

In the presented approach, the resulting parameters are dependent on the color calibration stage. If the color detection is poor or noisy, the resulting recognition system will be adapted to these specific conditions, with the consequence of having to retrain the system for each different lighting condition. This inconvenient is solved by ensuring accurate color detection. Our color detection system is accurate under different lighting conditions, and its calibration is performed in just about 15 minutes. In practice, we haven't had to perform the rule training when changing the lighting conditions.

The intention of this work is to present a method for the improvement of a vision system. Although we have just analyzed a particular blob-based vision system, we believe that a similar methodology can be applied for evolving other vision systems, such as grid based or corner based. Our intention is not necessarily to assess improvements of our vision system with respect to others, but to show that the result of this learning platform performs similarly. In a future work we expect to evaluate our system in terms of standard quantitative measures, using larger data sets. We also aim at comparing our visual system with those which are known to be successful within the RoboCup domain.

Acknowledgements

This research was partially supported by FONDECYT (Chile) under Project Number 1030500. We thank the reviewers for valuable comments on how to improve this paper.

References

1. Andre, D.: Automatically Defined Features – the Simultaneous Evolution of 2-Dimensional Feature Detectors and an Algorithm for Using Them. In: Kinnear, K. (eds.): Advances in Genetic Programming. MIT Press, Cambridge, Massachusetts (1994) 477-494.
2. Chalup, S., Creek, N., Freeston, L., Lovell, N., Marshall, J., Middleton, R., Murch, C., Quinlan, M., Shanks, G.: When NUbots Attack! The 2002 NUbots Team Report. Newcastle Robotics Laboratory. The University of New Castle. NSW 2308, Australia (2002) `http://robots.newcastle.edu.au/NUbotFinalReport.pdf`
3. Chen, S., Siu, M., Vogelgesang, T., Fai Yik, T., Hengst, B., Bao Pham, S., Sammut, C.: The UNSW RoboCup 2001 Sony Legged Robot League Team. In: Birk, A., Coradeschi, S., Tadokoro, S, (eds.): RoboCup 2001: Robot Soccer World Cup V. Springer-Verlag Berlin, Germany (2002) 39-48.
4. Cliff, D., Husbands, P., Harvey, I.: Evolving Visually Guided Robots. In: Meyer, J. A., Roitblat, H., Wilson, S. (eds.): Proceedings of the Second International Conference on Simulation of Adaptive Behaviour (SAB92). MIT Press Bradford Books, Cambridge, MA (1993) 374-383.
5. Cliff, D., Harvey, I., Husbands, P.: General Visual Robot Controller Networks via Artificial Evolution. In: Casement, D. (eds.): Proceedings of the Society of Photo-Optical Instrumentation Engineers Conference (SPIE-93), Session on Intelligent Robots and Computer Vision XII: Algorithms and Techniques, Boston, MA (1993)
6. Johnson, M., Maes, P., Darrell, T.: Evolving Visual Routines. In: Brooks, R., Maes, P. (eds.): Artificial Life IV. MIT Press (1994) 198-209.
7. Koza, J.: Genetic *Programming*. MIT Press, Cambridge, Massachusetts (1992)
8. Koza, J.: *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge, Massachusetts (1994)
9. Köppen, M., Teunis, M., Nickolay, N.:A Framework for the Evolutionary Generation of 2D-Lookup Based Texture Filters. In: Proceeding of Lizuka Conference, Lizuka, Japan (1998) 965-970
10. Martin, M. C.: Genetic Programming for Robot Vision. In: SAB 2002, the Seventh International Conference on the Simulation of Adaptive Behaviour (2002).
11. Martin, M.C.: Genetic Programming for Real World Robot Vision. In: Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), EPFL, Lausanne, Switzerland, September-October (2002) 67-72
12. Nguyen, T., Huang, T.: Evolvable 3d Modeling for Model-Based Object Recognition Systems, In: Kenneth, E. (eds.): Advances In Genetic Programming. MIT Press (1994) 459-475
13. Teller, A., Veloso, M.: PADO: A New Learning Architecture for Object Recognition. In: Ikeuchi, K., Veloso, M. (eds.): Symbolic Visual Learning, Oxford University Press (1996) 81-116
14. Veloso, M., Lenser, S., Vail, D., Roth, M., Stroupe, A., Chernova, S.: CMPack-02: CMU's Legged Robot Soccer Team. Carnegie Mellon University Report (2002) `https://www.openr.org/page1_2003/code2002SDK/CMU/cmu_teamdesc.pdf`