Wreath Product Cognitive Architecture (WPCA)

Anshul Joshi[1] and Thomas C. Henderson[2]

***Abstract*— A Belief-Desire-Intention (BDI) framework closely resembles human practical reasoning approach in day-to-day life, and is a well-studied architecture. The wreath product cognitive model, first described by Leyton is an abstract, although powerful, model which closely couples perception and actuation for representing shape. However, no implementation of the wreath product model exists. Our work is an attempt to combine the wreath product knowledge representation mechanism with a BDI architecture that works in a real-world setting. A prototype implementation of this combination is demonstrated on an *iRobot Create* differential-drive robot, with a *Kinect One* structural sensor, in an indoor environment. The effectiveness of our framework is demonstrated by its accuracy for mapping the environment and localization of the robot for navigation purposes.**

I. INTRODUCTION

We have examined a representation which features combined action and perception signals; i.e., instead of having a purely geometric representation of the perceptual data, we include the motor actions, for example, aiming a camera at an object, which produces actions that generate the particular shape. This generative perception-action representation uses Leyton's cognitive representation based on wreath products [27]. The wreath product is a special kind of group which captures information through symmetries on the sensorimotor data. The key insight is the bundling of actuation and perception data together in order to capture the cognitive structure of interactions with the world. This involves developing algorithms and methods: (1) to perform symmetry detection and parsing, (2) to represent and characterize uncertainties in the data and representations, and (3) to provide an overall cognitive architecture for a robot agent. We have previously demonstrated these functions in 2D text classification [21], and in this work we show it on 3D spatial data acquired by a real robot operating in an indoor environment, that uses this cognitive architecture, and maps the environment and localizes itself in that environment. The cognitive architecture called the *Wreath Product Cognitive Architecture* is developed to support this approach.

We have previously proposed innate theories of symmetry as the cognitive basis for embodied robot agents [19], [20] and more recently, a specific cognitive architecture based on *Bayesian Symmetry Networks* [18], [23]. This representation builds on the framework layed out by Leyton [26], [27] wherein he proposes that the wreath product captures the notion of a specific concept which is a representation of what something is or how it works; this may capture either a specific instance of an existing thing or an abstract description of a class of related objects. For Leyton, the wreath product provides the basis for concept representation, where a wreath product is a group formed by a splitting extension of the direct product of the fiber group which is acted on by a control group (usually a permutation group) and is derived from related perception and actuation. The distinctive feature of his representation is that it is based on how the set of features comprising the object to be represented is generated – it is a generative theory of shape. Thus, the actuation control sequences are part of the description of an object and determine the control group hierarchy. This is important because objects are expressed in terms of the specific embodiment of the robot agent perceiving them. Our contributions in this regard are as follows: (1) We implement a powerful representation - the wreath product representation - which works practically, and for which no implementation exists yet, (2) we demonstrate the effectiveness of this representation for an absolutely essential, yet non-trivial, mobile robot functionality – localization in an indoor environment – using wreath products as landmarks.

II. RELATED WORK

The first problem to be addressed for robot autonomy is that of building a cognitive framework. In recent years robotics researchers have understood the importance of developing cognitive abilities of robots, rather than explicitly programming the robots with the knowledge and algorithms to process that knowledge for achieving results, and a lot of research has been devoted to achieve this. For example, Beeson [2] has explored using *cognitive maps* as analogous to human spatial mapping process using the *Hybrid Semantic Spatial Hierarchy*. Desai et al. [7], [8], [9], [10] have used affine feature descriptors for the purpose of autonomous navigation of an Unmanned Ground Vehicle (UGV). Krueger et al. [25] have proposed an *Object-Action Complex* (OAC) as the basis for closely coupling different objects and the actions associated with them. Interested readers can refer to [1], [16], [17] for more examples. Various paradigms of cognitive frameworks have also been defined, each having its own advantages and disadvantages (see Vernon et al. [31] for an excellent overview of cognitive architectures). The second problem to be addressed for robot autonomy is that of Simultaneous Localization And Mapping (SLAM) deals with the problem of navigating within an environment as

*This work was supported, in part, by AFOSR grant AFOSR-FA9550-12-1-0291, and the University of Utah Graduate School Fellowship Program

[1]Anshul Joshi is a PhD Candidate at the University of Utah, Salt Lake City. 201 Presidents Cir, Salt Lake City, UT 84112, USA. joshi@cs.utah.edu

[2]Thomas C. Henderson is a Professor in School of Computing, University of Utah. 201 Presidents Cir, Salt Lake City, UT 84112, USA tch@cs.utah.edu

well as mapping it at the same time under motion and sensing uncertainties. A significant amount of research in the robotics community is devoted to solving this problem (see [11] for a survey on current SLAM techniques). Our work in this regards to cognitive framework and localization method deals with creating a cognitive framework that incorporates the wreath product representation and based on a well-studied architecture; the belief-desire-intention (BDI) architecture, and is practical. We illustrate its design and performance in this work which is divided into sections as follows. Section III gives the basics of the BDI architecture with a description of high-level functionality of all blocks. Section IV elaborates on the algorithms that form the core of our framework. Section V demonstrates the localization algorithm used to measure the accuracy of our system. Section VI describes the experimental setting, and the accuracy of results obtained. Section VII concludes with brief explanation of potential for future developments.

III. BDI ARCHITECTURE

We use a Belief Desire Intention (BDI) framework which has been studied in detail by researchers attempting to create a practical and rational reasoning agent (see [28], [32], [33]). Our main contribution in this work is to represent *beliefs* about the environment as wreath products, and to exploit these beliefs in a mapping and localization scenario. The layout of the *Wreath Product Cognitive Architecture* (WPCA) is shown at a high level in Figure 1. This architecture is based on the BDI model, and allows object representations to be constructed from symmetry groups discovered in sensorimotor data, and combined to form wreath products.

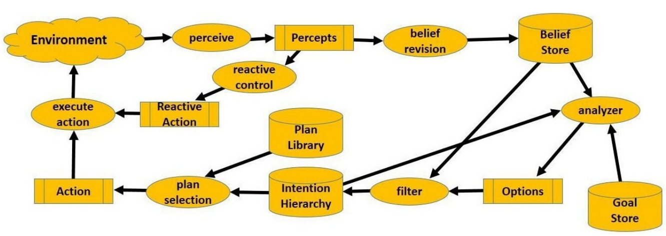

Fig. 1. BDI Architecture

The WPCA consists of:

- a set of repositories:
  - *belief store*: maintains all beliefs about the world.
  - *goal store*: asserts the goals of the robot agent as well as their priorities.
  - *intention hierarchy*: keeps a small number of currently selected most important specific goals.
  - *plan library*: consists of processes designed to achieve specific goals.
- a number of short-term memories:
  - *percepts*: information derived from sensorimotor data
  - *options*: a set of possible intentions which are then filtered to get a small set of intentions.
  - *action*: selected agent action (for example, take data, move, communicate).
  - *reactive action*: emergency control of robot (for example, obstacle avoidance).
- and a set of processes:
  - *perceive*: acquire and convert raw data into percepts (the first step toward wreath product construction).
  - *belief revision*: updates beliefs, including: add new beliefs, revise old beliefs and update uncertainties.
  - *analyzer*: considers beliefs, goals and current intentions and produces a new set of intentions.
  - *filter*: selects a small set of intentions from the options.
  - *plan selection*: given the current intentions, find appropriate plans to achieve them.
  - *reactive control*: interrupt normal cognitive processing in emergency situations.
  - *execute action*: robot platform level control.

The mapping from *percepts* to *action* for the robot system is called from a higher-level module that interfaces the WP architecture to the hardware. The blocks *belief revision*, *analyzer*, *options* generation, *filter*, *plan selection*, *action generation* are part of the robot's cognitive function which processes perceptual data (executes reactive behavior if deemed necessary), revises its beliefs based on this data, analyzes the current intentions, goals, and beliefs to generate options (actions that can be taken), and then selects the plan that lays out the sequence of steps (actions) needed to achieve a particular goal. For example, one of the goals in our framework is to *discover a world frame*, and the plan provides a sequence of actions which gather data from its environment searching for a world frame (a corner where two walls and the floor meet to form an orthogonal bases), and moves the robot around in its environment until such a world frame is found. Every plan in our framework is a finite state machine, which assesses current beliefs and recommends a particular action. Actions include taking data, translating (forward or backward), and rotating in place by a given angle (positive or negative rotation). For more details on the workings of various blocks not discussed here, please refer to [22].

IV. ALGORITHMS

We now elaborate on the design of the BDI framework. The most significant modules are described at a high level in the following subsections, and the algorithms are simplified and details about various helper functions are omitted for readability. Interested readers can find more details in [22]. Significant functionalities include the *Environment* module and the *WP BOT* module.

A. Environment Interface Module

The environment interface module (consisting of the *perceive* and *action* functions) serves as the interface between the robot brain and the environment, providing access to its sensor and actuators. This module interfaces the WP architecture to the hardware, i.e., the robot, and the Kinect, and thus handles the image acquisition module (which collects depth and RGB data from the Kinect), execution of actions (robot motion), other housekeeping activities, such as termination of execution upon completion, and logging all these processes. For the purpose of this work, this indoor environment is a cluttered environment of a lab, or an open area (atrium/lobby) inside a building.

B. Robot Brain: WP BOT

WP BOT is the brain of the robot that utilizes the BDI paradigm. It is comprised of the WP discovery, localization, belief revision, analyzer, filter, and plan selection and action generation functions. At the start, beliefs are initialized with the innate knowledge available to the robot, and the variables that will persist throughout the execution of this program. The percepts are passed on as arguments to the DATA_TO_WP function (discussed in subsection IV-C below), which processes these percepts to create wreath product sets (WP_SETS) that contain (possibly newly-discovered) wreath products. The robot starts its life with little innate knowledge. For example, part of the innate knowledge is the transform that maps the camera frame to the robot frame. Based on this, the robot will first discover the floor vector (i.e., the vector of the plane that is pointing *up*), to identify the floor plane, which is a special plane on which the robot moves, and updates beliefs with this information. Once the floor plane vector is discovered, the robot will try to discover the world frame, which is any corner in the surroundings that has 3 orthogonal walls (one of which needs to be the floor) that meet at a single point.

C. Data To Wreath Products: Wreath Product Construction Cycle

This function transforms range data from the depth sensor into wreath products. The process starts by building planes from 3D data points in the camera frame that are segmented using the RANSAC (RANdom Sample And Consensus) algorithm ([14], [34]). For example, in Figure 2 (a), the three orthogonal planes, found during the search for world frame, segmented using RANSAC are shown in different colors, along with the axes found (which are the normals to the three planes located at the origin). Even though these normals might be very close to, but not exactly orthonormal, they can converted into three orthonormal vectors (M_{ortho}) using the following equation.

$$M_{ortho} = M(M^T M)^{-\frac{1}{2}} \tag{1}$$

where

$$M = [R_x^T R_y^T R_z^T] \tag{2}$$

is composed of the three rotational components R_x, R_y, R_z for transforming points from robot frame to world frame (this orthonormal transform is used later during merging beliefs). Each plane point is then transformed from the camera frame to the robot frame (using the innate knowledge of the transform ${}^R T_C$ - camera frame to robot frame). For each plane, the plane parameters are found, namely, the plane normal n_{pi} using singular value decomposition (SVD), distance to the plane from origin d_{pi}, and error, ϵ_{pi}, of the plane points fit to the plane. Duplicate planes are removed by merging planes that have similar surface normal and similar distance from the origin. Thereafter, lines (R) and points (E) are found by intersection of two (i, j), or three $[R \times R]$ planes (i, j, k) respectively. More details on the E, R, and $[R \times R]$ notation can be found in [24]. Figure 2 (b) shows a simple illustration of the WPs discovered by this process. The yellow star has been added which signifies the location of the world frame origin. The plane points (in black) are sparsely plotted to show the 3 planes (denoted by $[R \times R]$ and green dotted lines added for demarcation). Each plane's normal is shown as blue arrows. Lines R's are intersection of pairs of planes and are shown in red, whereas points E's are intersections of three planes and are shown as blue dots. This particular image was generated from a set of beliefs that contained 12 innate beliefs (robot pose, camera to robot transform, gravity vector, and other information), and an additional 11 beliefs were discovered (four planes, five lines, and two points), for a total of 23 beliefs.

Since an indoor office environment is mostly comprised of planar surfaces, this algorithm effectively finds most of the planes, lines, and points, and their parametric information, if any. All these planes, points, and lines will be added to the WP_set. Note that the superscript R - not to be confused with the wreath product R which is a line - of a wreath product signifies that all these WPs have been transformed from the camera reference frame to the robot reference frame; they will be transformed into the world reference frame when the world frame is discovered, in the MERGE function.

D. Belief Revision

This function merges newly discovered beliefs (as wreath products) with existing beliefs. Newly discovered WPs are transformed into the world reference frame - denoted by superscript W - where they are compared against the existing planes, lines, and points in beliefs to check if they are duplicates, and to merge them if so. Functions have been developed to match planes, lines and points with their counterparts in the existing beliefs, and those that match are merged. The beliefs are then updated accordingly.

V. LOCALIZATION ALGORITHM

Our localization algorithm (based on [15], [30]) works in two steps: motion localization, and WP landmark localization. These two steps are elaborated in the following subsections.

A. Motion Localization

This procedure updates the robot pose based on control commands sent to the robot. The algorithm accepts the current state μ, covariance Σ, and control u_t as the arguments. Based on whether a translation motion occurred ($v_t \neq 0$) or a rotational motion occurred ($\omega_t \neq 0$), the state will be updated accordingly. This algorithm is a standard motion model update (more details in [30]). Since the robot has

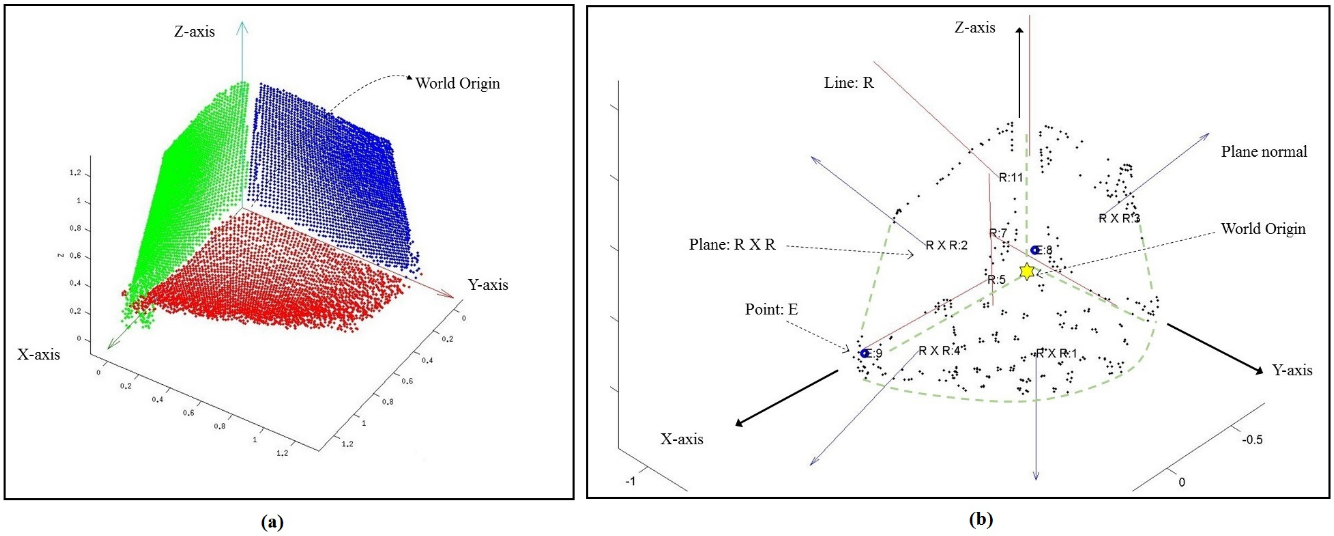

Fig. 2. WP discovery from data: (a) Shows the different planes segmented using RANSAC. (b) Shows the detailed WPs discovered.

only two discrete motions - translation and rotation - and not a combination of both at the same time, this model is simpler than the motion update algorithm of robots with more complicated drive systems (see [30, Chapter 5] for more details on this).

B. Wreath Product (WP landmark) Localization

WP landmark correspondence matches newly discovered landmarks, Z_t (observations in current timestep t, transformed from robot frame into the world frame), to existing WP landmarks in beliefs (LMs), and adds the indexes to correspondence C if they are close enough. We localize the robot based on the geometric constraints that must be satisfied if the type of WP landmark correspondence is known. For example, lines (parallel to the floor) and planes not parallel to floor plane with intersect with the floor plane, to form line constraints. Lines parameters from such intersections are determined for each line; note that we are determining on which side of these lines the robot lies. Once this side is determined, the robot has to be on a line that is at a certain distance from, and parallel to this observed line. For lines not parallel to the floor, we determine intersection of such lines or vertical planes with the floor. If multiple constraint lines are present, pairs of such constraint lines will intersect to give us multiple possible locations the robot might be in (one per pair of intersecting lines), whereas for single lines the closest point on the constraint line is our possible location (all these possible locations will be added to μ hypotheses, including the one determined by the motion localization). Similarly, lines intersecting the floor planes (for example, vertical lines) will give us points, and coupled with already known point landmarks signify that the robot is at a certain distance from these points, i.e., the robot has to be on the circle(s) with these point(s) as center(s). For a single circle constraint, the closest point on this circle to the current robot location is an additional μ hypothesis. If multiple circles are present, intersection of pairs these circles will give us additional μ hypotheses similar to the case with multiple lines. Orientation estimate θ hypothesis is simply the difference between the robot orientation with respect to an existing landmark, and its orientation with respect to a newly observed landmark ($\theta_{curr}+\theta_{z_t}-\theta_{lm}$), added to current μ_θ hypotheses. Multiple hypotheses ($\mathcal{H}_\mu$ for location and $\mathcal{H}_\theta$ for orientation) might be generated, as mentioned above, depending on how many constraint circles and constraint lines are discovered as explained above. These hypotheses are combined using weights assigned to the hypotheses based on their uncertainties, that are determined using standard applied optimal estimate, and are based on the variance associated with each hypothesis (see [15] for more details on this technique).

VI. EXPERIMENTAL SETTING AND RESULTS

Sprunk et al. [29] define a detailed benchmarking protocol for evaluation of robot indoor navigation algorithms. We use their localization performance as a benchmark to quantify the performance of our representational framework. In addition, we also refer to the localization performance of Biswas et al. [3], [4], [5], [6], Microsoft Research (MSR), and Endres et al. [12], [13] for benchmark values.

A. Performance Measures

Tables I and II show the performance measures (in terms of localization error in x, y-location in meters and angular error (orientation in degrees), respectively, for the mean, median, minimum, and maximum error encountered during test runs. Our system (BDI) is compared against Sprunk's benchmark system which uses the Pioneer P3-DX robot platform, MSR's P1 robot platform, Biswas' Fast Sampling Plane Filtering (FSPF) algorithm using the Kinect, and Endres' hand-held SLAM implementation using the Kinect on a dataset. Specifically, the system that resembles our framework the most, in terms of the sensor and spatial features used, is the FSPF system from CMU. (Statistics

System→	**WPCA**	Pioneer	MSR	FSPF	Endres'12
Mean	0.0989	0.22	0.23	0.7	0.097
Median	0.0948	X	X	1.08	X
Min	0.0511	0.12	0.03	0.17	0.034
Max	0.2574	0.32	0.43	3.47	0.16

TABLE I

LOCATION (x, y POSE) ERROR (IN METERS) COMPARISON BETWEEN SYSTEMS.

System→	**WPCA**	Pioneer	MSR	FSPF	Endres'12
Mean	10.4221	X	0.5	X	3.39
Median	8.9267	X	X	X	X
Min	2.3491	X	X	X	1.84
Max	29.1521	X	2.5	X	4.94

TABLE II

ANGULAR (ORIENTATION) ERROR (IN DEGREES) COMPARISON BETWEEN SYSTEMS.

that were not available for some of these implementations has been marked with an 'X.'). We observe that our system performs better than the Pioneer, MSR and FSPF systems, and is comparable to the Endres'12 system, with respect to the localization error. Our system does not perform so well for the orientation error, compared to the systems that provided orientation error data. However, it must be noted that our algorithms have very high tolerances for landmark correspondence since landmark rediscovery is more important to us than filtering out landmarks that are fairly close to one another, but which might not meet tight tolerances, which tends to oscillate our orientation measures significantly.

VII. CONCLUSIONS AND FUTURE WORK

We have demonstrated a novel practical implementation of a BDI architecture using wreath product representation for environmental data. We have also shown that this implementation works very well in the context of robot localization, one of the most essential functionality of a mobile robot, and a precursor to other intelligent capabilities. The performance of our localization algorithm using wreath products as landmarks compares fairly well to localization algorithms on other systems similar to ours; it performs better than other systems in terms of localization error, but does not perform well in terms of orientation error which can be attributed to trade-off between landmark correspondence discovery, and matching thresholds. Future work in this regard would involve improving orientation accuracy without negatively affection location accuracy or landmark rediscovery. One powerful capability (discussed in our other work) involves converting beliefs into a linear representation - for e.g., a string - for sharing and recovery of WPs from data, using context-free grammar and (deterministic) pushdown automaton (PDA). WPs can also be converted to plans, for e.g., using the line representation R to create a motion plan for the robot. These capabilities will be addressed in future.

REFERENCES

[1] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa and M. Ogino, and C. Yoshida. Cognitive Developmental Robotics: A Survey. *IEEE Transactions on Autonomous Mental Development*, 1(1):12–34, 2009.

[2] P.F. Beeson. *Creating and Utilizing Symbolic Representations of Spatial Knowledge using Mobile Robots* . PhD thesis, University of Texas, Austin, TX, 2008.

[3] J. Biswas. *Creating and Utilizing Symbolic Representations of Spatial Knowledge using Mobile Robots*. PhD thesis, Carnegie-Mellon University, Pittsburgh, PA, 2014.

[4] J. Biswas and M. M. Veloso. Wi-fi Localization and Navigation for Autonomous Indoor Mobile Robots. In *IEEE International Conference on Robotics and Automation*, Anchorage, AK, 2010.

[5] J. Biswas and M. M. Veloso. Depth Camera based Indoor Mobile Robot Localization and Navigation. In *IEEE Conference Robotics and Automation*, St. Paul, MN, 2012.

[6] J. Biswas and M. M. Veloso. Multi-sensor Mobile Robot Localization for Diverse Environments. In *RoboCup 2013: Robot World Cup XVII: 468-479*. Springer, Berlin, 2013.

[7] A. Desai and D.J. Lee. Using Probabilistic Model as Feature Descriptor on a Smartphone Device for Autonomous Navigation of Unmanned Ground Vehicles. In *SPIE Electronic Imaging, Intelligent Robots and Computer Vision XXX: Algorithms and Techniques*, volume 9025-18, San Francisco, CA, February 2014.

[8] A. Desai, D.J. Lee, and D. Ventura. Matching Affine Features with SYBA Feature Descriptor. In *Proceedings of the IEEE International Conference on Visual Computing*, Las Vegas, NV, December 2014.

[9] A. Desai, D.J. Lee, and D. Ventura. An Efficient Feature Descriptor Based on Synthetic Basis Functions and Uniqueness Matching Strategy. *Computer Vision and Image Understanding*, 142:37–49, January 2016.

[10] A. Desai, D.J. Lee, and C.N. Wilson. Using Affine Features for an Effiicient Binary Feature Descriptor. In *Proceedings of the IEEE Southwest symposium on Image Analysis and Interpretation*, pages 49–52, San Diego, CA, April 2014.

[11] H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.

[12] F. Endres, J. Hess, J. Sturm, D. Cremers, , and W. Burgard. 3-D Mapping With an RGB-D Camera. *IEEE-T Robotics*, 30(1), February 2014.

[13] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard. An Evaluation of the RGB-D SLAM System. In *IEEE International Conference on Robotics and Automation*, Saint Paul, MN, May 2012.

[14] M.A. Fischler and R.C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*.

[15] A. Gelb, J.F. Kasper, R.A. Nash, C.F. Price, and a.A. Sutherland, editors. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.

[16] G.H. Granlund. Organization of Architectures for Cognitive Vision Systems. In *Cognitive Vision Systems*, pages 37–55. Springer, 2006.

[17] F. Guerin. Learning Like a Baby: a Survey of Artificial Intellgence Approaches. *The Knowledge Engineering Review*, 26(2):209–236, 2011.

[18] T.C. Henderson, N. Boonsirisumpun, and A. Joshi. Symmetry Based Semantic Analysis of Engineering Drawings. In *Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, Beijing, China, September 2014.

[19] T.C. Henderson, E. Cohen, E. Grant, A. Joshi, M.T. Draelos, and N. Deshpande. Symmetry as a Basis for Perceptual Fusion. In *Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, Hamburg, Germany, 2012.

[20] T.C. Henderson, H. Peng, C. Sikorski, N. Deshpande, and E. Grant. Symmetry: A Basis for Sensorimotor Reconstruction. Technical Report UUCS-11-011, The University of Utah, May 2011.

[21] Thomas C. Henderson, Narong Boonsirisumpun, and Anshul Joshi. Actuation in Perception: Character Classification in Engineering Drawings. In *Proceedings IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, San Diego, September 2015. IEEE.

[22] A. Joshi. *Wreath Product Cognitive Architecture*. PhD thesis, University of Utah, Salt Lake City, UT, 2016.

[23] A. Joshi, T.C. Henderson, and W. Wang. Robot Cognition using Bayesian Symmetry Networks. In *Proceedings of the International Conference on Agents and Artificial Intelligence*, Angers, France, 2014.

[24] A. Joshi and Thomas C. Henderson. Generative cognitive representation for embodied agents. In *Multisensor Fusion and Information Integration (MFI)*, Beijing, China, 2014.

[25] N. Krueger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Woergoetter, A. Ude, T. Asfour, D. Kraft, D. Omrcen, A. Agostini, and R. Dillmann. Object-Action Complexes: Grounded Abstractions of Sensory-Motor Processes. *Robotics and Autonomous Systems*, 59(1):23–40, 2011.

[26] M. Leyton. *Symmetry, Causality, Mind*. MIT Press, Cambridge, MA, 1992.

[27] M. Leyton. *A Generative Theory of Shape*. Springer, Berlin, 2001.

[28] A.S. Rao and M.P. Georgeff. BDI Agents: from Theory to Practice. In *ICMAS*, volume 95, pages 312–319, 1995.

[29] C. Sprunk, G. Parent, L. Spinello, G. D. Tipaldi, W. Burgard, and M. Jalobeanu. An Experimental Protocol for Benchmarking Robotic Indoor Navigation. In *International Symposium on Experimental Robotics (ISER)*, pages 487–504, Marrakech, Morocco, 2016. Springer.

[30] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, Cambridge, MA, 2005.

[31] D. Vernon, G. Metta, and G. Sandini. A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development o f Mental Capabilities in Computational Agents. *IEEE Transactions on Evolutionary Computation*, 11(2):151–180, 2007.

[32] G. Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA, 1999.

[33] M.J. Wooldridge. *Reasoning about Rational Agents*. Intelligent Robots and Autonomous Agents. The MIT Press, Cambridge, MA, 2000.

[34] M. Zuliani. The Multiransac Algorithm and Its Application to Detect Planar Homographies. In *Proceedings of the IEEE Conference on Image Processing*, September 2008.