A Q-Learning Approach to Decision Problems in Image Processing

Alexandru Gherega, Radu Mihnea Udrea
University 'Politehnica' of Bucharest
Bucharest, Romania
alex.gherega@gmail.com, mihnea@comm.pub.ro

Monica Rădulescu
R&D Softwin
Bucharest, Romania
mradulescu@softwin.ro

Abstract—Decision making in a sequential and uncertain manner is still one of the major problems that all computer science research fields relate to, either by trying to find the optimal solution for it or needing such a solution to obtain optimal results for some other stated problem. This paper's objective is to address the possibility of using reinforcement learning for decision problems in data processing environments. A short review of current uses for reinforcement learning solutions into different fields is depicted as well as a particular case in image processing methods. A solution is proposed for a specific decision problem in the field of image processing. Our implementation shows the results of a reinforced parameterization method for edge detection using Q-Learning.

Keywords-reinforcement learning; Q-Learning; computer cognition; adaptive and learning systems.

I. INTRODUCTION

Reinforcement learning [1][2] is a way of discovering how to reach a specific desired state in order to maximize a numerical reward signal, i.e., how to map situations to actions based on the interactions with the environment. The learning entity is not told which actions to take, as in other forms of machine learning, but instead must discover which actions yield the best reward by random exploration. Actions may affect not only the immediate reward but also the next selected action and through that all subsequent rewards. These two characteristics, trial-and-error search and delayed reward, are the two most important features of reinforcement learning.

In the standard reinforcement learning model, described in Figure 1, an agent is interconnected to its environment via perception and interaction. On each interaction step the agent receives some information regarding the current state of the environment. The agent then chooses an action to generate as output. The action changes the state of the environment and the value of this state transition is communicated to the agent through a scalar reinforcement signal. The agent's desired behavior is to choose actions that tend to increase the long-term sum of reinforcement signals. An action selection policy is used by the reinforcement learning agent in order to choose the appropriate action to change the current state. The agent must find a trade-off between immediate and long-term rewards. It must explore the unknown states, as well as the states that maximize the reward based on current knowledge. A balance between the exploration of unknown states and the exploitation of known, high reward states is needed. There is a wide variety of reinforcement learning algorithms: SARSA, TD-Gammon, SARSA(λ) [1], Q-Learning [3], etc.

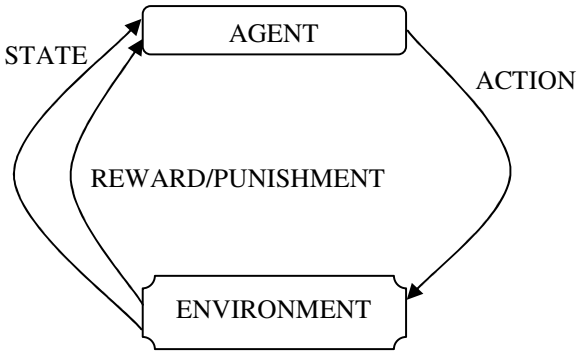


Figure 1. The components of a reinforcement learning agent

In this paper, an edge detector is automatically parameterized using the popular Q-Learning algorithm [3][4][5], which is a reinforcement learning technique that memorizes an action-value function and follows a fixed policy thereafter. One of the biggest advantages of Q-Learning is that it is able to compare the expected rewards without requiring a prior well known model of the environment. One of the policies used by Q-Learning is the Boltzman policy [3], which estimates the probability of taking an action with respect to a certain state. Other well known policies for Q-Learning are the ε-greedy and greedy policies [5]. In the greedy policy, all actions may be explored, whereas the ε-greedy selects the action with the highest Q-value in the given state with a probability of $(1 - \varepsilon)$ and the rest with probability of ε. For the approach proposed, in this paper a greedy policy was used.

Reinforcement learning serves both as a theoretical tool for studying a way entities learn to act under a certain environment as well as a practical computational tool for constructing autonomous systems that improve themselves based on cumulated experience [6][7]. These applications range from robotics and industrial manufacturing to

 ISBN: 978-1-61208-195-3

combinatorial search problems such as computer game playing [8][9][10]. Practical applications provide a test for efficiency and utility of learning algorithms. They are also an inspiration for deciding which components of the reinforcement learning framework are of practical importance.

Image processing represents one of the particular applications of the more general field of two-dimensional signal processing. Due to multimedia market explosion, digital image processing has become an interest area and attracts a lot of researchers from other computer science fields.

Modern digital technology has made it possible to manipulate multi-dimensional signals with systems that range from simple digital circuits to advanced parallel computers.

Digital image processing methods can be classified as follows: image processing (i.e., a bundle of operations such as filtering, labeling, segmentation, object detection, etc.), image analysis (i.e., a bundle of all measurements based on the information contained in the image itself, e.g., PSNR) and image understanding (i.e., collected regions and objects from an image are interpreted and classified, based on their content, for future processing). From these, image processing and image understanding involve solving decision making problems.

Decision making for image processing is a cumbersome process. Because an image's data may be large and highly redundant, it can be subject to multiple interpretations. In order to reduce the dataset size used in image processing, feature extraction methods are used. The use of certain features, that select the relevant information from the input data, simplifies the processing tasks and reduces the resources' costs. Feature extraction is used for a large range of image processing algorithms: face recognition, segmentation, image database indexing and retrieval, watermarking, medical imaging, image searching, etc. Some works divide the used features into different groups: color features, texture features and shape features [11].

Some of the most commonly used features are object size (area, volume, perimeter, surface), object shape (Fourier descriptors, invariant moments, shape measurements, skeletons), object color (space, integrated optical density, absolute and relative colors), object appearance/texture (co-occurrence matrices, run lengths, fractal measures, statistical geometric features), distribution function parameters (moments: mean, variance, median, inter-quartile range) [11]. The main issues with feature extraction algorithms are lack of flexibility and lack of adaptability with respect to input images and user requirements.

The rest of this paper is organized as follows. Section 2 presents a brief overview of reinforcement's learning success in different fields. In Section 3, a short overview of reinforcement learning usage in Image Processing and Computer Vision is reviewed. In Section 4, the proposed solution for automatic parameterization using Q-Learning shows the integration of reinforcement learning with a basic image processing technique (i.e., Sobel edge detector). Section 5 depicts the obtained results and the encountered implementation problems. The final section elaborates on the reached conclusions.

II. DIVERSITY OF REINFORCEMENT LEARNING APPLICATIONS

Current research reveals the potential of reinforcement learning techniques, in distributed environments with a high level of dynamism, for resources' allocations that induce a balanced utilization of the system's capabilities and maximizes the number of served applications [8]. Due to the fact that reinforcement learning has low scalability and the performances from the online training can be extremely reduced, a hybrid method of reinforcement learning and neural networks was proposed in [12] to address this issue.

In [13], a robot navigation model based on environment reference points is proposed. Some of the subcomponents of the navigation system are competing for the image acquisition system. Busquets et al. [13] use an auction based mechanism for solving competition and a reinforcement learning based method for tuning the auction functions' parameters. A doubled performance was achieved compared with the case of manually coding a navigation policy.

Reinforcement learning is useful in applications for control and administration systems as well. An implementation model for memory controllers is proposed in [14] based on the idea of self-optimization. The model's goal is to achieve efficient utilization of the transmission bus between a DRAM memory and a processor unit. Ipek et al. [14] use a reinforcement learning based implementation of an adaptive self-optimizing memory controller able to plan, learn and permanently adapt to processing requests. Ipek's et al. results show a performance improvement around 15%-20% over one of the best planning policy – First-ready first-come-first-serve.

Reinforcement learning techniques are successfully used in different medical areas such as medical imaging [15][16], individualization of pharmacological anemia management [17] or brain stimulation strategy for the treatment of epilepsy [18]. Usually, the information gathered from medical processes forms an excellent dataset for reinforcement learning. For this reason, more and more methods are being developed for decision making processes, feature extraction, separation policy's confidence interval calculation, etc.

The reinforcement learning paradigm was successfully used in autonomic systems for human interaction. Such an example, of human subject interaction, is a system for establishing a verbal dialogue with human subjects [19]. Henderson et al. [19] propose a hybrid learning method based on reinforcement learning and supervised learning (SL), which produced better results compared to other dialogue systems (40%). Just using SL offers a 35% improvement, where as just using reinforcement learning achieves performance under the level of other systems.

 ISBN: 978-1-61208-195-3

III. REINFORCEMENT LEARNING INTEGRATION IN IMAGE PROCESSING AND COMPUTER VISION

In the work of Sahba et al. [16], a reinforcement learning agent based segmentation technique is proposed for medical imaging. Due to a high level of noise, missing or diffuse contours and poor image quality, it is quite difficult to apply segmentation onto medical images.

The reinforcement learning agent's goal is to receive the highest reward by discovering the optimal parameters for each processing stage. The algorithm proposed by Sahba et al. consists of two stages: an offline stage and an online stage. During the first stage, the agent acquires knowledge – stored as a matrix – using a training set of images and segmentations of these images obtained by other methods. The behavior learned during the offline stage is applied by the reinforcement learning agent during the online stage in order to segment images from a test set. The algorithm implemented by Sahba et al. in [16] is a basic segmentation approach. The authors clearly state that their approach should not be compared with existing segmentation methods as the proposed method is just a prototype meant to show the possibility of using reinforcement learning with segmentation techniques.

Another approach for the same image segmentation methods is presented in [20]. In this paper, the results of segmentation are used in a complex object recognition algorithm. Current segmentation and object recognition computer vision systems are not robust enough for most real-world applications. In contrast, the system proposed in [20] achieves robust performance by using reinforcement learning to induce a mapping from input images to corresponding segmentation parameters. Using the reinforcement learning paradigm in an object recognition system is the major contribution in [20]. Through their research, Peng and Bhanu [20] show that, for a set of images, adaptive systems could be used for autonomous extraction of the segmentation criteria and for automatic selection of the most useful characteristics. The result is highly accurate recognition system.

The results of using reinforcement learning in the field of image processing do not stop to segmentation mechanisms. Decisions processes are an important part of edge detectors algorithms as well. In [21], the authors show the experimental results obtained by using a reinforcement learning based neural network for an edge detection algorithm. Similar to standard edge detection algorithms, a part of the image is used as input, in this case for the neural network. A key difference from standard edge detection algorithms is that the neural network doesn't use specific per-image parameters. The experimental results were compared with results from Canny and Sobel edge detection operators. The comparison underlines advantages in accuracy and efficiency.

Reinforcement learning is also used in text detection applications. In [22], ten parameters of text detection in video images are optimized using reinforcement learning.

Reinforcement learning was successfully used to discover the parameters needed in an image retrieval system. The research carried out by Srisuk et al. [23] uses a template comparison mechanism together with a reinforcement learning agent endowed with a Bernoulli policy in order to extract the necessary parameter for such a system.

The main issue with all methods depicted in this section is to eliminate the need of ground truth-based images. Although useful when evaluating the proposed frameworks, their use may be questioned when using online learning methods. In conclusion, the rewards must come directly from result evaluation (e.g., an optical character recognition (OCR) engine, image quality measurements). This will allow for online parameter optimization, which fully utilizes the benefits of reinforcement learning.

In some cases, the rewards come from the system's user. The work presented in [24] describes an approach to a custom image search engine, designed for e-learning. The proposed system takes the user's response as the reward for a reinforcement learning agent. The agent selects a set of images and suggests them to the user. The user chooses from this set the ones which bare any significance to it's search. As a result of this interaction the system learns a user's profile. On a future search the agent will compare the learned set with the returned result set and will chose only those images that match the user's preferences. In order to achieve a seamless interaction between the proposed system and the end users, the application in [24] is endowed with a graphical user interface as well.

A reinforcement learning agent is also used in face recognition algorithms [25] or in multiple filtering applications [26]. Most of all, applications for medical imagining and image retrieval were the first to use reinforcement learning due to the fact that a lot of bad quality images have to be processed, using variable parameters and human feedback.

IV. THE PROPOSED SOLUTION OF AUTOMATIC PARAMETERIZATION USING Q-LEARNING ALGORITHM FOR EDGE DETECTION

An edge detector is a simple algorithm to extract edges from an image. Applying an edge detector to an image generates a set of connected curves which follow the boundaries of objects (real objects, surfaces, texture changes, etc.). Edges play quite an important role in many image processing applications, particularly for machine vision systems that analyze scenes of man-made objects.

Most edge detection methods use either first-order or second-order derivative. The first order derivative of a point is considered to give a measure of the fortitude of an edge at that point while its local maximum's direction estimates the edge's local direction. The Sobel, Canny and Prewit edge detectors are just a few examples using the first-order derivative.

With the second-order derivative edge detection methods search for zero crossings which are local maxims of the gradient, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression (e.g., Marr-Hildreth) [27]. Most edge detection methods are based on a threshold that says how large the intensity change between two neighboring pixels must be in order to say that

 ISBN: 978-1-61208-195-3

there should be an edge between these pixels. Most of the times, the threshold depends on the image, its noise level or its details. As a result, the threshold is chosen manually, which makes these methods difficult to apply on large image datasets. The advantages of existing methods are simplicity and fast computing. Since they are part of other complex algorithms most of the times, it's important they are as fast and independent as possible.

In this section, an automatic parameterization for the Sobel edge detector is proposed (Figure 3). The application was developed using our in-house built Q-Learning framework, through which we try to attain more flexibility and scalability with our exploration of Q-Learning based applications. The general architecture for the Q-Learning framework is depicted in Figure 2.

The actual framework application is represented on the Framework Layer section and contains the following major components: a general learning agent based on the Reinforcement Learning paradigm (RL Agent), the Q-Learning implementation (QL Impl), a dictionary data structure (Policy) containing the parameterization for the Q-Learning algorithms – α (learning rate) and γ (discount factor), an extensible set of objective functions used when updating the Q-values. The usual updating function is the *maximum* function, as stated in the general theory – see equation (2), yet other functions could be used, e.g., *minimum*, *summation*, *mean*. Based on the previous components, the most important element of our framework is defined and implemented: the Q-Learning agent (QL Agent).

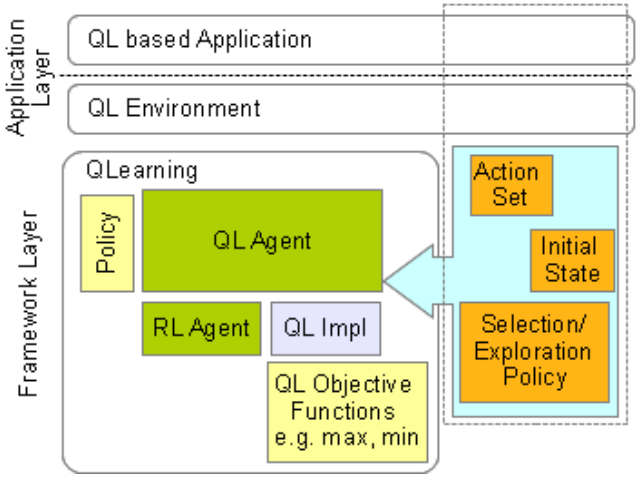


Figure 2. Q-Learning framework architecture

In order for the agent to interact with the environment, it needs to know a set of actions, start from an initial state and be endowed with an exploration/exploitation strategy. These latter components cannot be implemented in the framework and they are left as abstract elements, since they are tightly coupled with the application and environment particularities. This is suggested, in the architecture depicted in the Figure 2, by enclosing the action set, initial state and exploration policy into a dashed square that protrudes into the application layer.

The environment is both part of the framework as well as the application in the following sense:

- the way the Q-Learning agents are initialized, introduced and interlinked in/with the environment induces certain capabilities that the environment component must support (i.e., agents are an active part of the environment);
- each application depends on and exposes a certain environmental universe.

Using these components, the upper applications' instances are implemented based on the problem they try to approach, e.g., in this paper we used the framework to implement the application for finding an optimal edge detection threshold.

The model depicted in Figure 3, for the proposed automatic parameterization application, contains a Sobel edge detector, an evaluation module based on Pratt Figure of Merit [28] and a reinforcement learning agent that learns to find the optimal edge detection threshold for multiple images. A greedy policy Q-Learning algorithm is used. States are represented as values for the searched parameter (i.e., the edge detection threshold), the only two possible actions are increment and decrement of state's value and rewards are computed using Pratt Figure of Merit [28]. The Sobel edge detector extracts edges from the input images using a threshold value provided by the reinforcement learning agent. The extracted edges are assessed by the evaluation module and the result is used as a reward for the reinforcement learning agent, which takes the corresponding action and provides another edge detector threshold. The learning process continues until the optimal threshold is found.

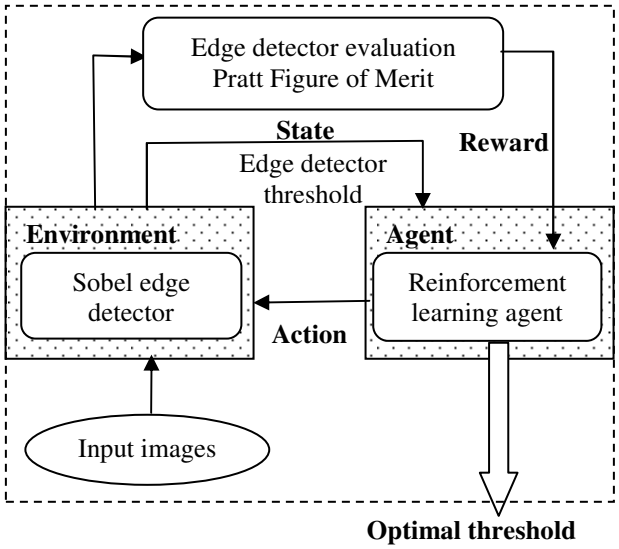


Figure 3. The proposed automatic parameterization for the Sobel edge detector

Pratt Figure of Merit is a well-known measure used to compare edge detectors' outputs. It attempts to balance three types of errors that can produce erroneous edge mappings: missing valid edge points, disconnected edge points and

 ISBN: 978-1-61208-195-3

misclassification of noise fluctuations as edge points. Pratt Figure of Merit uses a known edge for comparison. We consider a known edge the edge resulted from the output of a Canny edge detector based on a manual chosen threshold. For different edge detection thresholds, we visually analyze the resulting edges for the Canny detector and we chose the threshold that achieves the optimal possible edge.

The Figure of Merit is defined as:

$$R = \frac{1}{I_N} \sum_{i}^{I_A} \frac{1}{1 + \alpha d_i^2} \quad (1)$$

In the above equation, I_N is the maximum of I_A and I_I. The I_A value represents the total number of test edge pixels. The I_I value represents the total number of known edge pixels in the image. α is a scaling constant, while d_i is the distance from an actual edge point to the nearest known edge point. The scaling factor is used to penalize edge points that are clustered but away from the known edge pixels. After experimenting with other values we decided to use $\alpha = 0.9$, the value coined by Pratt in [28], a value that drastically penalizes pixels faraway from the known edge.

Q-Learning algorithm updates its Q-values using:

$$Q(s_t, a_t) = Q(s_t, a_t)(1 - \alpha) + \alpha [r_{t+1} + \gamma^{\max}_{a} Q(s_{t+1}, a)] \quad (2)$$

where r_t is the reward given at time t, α ($0 < \alpha <= 1$) the learning rate, may be the same value for all pairs. The discount factor γ is such that $0 \leq \gamma < 1$.

In this paper, a greedy policy is used. We use $\alpha = 1$ and $\gamma = 0.85$. The learning rate is used to establish the influence of the new acquired information against past experiences. A small factor will make the agent learn very little and rely solely on past experience – acquired or given as input premises, while a big factor would make the agent consider only the most recent information. The discount factor establishes the impact of future rewards. A null factor will make the agent consider just current rewards, while a big factor will make it achieve a high long-term reward.

V. EXPERIMENTAL RESULTS

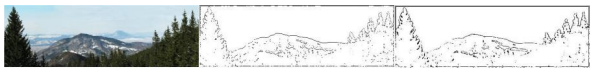
During the offline stage, the reinforcement learning agent is trained using three input images. In accordance with the Q-Learning algorithm, a total of 512 Q-values are computed during this stage with respect to each reward received for each taken action. For this stage the actions and the initial state are randomly selected. During the online stage we use the computed Q-values to determine the optimal threshold necessary for the edge detection algorithm. Starting from a randomly selected initial state, the agent chooses the action that will give a maximum reward.

The obtained optimal threshold is 56. This threshold can only be evaluated by analyzing the results produced by basic edge detectors with respect to it. We visually test the threshold by using it with Canny and Sobel edge detectors first on the training images (Figure 4b, 4c, 5b, 5c, 6b, 6c) and then on a test image (Figure 7). We used different kinds of real life images (landscapes, cartoons, portraits) to test the optimal threshold in vary conditions. Because edge detection methods are widely used in algorithms which address natural as well as artificial, medical and/or binary images, they must prove efficient in real life working scenarios.

One can notice that the edges, obtained using the automatically computed threshold, are continuous and have a small number of offset pixels. It can also be observed that the edges are not smeared.

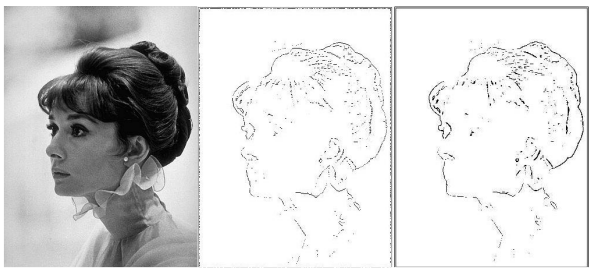
a b c

Figure 4. a. Original image. b. result of edge detector Canny with a 56 threshold. c. result of edge detector Sobel with a 56 threshold



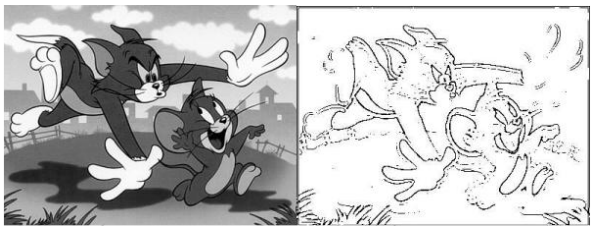
a b c

Figure 5. a. Original image. b. result of edge detector Canny with a 56 threshold. c. result of edge detector Sobel with a 56 threshold



a b c

Figure 6. a. Original image. b. result of edge detector Canny with a 56 threshold. c. result of edge detector Sobel with a 56 threshold

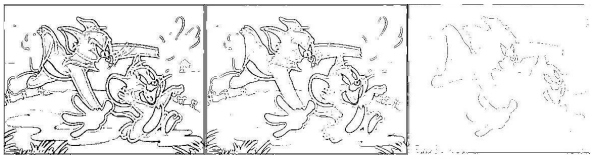


a b

Figure 7. a. Original image. b. result of edge detector Sobel with a 56 threshold

For additional evaluation of the detection threshold, we applied the Sobel algorithm to a test image and manually

selected the optimal threshold value. The obtained results are depicted in Figure 8. The optimal value chosen for the threshold was 52, a much close value to the one automatically achieved using Q-Learning algorithm.



a b c

Figure 8. Result of edge detector Sobel with a a. 29 threshold. b. 52 threshold c. 173 threshold

The proposed parameterization is an example of how reinforcement learning can be used to find optimal parameters for image processing algorithms. The presented results indicate that this solution can be efficient. Computing automatically the detection threshold for an edge detector provides the added benefit of being independent from the input image. As such, it can be used on multiple images with comparable performance. Although the threshold computed by our method may not be optimal for every image, it is sufficiently close enough to the optimal value. As such, the computed threshold provides at least a very good hint to what that optimal value should be.

The proposed algorithm is limited by the use of ground truth images as reference, images that must be known prior. For computing the set of rewards we used a set of reference contours extracted with the Canny detector. For natural images though, choosing the optimal threshold value by hand can be quite difficult to achieve. This issue could be solved by using an interactive system such that the reward, for a specific action, is given as a user's input.

VI. CONCLUSIONS

The paper's goal was twofold: to give a general overview of reinforcement learning as an optimization technique and to ascertain an insight over the benefits that can be drawn for decision making problems in the image processing field.

A short review of current research and the success of reinforcement learning techniques in various fields were presented and we studied the use of a Q-Learning approach to decision making with respect to an image edge detection problem.

We developed a reinforcement learning framework application using the Python programming language. Based on this framework the integration of Q-Learning algorithm was developed for the automatic parameterization of Sobel edge detector.

The obtained results show the potential of the proposed approach, while strongly indicating an improvement in speed, resources' utilization and usability as opposed to Sobel method.

We implemented the threshold discovery agent such that it learns - through the Q-Learning algorithm - to find an optimal threshold by using a greedy policy and a set of reference images. The results are good for test images of the same type as the references (e.g., nature, synthetic images, cartoons, etc.).

The use of reinforcement learning in image processing and computer vision is extending as a way of replacing human assistance with intelligent agents.

ACKNOWLEDGMENT

The work has been co-funded by the Sectoral Operational Program Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

REFERENCES

[1] R. Sutton and A.G. Barto, "Reinforcement learning: An introduction", The MIT Press, London, England, 2005, <retrieved: February, 2012>.

[2] L.P. Kaelbling and M.L. Littman, "Reinforcement learning: A survey", Computer Science Department, Brown University Providence, USA, 1996, <retrieved: February, 2012>.

[3] L. Busoniu, R. Babuska, and B. De Schutter, "Multi-agent reinforcement learning: An overview", Innovations in Multi-agent systems and applications, Studies in Computational Intelligence, Issue 310, 2010, pp. 183-221, <retrieved: February, 2012>.

[4] E.C. Mariano, P. Cuauhnahuac, and E.F. Morales, "Distributed reinforcement learning for multiple objective optimization problems", Congress on Evolutionary Computation, 2000, pp. 188 – 195, <retrieved: February, 2012>.

[5] T. G. Dietterich, "An overview of MAXQ hierarchical reinforcement learning", Proc. 4th Symposium on Abstraction, Reformulation and Approximation (SARA 2000), Lecture Notes in Artificial Intelligence, New York, 2000, pp. 26–44.

[6] F. Samreen and M. Sikandar Hayat Khiyal, "Q-Learning scheduler and load balancer for heterogeneous systems", Journal of Applied Sciences, 2007, pp. 1504 – 1510.

[7] M. Launer and M. Riedmiller, "An algorithm for distributed reinforcement learning in cooperative Multi-agent systems", Proc. 17th International Conference on Machine Learning (ICML 2000), 2000, pp. 535 – 542.

[8] S. M. Thampi and C. Sekaran, "Review of replication schemes for unstructured P2P networks", Proc. IEEE International Advance Computing Conference (IACC 2009), 2009, pp. 194 – 800, <retrieved: February, 2012>.

[9] A. Galstyan, K. Czajkowski, and K. Lerman, "Resource allocation in the grid using reinforcement learning", Proc. 3rd International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2004), vol. 3, 2004, pp. 1314 – 1315, <retrieved: February, 2012>.

[10] K. Verbeeck, J. Parent, and A. Nowé, "Homo egualis reinforcement learning agents for load balancing", Proc. Workshop on Radical Agent Concepts (WRAC 2002), 2002, pp. 81-91.

[11] P. Vallotton, "Image analysis – Feature extraction", Commonwealth Scientific and Industrial Research Organisation (CSIRO 2008) Mathematical and Information Sciences, Australia, 2008.

[12] G. Tesauro, N.K. Jong, R. Das, and M.N. Bennani, "On the use of hybrid reinforcement learning for autonomic resource allocation", Cluster Computing 10(3), 2007, pp 287-299.

[13] D. Busquets, R.L. de Mantaras, C. Siera, and T.G. Dietterich, "Reinforcement learning for landmark-based robot navigation", Proc. 1st International Joint Conference on Autonomous Agents and Multi-

 ISBN: 978-1-61208-195-3

agent Systems: part 2 (AAMAS 2002), 2002, pp. 841 – 843, <retrieved: February, 2012>.

[14] E. Ipek, O. Mutlu, J.F. Martinez, and R. Caruana, "Self-Optimizing memory controllers: A reinforcement learning approach", Proc. 35th International Symposium on Computer Architecture (ISCA 2008), 2008, pp. 39 – 50, <retrieved: February, 2012>.

[15] S.B. Magalhaes, B. Netto, V.C. Rodrigues, A.S. Correa, A. Cardoso de Paiva, and N. Areolino de Almeida, "Application on reinforcement learning for diagnosis based on medical image", Reinforcement Learning, cap 20, I-Tech Education and Publishing, 2008, pp. 379 – 398, <retrieved: February, 2012>.

[16] F. Sahba, H.R. Tizhoosh, and M. Salama, "Application of reinforcement learning for segmentation of transrectal ultrasound images", Biomed Central Medical Imaging, 2008.

[17] A.E. Gaweda, K.M. Muezzinoglu, and G.R. Aronoff, "Individualization of pharmacological anemia management using reinforcement learning source" Special issue: Joint Conference on Neural Networks (IJCNN 2005), 2005, pp. 826 – 834.

[18] A. Guez, R.D. Vincent, M. Avoli, and J. Pinea, "Adaptive treatment of epilepsy via batch-mode reinforcement learning", The Association for the Advancement of Artificial Intelligence (AAAI 2008), 2008, pp. 1671-1678, <retrieved: February, 2012>.

[19] J. Henderson, O. Lemon, and K. Georgil, "Hybrid reinforcement/ supervised learning of dialogue policies from fixed data sets", Computational Linguistics, vol. 34, Issue 4, 2008, pp. 471 – 486, <retrieved: February, 2012>.

[20] J. Peng and B. Bhanu, "Closed-Loop object recognition using reinforcement learning", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 20, Issue 2, 1998, pp. 139 – 154, <retrieved: February, 2012>.

[21] N. Siebel, S. Grunewald, and G. Sommer, "Created edge detectors by evolutionary reinforcement learning", Evolutionary Computation, IEEE World Congress on Computational Intelligence (WCCI 2008), Hong Kong, 2008, pp. 3553 – 3560, <retrieved: February, 2012>.

[22] G.W. Taylor and C. Wolf, "Reinforcement learning for parameter control of text detection in images from video sequences", Proc. International Conference on Information and Communication Technologies (ICICT 2004), Lyon, 2004, pp. 517 - 518, <retrieved: February, 2012>.

[23] S. Srisuk, R. Fooprateepsiri, M. Petrou, and S. Waraklang, "A general framework for image retrival using reinforcement learning", Proc. Image and Vision Computing (IVC 2003), New Zealand, 2003, pp. 36 - 41, <retrieved: February, 2012>.

[24] M. Shokri, H. Tizhoosh, and M. Kamel, "Reinforcement learning for personalizing image search", LORNET Annual E-Learning Conference on Intelligent Interactive Learning Object Repositories, 2006, <retrieved: February, 2012>.

[25] M. Harandi, M. Ahmadabadi, and B. Araabi, "Face recognition using reinforcement learning", Proc. International Conference on Image Processing (ICIP 2004), vol. 4, 2004, pp. 2709 – 2712, <retrieved: February, 2012>.

[26] F. Sahba, H.R. Tizhoosh, and M. Salama,"Using Reinforcement Learning for filter fusion in image enhancement", Proc. Computational Intelligence (CI 2005), 2005, pp. 262 – 266.

[27] D.K. Sharma, L. Gaur, and D. Okunbor, "Image compression and feature extraction using Kohonen's self-organizing map neural network", Journal of Strategic E-Commerce, 2007, <retrieved: February, 2012>.

[28] W.K. Pratt,"Digital image processing", John Wiley and Sons, New York,1991.

 ISBN: 978-1-61208-195-3