Progress in Building a Cognitive Vision System

D. Paul Benjamin

Pace University, 1 Pace Plaza, New York, New York 10038, 212-346-1012
benjamin@pace.edu

Damian Lyons

Fordham University, 340 JMH, 441 E. Fordham Rd., Bronx, NY 10458, 718-817-4485
dlyons@fordham.edu

Hong Yue

Pace University, 1 Pace Plaza, New York, New York 10038
yh19243n@pace.edu

ABSTRACT

We are building a cognitive vision system for mobile robots that works in a manner similar to the human vision system, using saccadic, vergence and pursuit movements to extract information from visual input. At each fixation, the system builds a 3D model of a small region, combining information about distance, shape, texture and motion to create a local dynamic spatial model. These local 3D models are composed to create an overall 3D model of the robot and its environment. This approach turns the computer vision problem into a search problem whose goal is the acquisition of sufficient spatial understanding for the robot to succeed at its tasks.

The research hypothesis of this work is that the movements of the robot's cameras are only those that are necessary to build a sufficiently accurate world model for the robot's current goals. For example, if the goal is to navigate through a room, the model needs to contain any obstacles that would be encountered, giving their approximate positions and sizes. Other information does not need to be rendered into the virtual world, so this approach trades model accuracy for speed.

Keywords: machine vision, virtual world, 3D model

1. Introduction

Robots are needed that can help people perform tasks in settings that are dangerous or repetitive and dull, in homes, in hospitals and in factories. This requires sophisticated human-robot interaction in which the robot can comprehend and predict human motions, and plan responses that are cooperative and avoid harm.

Our system uses a highly accurate physically realistic simulator coupled with stereo vision for 3D modeling and navigation that enables a robot to model and respond to human movements. The goal is to produce a system that will be fast and will use relatively inexpensive equipment [Benjamin 2012, 2010, 2008].

The main research objectives of our work are to:

1. Demonstrate the efficient use of stereo vision to recognize objects, people, and motions, and

render them accurately into the simulator,

2. Demonstrate the use of a 3D simulator with realistic physics to navigate.

A primary goal is to demonstrate smooth interaction and cooperation with humans and other robots in a range of settings. The settings to be used include navigating alongside a walking person, navigating through walking people, and crossing a busy street while avoiding moving vehicles.

The usual approach to the use of vision in robotics is to attempt to solve two problems [Barnes1997]:

(a) Process visual data to extract all the objects and motions in the environment,

(b) Identify the results from (a) that are important and relevant to the current task.

Unfortunately, both of these steps are very expensive computationally. The first step requires processing an enormous amount of visual data, especially when the environment is very dynamic. The second step is a difficult data mining problem.

A-priori information includes generic 3D models for walls, buildings, and common outdoor and indoor objects as well as any specific 3D map and object information for the scene. By modifying the a-priori data, the system can be quickly adapted to a wide range of scenes and situations. Our approach to this complexity issue is to leverage goal-directed rendering: the robot first decides which aspects of its environment are relevant, based on its goals. This information is used to focus the cameras on specific regions of the environment and extract only the information needed for the goals. This is important because it has the potential to be faster and less expensive than current approaches. Our current system runs on a laptop in real time. This is an original approach that is potentially transformative, because it creates the possibility of a small, inexpensive 3D modeling system that can be ubiquitous, e.g., it can be on wheelchairs, and it can be placed in every corridor of a hospital or a factory.

2. 3D Mental Models

Computer vision has had a difficult time reproducing the human ability to understand visual scene information across a wide range of applications domains and environmental conditions. There is evidence from cognitive psychology [Oliva & Torralba 2008] that effectively leveraging context is a key aspect of this human facility. However, while there has a strong bottom-up Marr-based stream of vision research [Marr 1982], the use of context has also been recognized in computer vision for a long time: at least from the Univ. of Mass. VISIONS project [Hanson & Riseman 1978] and more recently to the linguistic-inspired Bag of Words approaches (e.g., [Csurka et al. 2004]) global extensions of scale-invariant features (e.g., [Mortensen et al. 2005]) and others [Marques et al 2011]. But in general these approaches still view scene recognition as a ‘recognize the snapshot’ problem, with little input from ongoing, long term objectives and tasks of the system. The scene understanding problem for a human is one of an embedded system leveraging sensing to fulfill its goals: sensing is strongly biased in the service of task and how the agent’s and other agents’ actions are expected to play out in the physical world.

Recent evidence in cognitive psychology [Shanahan 2006] and neuroscience [Pezzulo 2011] supports the proposition that simulation, the “re-enactment of perceptual, motor and introspective states” is a central cognitive mechanism that helps to provide context for planning. Shanahan [Shanahan 2006] 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 [Barsalou 2009] proposes that distributed structures in the brain's feature and association areas learn to recognize categories of experience. He proposes that these simulators can recreate small subsets of their content in what he refers to as "situated conceptualizations", which are embodiments 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 [Pezzulo 2011]. 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. [Cassimatis 2004], 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 [Chella 2007][Macaluso 2007] base their cognitive robot architecture CiceRobot on the concept of emulators as developed by Gärdenfors [Gardonfors 2004]. They 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 [Pezzulo 2011] argues that the evidence in favor 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 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 [Xiao 1995] 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 [Benjamin 1996].

There has been work on extending cognitive models to interact with user interfaces, with the goal of automatically testing and improving user interface design [Ritter 2007, St. Amant 2005]. Like our work, this work is based on a model of human visual processing and is incorporated into a unified cognitive architecture. The cognitive architecture can see the computer screen and

interact with the software similar to the manner of humans. This permits evaluation of the ease of interaction, and of the time required to accomplish tasks through the interface. This work is valuable to us as a prototype; however, this work is limited only to user interfaces and is not designed to be applied to more general environments, such as computer vision in a mobile robot. In particular, there is no virtual world.

The Soar cognitive architecture has been extended twice with visual mechanisms to give Soar the ability to control robots [Laird 2009]. The first is Soar/SVI [Lathrop 2009], which gives Soar the ability to create and reason about spatial representations and abstractions (imagery). The second is Soar/SVS [Wintermute 2010, Wintermute 2011], which adds the ability to simulate the effects of actions in the environment. This body of work is the most closely related work to ours.

Soar/SVI and Soar/SVS do provide the ability for Soar to reason about spatial predicates, which are created in a goal-independent way from the raw visual input and placed in Soar's working memory. This permits Soar to use spatial information in its task planning. However, the visual memory is not a full 3D virtual world, and their research does not examine how goal-directed inference interacts with perception. Wintermute states: "Theoretically, memories and processes inside SVS, with influence from symbolic processing in Soar, should segment and recognize objects and estimate 3D spatial structure based on 2D visual information. As we do not address the veridical perception problem, the system does not attempt this." [Wintermute 2010, p.42] The perception problem is precisely what we are addressing in our research, and we are investigating issues such as how knowledge about a situation affects the accuracy of stereo disparity and object tracking. In this respect, our work is complementary to Soar/SVI/SVS.

There is also much work on scene classification [Borji 2014] but this is not really similar to our work. That work classifies individual images according to content, e.g. "living room scenes" versus "swimming pool scenes". But we are not interested in just assigning a class label to an image; we want to comprehend the behaviors in a sequence of images (video) so that we can understand what might happen next.

There is considerable work in the psychological literature on focus of attention in humans [Riche 2013] and we have found it a source of inspiration. It presents a range of models of saliency based on large amounts of data. However, that work does not attempt to connect their models to task representation and problem solving, as we are doing.

3. System Architecture

3.1 Motivation

Our vision system architecture is directly inspired by the cognitive and neurobiological structure of the human vision system, and the goal of our work is to develop an appropriate set of abstractions for a computational implementation of the human vision system and measure their effectiveness.

The human vision system does not apply equal computational resources everywhere in its visual field, but instead focuses on and analyzes just a small portion of the visual field at each moment; this is called a *fixation* [Rayner 1995]. After extracting the needed information from that region of the visual field, the vision system rapidly moves the eyes to a new region of the visual field for the next fixation. These rapid movements are *saccades*, which are quick movements across the visual field, and *vergences*, which change the depth of focus [Rayner 1995]. The effect of this organizational structure is to permit efficient use of limited

computational resources. Instead of fully processing all of the sensory input and then discarding everything that is not relevant to the goals, this organization applies computational resources only to the parts of the sensory input that are likely to be relevant to the agent's goals. The key is to organize the search of the visual field in a manner that effectively gathers useful information.

Much work has been done on measuring the functioning of the human vision system and of its system of saccades and vergences [Rayner 1995], but there has not been a computational implementation that connects the actions of the vision system to the goals of the agent.

Our research hypothesis is that the movements of the vision system are those that are necessary to build a sufficiently accurate 3D world model for the robot's current goals. For example, if the goal is to navigate through a room, the world model needs to contain any obstacles that would be encountered, giving their approximate positions and sizes. The vision system needs to search for this information; other information does not need to be rendered into the virtual world.

In this way, our system prunes the information at the perception stage, using its knowledge about the agent's goals and about objects in the world and their dynamics to decide where to look and what type of information it is looking for. This is in contrast to the usual approach of gathering lots of sensory information, processing it all and rendering it into a world model in a goal-independent manner, then deciding which information is necessary for decision making. This latter approach wastes a great deal of processing time processing information that is discarded in the decision making process. We are designing a fast, inexpensive vision system by emulating the organization of the human vision system.

3.2 Implementation Overview

Typical robot vision systems connect their cameras directly to their world models, so that sensory data is processed and modeled in a fixed way directly in the world model. Reasoning is then performed on the world model to extract meaning about the scene. This type of architecture treats perception as a separate process from reasoning, and typically the implementation reflects this, e.g. a computer vision module processes the vision data and puts symbolic representations of the recognized objects and their relationships in the world model, and the reasoning engine manipulates these symbols to plan and learn. The reasoning engine does not alter the representation of the visual data.

In contrast, our virtual world is not connected directly to visual input. Sensory data is placed directly in the working memory of the reasoning engine (the Soar cognitive architecture) after some low-level processing; the reasoning engine's principal task is to reason about how to model and interpret the data. It does this by repeating the following five steps:

> It senses in the virtual world, using the same position and orientation as in the real world, and using the same sensors. It grabs graphics input from PhysX, and if it is also modeling range data, it grabs distance data from PhysX in the directions of the actual range sensors.
>
> It compares the virtual sensory data with real sensory data using the MMD (Match-Mediated Difference) [Lyons 2009][Lyons 2009b], which uses a least-squares measure to find the degree of disagreement. It aligns the real and virtual images with an affine map, then finds a set of matched key points and places a normalized Gaussian at each of them to detect differences. The regions of greatest difference are placed in working memory.

A region of difference is selected by the reasoning engine according to the robot's goals and the degree of disagreement. The vision system saccades to that region and fixates.

Stereo disparity, color segmentation, and optical flow are computed only in the small region of focus. Restricting the computation to this small region permits the use of highly accurate but computationally expensive algorithms for these computations, e.g. disparity of disparities.

The reasoning engine analyzes the information from the fixations and determines the needed adjustment to the virtual world to reduce the disagreement. If an object has disappeared from the real world, it is removed from PhysX. If a new object has appeared, the information from the fixation is input to the object recognition database, and a mesh model of the best match is rendered into the virtual world. If an object's motion has changed, its process model is updated.

In this way, perception becomes a problem-solving process; the system's goal is to build a 3D representation that is useful for the given task, and this goal determines the focus of attention. This enables all the knowledge of the system to be brought to bear on perception. The result is a working physical model of the observed scene. PhysX can run this virtual world faster than real time to predict the consequences of various actions and evaluate them. In this way, it implements simulation-based perception and control.

Figure 1 shows the block diagram of the overall system concentrating on the lower level perceptual and simulation system. The core functionality of the system is in the four shaded boxes. The camera module generates an image of the environment as seen by the robot at pose P_r. The simulation module generates a synthetic view of the environment as predicted by the simulation with the robot at pose P_S. Both images are fed to the match-mediated difference module, which calculates the affine transform from one to the other producing an 'error' measure H_e and a match-mediated difference (MMD) image. The error information is used to modify the pose of the simulation and improve the localization of the robot with respect to the simulation. The difference mask is used to determine what scene elements are not where they were expected (new objects, missing objects or misplaced objects).

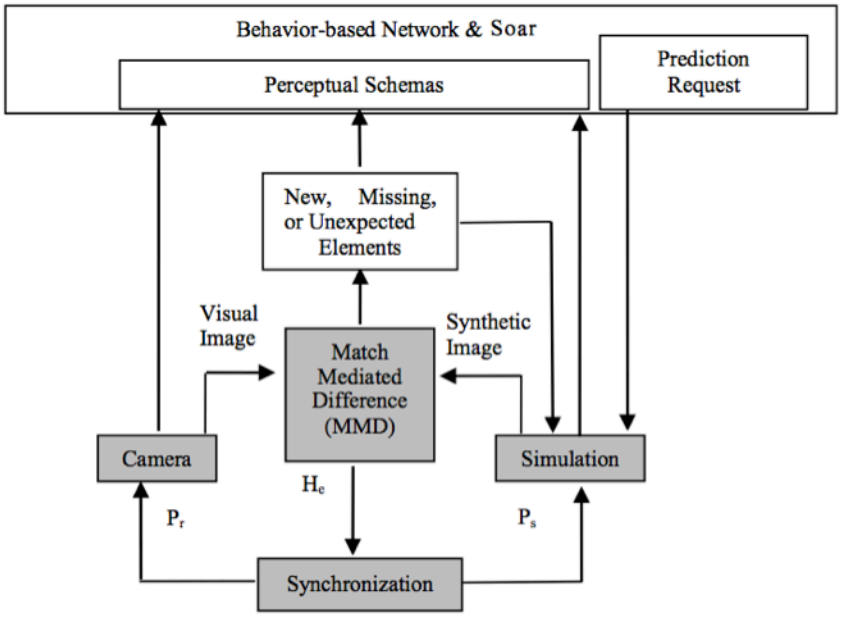

Figure 1: System Block Diagram

The four shaded boxes comprise the autonomous core of the system and function in two modes:

1. Synchronization Mode: In synchronization mode, the system continually compares one image from the camera module with one image from the simulation module generating an error homography H_e and an MMD image. The error homography is used to calculate a new simulation pose P_S. The difference image is used to identify new, unexpectedly placed or missing scene elements. This information is used to add or remove simulation elements to improve the visual correspondence of the simulation with the environment. For example, an unexpected object will appear as a difference region. In response, an object will be introduced into the simulation at this position. The texture information from the difference region is then used to texture the surface of the object. The simulation is constrained to step through time at the same rate as the robot ("real-time").

2. Prediction Mode: In prediction mode, the state of the simulation is first stored, then the simulation is allowed to step through time at a faster rate than the robot (super "real-time") for a specific duration. The synchronization with the camera module is disabled for the duration. At the end of the duration, the camera image is compared with the information from the camera module to extract predicted scene element locations. The current state of the simulation is then restored.

The interface between the behavior-based and Soar levels of the system is implemented using the concept of a Perceptual Schema [Arbib 2003], acting as a visual marker [Wasson, Kortenkamp & Huber 1999] or anchor [Coradeschi & Saffioti 2001] that ties a scene element or elements to perceptual concepts that play a role in behaviors. Perceptual schema markers can be placed on real-scene elements and then track these elements, reporting information to the behaviors in which they play a role. They can also be placed on corresponding simulated scene elements, allowing for the easy extraction of predicted object behaviors when operating in prediction mode. A perceptual schema could include active motion of the sensors, and even motion of the robot itself: a visual routine in the sense of Ullman [Ullman 1984].

For additional detail as well as experimental results about the MMD module and Synchronization module, see [Lyons 2009, 2010, 2011, 2012].

3.3 Current Status of the Vision System

The system currently can model static environments and environments with simple dynamics, such as a bouncing ball. These are modeled in real time with the entire system running on a laptop, and with sufficient accuracy that a Pioneer robot can navigate through the world. The simplicity of the static world results in a very simple visual search strategy: the vision system saccades to the largest difference that is in front of the robot.

The MMD is robust to variations in lighting and color and level of detail between the real and virtual worlds. The types of objects that can be recognized and rendered are limited to a few simple chairs and tables, balls, and the Pioneer robot; these are hand-coded in a small library.

A number of videos showing the operation of the system are available [PaceVids].

4. Current Work

We are focusing on two main topics at this point:

1 - The speed must be substantially increased. As the environments become more dynamic and contain more objects, the computational cost of 3D modeling becomes prohibitive.

2 - Currently the system uses a small hand-coded library of objects that it can recognize. This must be replaced with the ability to use one of the large online libraries of 3D models.

4.1 Increasing the Speed

The speed of the system is a fundamental consideration. Many approaches to 3D modeling work well in theory and on small examples, but are too expensive computationally on real problems. In our current system, we have encountered two main bottlenecks: the MMD and the number of degrees of freedom resulting from too many moving objects.

We are addressing the first bottleneck by implementing the MMD on the laptop's GPU. The current implementation is CPU-based, and can yield at most four difference calculations per second. The MMD calculations are based on the computation of local Gaussians, and we are confident that the MMD calculations can be parallelized to a large extent. Even an ordinary laptop usually has a GPU with hundreds of cores, and we expect a GPU-based implementation to eliminate this bottleneck. Our goal is for the MMD calculations to be a negligible fraction of the total computational time; the MMD must be run at least ten times per second, so we are aiming for an MMD that runs in under 10 milliseconds. We are using CUDA, which is Nvidia supported and interfaces easily with PhysX, which is also from Nvidia.

The second bottleneck arises from the fact that the system currently must visualize every situation, even if it is very similar to ones it has seen before. For example, suppose a robot bounces a ball off a wall and it bounces back to the robot. Every time this is done the robot must visualize the path of the ball and predict that it will bounce back. This is unnecessary. Furthermore, in more complex environments it can be the case that parts of the environment behave in ways the robot has seen before; re-visualizing these behaviors can waste time that is needed to analyze other parts of the environment. It is desirable for the robot to learn from experience, so that it can know the results of an action without having to visualize and deliberate every time. This is the approach we will take to handling the growth of complexity as the

environments become more dynamic.

This is one primary motivation for using Soar as the reasoning engine. Soar possesses an *episodic memory* [Gorski & Laird 2011], which contains a complete history of the experience of the agent. These episodes can be triggered for storage and later retrieved by *cues* that exist in working memory. Each cue is a subset of working memory that can match one or more episodes and cause them to be retrieved from long-term memory. Once an episode is retrieved, the agent can access temporally related episodes that occurred before and after it. This permits the agent to bypass 3D model construction and visualization, and quickly "step through" the results of the previously seen sequence of episodes. In this way, the system can shift over time from deliberation to reaction.

Effective implementation of an episodic memory for 3D behaviors requires identifying a set of important working memory structures that correspond to features of meaningful episodes, so that Soar will know which episodes to store, and will retrieve them at appropriate times. These features could include sudden changes in direction, velocity or acceleration of objects in the environment (especially those resulting from collisions), as well as the appearance and disappearance of objects. In addition, these features are likely to include relevant goals, such as wanting to determine the future position of an object or wanting to choose a direction to move. This part of our work consists of measuring the effect of a variety of such features on the reduction of time needed to classify observed behaviors and on the accuracy of the classification, and designing an effective cue system. If the cues are too general, then too many episodes will be retrieved, and too often, resulting in a slowup of the system. If the cues are too specific, then too few episodes will be retrieved to produce much speedup. One promising idea is learning which cues to use for retrieval [Gorski & Laird 2011].

4.2 Object Recognition

The current object recognition library is small, containing fewer than a dozen objects. Each object is represented by a set of images from eight angles, together with a mesh model. Objects are recognized using a Haar classifier, and the corresponding mesh is rendered into the virtual world with the appropriate pose. This was sufficient for our initial work, but we need to enable our vision system to recognize a large variety of objects without a lot of hand coding on our part.

We will implement a new method based on the approach of Lai and Fox [Lai & Fox 2010]. Their object detection system is based on matching 3D point clouds against Google's 3D Warehouse. Their algorithm minimizes simultaneously the classification distances of both the virtual representations of objects and their real representations. This permits their algorithm to use a large labeled database of virtual 3D objects for identification in the real world, and the database is public, open source, and continually growing.

Their code is slow (80 seconds to classify an entire outdoor scene) but their approach is "highly parallelizable" via GPU implementation [Lai & Fox 2010]. In addition, our task is made much easier as we are using video rather than separate individual scenes as they do; we do not need to start from scratch for each scene. We already know the ground and most of the scene and can focus on the small region where a difference has been detected to identify the object in that region of interest; this computation does not even need to be done when there are no new objects. This alone should speed the computation by a large amount.

Once an object is recognized, an associated mesh will be retrieved, if it exists, and rendered into the virtual world. If there is no previous mesh for this object, the point cloud will be

smoothed and filled in and turned into polygonal form as described in [Rusu 2008] then rendered as PhysX mesh and added to the library of mesh objects.

5. Summary

A wide range of computer vision techniques have been developed over the past several years to recognize objects and people, to classify their motions, and for robot localization and mapping, with the goal of creating robots that can interact quickly and safely with people. This is a goal with tremendous societal impact, as it leads to a wide variety of applications that can affect peoples' lives in many ways, including their jobs, their healthcare, their transportation, and their security. These computer vision techniques work well in simple environments containing few moving objects, but their performance degrades rapidly as the environments become more realistic and dynamic. A new approach is needed to find methods that can scale with the complexity of the real world.

Our approach is based on research in cognitive psychology that indicates that the use of 3D models in spatial reasoning is fundamental, occurring even in people who have been blind since birth [Ungar 2000]. Our design for a vision system emulates the human vision system, and explores the connections between the measurable searching motions of the human vision system - saccades and fixations - and the goal of maintaining an accurate 3D model of the environment.

REFERENCES

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

[Barnes 1997] Embodied computer vision for mobile robots, N. Barnes and Zhi-Qiang Liu, Intelligent Processing Systems, 1997. ICIPS '97. 1997 IEEE International Conference on (Volume:2) pp. 1395 – 1399, 1997, DOI: 10.1109/ICIPS.1997.669238

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

[Benjamin 2012] D. Paul Benjamin, Damian Lyons, John V. Monaco, Yixia Lin, and Christopher Funk, "Using a Virtual World for Robot Planning", SPIE Conference on Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, April 2012. http://csis.pace.edu/robotlab/pubs/SPIE2012.pdf

[Benjamin 2010] D. Paul Benjamin and Damian M. Lyons, "A Cognitive Approach to Classifying Perceived Behaviors", SPIE Conference on Multisensor, Multisource Information Fusion, 2010. http://csis.pace.edu/robotlab/pubs/SPIE2010.pdf

[Benjamin 2008] D. Paul Benjamin, Deryle Lonsdale, Damian Lyons and Siddtharth Patel, "Using Cognitive Semantics to Integrate Perception and Motion in a Behavior-Based Robot", Proceedings of the 2008 ECSIS Symposium on Learning and Adaptive Behavior in Robotic Systems (LAB-RS 2008), IEEE Computer Society, August 6-8, 2008, Edinburgh, Scotland. http://csis.pace.edu/robotlab/pubs/LABRS2008.pdf

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

[Borji 2014] Ali Borji and Laurent Itti: "Human vs. Computer in Scene and Object Recognition", CVPR 2014: 113-120.

[Cassimatis 2004] Cassimatis, N., Trafton, J., Bugajska, M., Schulz, A., "Integrating cognition, perception and action through mental simulation in robots," Robotics and Autonomous Systems **49**, pp.13-23 (2004).

[Chella 2007] Antonio Chella, Marilia Liotta, Irene Macaluso, "CiceRobot: a cognitive robot for interactive museum tours," Industrial Robot: An International Journal, **34** No. 6, pp.503 – 511, (2007).

[Coradeschi & Saffioti 2001] Coradeschi, S., Saffiotti, A., (2001), "Perceptual Anchoring of Symbols for Action" Int. Joint. Conf. on AI, Aug 4-10, Seattle WA.

[Csurka et al. 2004] Csurka et al. (2004) "Visual categorization with bags of keypoints." Workshop on statistical learning in computer vision, ECCV. Vol. 1.

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

[Gorski & Laird 2011] Nicholas A. Gorski, John E. Laird, "Learning to use episodic memory", Cognitive Systems Research, **12**, Issue 2, June 2011, Pages 144-153, ISSN 1389-0417, http://dx.doi.org/10.1016/j.cogsys.2010.08.001.

[Hanson & Riseman 1978] Hanson, A. and E. Riseman. (1978b). "VISIONS: A computer System for Interpreting Scenes" in Computer Vision Systems (A. Hanson and E. Riseman, Ed.), New York: Academic Press.

[Lai & Fox 2010] Kevin Lai and Dieter Fox, "Object Detection in 3D Point Clouds Using Web Data and Domain Adaptation", International Journal of Robotics Research, **29**, 8, pp.1019-1037, 2010.

[Laird 2009] John E. Laird, "Toward Cognitive Robotics", Proc. SPIE 7332, Unmanned Systems Technology XI, 2009. http://dx.doi.org/10.1117/12.818701

[Lathrop 2009] Lathrop. S. D. and Laird, J. E. 2009. "Extending Cognitive Architectures with Mental Imagery". In Proceedings of the 2nd Conference on Artificial General Intelligence.

[Lyons 2012] Damian Lyons, P. Nirmal and D. Paul Benjamin, "Navigation of Uncertain Terrain by Fusion of Information from Real and Synthetic Imagery", SPIE Conference on Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, April 2012. http://csis.pace.edu/robotlab/pubs/LyonsNirmalBenjamin2012.pdf

[Lyons 2011] "A Relaxed Fusion of Information from Real and Synthetic images to Predict Complex Behavior", Damian Lyons and D. Paul Benjamin, SPIE Conference on Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, April 2011. http://csis.pace.edu/robotlab/pubs/LyonsBenjamin2011.pdf

[Lyons 2010] Damian 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)

[Lyons 2009] Damian M. Lyons and D. Paul Benjamin, "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). http://csis.pace.edu/robotlab/pubs/SPIE_IRCV2009.pdf

[Lyons 2009b] Damian M. Lyons and D. Paul Benjamin, "Robot Video Tracking by Comparing Real and Simulated Video Scenes", Conference on Intelligent Robots and Computer Vision, SPIE, San Jose, CA, January 2009.

[Macaluso 2007] Macaluso, I., and Chella, A., "Machine Consciousness in CeceRobot, a Museum Guide

Robot," Proceedings, AAAI Fall 2007 Symposium, Arlington VA, (2007).

[Marques et al. 2011] Marques, O., Barenholtz, E., and Charvillat, V. (2011) "Context modelling in computer vision: techniques, implications and applications", Multimedia Tools and Applications **51**:303- 339.

[Marr 1982] Marr David, (1982) Vision. W. H. Freeman, San Francisco.

[Mortensen et al. 2005] Mortensen, E., Deng, H., Shapiro, L., (2005) "A SIFT Descriptor with Global Context" Int. Conf. on Computer Vision and Pattern Recognition (CVPR).

[Oliva & Torralba 2008] Oliva, A., and Torralba, A., (2008) "The role of context in object recognition", TRENDS in Cognitive Sciences, **11**, No. 12.

[PaceVids] Pace University Robotics Lab Videos, http://csis.pace.edu/robotlab/videos.html

[Pezzulo 2011] Pezzulo, G., et al., "The mechanics of embodiment: a dialog on embodiment and computational modeling," Frontiers in Psychology, **2,** A5, January (2011).

[Rayner 1995] Rayner, K., "Eye movements and cognitive processes in reading, visual search, and scene perception", In J. M. Findlay, R. Walker, & R. W. Kentridge (Eds.), Eye Movement Research: Mechanisms, Processes, and Applications (pp. 3-21). New York: Elsevier, 1995.

[Riche 2013] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit, "Saliency and Human Fixations: State-of-the-Art and Study of Comparison Metrics", in Proc. ICCV, 2013, pp.1153-1160.

[Ritter 2007] Ritter, F. E., Kukreja, U., & St. Amant, R. (2007). Including a model of visual processing with a cognitive architecture to model a simple teleoperation task. Journal of Cognitive Engineering and Decision Making, **1**(2), 121-147.

[Rusu 2008] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. "Towards 3D Point cloud based object maps for household environments", *Robot. Auton. Syst.* **56**, 11 (November 2008), 927-941. DOI=http://dx.doi.org/10.1016/j.robot.2008.08.005

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

[St.Amant 2005] St. Amant, R., Riedl, M. O., Ritter, F. E., & Reifers, A. (2005). Image processing in cognitive models with SegMan. In Proceedings of HCI International, 2005. (Invited.) Volume 4 - Theories Models and Processes in HCI. Paper # 1869.

[Ullman 1984] Ullman, S. (1984) Visual routines. Cognition **18**:97-159.

[Ungar 2000] Ungar, S., Cognitive mapping without visual experience. In Kitchin, R. & Freundschuh, S. (eds), Cognitive Mapping: Past Present and Future, London: Routledge.

[Wasson, Kortenkamp & Huber 1999] G. Wasson, D. Kortenkamp, and E. Huber, (1999) "Integrating Active Perception with an Autonomous Robot Architecture" Journal of Robotics and Autonomous Systems **29**: 175-186

[Wintermute 2010] Wintermute, S. (2010). Abstraction, Imagery, and Control in Cognitive Architecture. PhD Thesis, University of Michigan, Ann Arbor.

[Wintermute 2011] Wintermute, S.: Imagery in Cognitive Architecture: Representation and Control at Multiple Levels of Abstraction, Cognitive Systems Research, **19-20**,1-29, 2011.

[Xiao 1995] 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).