Navigation of uncertain terrain by fusion of information from real and synthetic imagery

Damian M. Lyons^a, Paramesh Nirmal^a, and D. Paul Benjamin^b, ^aFordham University, Robotics and Computer Vision Laboratory, Bronx NY 10458; ^bPace University, Department of Computer Science, New York NY 10023

ABSTRACT

We consider the scenario where an autonomous platform that is searching or traversing a building may observe unstable masonry or may need to travel over unstable rubble. A purely behaviour-based system may handle these challenges but produce behaviour that works against long-terms goals such as reaching a victim as quickly as possible. We extend our work on ADAPT, a cognitive robotics architecture that incorporates 3D simulation and image fusion, to allow the robot to predict the behaviour of physical phenomena, such as falling masonry, and take actions consonant with long-term goals.

We experimentally evaluate a cognitive only and reactive only approach to traversing a building filled with various numbers of challenges and compare their performance. The reactive only approach succeeds only 38% of the time, while the cognitive only approach succeeds 100% of the time. While the cognitive only approach produces very impressive behaviour, our results indicate how much better the combination of cognitive and behaviour-based can be.

Keywords: cognitive robotics, navigation, sensory fusion.

1. INTRODUCTION

In previous work [10][12][13] we have considered a scenario where an autonomous platform that is searching an area for a target may observe unstable masonry or may need to travel over, by or through unstable rubble. One approach to allow the robot to safely navigate this challenge is to provide a general set of reactive behaviors that produce reasonable behavior under these uncertain and dynamic conditions. However, this approach may also produce behavior that works against the robot's long-term goals, e.g., taking the quickest or safest route to a disaster victim or out of the building. In our work we investigate combining a behaviour-based approach with the Cognitive Robotics paradigm of rehearsal to produce a hybrid reactive-deliberative approach for this kind of scenario.

We have proposed an approach, ADAPT, that leverages the state of the art in physics-engines for gaming so that for a robot reasoning about what actions to take, perception itself becomes part of the problem solving process. A physics-engine is used to numerically simulate outcomes of complex physical phenomena. The graphical image results from the simulation are fused with the image information from robot's visual sensing in a perceptual solution to the prediction of expected situations.

In the current paper, we extend our prior static work to handle the motion of a mobile platform through an indoor environment. We will assume that the static components of the scene, walls and floors, have been modelled in the physics engine. However, dynamic objects can appear, are identified as unexpected, and the physics engine is used to predict their behaviour. The robot can then select motion that is both safe and works towards long-term goals. We make two simplifications to the problem. First, we will assume that the only unmodeled objects are those whose shape and behaviour can be classified in one image; for example, an unstable or collapsing column of masonry. A rolling, bouncing object would not meet this assumption (though we showed in [12] for an otherwise static scene our approach will handle this). Since this assumption is difficult to implement in a physical evaluation framework, we introduce a second simplification in which the robot traverses a (second) simulated scene rather than a real scene.

In section 2 we review the current literature related to this problem. In section 3 we describe the building traversal scenario, and in section 4 present the cognitive navigation architecture that we have developed to address the problem. To demonstrate the performance benefits of our approach, in section 5 we present an experimental comparison of a reactive only and a cognitive only approach to the building traversal problem (inspired by Braun et al. [4]). Results are gathered for 30 traversals of a building with a random number (between 1 and 4) of randomly appearing (size, color and initial velocity) unstable obstacles. The results show that the reactive only approach will (as expected) fail in cases

where the obstacles are moving too quickly to be avoided (62% of the cases in our evaluation), whereas the cognitive approach works in every case. However, our separation of reactive and cognitive approaches is purely to allow us to compare performance for this scenario; we will argue that for a real scenario robot platform will need both capabilities and we can estimate the performance improvement by including both.

2. PRIOR WORK

A central idea in cognitive robotics is that that simulation, the 're-enactment of perceptual, motor and introspective states' [17] is a key cognitive mechanism. For example, Cassimatis et al. [5] integrate a 3D simulation as a 'specialist' within the Polyscheme/Polyrobot architecture. Shanahan [17] has proposed a neurologically plausible cognitive architecture that includes rehearsal. Hall's Internal Rehearsal System [8] is based on Shanahan's concepts, leveraging a 3D simulator to select collision free robot arm motions. In these systems, the simulator is an inherent module of the architecture and communicates with the other modules by an agreed upon common language or protocol.

Macaluso and Chella [15] also include a simulator in their CiceRobot robot architecture. However, in CiceRobot the simulator generates a set of 2D scenes that are then compared to real video to help localize the robot. However, the comparison only involves comparisons of artificial visual location markers, not of the scene or the scene contents. In previous work we have introduced ADAPT [3], an architecture for cognitive robotics. ADAPT merges RS [11], a language for specifying and reasoning about sensory-based robot plans with SOAR [9] a widely used cognitive architecture. The visual component of the perceptual apparatus for ADAPT, introduced in [10] and developed in [12][13], leverages the state of the art in physics-engines for video gaming to provide a fast and effective 3D simulation system. A physics-engine is a module used in video game software to ensure that the virtual game world obeys the laws of physics and looks realistic to the game player. The physics engine numerically simulates the outcomes of complex physical phenomena, such as falling masonry and collapsing ceilings as well multiple colliding and bouncing objects. The state of the art in physics-engines for gaming is such that these complex multibody simulations can be carried out very quickly¹.

Physics-engine software typically commits to a fairly standard taxonomy of object shapes and properties. To avoid the issue of having a behaviour-based robot commit to or be limited by this exact same description of its environment, we followed Macaluso and Chella [15] in restricting the interaction between the simulation and robot to the visual comparison of the output image from the simulation and the robot's camera input image. This has the advantage of keeping the robot control software and the simulation software quite separate (so it is easy to adopt new and better physics-engine software as it becomes available). However, it also separates the two at a semantic level, so that the robot's understanding and representation of its environment can be quite different from that of the physics-engine. Rather than looking for artificial landmarks to aid localization, as Macaluso and Chella did, our objective here is to compare natural scene content between the real and synthetic images.

While fusing multiple real images poses some difficult challenges, fusing real and synthetic images posed a whole new set of problems. In [10][12] we introduced an approach, called the *match-mediated difference* (MMD), to combining the information from real and synthetic images for static scenes containing real and/or simulated stationary and moving objects. We leveraged Itti and Arbib's [14] concept of the minimal subscene (developed for discourse analysis) to capture how the robot modelled the scene being viewed, how it deployed the simulation and the MMD operation to determine unexpected scene elements and how it requested and assimilated simulation predictions. The minimal subscene contains a network of interconnected processes representing task and perceptual schemas [1].

In this paper, we continue to leverage the minimal subscene concept to include the introduction of novel objects seen by the robot and to select a response to them. In schema theory [1], a perceptual schema interprets sensory input in terms of the ongoing task. So, rather than exploiting a saliency or a Bayesian surprise measure to notice such objects (e.g., [16]), in our approach we will interpret any sensory input that is not expected using the perceptual schemas of the minimal subscene (to check for a potential obstruction) and either used or dismissed as they decide.

3. BUILDING TRAVERSAL SCENARIO

Robot systems are increasingly involved in search and rescue situations. We are specifically interested in search and rescue situations within buildings. Because the search and rescue is typically a response to a disaster situation – a gas main explosion, the aftermath of a natural disaster, and so forth – the building traversed may be in poor condition. Currently, robots are used as remote eyes and hands for rescue personnel; that is, the robots are teleoperated by humans

¹ http://www.geforce.com/hardware/technology/physx

from a remote location. Because the teleoperation user interface poses a severe challenge to rescue personnel, and because the disaster environment is not conducive to stable and continuous wireless connectivity, a robot capable of useful autonomous action (while keeping rescue personnel 'in the loop' whenever possible) would have major advantages over a teleoperated vehicle in terms of ease of use and effective deployment.



Figure 1: Example of a building traversal scenario: (a) perspective view (b) floor plan

We consider an example scenario, as shown in Figure 1 to illustrate the kind of decisions with which an autonomous robot will be confronted. The robot has been deployed to a building that has been involved in a disaster. The robot has been assigned the mission of following a traverse through the building to a specific location and then exiting the building (Figure 73 shows the specific building and traversal route for the experimental evaluation in this paper). For example, the robot may be searching for potential victims, or the robot may have been tasked with going to the known location of a victim for the purpose of bringing aid or communications, or the robot may be searching for a specific object, an explosive or biological weapon. In Figure 1 (a), the robot has just entered a room. The room is filled with debris, some of which is unstable. The robot needs to make a decision about how to traverse the room safely and in a manner that allows it to complete its mission. In Figure 1(b) a section of the floor plan is shown indicating the route the robot should follow. The robot needs to exit the room by the unstable masonry on the shortest, or main route to the goal.

If the unstable masonry column in the room begins to collapse, the robot will need to move to stay safe. If the avoidance motion is purely reactive, then the robot may end being hit by the column (if it cannot react in time) or else may spend time avoiding the column, until such time as the column settles and the robot can determine if its path has been blocked. A better solution is for the robot to recognize that the unstable masonry poses a potential problem, and rather than continuing on its path, it should determine whether the masonry will cause a blockage if it collapses. This prediction has the advantage of allowing the robot to quickly switch to the alternate route, and not wasting time dealing with the column. The prediction needs to be accurate and fast. Switching to the alternate route whenever a pile of unstable masonry is seen could easily be as inefficient as waiting to react to the masonry, so the alternate path should only be chosen if the prediction is that the path is blocked. Similarly, unless the prediction is faster than real-time, it would more efficient to just wait to see what happens and react to that.

The example of a single column of falling masonry is just one kind of dynamic event that a robot traversing a disaster building may encounter. Other examples include parts of the roof falling in, unstable heaps of rubble on the floor, and objects that bounce and collide with others before settling. Any proposed approach to this problem would need to be able, at least in principle, to handle all these situations.

4. NAVIGATION ARCHITECTURE

The navigation architecture introduced here is heavily based on the ball tracking architecture we introduced in [12]. However, the minimal subscene module in this case includes (see Figure 2):

- a navigation task schema that directs the robot to follow waypoints,
- a waypoint perceptual schema representing the list of waypoints and alternates for traversing the building, and
- an obstruction perceptual schema that is invoked when unexpected objects are detected and which directs the prediction and blockage checking.

Minimal Subscene



Figure 2: A Cognitive Navigation Architecture

4.1 Navigation of Building Traversal Way-points

The minimal subscene module is shown at the top of Figure 2. It contains the navigate, waypoint and obstruction schemas. The navigate building task schema requests waypoints along the building traverse from the waypoint perceptual schema. For each waypoint, the navigate schema issues the motion commands to the robot to drive it towards that waypoint. Arrival at a waypoint is recognized only by odometry in this implementation. The closest obstruction to the robot is represented by the potential obstruction perceptual schema. Travel to the intermediate goal of a waypoint while avoiding an obstruction is modeled using Arkin's potential field approach to local obstacle avoidance [2]: A positive potential field attracts the robot to the goal while a negative potential field repels the robot from the obstruction.

In the case that the obstruction schema signals that the way is blocked (how this is deduced is explained in section 4.4), the navigation schema requests an alternate waypoint from the waypoint schema. It is assumed that the waypoint schema has all the information on the waypoints in the building, including alternate paths if they exist. This schema could be implemented with dynamic path-planning or by a simple table of waypoints and alternates. We have implemented it as the latter since it is somewhat outside the scope of our problem area.

4.2 Fusion of Real and Synthetic Imagery for Synchronization

The schemas within the minimal subscene depend on lower-level visual and motor routines to implement their behaviors. The simulation functionality is also a lower-level functionality, directed and used by schemas in the minimal subscene. The motion commands from the navigate schema are sent to the motor control module of the robot, causing it to move towards the next waypoint. The same information is *also* sent to the simulation module, causing the simulated position of the robot to change in tandem with its actual motion. Of course, the actual motion of the robot may deviate from its commanded motion, because of unmodelled interactions with the terrain such as slippage. For this reason, the location of the robot in the simulation module may drift from the corresponding real-world location.

The images from the robot camera I_r and simulation camera I_s are fused using the MMD module [10]. The first output of the MMD operation is the image homography that maps between the real and synthetic images H_e . The other outputs are two sets of image difference regions: C_{sr} represents objects missing from the real scene but present in the synthetic image from the simulation, and C_{rs} represents objects in the real scene and not in the synthetic scene.

$$MMD(I_r, I_s) = (H_e, C_{sr}, C_{rs})$$

Each $c_i \in C_{sr}$, $C_{rs} \subseteq \{0 \dots U_m\} \times \{0 \dots V_m\} \subseteq N^2$ represents an image region, a set of pixels in the image identified by their natural number image coordinates $\{0 \dots U_m\} \times \{0 \dots V_m\}$. The homography information is used to incrementally

correct the simulation position using the formula (based on [7]):

$$H_e = K_s^{-1} \left(R - \frac{\tau n^T}{d} \right) K_r$$

Where H_e is the homography from the MMD module, K_s and K_r are the synthetic and real camera matrices, *n* is the normal to, and *d* the distance from the scene, and R and τ are the relative rotation and translation of one camera with respect to the other. For more detail on this part of the problem, see [12].

4.3 Finding Unexpected Objects

The MMD operation was developed in [10] to effectively fuse information from real and synthetic views of the same scene. The constraint that the real and synthetic scenes have similar appearance was relaxed in [13] so that only the structure needed to be similar. We use this version of the MMD operation here. Figure 3 (a-h) shows the MMD stages in the identification of an unexpected object, and Figure 3 (i-p) shows the same situation with no unexpected scene elements. Any unexpected object difference information (i.e., in C_{rs}) generated by the MMD operation is directed to the obstruction schema, which determines if the difference region could be a potential object using the rule:

$$C_{obs} = \frac{MAX}{i} \quad A(c_j \in C_{rs}) \quad and \ c_j > \tau_A$$

where A(c) is the area of the region *c*. This reasoning is based purely whether the largest difference region exceeds a threshold size τ_A . If it does, then the schema attempts to fit a *rectangular prism* to the region and locate it accordingly in the room. The ellipse fitting function *Ell(c)* returns the major and minor axis and major axis angle of the best fit ellipse for the region *c*, and this information is used as the height, width, and depth of the prism



PRISM(m1, m2, m2), where
$$Ell(C_{obs}) = (m1, m2, \theta)$$

Figure 3: Intermediate stages in MMD processing when presented with an unexpected object (a-h; and no object (i-p); (a) real scene with object, (e) corresponding simulation scene with corner points shown, (b) warped image difference, (f) match-mediated difference mask, (c,g) Canny edge images, (d) real edge image anded with warped difference image divided by difference mask (MMD operation), (h) simulation edge image MMD; (i-p) are the same steps but for no unexpected scene elements.

The angle of the new prism with respect to the vertical is given by θ , and the position of the prism in the simulation scene is obtained using the camera calibration information, the centroid $ct(C_{obs})$, and the assumption that the prism rests

on the floor. This is the point at which the assumption we made initially that everything that needs to be known about the object can be gleaned in one image, is used; e.g., the angle will determine the prism's stability.

The obstruction schema directs the simulation module to add the new object to the simulation at the appropriate location in the scene. Once the object is added, the schema directs the simulation to go in 'fast-forward' mode, to simulate what happens in the near future (a fixed time horizon of a few seconds). If the object was unstable, then it will fall, possibly interacting with other objects, eventually coming to rest. The time horizon was selected to allow sufficient time for the object to settle.

4.4 Navigation Impact of Predicted Behavior

The MMD operation is used a second time by the obstruction schema, this time comparing the 'fast-forward' simulation scene to the actual scene. The decision about whether the route to the waypoint is obstructed is made by first examining the MMD 'missing object' difference but using the same rule as before (See Figure 4):



Figure 4: MMD comparison of real (a) and predicted scenes (b) producing difference (c) used for navigation.

The predicted location of the object is then analyzed to see if it blocks the route to the waypoint. This analysis is done by visual inspection. The center bottom of the image $(U_m/2, V_m)$ is used as a reference to the current location of the robot. The waypoint schema marks the location of the next waypoint in each simulated visual image with a distinctive visual beacon (a saturated red sphere in Figures 4 and 5). This allows the waypoint to be easily identified for analysis. The route is considered not blocked as long as:

$Line((U_m/2, V_m), (B_u, B_v)) \cap C_{obs} = \emptyset$

where Line(p,q) is the set of points in the line segment between p and q and (B_u, B_v) is the location of the beacon. Of course its possible that even if the intersection is not empty, it may be feasible to navigate around the obstacle and over to the waypoint. We can check the distance from the obstacle to the wall, to catch this case. However, here, all the obstacles were placed so as to block the doorways if they fell, so this additional position test was not employed.



Figure 5: Illustration of blockage calculation

5. EXPERIMENTAL COMPARISON

To evaluate the benefits of this cognitive approach to navigation in a dynamic environment, we developed a performance test: a simulated building through which the robot needs to navigate and which has various kinds of dynamic objects that need to be considered. To simplify the comparison, both the real and synthetic videos *are in fact simulated*. We use the NVIDIA PhysX physics engine with its basic OpenGL rendering routines. Two copies of the simulation are started: a parent, 'real imagery' simulation and a child 'synthetic imagery' simulation. Both copies read a building description and construct the building according to the description. However, the real building has a wide variety of possible appearances achieved by modifying the colors and textures on the walls, whereas the synthetic building always has the same appearance. While comparing two simulation-generated, synthetic images is much easier than comparing real and

synthetic images, by allowing the appearance to vary widely we can demonstrate the MMD operations robustness in a manner similar to that encountered when we compared real and synthetic images [13]. The parent simulation communicates the same way that the real robot would communicate with the simulation, that is, as shown in Figure 2.

5.1. Experimental Scenario

A 15 room building was designed so as to present space for the robot to be confronted with 'challenges' and be able to respond to the challenges by either continuing a traverse through the building or selecting an alternate path. Figure 7(a) shows the simulation model of the building from above. The entrance and exit doorways are on the bottom right and left. There are several large rooms with multiple doors which are the areas in which the robot can respond to challenges. There are also a number of smaller rooms which offer alternate routes through the building. Figure 7(b) shows the main waypoints (solid arrows) and alternate routes (dashed arrows). This information is stored in the waypoint schema.

(b)

(a)





Figure 7: 15 Room Building floor plan used in evaluation (a) overhead perspective of simulation model of building, (b) traverse waypoints (solid) and alternates (dashed).

The robot makes the traverse of the building in either reactive mode (with the feedback from the simulation disengaged, so that no predictions are offered) or in cognitive mode (using predictions). For each run, the simulation building always appears the same. However, the real building can be altered dramatically as follows:

- 1. From one to four unstable columns of masonry can be placed as challenges, one in each of the large rooms.
- 2. The masonry can vary in color, in size and in initial velocity (how fast it falls).
- 3. The background wall colors can vary in color and texture.

A series of 30 runs was conducted with random variations in the real building characteristics. On the reactive runs, the robot was capable of avoiding falling obstacles as long as they were not falling too quickly using its potential field obstacle avoidance. On the cognitive run, the robot always checked its visual imagery and in the case of an unexpected object, carried out and acted on the basis of a prediction.

5.2. Results

The results of the 30 runs for reactive and cognitive trials are shown in Table 1. The average number of challenges in both trials was similar (2.8 reactive and 2.7 cognitive). The reactive trial was only successful 38% of the time; the remainder of the time the robot was hit by falling masonry. The cognitive trial was successful 100% of the time.

I able 1: Evaluation results		
	Reactive	Cognitive
Number of Trials	30	30
Number of Challenges	Av=2.8, 1SD = 1.0	Av=2.7, 1SD=1.2
Percent Successful	38%	100%
Distance Travelled	Av=1929,1SD=998	Av=2952, 1SD=235
as a % of max	73%	89%
Number of Waypoints	Max=19,Min=5	Max=23, Min=19
as a % of max	75%	94%
Number of Detours	0	Av=2.7, 1SD=1.2
Number of Corrections	Av=10, 1SD=5	0

The number of cognitive detours is the same as the number of cognitive challenges: every challenge was detoured around. The number of waypoints as a percentage of the maximum was 94% meaning that the robot was very frequently taking a long route. This evidence is supported by the distance travelled. However, we know from the reactive case, that

perhaps a third of these cases could have been shortened if we had allowed a combined reactive and cognitive ability. Recall, we purposely separated these for the purpose of testing.

6. CONCLUSIONS

This paper has presented a cognitive architecture for navigating a building in which the robot may encounter situations that require it to balance short and long term mission objectives. Our previous work on fusing information in real and synthetic imagery was leveraged to allow the robot to predict the outcome of complex events such as falling masonry and to take appropriate action. We conducted an evaluation, comparing a reactive only building traverse with a cognitive only building traverse. While the reactive runs were only occasionally successful, the cognitive runs were always successful. However, our results showed that when recombined the combination has the potential to be better than either.

The work presented here did not attempt to track moving objects, or indeed include any kind of object other than falling masonry columns. In addition, the falls were always terminal in that they blocked the doorways when done. Including both of these cases would allow a testing of the full, combined reactive and cognitive architecture, rather than the 'artificial lesion study' reported here.

The appearance variability in this study was limited enough that the MMD operation never failed. In fact, with issues of contrast and lighting, we would expect that the operator will sometimes fail. The only effective way to measure this is with comparison to real video as was used in our earlier papers. Although no saliency or surprise information is currently being used in our implementation, this (e.g. Maier and Steinbach [16]) offers a good alternative for making the MMD robust in difficult contrast and lighting situations.

References

- [1] Arbib, M.A., "The Handbook of Brain Theory and Neural Networks." MIT Press (2003).
- [2] Arkin, R.C., "Behaviour-based Robotics." MIT Press (1998).
- [3] Benjamin, D.P., Lonsdale, D., and Lyons, D.M., "Embodying a Cognitive Model in a Mobile Robot." Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision, Boston, October (2006).
- [4] Braun, J., Bergen K., and Dasey, T., "Inner rehearsal modeling for cognitive robotics," In: *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, SPIE Defense and Security Symposium*, Orlando (Kissimmee), FL, April (2011).
- [5] Cassimatis, N., Trafton, J., Bugajska, M., Schulz, A., "Integrating cognition, perception and action through mental simulation in robots." *Robotics and Autonomous* Systems N49, pp13-23 (2004).
- [6] Antonio Chella, Marilia Liotta, Irene Macaluso, "CiceRobot: a cognitive robot for interactive museum tours." Industrial Robot: An International Journal, V34 N6, pp.503 – 511, (2007).
- [7] Faugeras, O., and Lustman, F., "Motion and structure from motion in a piecewise linear environment," *Int. J. Patt. Recog & AI.* 2(3) (1998).
- [8] Hall, J., "Internal rehearsal for a cognitive robot using collision detection," MS Thesis, Dept. of Elec. Eng. Vanderbilt Univ. (2007).
- [9] Laird, J., Newell, A., Rosenbloom, P., "Soar: An Architecture for General Intelligence." Art. Int. 33 (1987).
- [10] Lyons, D.M., and Benjamin, D.P., "Locating and Tracking Objects by Efficient Comparison of Real and Predicted Synthetic Video Imagery." SPIE Conf. on Int. Robots and Comp. Vision, San Jose CA, Jan. (2009).
- [11] Lyons, D., and Arkin, R.C., "Towards Performance Guarantees for Emergent Behavior." *IEEE Int. Conf. on Robotics and Automation*, New Orleans LA, April (2004).
- [12] Lyons, S. Chaudhry, Marius Agica and John Vincent Monaco, "Integrating perception and problem solving to predict complex object behaviors." In: *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, SPIE Defense & Security Symposium*, Orlando (Kissimmee), FL, April (2010)
- [13] Lyons, D., and Benjamin, D.P., "A relaxed fusion of information from real and synthetic images to predict complex behavior," In: *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, SPIE Defense and Security Symposium*, Orlando (Kissimmee), FL, April (2011).
- [14] L. Itti, M. A. Arbib, "Attention and the Minimal Subscene," In: Action to Language via the Mirror Neuron System, (M. A. Arbib Ed.), pp. 289-346, Cambridge, U.K.:Cambridge University Press, (2006).
- [15] Macaluso, I., and Chella, A., "Machine Consciousness in CeceRobot, a Museum Guide Robot." Proceedings, *AAAI Fall 2007 Symposium*, Arlington VA, (2007).
- [16] Maier, W., and Steinbach, E., "Surprise-Driven Acquisition of Visual Object Representations for Cognitive Mobile Robots," *Int. Conf. Robotics & Automation*, Shanghai China (2011).
- [17] Shanahan, M.P., "A Cognitive Architecture that Combines Internal Simulation with a Global Workspace." Consciousness and Cognition, vol. 15, pages 433-449, (2006).