Proceedings of the 2007 IEEE Symposium on Computational
Intelligence in Image and Signal Processing (CIISP 2007)

Application of Opposition-Based Reinforcement Learning in Image Segmentation

Farhang Sahba[1,3], Hamid R. Tizhoosh[1,3], Magdy M. M. A. Salama[2]
1 Department of Systems Design Engineering, University of Waterloo, Waterloo, Canada.
2 Department of Electrical and Computer Engineering, University of Waterloo, Watrloo, Canada.
3 Pattern Analysis and Machine Intelligence Laboratory (PAMI), University of Waterloo, Waterloo, Canada.

***Abstract*— In this paper a method for image segmentation using an opposition-based reinforcement learning scheme is introduced. We use this agent-based approach to optimally find the appropriate local values and segment the object. The agent uses an image and its manually segmented version and takes some actions to change the environment (the quality of segmented image). The agent is provided with a scalar reinforcement signal as reward/punishment. The agent uses this information to explore/exploit the solution space. The values obtained can be used as valuable knowledge to fill the Q-matrix. The results demonstrate potential for applying this new method in the field of medical image segmentation.**

I. Introduction

Image segmentation plays a pivotal rule in many computer vision applications. In most cases due to various factors the object is difficult to segment. In methods which rely on learning techniques the lack of a sufficient number of training samples is an obstacle especially when the samples are being manually prepared by an expert. Therefore, a more universal approach should require a minimum number of training data set. Considering the above factors a new agent-based algorithm based on reinforcement learning (RL) is introduced. The most important concept of RL is learning by trial and error based on interaction with the environment [2], [3]. It makes the RL agent suitable for dynamic environments. Its goal is to develop an *action policy* that controls the behavior of the dynamic process, guided by signals (reinforcements) that indicate how well it has been performing the required task. In the case of applying this method to image segmentation, the agent takes some actions as parameter adjustment to change its environment which is the quality of the segmented object. First, the agent gets the image and takes some actions. Then it receives an objective reward or punishment obtained based on comparison of its result with the manually segmented version (gold image). The agent tries to learn which actions can gain the highest reward. After this stage, based on the accumulated rewards, the agent has appropriate knowledge for similar images. In our algorithm we use this reinforced adjustment to control the local processing parameters. We segment the prostate in Transrectal Ultrasound (TRUS) images as a case study [1]. A potential obstacle when we apply RL agents into image-based applications is the large number of state-action pairs involved in our problem. In such cases it is usually difficult to derive the state-action information especially when we need to store past experiences. Therefore we need to speed up the learning process. The opposition-based leaning is the method that can be applied for this purpose [8].
The main purpose of this work is to demonstrate the ability that a reinforcement learning agent can be trained using a very limited number of samples and also can gain extra knowledge during the segmentation process. This is a major advantage in contrast to other approaches (like supervised methods) which either need a large training set or significant amount of expert or apriori knowledge.

This paper is organized as follows: Section II is a short introduction to reinforcement learning. Section III briefly introduces the opposition-based theory. Section IV describes the problem statement and proposed method. Section V presents the results and the last part, section VI, concludes the work.

II. Reinforcement Learning

Reinforcement learning (RL) is based on the idea that an artificial agent learns by interacting with its environment [2], [3]. It allows agents to automatically determine the ideal behavior within a specific context that maximizes performance with respect to predefined measures. Several components constitute the idea behind reinforcement learning. The RL agent is the decision-maker of the process and attempts to take an action recognized by the environment. It receives a *reward* or *punishment* from its environment depending on the action taken. The RL agents discover which actions bring more reward using *exploration* and *exploitation* by receiving the information concerning the state of the environment. At the beginning of the learning process the RL agent does not have any knowledge about how promising taking different actions are [2]. It takes the various actions, and observes the results. After a while, the agent has explored many actions which bring the highest reward and gradually begins to exploit them. In fact, the agent acquires knowledge of the actions and eventually learns to perform the actions that are the most rewarding. During this process it tries to meet a certain goal relating to the state of the environment. The reward and punishment could be defined objectively when they are defined using a function; or gained subjectively when they are given to the agent by an experienced operator. Reinforcement learning learns online, and can continuously learn and adapt while performing the required task. This behavior is useful for the cases where learning samples are difficult or impossible to obtain [2], [6].

1-4244-0707-9/07/$25.00 ©2007 IEEE

The design of RL agents is based on the definition of the problem at hand. Figures 1 shows the general components of reinforcement learning. The agent, which is the decision maker of the process, observes the state of the environment. Then it takes an action based on the former experience associated with the current observation and accumulated reinforcement (reward/punishment). Finally, The agent receives a reward or punishment from its environment depending on the action taken.

Q-Learning, a popular technique proposed by Watkins in 1989, is an iterative method for action policy learning [2], [4]. This method works based on estimating the values of state-action pairs [3].

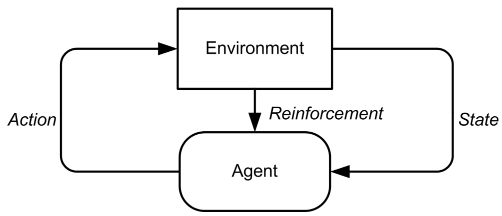

Fig. 1. A general model for Reinforcement learning agent.

III. Opposition-Based Theory

The Opposition-Based Learning (OBL) [7], [8] provides a practical scheme for extension of existing learning algorithms. The idea can be employed to extend RL agents to shorten the exploration time. Tizhoosh [7] has introduced a new class of RL algorithms based on opposition. By considering states and opposite states, and actions and opposite actions simultaneously multiple updates can be made. This leads to a shorter exploration period. Therefore a desirable level of accuracy can be achieved in a shorter time.

Learning, optimization and search are fundamental tasks in the machine intelligence research. Whenever we are looking for the solution x of a given problem, we usually make an estimate x_i. This estimate is not the exact solution and could be based on experience or a totally random guess. In some cases we are satisfied with the estimate x_i and sometimes we try further to increase the result accuracy if possible. In many cases the learning begins at a random point. We begin from scratch and move toward a solution.

The action policy of reinforcement agents is initially based on randomness. The random guess, if not far away from the optimal solution, can result in a fast convergence. However, it is natural to state that if we begin with a random guess, which is very far from the existing solution, let say in worst case it is in the *opposite location*, then the approximation, search or optimization will take considerably more time, or in worst case becomes intractable. Of course, in absence of any apriori knowledge, it is not possible to make the best initial guess. Logically, we should be looking in all directions simultaneously, or more concretely, in the opposite direction. searching in opposite direction could be beneficial from an algorithmic point of view as well. If we are searching for the solution x, and if we agree that searching in opposite direction could be advantageous for some cases, then calculating the opposite number $\breve{x}$ is the first step [7], [8].

Definition (Type-I Opposition) - Let $P = (x_1, x_2, \cdots, x_n)$ be a point in an n-dimensional space, where $x_i \in [a_i, b_i]$ with $a_i, b_i \in R$. The type-I opposite point $\breve{P} = (\breve{x}_1, \breve{x}_2, \cdots, \breve{x}_n)$ is then completely defined where

$$\breve{x}_i = a_i + b_i - x_i. \tag{1}$$

Figure 2 illustrates the definition for one-dimensional case based on the distance of the opposite guess from the interval boundaries.

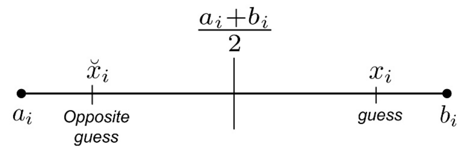

Fig. 2. A demonstration for one-dimensional opposition based on the distance of the opposite guess from the interval boundaries.

As mentioned, reinforcement learning is based on interaction of an intelligent agent with the environment by receiving reward and punishment. In this sense, reinforcement learning is a type of weakly supervised learning. In order to explain how the concept of opposition-based learning can be used to extend reinforcement agents, we focus on the simplest and most popular reinforcement algorithm, Q-Iearning.

In this algorithm, the amount of time needed for convergence is proportional to the size of the Q-matrix. A larger Q-matrix, resulting from a larger number of states and/or a greater number of actions requires more time to be filled.

Generally, the RL agents begin from scratch and make stochastic decisions, explore the environment, find rewarding actions and exploit them. Specially at the beginning of the performance of the RL agents is poor due to lack of knowledge about which actions can control the environment in desired direction. Whenever the RL agent takes an action it should also consider the *opposite action* and/or *opposite state*. This will shorten the state-space exploration and consequently accelerate the convergence. Of course the concept of opposition can be applied if opposite actions and opposite states are meaningful in the context of the problem at hand. In regard to action a and state s and the existence of their opposites $\breve{a}$ and $\breve{s}$, following cases can be distinguished [8]:

- Opposite action $\breve{a}$ and opposite state $\breve{s}$ are given: at least four cases can be updated per state observation.
- Only $\breve{a}$ can be defined: two cases can be updated per state observation.
- Only $\breve{s}$ can be defined: two cases can be updated per state observation.
- Neither $\breve{a}$ nor $\breve{s}$ can be given: application of opposition concept not straightforward.

Assuming that opposite actions and opposite states both exist, then at least four state-action pairs can be updated in each iteration. In general, if action a is rewarded for the state s, then a is punished for the opposite state $\breve{s}$, the opposite action $\breve{a}$ is punished for s and rewarded for $\breve{s}$ (Figure 3)[8].

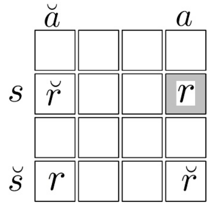

Fig. 3. Time saving in RL: the action a is rewarded for the state s. The opposite cases are updated simultaneously without explicit action-taking.

In order to make additional updates as described, the RL agent has to know how to find opposite actions, and opposite states. Clearly, this will depend on the application at hand. Whereas for some applications opposite actions are straightforward, this may not be the case for other ones. Nonetheless, general procedures may be defined to facilitate this. A degree of opposition $\breve{\varphi}$ can be defined to measure in how far two actions a_1 and a_2 are the opposite of each other [8]:

$$\breve{\varphi}(a_1|_{s_i}, a_2|_{s_j}) = \eta \times \left[1 - e^{\left(-\frac{|Q(s_i,a_1)-Q(s_j,a_2)|}{\max_k\left(Q(s_i,a_k),Q(s_j,a_k)\right)}\right)}\right], \tag{2}$$

where η is the state similarity and can be calculated based on state clustering [8].

Considering action a_t when visiting state s_t, opposition-based Q-learning for opposite action can be defined as given in Table I.

TABLE I

PSEUDO CODE OF OB Q-LEARNING FOR OPPOSITE ACTION [7].

Initialize $Q(s, a)$ randomly
Repeat (for each episode)
  Initialize s
  Repeat (for each iteration of episode)
    Choose a from s using policy π derived from Q
    Take action a, observe reward r, and next state s'
    $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$
    Calculate the degree of opposition(optional)
    Determine opposite actin $\breve{a}$ and next state s''
    Calculate the opposite reward $\breve{r} = p$
    Determine next opposite action $\breve{a}'$ from s''
    $Q(s, \breve{a}) \leftarrow Q(s, \breve{a}) + \alpha[\breve{r} + \gamma \max_{a'} Q(s'', a') - Q(s, \breve{a})]$
    $s \leftarrow s'$
  until s is terminal

IV. PROBLEM STATEMENT AND PROPOSED APPROACH

Recently some methods are introduced in the literature which use a reinforcement learning schema for image-related problems [6], [10], [11], [12], [17]. In this paper, we present the continuation of our work in [16] using new features. It enables us to implement the task of segmentation in ultrasound image (as a case study) more effectively. In our proposed approach we treat the segmentation task locally. The image is divided in $R_S \times C_S$ sub-images (R_S rows and C_S columns) and the RL agent works on each of them separately. We first threshold the sub-images using local values. Due to some disturbing factors such as speckle and low contrast, we usually have artifacts after thresholding. Therefore, we use morphological opening as a second stage to post-process each thresholded sub-image. The reinforcement learning agent determines the local thresholding value and the size of structuring element for each individual sub-image.

When we apply RL agents into image-based applications, a large amount of state-action pairs involves in our tasks. In these applications usually two following problems may occur:

- Massive memory requirement and,
- Visiting enough state-action pairs in a nonrealistic time to fill the corresponding table.

We use the opposition-based learning to overcome these problems. To construct the RL agent, three components; states, actions and reward must be defined. The agent starts its work using an ultrasound image and its manually segmented version. It works on each sub-image and using the gold standard (obtained from the manually segmented version) explores the solution space for that sub-image. During this time the RL agent changes the local thresholding values and the size of structuring element for each sub-image individually. By taking each action the agent receives corresponding reward/punishment for that state-action pair and updates the corresponding value in Q-matrix. After this process the agent has explored many actions and tries to exploit the most rewarding ones. This method is specifically useful where there are several images having inherently the same characteristics. In such a case, instead of parameter adjustment for each individual input image or using a large training data set to cover all possible cases, we can use some of them and acquire knowledge to segment the other ones. It is also useful to gain extra knowledge when the agent tries to segment new images. Figure 4 shows the general model and its components used in our proposed approach.

A. States

We considered the idea of the state containing some parameters representing the quality of the image. These parameters can be chosen among various shape and/or boundary properties such as area, Euler number, compactness, convexity, slope density function, signature and so on. They must reflect the quality of each sub-image after thresholding and post-processing. The solution to how we must choose the features is highly dependent on the specifications of the problem at hand. For our application the following features extracted from the largest object in each sub-image are used to define the states:

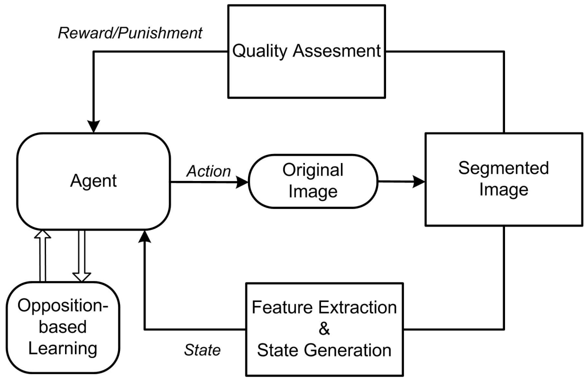

Fig. 4. Model used in proposed approach.

1) Area: The area A of the object is used as one the feature to define the state. We calculate the normalized area with respect to the whole area of the sub-image as follow:

$$A_{\Delta} = \frac{A_{subimage} - A_{object}}{A_{subimage}}. \tag{3}$$

2) Compactness: The compactness defined as:

$$\Psi = \frac{P^2}{A} \tag{4}$$

where P is the perimeter of the object in the sub-image [5].

3) Relative coordinate of each sub-images: We suppose that the geometric center of the prostate in the original image is given by the user. The $X-Y$ relative location of each sub-image with respect to the location of the geometric center is used as a state parameter.

4) The number of the objects: The last parameter used in state definition is the number of revealed objects, N_O after thresholding.

B. Actions

The actions are defined as changing of the threshold value and size of structuring element for each individual sub-image. The agent increases or decreases the assigned local thresholding value for each sub-images by adding/subtracting a specific value ($\pm\delta$). We can take some predefined values $(\tau_1, \tau_1, ..., \tau_n)$ between the maximum and minimum gray levels in each iteration. For morphological operator the agent may increase/decrease the size of structuring element in a specific interval or choose among some predefined values $(\upsilon_1, \upsilon_2, ..., \upsilon_n)$.

C. Reward/Punishment

The rewards and punishments can be defined based on a criterion representing how well the object has been segmented in each sub-image. Several criteria can be used for this purpose. A straightforward method is to compare the results before and after action based on the quality of the segmented objects. To measure this for each sub-image we note how much the quality is changed after the action. In each sub-image, for improving the quality of segmented object the agent receives rewards, otherwise it will be punished. A general form for the reward/punishment function can be represented as follow:

$$reward = \begin{cases} \epsilon_1 D_{\Delta} & D_{\Delta} \geq 0, \\ \epsilon_2 D_{\Delta} & D_{\Delta} \leq 0 \end{cases} \tag{5}$$

In this equation D_{Δ} is a measure indicating the difference between the quality before and after taking the action and ϵ_1 and ϵ_2 are the constant values.

D. Opposition-based Learning Procedure and Testing

In the case of applying opposition-based reinforcement learning to the task of image segmentation we may use it for states and/or actions. This is highly dependent on how the state-action pairs are defined. As mentioned, the goal is to find appropriate local parameters for each sub-image so that we can segment the whole image. These parameters are the value of threshold and the size of structuring element for a morphological operator.

After each iteration the RL agent has scanned the whole image. Based on the quality before and after the action taken the agent receives reward/punishment for each sub-image and updates its knowledge. We know that a problem arises when the number of states/actions increases. Because the RL agent must visit all sub-images, it takes too long to try various actions especially when the agent is rather in exploration mode. Using opposition-based learning we can update the RL agent more rapidly. For our case, this is done for actions. For example if the action is to increase the thresolding value, the opposite action can be defined as decreasing it, or if the action is choosing a specific value among some predefined values the opposite action can be define based on the relative distance of others with respect to the current value. Generally speaking, we can define the degree of being opposite based on their distance to the current situation (equation 2).

The states and actions are based on what we designed in section IV-A and IV-B, respectively. In the offline stage (exploration mode) the perfect output image is available using manually segmented version. For reward/punishment function, we use the same equation 5 and for the quality measure of each sub-image we calculate in how far the similarity with the perfect output image is changed after the action taken. To measure this similarity we can calculate the percentage of the pixels that are the same in the perfect output image and the image segmented by the RL agent.

During this procedure, the agent must explore the parameter space. It can be achieved using the Boltzman policy with a high temperature or ε-greedy policy [3]. In a standard Q-learning, after a sufficiently large number of iterations, the Q-matrix is filled with appropriate values. It means that the agent can estimate the best action for each given state. In the case when we use opposition-based learning, the training time is reduced during exploration mode. That is because the Q-matrix is filled more rapidly using extra updates.

For new samples, the agent takes its action based on the knowledge it has previously gained. It finds the appropriate thresholding value and the size of structuring element for each sub-image such that the prostate can be correctly segmented.

We may want to apply an objective evaluation for new images (depends on application at hand). We can use the signature of the extracted object and compare to the standard signature of the object we are looking for. A signature is a functional representation of a contour, generated by various techniques [13]. Generally, a signature is based on the distance versus angle. Because we have the geometric center of the prostate in original image (given by the user) we can calculate the distance from the points on the boundary to the geometric center of the object as a 2π periodic function. One angle θ is assigned to a distance d represented as a 2D function $f(\theta, d)$. We may normalize d to make this transformation scale invariant. Using this method we can find significant irregular parts and use their information to update the RL agent for new images in an objective manner [16]. As stated, this evaluation is not necessary and is applied where acceptable results (in terms of correct updates) can be achieved.

Finally the agent can recognize the optimal values and segment the prostate in new samples.

V. Results and Discussions

In this section we present and discuss the results of the proposed approach. We implemented an ε-greedy policy to explore/exploit the solution space. Considering the size of the prostate in TRUS images we empirically choose $R_S = 4$ and $C_S = 5$ (20 sub-images). The number of discrete levels for total states was set to 240. The threshold action is defined by taking 4 predefined values (equally spaced) between the local maximum and minimum gray levels in each sub-image. For the post-processing action we chose the size of structuring element among values $0, 5, 15$.

Three manually segmented images were used to train the system. The RL agent was trained using a total of 6000 iterations for all sub-images for a standard Q-learning. For simplicity we calculate the reward/punishment based on $\epsilon_1 D_\Delta = 0, \epsilon_2 D_\Delta = 10$.

After performing the procedure the Q-matrix was filled with appropriate values. In fact, the agent gained enough knowledge to recognize the optimal values for each sub-image.

The method was applied on 20 sample images from two patients. In all cases, the agent could segment the prostate and terminate the process using the standard Q-learning.

Then the algorithm was applied again using the opposition-based reinforcement learning. In the learning loop, the opposite actions are calculated and the agents knowledge is updated.

Figure 5(a) and 5(b) show a sample image and its manually segmented version. Also Figure 5(c) and 5(d) illustrate the results for standard Q-learning and opposition-based learning. Table II shows the results as learning time reduction (LTR) for opposition-based Q-learning compared to standard Q-Learning. As we can see in the mean value, a considerable reduction in learning time is achieved. This can be very valuable for many applications.

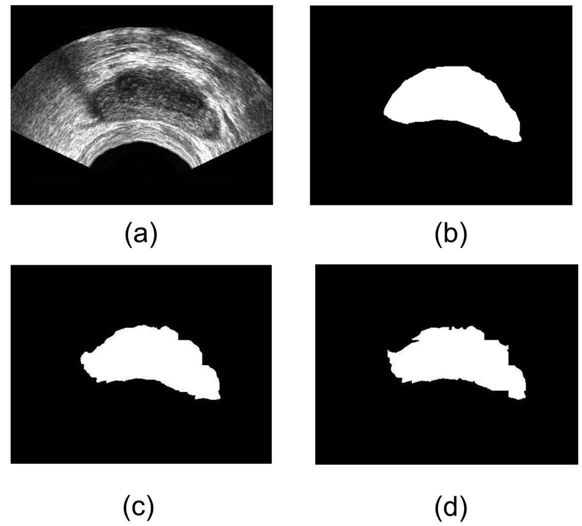

Fig. 5. (a) Sample ultrasound image, (b) manually segmented version, (c) segmented using standard Q-learning, (d) segmented using Opposition-based Q-learning

TABLE II

Learning Time Reduction (LTR) for opposition-based Q-learning compared to standard Q-Learning. The results are represented for Image 1-20 (I 1-20).

I	**LTR(%)**	I	**LTR(%)**	I	**LTR(%)**	**I**	**LTR(%)**
1	25	**2**	14	**3**	22	**4**	20
5	23	**6**	17	**7**	13	**8**	16
9	22	**10**	21	**11**	17	**12**	23
13	21	**14**	20	**15**	18	**16**	22
17	20	**18**	22	**19**	21	**20**	17
Mean	**19.72**					**Std**	**3.12**

Figure 6 shows the results for proposed opposition-based reinforcement learning on 8 sample images. For all images the error in final segmented object is defined as:

$$e = 100 \times \frac{N_{missclassed}}{N_T}, \tag{6}$$

Where $N_{missclassed}$ and N_T are the number of mis-classed pixels and the total number of pixels, respectively.

Table III shows the average errors for both standard Q-learning and opposition-based Q-learning.

TABLE III

Mean error values for opposition-based Q-learning and standard Q-Learning.

	Standard Q-learning (%)	Opposition-based Q-learning (%)
Mean	6.4	8.6
Std	2.3	3.1

There may be a tradeoff between accuracy and speed. Considering the results in terms of visual appearances and the amounts of errors, they can be well acceptable to use as the inputs of a fine tune segmentation algorithm. For instance, these results can be used as initial snake for the well-known method introduced in [14] or as a coarse estimation for the methods introduced by authors in [15]. In some cases that the original image has good quality, the results of proposed approach may be used to make the final segmentation.

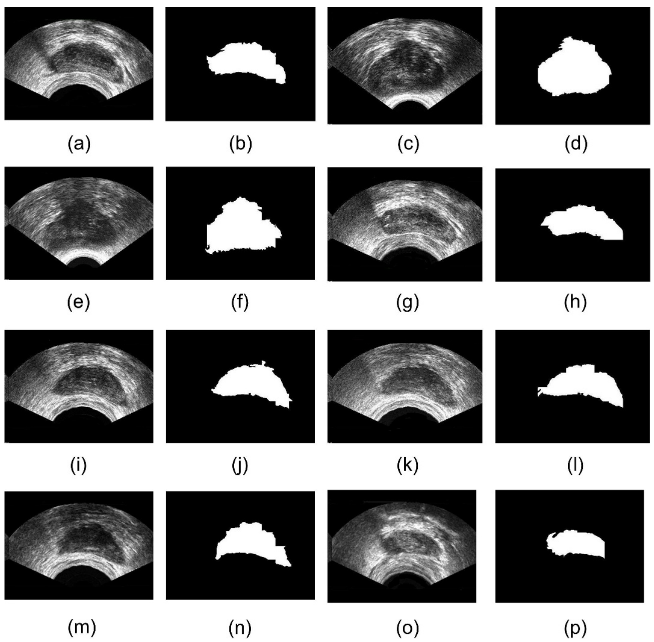

Fig. 6. The results of proposed approach for some test images (a) - (p). First and third columns are original images, second and fourth columns are the results.

VI. Conclusions

In this work, an opposition-based reinforcement learning method as a novel idea for image segmentation was proposed and some results were illustrated. First, the image is divided in some sub-images. In each sub-image, the agent takes some actions as changing the thresholding value and the size of structuring element to change the quality of the segmented parts.

In a standard Q-learning, the Q-matrix was filled after a large number of iterations, but when we used opposition-based learning the training time was reduced considerably.

After the offline stage, the agent took actions with maximum reward for each sub-image. It was able to choose the appropriate values for the input image with similar characteristics based on its accumulated knowledge. The proposed method can be trained for various object segmentation tasks including applications in medical image segmentation to achieve an acceptable level of performance. In fact combining of two ideas, reinforced segmentation and opposition-based learning, can give an important advantage for these applications which generally have huge amount of data. The idea in this method has the potential to be used as the main segmentation approach, or as an interim stage to serve other segmentation methods. The method was applied to some ultrasound images to show its effectiveness.

Our future work will concentrate on extension of the algorithm.

Selection of various numbers for sub-images and applying of standard processes such as boundary refinement on the extracted object to have a well-shaped in the final result will be investigated. Also, more appropriate quality measures, usually used in medical imaging, must be apply to evaluate the performance more accurately.

References

[1] M. F. Insana and Brown D. G., *Acoustic scattering theory applied to soft biological tissues*, Ultrasonic Scattering in biological tissues, CRC Press, Boca Raton,pp 76-124, 1993.

[2] R.S. Sutton, and A.G. Barto , *Reinforcement Learning*, MIT Press, Cambridge, MA, 1998.

[3] S. Singh, P. Norving, D.Cohn, *Introduction to Reinforcement Learning*, Harlequin Inc.,1996.

[4] C.J.C.H. Watkins, P. Dayan,"Q-Learning" *Machine Learning*, 8, 279-292, 1992.

[5] R. C. Gonzalez, R. E. Woods , *Digital Image Processing*, Prentice Hall, 2nd Edition, 2002.

[6] Tizhoosh, H.R., Taylor, G.W.," Reinforced Contrast Adaptation", *International Journal of Image and Graphic*, vol. 6, No. 3, pp. 377-392, 2006.

[7] Tizhoosh, H.R.," Opposition-based learning: A new scheme for machine inteligence", *International Conference on computational Inteligence for Modeling Control and Automation CIMCA'05*, Vienna,Austria,2005.

[8] Tizhoosh, H.R." Opposition-based Reinforcement Learning", *Journal of Advanced Computetional Inteligence and Inteligent Informatics*, vol 10, no. 4, pp. 578-585, 2006.

[9] Tizhoosh, H.R.," Reinforcement learning based on actions and opposit actions", *ICGST International Conference on Artificial Inteligence and Machine Learning AIML-05*, Cairo, Egypt, 2005.

[10] M. Shokri., H. R. Tizhoosh, " Q(λ)-based Image Thresholding",*Canadian Conference on Computer and Robot Vision*, pp.504-508 ,2004.

[11] F. Sahba., H. R. Tizhoosh, "Filter Fusion for Image Enhancement Using Reinforcement Learning", *IEEE Canadian Conference Electrical and Computer Engineering 2003*, Montreal, May 2003.

[12] F. Sahba, H. R. Tizhoosh, M. M. A. Salama, " Using Reinforcement Learning for Filter Fusion in Image Enhancement", *The Fourth IASTED International Conference on Computational Intelligence*, July 2005, Calgary, Canada.

[13] Holt, R. J. and Netravali, A. N., " Using Line Correspondences In Invariant Signatures For Curve Recognition", *Image and Vision Computing*, Vol. 11, No. 7, Sept. 1993. Pp. 440-446.

[14] H. M. Ladak, F. Mao, Y. Wang, D. B. Downey, D. A. Steinman, A. Fenster, " Prostate boundary segmentation from 2D ultrasound images", *Medical Physics* 27 , pp. 1777-1788, 2000.

[15] F. Sahba, H. R. Tizhoosh, M. M. A. Salama, " A Coarse-to-fine approach to prostate boundary segmentation in ultrasound images", *BioMedical Engineering OnLine*, 4:58, October 2005.

[16] F. Sahba, H. R. Tizhoosh, M. M. A. Salama, "A Reinforcement Learning Framework for Medical Image Segmentation", *Proceeding of the IEEE World Congress on Computational Intelligence (WCCI)*, pp. 1238-1244, July 2006, Vancouver, Canada.

[17] F. Sahba, H. R. Tizhoosh, M. M. A. Salama, "Increasing object recognition rate using reinforced segmentation", *The IEEE International Conference on Image Processing (ICIP)*, October 2006, Atlanta, United State.