Building Internal Scene Representation in Cognitive Agents

Marek Jaszuk[1] and Janusz A. Starzyk[2,1]

[1] University of Information Technology and Management, ul. Sucharskiego 2,
Rzeszów, Poland,
[2] School of Electrical Engineering and Computer Science, Ohio University, Athens,
OH 45701 USA
marek.jaszuk@gmail.com,starzykj@ohio.edu

Abstract. Navigating in realistic environments requires continuous observation of a robots surroundings, and creating internal representation of the perceived scene. This incorporates a sequence of cognitive processes, including attention focus, recognition of objects, and building internal scene representation. The paper describes selected elements of a cognitive system, which implement mechanisms of scene observation based on visual saccades, followed by creating the scene representation based on a distance matrix. Such internal representation is a foundation for scene comparison, necessary for recognizing known places, or changes in the environment.

Keywords: visual saccades, scene model, episodic memory

1 Introduction

Humans are capable of intelligently acting in complex environments. For instance, we can find our destination in a city, interact with other people to exchange information or arrange objects in a room in a suitable way. In these tasks, we outperform current men built systems such as autonomous robots. Thus it is highly desirable to develop artificial agents, which will be able to support us in performing a rich variety of tasks, both in our daily environment, as well as in dangerous and inaccessible places.

For this purpose we devise computer models of intelligent information processing called cognitive systems [1]. Such systems should be able to collect and process a stream of sensory data to create internal representation of the environment, and undertake the proper actions. The original foundations for cognitive systems come from psychological theories and are also a part of artificial intelligence research. Such systems are composed of various components that mimic different mental functions. Among their most important elements are: sensory and motor functions, as well as different kinds of memory, including semantic, episodic, procedural, and working memory.

It is known that to carry out a successful navigation in complex environments, mobile robots must acquire and maintain internal representation of the environment. This is not a trivial task and many factors affect the reliability of such models. In the presented paper we try to solve this problem, by introducing a saccade based algorithm of building a distance based scene model. We also introduce a scene comparison method, based on comparing the distance matrices.

The paper is organized as follows: Sec. 2 discusses visual saccades and the discussed cognitive system architecture; In Sec. 3 we discuss building the scene model from the data delivered by a sequence of visual saccades; Sec. 4 presents a method designed for comparing scenes represented in the form of distance matrices; In Sec. 5 we present the VEEMA simulation, where the experiments are carried out; In Sec. 6 we present sample experimental results demonstrating the performance of the scene comparison method.

2 The Fundamentals

2.1 Visual Saccades

Visual saccade is a fast eye movement, from one focus point to another [2]. Saccadic movement is observed also in other senses, e.g. as a fast shift in frequency of received signal or other quick change in perception. The main purpose of saccadic movements is to identify objects with high saliency, which are potentially important for the observer. The visual signal from the environment is initially filtered in order to identify regions with outstanding visual features. Next, attention is switched rapidly between such regions in a sequence of saccadic movements. After focusing attention on a given region, the observer is able to acquire more precise visual data needed to recognize the object. Additionally, the observer should be able to assess the distance between particular points on which the attention is focused. Humans realize this task through stereovision. In robots, it can be realized much easier and more precisely than in humans, using several techniques, depending on the robots construction.

In robotic systems there are two possible ways of implementing visual saccades. The first of them relies on the camera movements, which resembles the biological reality [3, 4]. The main difficulty of this approach is that it requires a special camera steering system, which will be able to scan the environment with sufficient speed and accuracy. This kind of fast motion is necessary to identify enough details of the environment, to allow the robot to respond to changing conditions in real time. The other realization of visual saccades is implemented by software analysis of images from static camera mounted on a robot. This approach is much simpler to implement, because it does not require any special hardware, it yields high speed saccades and can be used practically on all mobile robots equipped with a camera. Robot can still move its camera to follow changes in the environment, but this motion may be better controlled and proceed at a lower speed than the one required for visual saccades.

2.2 The Cognitive System

The presented work focuses on selected parts of a cognitive system based on motivated learning [5]. The system is based on the idea of embodied intelligence, where whole learning comes from interaction with the environment instead of scripted algorithms designed by engineers [6]. The embodied agent interacts with the environment through a set of sensors and actuators, and the reaction of the environment to the undertaken actions is the source of information necessary for learning skills, that the agent needs to survive. The structure of the discussed cognitive system represented on a very general level is shown in Fig. 1. Its central part contains emergent motivations, and the goal creation system, which are responsible for stimulating the agent to act and learn new skills. The set of sensors acquires the raw stream of data from the environment. Subconscious attention switching responds to salient features of the sensory data in order to focus the agents attention on selected objects. Semantic and episodic memory are both long term memories. Semantic memory stores general facts about the world, learned from repeating experiences, while episodic memory is responsible for storing the experiences and situations that the agent cognitively perceived. The working memory is responsible for processing the filtered and semantically recognized data, building the internal representation of experiences, and storing them in the form of episodes.

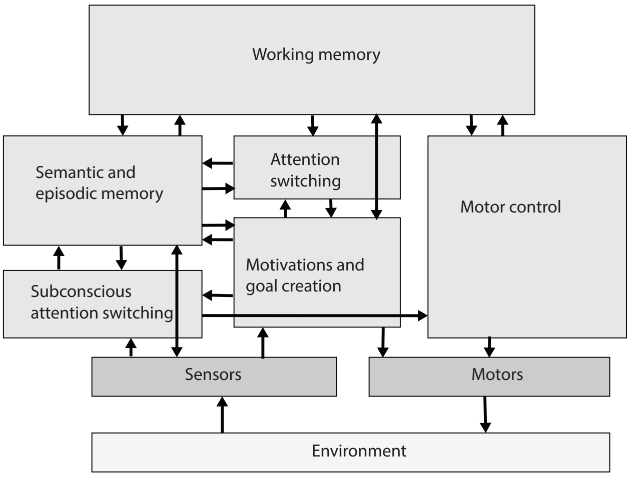

Working memory
Semantic and episodic memory
Attention switching
Motor control
Motivations and goal creation
Subconscious attention switching
Sensors
Motors
Environment

Fig. 1. The general structure of the considered cognitive system

Fig. 2 presents in more detail elements of the cognitive system model, responsible for building the environment representation, and storing it in the form of sequences of episodes in the episodic memory. The focus of our interest is visual data, so the role of sensors is played by cameras. The video stream registered by a camera contains huge amount of data, with various level of significance. To save the system against the flood of unimportant details, the stream needs to be filtered in order to find the elements, which are worth the agents attention. Thus the video data need to be initially processed in order to identify regions with high visual saliency. Such regions are more attractive to the agent, because they are more likely to contain information important for the agent. The agent then follows the visually salient elements of the video data in a sequence of saccadic movements. After focusing attention on a given region, the content of this region is analysed and recognized cognitively by the semantic memory. The sequence of recognized objects is delivered together with geometrical location data to the scene building module, where spatial model of the scene is created. The scene representation is the key element for the episodic management module, which identifies changes within the scenes, and decides about saving valuable experiences in the episodic memory.

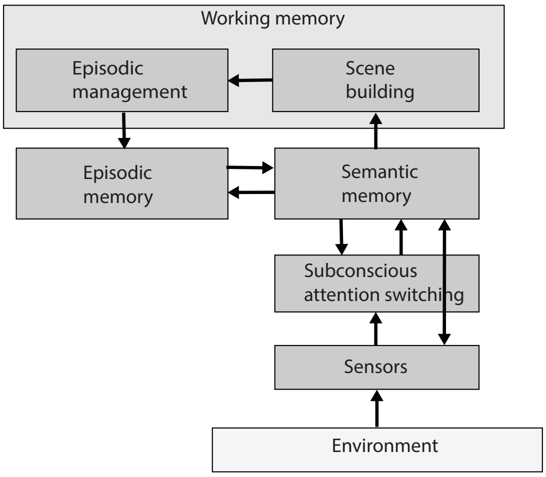


Fig. 2. The structure of the cognitive system elements, responsible for building the internal scene representation

There are a number of methods for reconstructing 3D geometry of the environment. Among the most commonly used are those based on multiple view techniques [7]. Our approach, however, is different, because it incorporates the element of cognitive recognition. In this way we avoid computationally expensive processing of raw pixel data. In our approach the data are replaced by relatively small number of cognitively recognized objects. In this way, recognition of known scenes is performed in a similar way like humans do - by comparing a small number of characteristic objects represented on symbolic level. Also storage of such symbolic data is much more efficient.

3 Building internal scene representation from a sequence of saccades

Building the scene representation requires identifying objects within the environment, and finding their spatial relations (location map). In humans the area, which can be consciously analysed and recognized during a single act of attention focus is relatively small. This imposes limitation on the size of the identified objects. To mimic this behaviour, we assume small size of objects in relation to the whole scene, which allows for treating their location as a point. This simplifies computing distances, and constructing spatial scene model. If a scene object is too large, it has to be observed in a sequence of saccades. Such object is represented in the agents memory as a complex object composed of a number of simple objects.

We have to note, that the scene is always perceived from the observers perspective, so the coordinates of objects are registered in the egocentric coordinate system of the agent. The problem is, however, that the origin of the system moves according to the moves of the agent, while the environment representation has to be independent of the temporary agent location, i.e. represented in the allocentric coordinates. Thus our method assumes that every scene is represented within its local coordinates. Actually, the internal scene representation will be based on the matrix of distances between objects. However, to compute the matrix it is necessary first to express the locations of objects within some coordinate system. The actual choice of this system is not that important because the resulting distance matrix is always the same. It is enough if the system is fixed in relation to the scene.

To illustrate the mechanism of creating the internal scene representation we will analyse a sample sequence of saccadic movements. For simplification we initially limit the considerations to a 2D environment map. This analysis can later be extended to a 3D case. When an agent enters a new scene, it starts from focusing attention on the object, which is the most visually salient (the precise description of analysing the visual saliency is not the subject of this paper and will be described separately). For practical reasons location of the first object perceived in the scene is the most convenient choice for the origin of the scene coordinates Fig. 3(a). The agent can assess the distance a_1 to the object, but it is currently of not much use, because the object is located in coordinates, which are irrespective of the agent location. Technically the assessment of the distance can be realized by using stereovision, range laser or other sensors, depending on the robots construction.

The next step is focusing attention on the second object in the scene, which is done again on the basis of visual saliency. The transition between the first and the second object is the first saccade made by the agent while observing the scene. After focusing attention on the second object, and recognizing it cognitively, the agent has to locate it within his own scene representation. This can be done if it is possible to determine the location of the second object with respect to the previous one. It is not possible to measure the distance directly, but it can be computed given that the agent can measure the distance from his location to both the objects (a_1 and a_2), and the angle α_1 between both the directions (Fig. 3(b)).

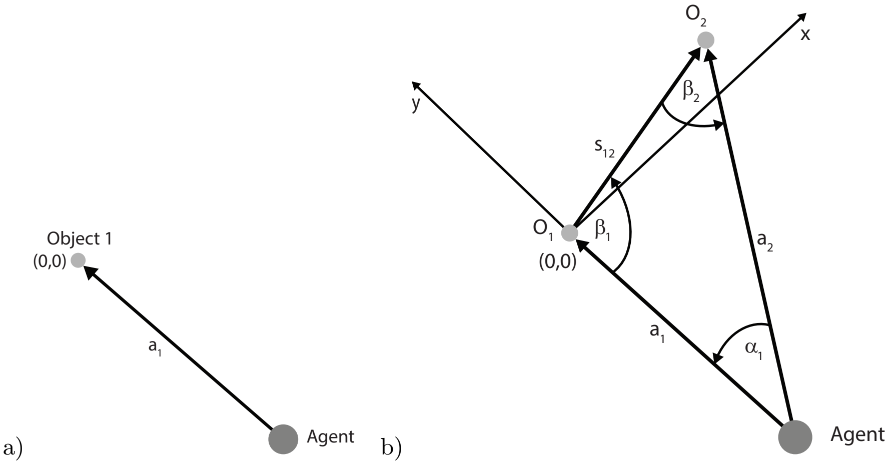


Fig. 3. The first stage of building the scene map: (a) perception of the first object - origin of the local coordinate system, (b) first saccade and location of the second object

The distance between O_1 and O_2, which is the length of saccade S_12 can be obtained from the law of cosines applied to the triangle with vertices in O_1, O_2, and the agent location:

$$s_{12}^2 = a_1^2 + a_2^2 - 2a_1a_2\cos\alpha_1. \tag{1}$$

To locate the second object in the internal scene representation, the direction of the axes of local scene coordinates is needed. We can choose the direction of the y axis as an extension of the line between the agent and O_1 (Fig. 3(b)). This defines the direction of the x axis as perpendicular to y. Given the axes we can compute the location of the second object within the local coordinates:

$$x_{O2} = s_{12}cos\left(\beta_1 - \frac{\pi}{2}\right), \tag{2a}$$

$$y_{O2} = s_{12}cos\left(\pi - \beta_1\right), \tag{2b}$$

We have to remember, that the agent can move while observing the scene. This results in different position, between observing object 1 and object 2. As a result Eqs. 1 and 2 would not be precise, because we would have to take into account the change of agents position between saccades. In realistic conditions, however, the transition of a robot in the environment is relatively slow, compared to the speed of saccadic scene observation. As a result the effect of agents transitions in the environment can be neglected without large negative effect on the precision of the internal scene model.

To make the description of the scene building mechanism complete, we have to demonstrate adding the third object to the scene. Locating the new object in the local coordinates is based on knowing positions of the two previously memorized objects. The agent, as in previous cases, locates a visually salient object and focuses attention on this object. It can also measure the distance to this object, and the angle α_2 between the directions of observation of object 2 and object 3 from the agents position (Fig. 4). Given the data, the length of saccade s_{23} can be computed analogously as in the first saccade (Fig. 4).

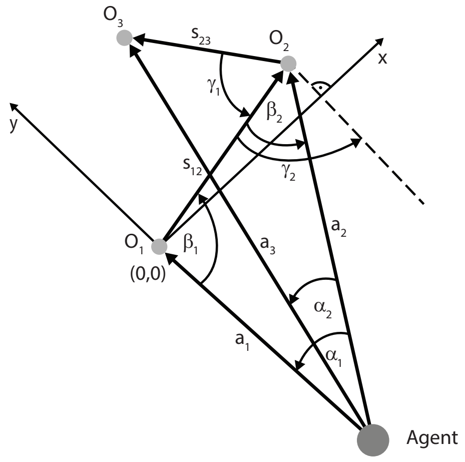


Fig. 4. Adding the third object to the scene map

To position object 3 in the local coordinates we need to know the angle $\gamma_1 + \gamma_2$. This will allow for computing the projections of s_{23} on the x and y axes. To find this angle we start from using the law of cosines to compute β_2 and $\beta_2 + \gamma_1$ directly, which after subtracting gives γ_1. We already know coordinates of object 2 so the γ_2 angle can also be computed from one of trigonometric functions. In this way we obtained $\gamma_1 + \gamma_2$. Now the coordinates of object 3 can be computed as follows:

$$x_{O3} = x_{O2} - s_{23} cos \left(\gamma_1 + \gamma_2 - \frac{\pi}{2} \right) , \tag{3a}$$

$$y_{O3} = y_{O3} + s_{23} sin \left(\gamma_1 + \gamma_2 - \frac{\pi}{2} \right) , \tag{3b}$$

The procedure described for object 3 is repeated for every new saccade. To locate the new object within the local scene coordinates only the most recent saccade is needed. Although some imprecisions are possible during measuring distances between objects within the scene, the described procedure allows to avoid cumulating errors in the scene representation building. Moreover, the scene model can be modified and corrected, when the agent moves within the environment, and observes the same objects from different positions.

In steering robots within realistic environments we are obviously interested in building 3D representation of the scene. This can be done by extending the described methodology to the 3D coordinates. The general procedure is in the 3D case very similar to the 2D case. In every step of the scene observation, the agent focuses attention on an object with high visual saliency, and measures distances to the perceived objects, together with angles between directions of observation of particular objects. The first observed object becomes the origin of the scene, and the subsequent objects are added in relation to the most recently added ones. The difference is the additional dimension, which has to be taken into account when calculating positions of new objects. This results in necessity of using data from three most recent saccades, while adding new object to the scene, instead of just two as it was in the 2D case. Detailed discussion of the 3D saccades, does not bring much new to understanding of discussed mechanism, and thus is omitted.

4 Distance Based Scene Comparison

Let us assume that D_1 and D_2 are the matrices of distances between objects in scenes SC_1 and SC_2 respectively, and that the order of rows and columns correspond to equivalent symbols in both scenes. In case a symbol is unique to a scene, we add a new row and column in both matrices. If one of the compared scenes has no such object, its distance matrix will have the distance to this object set to some large value like 2x the size of the scene diagonal. In this way both matrices always have the same size. Notice that D_1 and D_2 are symmetrical matrices with zeros on diagonal (an object has 0 distance to itself). In addition to the distance matrices we assign some significance to the scene objects. The significance results from the agents internal needs, and motivations, and allows for steering the agents attention not only by external objective factors, but also by internal perception of object significance. Every object in the scene has some significance and its significance may be different in different scenes in which it appears. For the purpose of scene comparison we define a mutual significance matrix as follows:

$$S_M = [s_{ij}]_{m\times m}\,, \text{where} \quad s_{ij} = \prod_{k=1,2} s_{ki}s_{kj} \tag{4}$$

and s_{ki} is i-th object significance in k-th scene. To normalize the distance similarity measure we calculate a distance difference matrix ΔD:

$$\Delta D = [\delta_{ij}]_{m\times m} = \begin{cases} \frac{\|d_{1,ij}-d_{2,ij}\|}{d_{diag}}, & \text{if} \quad d_{1,ij} \le d_{diag} \wedge d_{2,ij} \le d_{diag} \\ 1, & \text{if} \quad d_{1,ij} > d_{diag} \vee d_{2,ij} > d_{diag} \end{cases} \tag{5}$$

where $d_{diag} = max\left(d_{diag1}, d_{diag2}\right)$ is the maximum (diagonal) distance in both scenes. This guarantees that $0 < \delta_{ij} < 1$ for all i, j. If at least one of the two elements for which the distance difference is computed, does not exist in one of the scenes, such distance difference takes the maximal value 1. Then we calculate the normalized scene similarity as

$$S_{SC} = \frac{\sum_{i=1}^{m} \sum_{j=i+1}^{m} \left(1 - d_{ij}\right)^{p} s_{ij}}{\sum_{i=1}^{m} \sum_{j=i+1}^{m} s_{ij}}. \qquad (6)$$

This scene similarity has values between 0 (dissimilar) and 1 (identical). The p factor is an empirical constant, which is adjusted to control the sensitivity of the similarity measure - the larger p, the more sensitive is the measure to differences in the compared scenes.

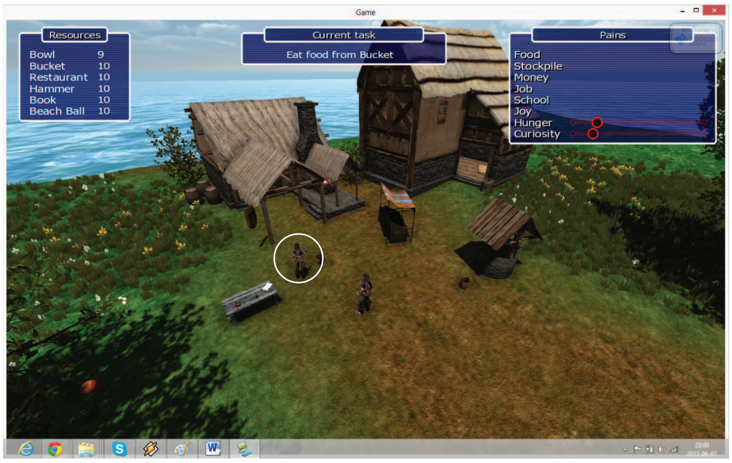
When the objects are not unique, which means that a scene may have a number of objects with the same symbol (i.e. they are indistinguishable), a similar approach can be applied. However, since in this case there is no unique ordering of the symbols due to repetitions, we must either examine all combinations of ordering of the repeated symbols that is NP hard, or use a heuristics, where we assign symbol pairs based on similarities of their corresponding distance vectors $d_{1,ij}$ and $d_{2,ij}$ within uniquely defined scene elements.

5 The VEEMA Simulation Environment

Our current implementation of the cognitive agent is designed to cooperate with a virtual environment called VEEMA (Virtual Environment with Embodied Motivated Agent). This environment is based on the NeoAxis 3D Game Engine [8]. The agent is able to explore the virtual world similarly as a robot could do it in the real world. The advantage of using the virtual environment is full control over the environment, which is impossible in the real world conditions. In this way we do not need to care about lighting conditions, or other technical details. Such a solution allows to concentrate on selected parts of the cognitive agent development problem to be solved. This approach is also cost effective, because it does not require buying a physical robot.

A screenshot of the VEEMA environment is shown in Fig. 5. The figure in the middle is the motivated agent exploring its environment. In the current implementation the agent is able to see only selected objects. The complete analysis of the video stream will be implemented in the future version of the agent.

In the current version, the environment explored by the agent is relatively simple, and far from complexity of real world environments. It was adapted to a motivated learning scenario [5], in which the agent is searching among a collection of resources, and learning some skills to survive in the environment. The agent can focus attention on objects placed in environment, and build the internal scene representation. The layout of objects seen by the agent is shown in Fig. 6(a). Some objects are located within the buildings in the scene Fig. 6(b),



Game

Resources

Bowl	9
Bucket	10
Restaurant	10
Hammer	10
Book	10
Beach Ball	10

Current task

Eat food from Bucket

Pains

Food
Stockpile
Money
Job
School
Joy
Hunger
Curiosity

Fig. 5. A third person camera view on the VEEMA environment. The white circle indicates the agent

but the buildings in the current implementation play only illustrative role, and are not recognized cognitively by the agent.

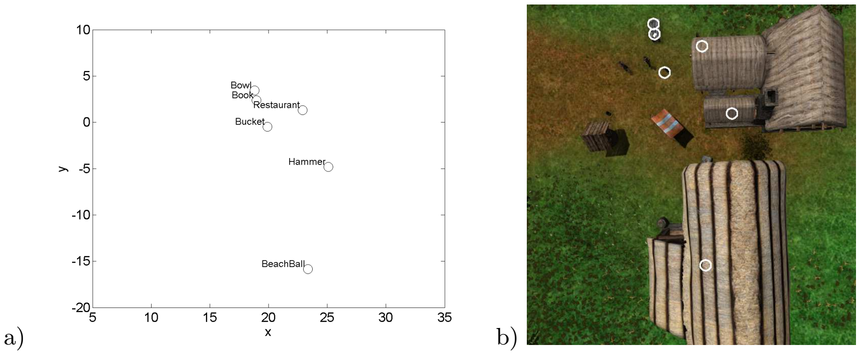


Fig. 6. The layout of objects in the considered learning scenario: (a) projection on the x-y plane, (b) view from the top camera with indicated locations of objects

6 Experimental Results

To test performance of the scene comparison algorithm described in Sec. 4 we constructed a number of experiments, in which two scenes with different locations and types of objects are compared.

Here we present one of the experiments, which shows how the similarity value changes, when the first scene consists of a fixed set of objects (6 unique objects), and the second scene consists initially of 2 objects, and then subsequent objects are added to the scene at the same positions as in the first scene (Fig. 7(a)). The numbers in the figure indicate type of objects. When the number of objects in the second scene reached the number of objects in the first scene (identical scenes $\equiv$ similarity measure=1), a sequence of new objects (not present in scene 1) was added until their number reached 11. The positions of newly added objects were chosen randomly. The final configuration of both scenes is shown in Fig. 7(b). Objects from the first scene are indicated by crosses, while objects from the second scene are indicated by circles.

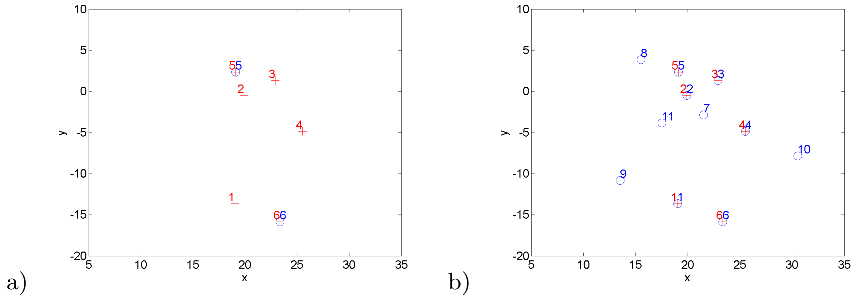


Fig. 7. The layout of objects in the experiment: (a) the first stage of experiment, (b) the final stage of experiment

The set of similarity values obtained in the experiment is shown in Fig. 8. As one can see, initially similarity increases monotonically with the increasing number of objects, and after reaching maximum (=1) decreases with the number of newly added objects to the second scene. This is what we expected from the similarity measure. The presented results are obtained for $p = 10$ in Eq. 6. This value seems quite accurate, because it allows for sharp distinctions between the scenes. However, it is possible that for more complicated scenes, slightly different values can be applied.

7 Conclusions

The presented paper discusses the problem of building internal environment representation within a cognitive system steering a mobile robot. A methodology was presented, which starts from saccadic scene observation. The data from a sequence of saccades are transformed into an internal scene model, which after memorizing in the episodic memory, can be further used for recognizing known places or complex objects. The internal scene representation is based on the matrix of distances between objects. Such representation allows for fast scene

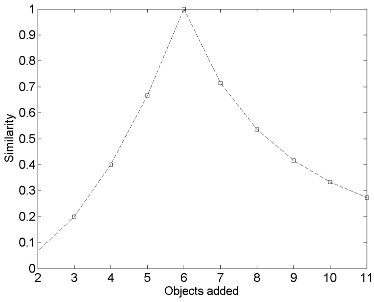


Fig. 8. Similarity for each step of the experiment

comparison, which can be used for comparing the currently observed scene with the contents of the episodic memory. The same mechanism can be applied to analysis of complex objects contained within scenes.

Sample experiments based on a virtual 3D environment were performed, in which the performance of the proposed similarity measure was demonstrated. The results show, that the proposed methodology gives satisfying results, however, the scene created in the virtual environment is simplified, and of much smaller complexity, than real world scenes. Thus the aim of future work will be extending the system to be able to process all the data delivered by a video stream, either registered in the virtual simulation or in real environment. The other research direction is developing a method for objects representation. In the current version of our system the objects have no internal structure.

Acknowledgement

The research is supported by The National Science Centre,
grant No. 2011/03/B/ST7/02518.

References

1. Langley, P., Laird, J.E., Rogers, S.: Cognitive architectures: Research issues and challenges. Cogn. Syst. Res. 10, 141-160 (2009)
2. Yarbus, A.: Movements of the eyes, Plenum Press, New York (1967)
3. Frintrop, S., Rome, E., Christensen H.I.: Computational Visual Attention Systems and their Cognitive Foundation: A Survey, ACM T. Appl. Percept. 7(1), 1-39 (2010)
4. Begum, M., Karray, F.: Visual Attention for Robotic Cognition: A Survey. IEEE Trans. Auton Ment. Devel. 3(1), 92-105 (2011)
5. Starzyk, J.A., Graham, J.T., Raif, P., Tan, A-H.: Motivated Learning for the Development of Autonomous Systems, Cogn. Syst. Res. 14, 10-25 (2012)
6. Brooks, R.: Intelligence without representation, Artif. Int. 47, 139-159 (1991)
7. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press, Cambridge (2003)
8. The NeoAxis Game Engine, http://www.neoaxis.com/