A relaxed fusion of information from real and synthetic images to predict complex behavior

Damian M. Lyons[a] and D. Paul Benjamin[b],

[a]*Fordham University, Robotics and Computer Vision Laboratory, Bronx NY 10458;*

[b]*Pace University, Department of Computer Science, New York NY 10023*

ABSTRACT

An important component of cognitive robotics is the ability to mentally simulate physical processes and to compare the expected results with the information reported by a robot's sensors. In previous work, we have proposed an approach that integrates a 3D game-engine simulation into the robot control architecture. A key part of that architecture is the Match-Mediated Difference (MMD) operation, an approach to fusing sensory data and synthetic predictions at the image level. The MMD operation insists that simulated and predicted scenes are similar in terms of the appearance of the objects in the scene. This is an overly restrictive constraint on the simulation since parts of the predicted scene may not have been previously viewed by the robot.

In this paper we propose an extended MMD operation that relaxes the constraint and allows the real and synthetic scenes to differ in some features but not in (selected) other features. Image difference operations that allow a real image and synthetic image generated from an arbitrarily colored graphical model of a scene to be compared. Scenes with the same content show a zero difference. Scenes with varying foreground objects can be controlled to compare the color, size and shape of the foreground.

Keywords: cognitive robotics, problem-solving, simulation, computer vision, sensory fusion.

1. INTRODUCTION

For robot systems to be able to carry out complex behaviors in an unstructured environment, they need to be capable of some level of cognitive reasoning. For example, a mobile robot mapping a disaster area should be able to assess how fragile a partially fallen structure appears to be, and where to move to avoid any danger if the structure should collapse. In previous work, we have developed a cognitive robotics approach that combined a 3D simulation with behaviour-based action and deliberation. The 3D simulation allows the robot to model and predict physical events, such as (many) objects falling and colliding, to determine how to act. We use the Nvidia PhysX[1] real-time physics engine with OpenGL rendering as our 3D simulation engine.

Our approach constrained the simulator to try to shape and color the components of the synthetic world as close to those in the real, sensed world as possible. If a robot is traversing an urban disaster site and predicts that a nearby column of masonry is leaning dangerously enough that it may collapse, there is no reason to ask that the simulator be able to color and shape the chunks of masonry in the collapsed wall. In the current work, we expand the difference operation so that graphical images (such as, e.g., Figure 2(d)) can be compared to real scenes (such as, e.g., Figure 2(a)). We begin with a review of the prior work in the next section. Section 3 reprises our existing work to set the scene for the extended comparison operation in Section 4. The final section summarizes our results and compares them to other results in the field.

2. PRIOR WORK

Recent evidence in cognitive psychology [19] and neuroscience [18] supports the proposition that simulation, the 're-enactment of perceptual, motor and introspective states' is a central cognitive mechanism. Shanahan [19]

[1] http://www.nvidia.com/object/physx_new.html

proposes a large-scale neurologically plausible architecture that allows for direct action (similar to a behavior-based approach) and also ‘higher-order’ or ‘internally looped’ actions that correspond to the ‘rehearsal’ or simulation of action without overt motion. Barsalou [2] proposes that distributed structures in the brain’s feature and association areas learn to recognize categories of experience. He calls these *simulators* and proposes that they can recreate (simulate) small subsets of their content in what he refers to as situated conceptualizations. A *situated conceptualization* is an embodiment of a simulation in a context: A situated conceptualization of a bicycle in a context for repair might be very different than in a context for riding, and would include additional simulators to complete the embodiment. Barsalou argues that by running the situated conceptualization as a simulation, the perceiver can anticipate future perception.

Cognitive functions such as anticipation and planning operate through a process of internal simulation of actions and environment [18]. Indeed there is a history in the field of Artificial Intelligence of using ‘simulated action’ as an algorithmic search procedure, e.g., game trees, though such an approach typically has problematic computational complexity. The *Polybot* architecture proposed by Cassimatis et al. [6], and based on his *Polyscheme* cognitive framework, implements planning and reasoning as sequences of ‘mental’ simulations that include perceptive and reactive subcomponents. The simulations include not just the effect of actions, but also the understood ‘laws’ of physics (e.g., will a falling object continue to fall) and are implemented as a collection of specialist modules that deliberate on propositions of relevance to the robot. Macaluso and Chella [7][15] base their cognitive robot architecture *CiceRobot* on the concept of *emulators* as developed by Gärdenfors [9]. The use a 3D robot/environment simulator coupled in a feedback loop with the robot controller. Control commands are sent to both simulation and robot. The simulator generates a set of 2D images of all expected scenes and these are compared to the actual visual input in order to determine which most closely represents the actual scene.

Pezzulo [18] argues that the evidence in favour of simulation suggests that the cognitive infrastructure for a robot should incorporate the perceptual and motor capabilities of the machine as fundamental tools in cognition. As just one example, consider that spatial terms are often used to give a grounded interpretation to more abstract concept and lead to standardized ways to view abstract concepts such as magnitude (‘higher’ values and ‘lower’ values). This should be contrasted with an approach that views a robot’s sensors as a (transparent) tool with which to fill an object database for plan construction, and a robot’s motors as a (transparent) way to cause change in the robot’s external environment.

Although AI uses algorithmic search in a space of simulated actions as a problem solving approach, the typical starting point in is a design selection of the state space to represent the problem and the world. This selection is problem oriented and independent of the motor and sensory skills of the problem-solving agent. As an example, consider Xiao and Zhang [20] integration of a simulation into a robotic assembly task planning architecture.

In addition to being contraindicated by the evidence from cognitive psychology and neuroscience, this integration approach adds two additional difficulties: First, there is no general way to link the data structures of a simulation with the sensory apparatus of the robot. Second, selection of search space can have a serious impact on finding a solution [3].

In previous work we have introduced ADAPT [4][5] an architecture for cognitive robotics. ADAPT merges RS [12], a language for specifying and reasoning about sensory-based robot plans with SOAR[10] a widely used cognitive architecture. RS, based on Arbib’s ‘schema theory’ [1], represents robot plans as networks of perceptual and motor schemas. We added a 3D simulation engine that allows physical scenarios to be simulated as part of planning and learning. In [11] [13], we developed a visual subsystem for ADAPT that allows the 3D simulation to communicate with the robot in a language common to its sensors – a visual image of the world. Our problem requires comparing the synthetic and real imagery to look for differences between actual and predicted object behaviours. We have developed an approach called the Match-Mediated Difference (MMD) image that allows effective comparison of real and synthetic views of a scene. The MMD image method also allows the real and synthetic camera poses to be synchronized. In [11] we showed how this approach could be used to follow the behaviour of a rolling target through a collision event.

However, that work demanded that the simulated scene and the actual scene be very close in appearance. That is a problem for three reasons: First, it is difficult to apply the texture-mapping approach of [11] to locations or objects in the simulation that have not yet been seen by the robot. Second, the cognitive psychology evidence [2] suggests that the level of detail in simulators varies significantly. Finally, it is difficult to justify every simulation needing to be photorealistic in its results. In this paper, we tackle the problem of extending the visual architecture of [11] [13] to provide a more relaxed comparison operation, allowing graphical 3D models to be visually compared to camera images.

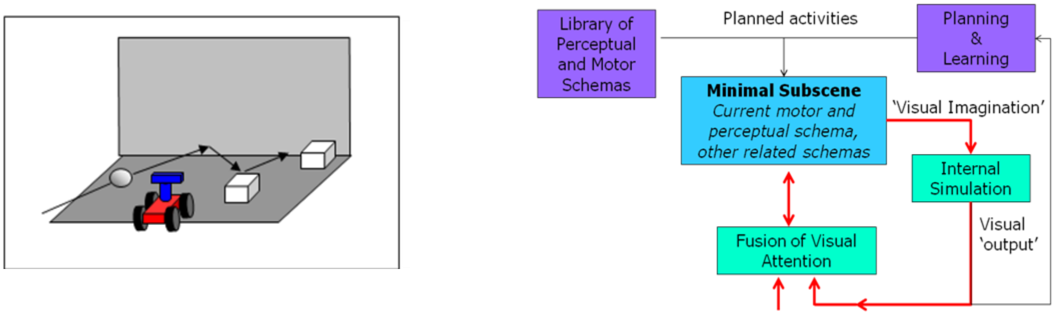
3. INTERNAL SIMULATION AND FUSION WITH VISUAL IMAGARY

In previous work, we have developed an approach designed to allow a cognitive robot system to reason about complex physical actions. As an example of such reasoning, we have chosen a task where a robot must predict the location of a moving object in order to intercept it. In the simplest example of such a scenario, visual tracking of a rolling object, e.g., a ball, can yield a robust solution (Mantz and Jonker [15]). If the scenario is expanded realistically to include the ball moving towards a wall or another unexpected agent then the unexpected collisions and sliding render tracking much more challenging. A purely tracking approach puts the robot in the position of always playing 'catch-up' with the target after a collision instead of predicting where it will be and moving there. This same issue can arise when a robot is operating in a complex dynamic environment, for example, an urban search and rescue robot moving in a field of semi-stable rubble which can topple and move in complex ways.

We start with a relatively simple scenario (see Figure 1(a)): A robot is positioned facing a wall. A ball is rolled across the field of view of the robot ultimately bouncing from the wall. The robot needs to intercept the ball after the bounce. Additional objects are placed by the wall so that ball bounces in a complex manner. In [11] we proposed and show results for a visual architecture designed for this task (Figure 1(b)). A brief introduction and review of that architecture is presented here to provide the context for the new results in the next section.

Itti and Arbib [14] define the *minimal subscene* as the middle ground between language and visual attention. Salient objects, the actions associated with them, and other objects associated with those actions are recursively gathered into the minimal subscene which then provides the context for discourse. We adopt this concept, and in our case, the minimal subscene provides a perceptual, problem solving context.

The minimal subscene is composed of a network of sensory and motor schema, put in place partially by the Soar module (top-down) and partially by ongoing perception (bottom-up). The elements of the subscene have corresponding elements in the simulation module. The fusion of visual attention module integrates the visual image generated by the simulation and the image from the video camera.


Library of Perceptual and Motor Schemas
Planned activities
Planning & Learning
Minimal Subscene
Current motor and perceptual schema, other related schemas
'Visual Imagination'
Internal Simulation
Visual 'output'
Fusion of Visual Attention

(a) (b)

Figure 1: (a) Predicting complex behavior (b) The Minimal Subscene

3.1 Synchronizing real and simulated scenes

Real and synthetic images of even identical looking scenes produce a large difference image because of the different methods of image generation. We developed the Match-Mediated Difference (MMD) to compare such images pairs effectively. The MMD method first looks for common corner features between both images. These matched features are used to first generate a homography mapping one image to the second H_e – this error homography then gives the camera pose correction. The synthetic camera pose is iteratively modified by mapping the error homography to a transformation of the camera pose until real and synthetic scenes are aligned. Secondly, the matched points are used to generate an MMD mask – if these points really correspond to the same features in both images, then we expect that the difference image should be zero close to these points. The MMD mask is used to enforce this constraint between the image pairs. Not only was the resulting comparison effectively able to align and compare real-synthetic image scenes with no foreground objects, and but also image scenes with expected and unexpected objects in the foreground [11][13].

3.2 Synchronizing real and simulated foreground

If any unexpected areas of difference are generated in the MMD comparison – that is, any area of difference not being already monitored by a perceptual schema in the minimal subscene of Figure 2(b), then a new perceptual schema is triggered to model and monitor and area and placed in the minimal subscene. The perceptual schema for a foreground object has the responsibility of both monitoring and modeling: monitoring the visual image for the object and interacting with the simulation to model the object behavior. The perceptual schema uses the MMD image information to adapt the simulation parameters of the object so that it more closely follows observed behavior, e.g., iteratively modifying the simulation velocity for a rolling target as in Lyons et al. [11][13].

4. RELAXING IMAGE FUSION

Our objective is to modify the image fusion operations so that the synthetic imagery need not be so similar to the actual camera imagery. Figure 2(a) shows the view from a robot camera showing one laboratory wall. Figure 2(b) is the standard 3D model used in our previous work, but shown from a different viewpoint so that the 3D structure is apparent. Figure 2(c) is a view of this model view-synchronized with the actual camera view. All our previous work used this model along with realistic-appearing foreground object such as boxes and balls. However, Figure 2(d) is a graphical model of the same scene and Figure 2(e) is a view of this model view-synchronized again with the camera view as was Figure 2(c). We would like to be able to compare Figures 2(a) and 2(e) using the approach we have used for comparing imagery such as Figures 2(a) and 2(c).

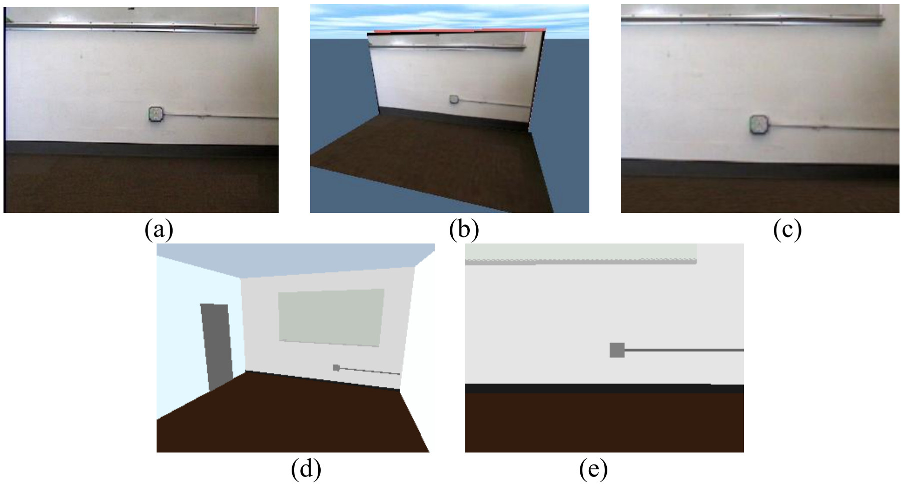


Figure 2: (a) Camera view (b) 3D synthetic texture-mapped model (c) in synchronized view (d) 3D synthetic graphic model (e) in synchronized view

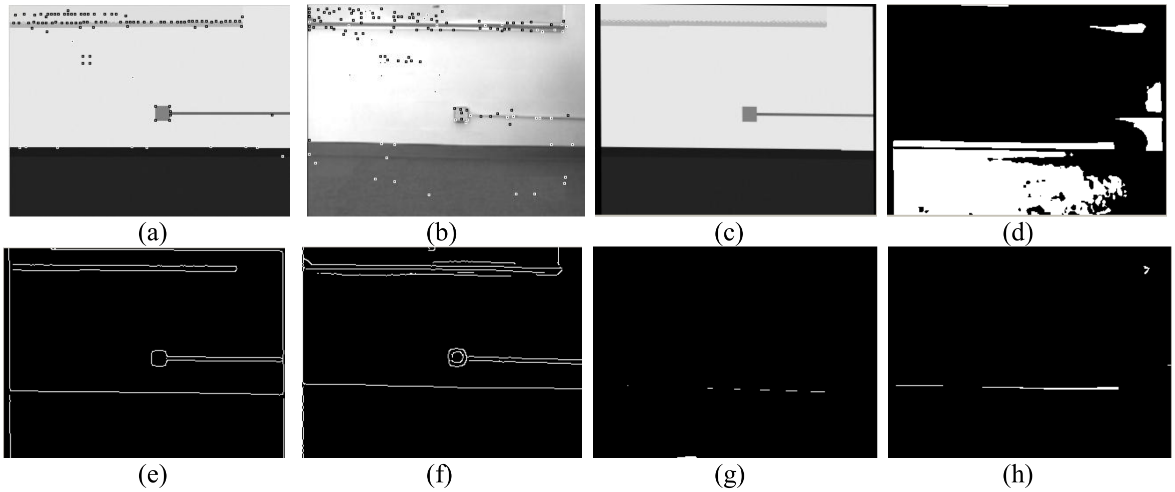


Figure 3: (a) Synthetic graphical scene with corner detections, (b) Camera image with corner detections, (c) Affine transformed synthetic scene, (d) Match-mediated difference; (e) and (f) show Canny edge images of (a) and (b), respectively, (g) and (h) are edge images (e) and (f) *anded* with the Match-mediated difference image (d), respectively.

Our relaxed constraint therefore is that the regions in the real and synthetic scene represent the same objects, though the two images may need to be aligned and the colors and textures of regions corresponding to the same objects might be quite different. The first aspect of this comparison, alignment, is not so difficult because the edge and corner structure of the two images remain similar with our relaxed scenario, and hence the corner detection and affine transformation developed in Lyons & Benjamin [11] are still sufficient. Figure 3(a)-(c) shows an example of the corner detections and affine transformation for an empty (no foreground objects) real and synthetic scene. However, as Figure 3(d) shows the match-mediated difference is not empty in this case because the appearances of the real and the graphical simulated scenes are just too different even enforcing the constraint of zero-difference at the match-points.

In the relaxed scenario, the region boundary information becomes crucial to defining the scene since we can no longer rely on the region interiors to give reliable difference results. Figure 3 (e) and (f) are edge images extracted from the real and synthetic images respectively. If some of the difference in Figure 3(d) coincides with these edges, then it may indicate missing or unexpected objects in the scene. In this example, Figure 3(g) and (h) shown that the only coincidence with the edge images is along the back wall due to a remaining small misalignment – a correct result for two images that show the same scene.

4.1 Relaxed fusion of foreground color

It is important to be able to tell when a simulated, predicted image differs in the way we have explained from the real camera image. Figure 4(a) is a camera view with a single foreground object. Figure 4(b) is a synthetic, texture mapped view which works well with the match-mediated difference based system in Lyons et al. [13]. However, both Figure 4(c) and (d) would present problems for that system.

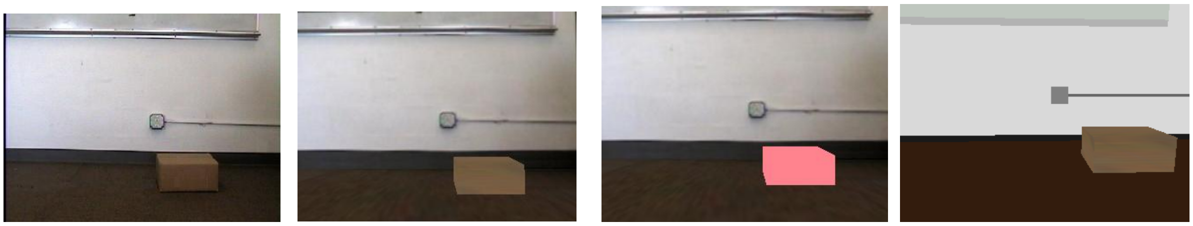


Figure 4: (a) Camera view (b) Synthetic view with textured background and object (c) Synthetic view with textured background and graphical object (d) Synthetic graphical model with textured object.

We develop a modified fusion operation to handle foreground color as follows. Consider comparing the synthetic image I_s in Figure 2(e) with the camera image I_r in Figure 4(a), that is, an empty synthetic scene to a real scene with an (unexpected) box. Let H_{rs} be the affine transformation to align the two images and let I_m be the match-mediated mask image from Lyons & Benjamin [11]. Let I_{cs} *(I_{cr}* respectively*)* be a color feature image. In our current work, for example, it is the sum of the UV channels in the LUV representation of I_s (I_r respectively).

Anding the camera edge image and MMD difference image produces a large area of difference which identifies the foreground. Figure 5(c) shows the edge image of the camera scene in Figure 2(e) (computed here using the Canny edge detector in the OpenCV library). The mask M_{fr} is as

$$M_{fr}(p) = \begin{cases} 1 & p \,\epsilon\, BB_{MMD} \\ 0 & else \end{cases}$$

where $\mathrm{BB}_{\mathrm{MMD}}$ is the bounding box of the difference region. We define the weighted color feature difference as:

$$I_{cd} = \frac{k_c \, M_{fr} \mid I_{cr} - H_{rs} \, I_{cs} \mid}{I_m} \tag{1}$$

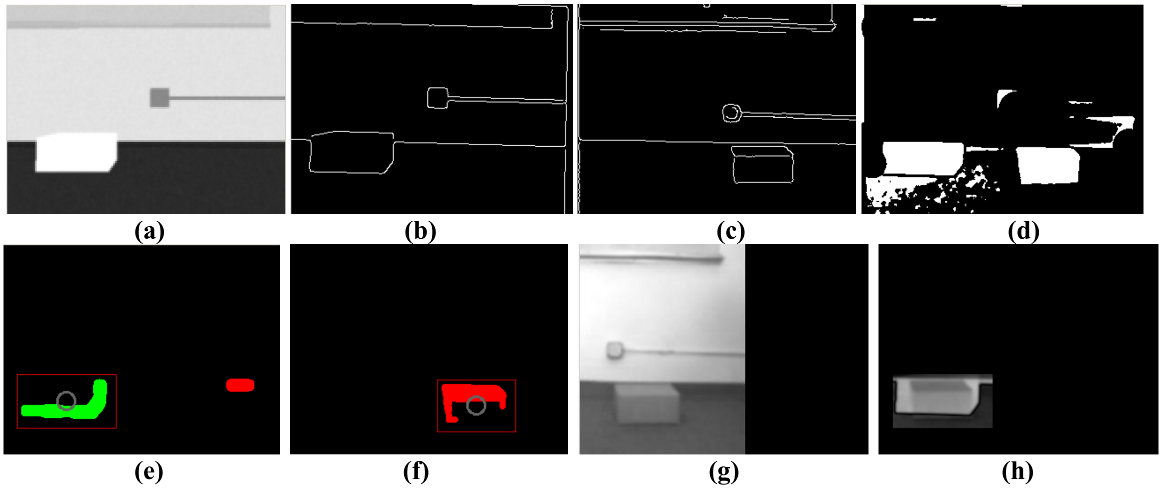
where k_c is the color weight. The absolute difference between the real and warped synthetic image on the top line of eq. (1) is multiplied by the foreground mask M_{fr} and the color weight k_c. The resulting image is divided by the MMD mask I_m (as is the intensity difference image in [11]). In the case where the two foregrounds have similar color as in Figure 4(a) and (b) the absolute difference is very small and result is a small color difference. In the case where the foreground colors are different, then the difference value depends on the color weight. The planning and deliberation modules can select a color weight based on whether color is an important difference feature to note in the simulated scenario.

4.2 Foreground size and shape

In the previous section, the real and synthetic scenes had markedly different visual appearance, but in terms of scene structure and scene content, the only difference was the foreground box in the left of the camera image in Figure 4(a) versus no foreground objects in the synthetic scene of Figure 2(e). If it were the other way around, we would need to use a mask M_{fs} (instead of M_{fr}) to restrict attention to just the synthetic foreground in eq. (1). M_{fs} is produced by anding the MMD difference image with the edge image for the synthetic image (Figure 5(b) shows the edge image for the synthetic scene in Figure 5(a)) and constructing a binary mask from the bounding box of the resulting difference region.

Note that some comparisons between the two scenes only make sense if there are foreground objects in *both* scenes. Figure 5(a) shows a synthetic, graphical scene with a light colored box of similar size and shape to the one in Figure 4(a) but on the opposite side of the image. Let's first consider implementing a visual comparison of the size of the two objects. A direct subtraction would of course produce two difference regions, one on each side of the difference image, as in Figure 5(d) below. We would prefer to have a single overlapped region whose size is related to the size difference between the two foreground objects.

By anding the MMD difference image (Figure 5(d)) with the two edge images (Figure 5(b) and (c)) and filtering for the small regions due to misalignment errors, we can produce the bounding box areas of the two foregrounds (Figure 5(e) and (f)).


(a) **(b)** **(c)** **(d)**

(e) **(f)** **(g)** **(h)**

Figure 5: (a) Synthetic graphical scene with foreground box, (b) Canny edge image of scene (d), (c) Canny edge image of scene in Fig. 4(a), (d) MMD difference image, (e) Synthetic foreground bounding box, (f) Real foreground bounding box, (g) H_{srs} applied to Fig. 4(a), (d) masked size difference.

To compare the two regions directly, we construct a homography H_{srs} that transforms the real image so that its foreground coincides in image position with the synthetic foreground (Figure 5(g)). We can now define the size difference image as:

$$I_{sd} = \frac{k_s \, M_{fs} \; | \; H_{srs} \, I_r - H_{rs} \, I_{cs} \; |}{I_m} \qquad (2)$$

where k_s is the size difference weight. Figure 5(h) shows this image for our running example with $k_s=1$.

We modify this operation to compare the *shape* of the two regions by first normalizing the size of the regions. The four corner points of the bounding box in each image is used to define a homography H_{prs} that maps the size of real image foreground region to the synthetic image foreground region.

$$I_{pd} = \frac{k_p \, M_{fs} \mid H_{prs} \, I_r - H_{rs} \, I_{cs} \mid}{I_m} \quad (3)$$

Where k_p is the shape difference weight. Figure 5(h) shows this image for our running example with $k_s=1$.

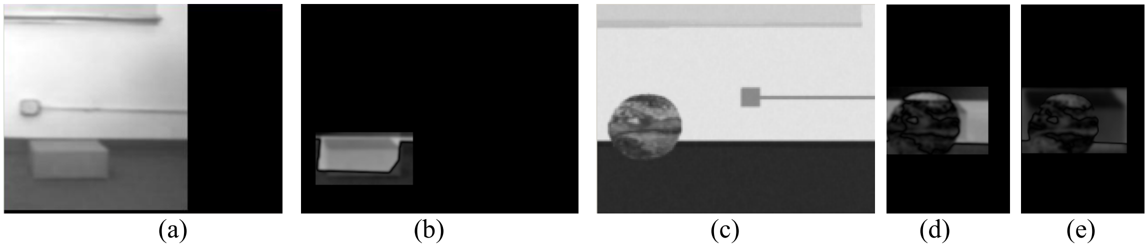


Figure 6: (a) H_{prs} applied to Fig. 4(a), (b) masked shape difference, (c) Ball foreground, (d) masked size difference with Fig. 4(a) $k_s=1$, (e) masked shape difference with Fig. 4(a), $k_p=1$.

5. CONCLUSIONS

We have presented here an extension of the visual architecture developed in Lyons et al. [11] for cognitive robotics. In this extension we can visually compare synthetic camera images generated by a 3D graphical simulation with camera images of real scenes. The comparison operation can be parameterized (by k_s, k_p, k_c) to ignore differences in the color of regions and/or in the shape or size of the foreground objects. This step was necessary to allow the comparison of simulation results with camera images of previously unseen areas or objects, to bring the architecture into line with evidence from cognitive psychology and neuroscience, and to make the process of simulation cheaper and faster.

Polyscheme/Polybot [6] is a framework for linking multiple specialists and representations into a single cognitive architecture. This is similar to our concept of including the 3D simulation as a separate and very different 'specialist' in our architecture. However, Polyscheme focuses on integrating the specialists to make inferences about a situation. All of the Polyscheme specialists need to be interfaced to the general framework; they need to agree on a 'lingua franca' or common language. The choice of such a language may impose unnatural constraints on the way specialists can communicate. We take a different approach to integration, inspired by cognitive modelling when we integrate the simulation and robot control at the level of the semantics of the natural visual image.

In the CiceRobot architecture, Macaluso and Chella [15] follow a similar approach to ours. They compare synthetic images from a 3D simulator to real camera images to localize the robot in a building. Although the general outline of CiceRobot is quite similar to the visual apparatus of ADAPT, it differs dramatically at the detailed level. Since the building is instrumented with carefully placed landmark wall-markings, the CiceRobot image comparison operation is simply the location and comparison of the landmarks in the synthetic and real images. In our case, there are no artificial landmarks, and image pair alignment and comparison is carried out by extracting 'naturally occurring' corners and edges from both scenes.

The next steps in our work including integrating this expanded image difference operation into the target tracking and prediction system described in Lyons et al. [13] as well as looking to some of ways in which this can be made more efficient and general. For example, the bounding-box approach to the foreground masks is coarse, since it includes some of the background scene around the object, an arbitrarily shaped pixel mask would be more accurate. A set of convex regions, which minimally cover the foreground regions, would need to be generated from the difference image to build the pixel mask. In terms of generality, we have considered here the case of a single foreground region in both synthetic and real images. This is sufficient for the rolling target application. Multiple foreground masks will require going beyond the use of an edge image to filter the difference image. An image segmentation will need to be used in that case. This raises a data association problem between the foreground regions in the synthetic and real images. One approach is to generate multiple associations (multiple images) as in the CiceRobot localization framework [15] and employ a particle filter.

References

[1] Arbib, M.A., "The Handbook of Brain Theory and Neural Networks*." (Ed. M.A. Arbib) MIT Press* (2003).

[2] Barsalou, L.W., "Simulation, situated conceptualization and prediction." *Phil. Tran. R. Soc. B*(2009) 364, 1281—1289.

[3] Benjamin, D.P., "Reformulating Theories of Action for Efficient Planning." in *Theories of Action, Planning and Robot Control: Bridging the Gap*, Chitta Baral (ed.), AAAI Press (1996).

[4] Benjamin, D.P., Lyons, D.,Lonsdale, D., "ADAPT: A Cognitive Architecture for Robotics**."** *2004 Int. Conf. on Cognitive Modeling,* Pittsburgh PA July (2004).

[5] 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).

[6] 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).

[7] 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).

[8] Coradeschi, S., Saffiotti, A., "Perceptual Anchoring of Symbols for Action." *Int. Joint. Conf. on AI*, Seattle WA (2001).

[9] Gärdenfors, P., "Emulators as a source of hidden cognitive variables." *Behavioral and Brain Sciences* 27(3): 403, (2004).

[10] Laird, J., Newell, A., Rosenbloom, P., "Soar: An Architecture for General Intelligence." *Artificial Intelligence* **33** (1987).

[11] Lyons, D.M., and Benjamin, D.P., "Locating and Tracking Objects by Efficient Comparison of Real and Predicted Synthetic Video Imagery." *SPIE Conf. on Intelligent Robots and Computer Vision*, San Jose CA, Jan. (2009).

[12] Lyons, D., and Arkin, R.C., "Towards Performance Guarantees for Emergent Behavior." *IEEE Int. Conf. on Robotics and Automation*, New Orleans LA, April (2004).

[13] 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 and Security Symposium*, Orlando (Kissimmee), FL, April (2010)

[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] Mantz, F., Pieter Jonker, "Behavior Based Perception for Soccer Robots." in: Goro Obinata, Ashish Dutta, Nagoya University (Eds),*Vision Systems* Advanced Robotic Systems, Vienna, Austria, April (2007).

[17] de la Puente, P., Rodriguez-Losada, D., Valero A., and Matia, F., "3D Feature Based Mapping Towards Mobile Robots Enhanced Performance in Rescue Missions." *IEEE/RSJ Int. Conf. on Int. Robots & Systems*, St. Louis USA, (2009)

[18] Pezzulo, G., et al., "The mechanics of embodiment: a dialog on embodiment and computational modelling." *Frontiers in Psychology*, V2 A5, January (2011).

[19] Shanahan, M.P., "A Cognitive Architecture that Combines Internal Simulation with a Global Workspace." *Consciousness and Cognition*, vol. 15, pages 433-449, (2006).

[20] Xiao, J., Zhang, L., "A Geometric Simulator SimRep for Testing the Replanning Approach toward Assembly Motions in the Presence of Uncertainties." *IEEE Int. Symp. Assembly and Task Planning*, (1995).