The Cognitive Symmetry Engine

Thomas C. Henderson, Anshul Joshi and Wenyi Wang
University of Utah

UUCS-13-004

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

10 September 2013

Abstract

We propose the use of symmetry theories as the basis for the interpretation of sensorimotor data and the creation of more abstract representations. Here we outline a cognitive architecture to implement such an approach and provide a set of specific mechanisms for 1-D, 2-D and 3-D sensorimotor processing. The overall goal is to integrate low-level sensorimotor data analysis and behavior with more abstract affordance representations. Sensorimotor affordance and cognition is an essential capability for self-learning robots. Given only minimal innate knowledge but well-defined sensorimotor cognitive mechanisms, a robot should be able to identify useful relations between its different actuators and sensors. Symmetry plays an important role in identifying invariant sensor-actuator signal relations, and these invariances can be effectively exploited if such relations are bundled for future use. We call these collections of simultaneous symmetries in actuator commands and sensed signals *Symmetry Bundles*. Along with the theoretical framework and semantics of Symmetry Bundles, we define new practical approaches to detect, classify and bundle the inherent symmetries present in signals in order to form useful affordances. The overall cognitive architecture is called the *Cognitive Symmetry Engine*.

1 Introduction

We explore the thesis that symmetry theory provides key organizing principles for cognitive robot architectures. Cognitive systems perceive, deliberate and act in unstructured environments, and the development of effective mental abilities is a longstanding goal of the AI and intelligent systems communities. As described by Vernon et al. [69], cognition ”can be viewed as a process by which the system achieves robust, adaptive, anticipatory, autonomous behavior, entailing embodied perception and action.” Their survey considers two basic alternative approaches to cognition: *cognitivist* (physical symbol systems) and *emergent* (dynamical systems), where the cognitivist paradigm is more closely aligned with disembodied symbol manipulation and knowledge representation based on a priori models, and the emergent paradigm purports dynamic skill construction in response to perturbations to the embodiment. Basically, cognitivists maintain that patterns of symbol tokens are manipulated syntactically, and through percept-symbol associations perception is achieved as abstract symbol representations and actions are causal consequences of symbol manipulation. In contrast, emergent systems are concurrent, self-organizing networks with a global system state representation which is semantically grounded through skill construction where perception is a response to system perturbation and action is a perturbation of the environment by the system. The emergent approach searches the space of closed-loop controllers to build higher-level behavior sequences out of lower ones so as to allow a broader set of affordances in terms of the sensorimotor data stream. An important aspect of this discussion which concerns us here is that raised by Krichmar and Edelman [31]: ”the system should be able to effect perceptual categorization: i.e. to organize unlabeled sensory signals of all modalities into categories without a priori knowledge or external instruction.” We address this issue and propose that certain fundamental a priori knowledge about symmetries is vital to this function.

Vernon later took up Maturana and Varela’s *enaction* conceptual framework for cognitive systems [68]. The goal there is to understand how to describe the role of development in making an agent act effectively and gain new skills. The five basic elements of enaction are: (1) autonomy, (2) embodiment, (3) emergence, (4) experience and (5) sense making. The last one is considered the most important: ”emergent knowledge is generated by the system itself and it captures some regularity or lawfulness in the interactions of the system, i.e. its experience. However, the sense it makes is dependent on the way in which it can interact: its own actions and its perceptions of the environments actions on it.”

This is the key issue addressed in this paper: it seems somewhat contradictory to say that ”regularity or lawfulness” are captured ”without a priori knowledge.” How can a law or regularity be recognized without knowing the law or rule? Our claim is that symmetries help characterize these regularities.

Our goal is to advance the state of the art in embodied cognitive systems. The requirement for cognitive ability is ubiquitous, and its achievement is an essential step for autonomous mental development. At its root, a cognitive architecture is a structural commitment to processes and representations that permit adaptive control in an operating environment that cannot be modeled completely a priori. A cognitive agent optimizes its behavior to achieve an objective efficiently by finding models that resolve hidden state information and that help it to predict the future under a variety of real-world situations. These processes involve monitoring, exploration, logic, and communication with other agents. It is necessary to create new theories and realizations for cognitive organization in complex, real-time systems that consist of interacting domain specific agents, each with rich internal state and complex actions in order to facilitate the construction of effectively organized cognitive infrastructure.

The proposed technical basis for this is symmetry operators used in perception, representation and actuation. Our specific hypothesis is:

The Domain Theory Hypothesis: We propose that robot affordance knowledge acquisition and perceptual fusion can be enabled by means of a common sensorimotor semantics which is provided by a set of group symmetry theories embedded a priori in each robot. These theories inform the production of structural representations of sensorimotor processes, and these representations, in turn, permit perceptual fusion to broaden categories of activity. The *Domain Theory* predicates:

1. a representation of an innate theory and inference rules for the theory,
2. a perceptual mechanism to determine elements of a set and operators on the set,
3. a mechanism to determine that the set and its operators are a model of the innate theory, and
4. mechanisms to allow the exploitation of the model in learning and model construction.

As pointed out by Weng [71], a major research question in autonomous mental development is ”how a system develops mental capabilities through autonomous real-time interactions with its environment by using its sensors and effectors (controlled by an intrinsic development program coded in the genes or designed in by hand).” Thus, a representation is sought derived from sensorimotor signals as well as the grouping of such signals as processing takes place. Note that this assumes that no coordinate frames exist in this setting; see [65] for a discussion of coordinate frames in biological systems. Asada et al. [2] give a good account of the development of body representations in biological systems and maintain that ”motions deeply participate in the developmental process of sensing and perception.” They review data ranging from spinal reflexes with fixed motor patterns, to motion assembly, to mixed motion combinations in the cerebrum. Lungarella [40] also has much to say on this issue, and of great interest here, states that ”spontaneous activity in newborns are not mere random movements ... instead organized kicks, arm movements, short phase lags between joints ... may induce correlations between sensing and motor neurons.”

Our proposed method is to detect and exploit various symmetries in the sensorimotor data in order to achieve the objectives. Symmetry [73] plays a deep role in our understanding of the world in that it addresses key issues of invariance, and as noted by Viana [70]:

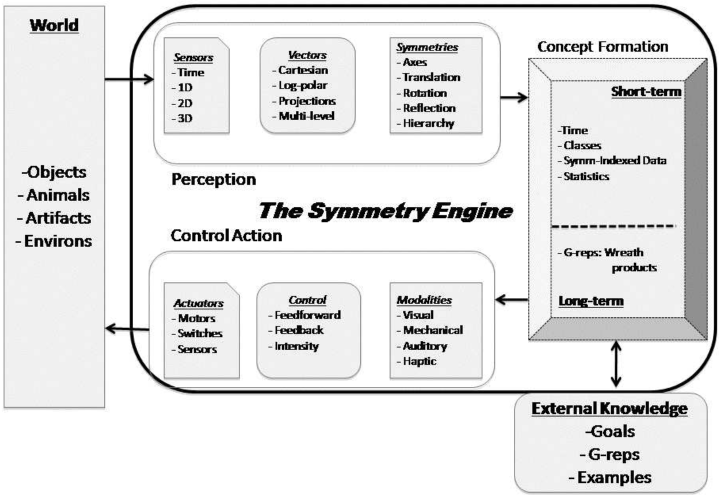


Figure 1: The Symmetry Engine. *Perception* requires an appropriate set of operators to construct *G-reps*; this includes vector constructors, symmetry detectors, and symmetry-based data indexing and variance operators. *Control action* requires the ability to map *G-reps* onto action sequences to achieve desired results in the world. *Concept Formation* operators allow the exchange of *G-reps* with other agents.

“Symmetry provides a set of rules with which we may describe certain regularities among experimental objects.” Symmetry to us means an invariant, and by determining operators which leave certain aspects of state invariant, it is possible to either identify similar objects or to maintain specific constraints while performing other operations (e.g., move forward while maintaining a constant distance from a wall). Operationally, the hypothesis is that *group theoretic representations (G-Reps)* inform cognitive activity. In related work, Leyton proposed *wreath products* [35, 36] as a basis for cognition. Leyton argues that the *wreath group product* is a basic representation for cognition as stated in the *Grouping Principle*: "Any perceptual organization is structured as an n-fold wreath product $G_1 \wr \ldots \wr G_n$" and proposes that "human perceptual and motor systems are both structured as wreath products." We loosely use that formalism (the operator $\wr$ indicates a group sequence), and plan to demonstrate that symmetry-based signal analysis and concept formation allow us to address (1) the sensorimotor reconstruction problem, (2) affordance learning, and (3) affordance representation and indexing for life-long experience. A schematic view of our proposed symmetry-based affordance architecture (the *Symmetry Engine*) is given in Figure 1. The successful demonstration of this approach will constitute a major advance in the field of cognitive autonomous agents, and will also motivate joint research programs into human cognition. For a more biologically motivated cognitive architecture which learns features for hierarchical models to recognize invariant objects, see [72] as well as other papers from the Honda research group on their approach to cognitive architecture [5, 6, 58].

Our major research thrusts to construct this robot cognitive architecture are:

- **Symmetry (Symbol) Detection**: This involves the recognition of symmetry tokens in sensorimotor data streams. Various methods are proposed for this in 1D, 2D and 3D data. Here symmetries are various invariant affine transformations between subsets of the data, including translation, rotation, reflection, scaling, etc. Also important is the detection of local and global symmetry axes.
- **Symmetry Parsing**: A collection of sensorimotor data gives rise to a set of tokens which must be parsed to produce higher-level nonterminal symbols (or concepts). We propose that a symmetry grammar is innate in the robot, but that experience informs its specific structure for a given robot.
- **Symmetry Exploitation**: Symmetries can be used to solve the sensorimotor reconstruction problem, to represent new concepts, and to discover and characterize useful behaviors.

1.1 Cognitive Architecture

Figure 2 provides a more detailed view of our current cognitive architectural implementation based on the *Symmetry Engine*. A particularly important feature is the Behavior Unit (in the middle of the figure). Behavior is encoded as *Object-Action Complexes* [1, 32] (in brief, an OAC is a triple (E, T, M) where the execution is specified by E, T is a prediction function on an attribute space, and M is a statistical measure of success of the behavior).

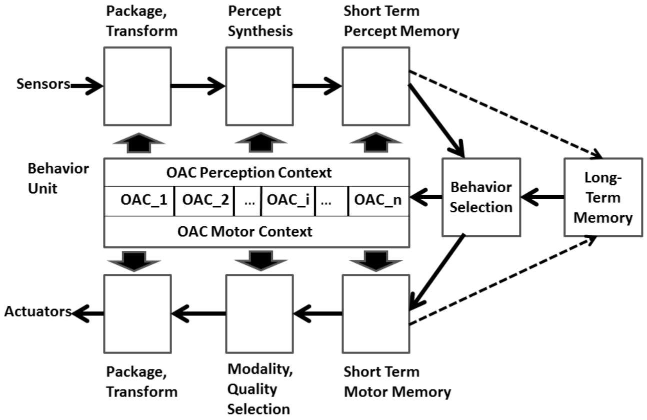


Figure 2: General Cognitive Framework Architecture.

In Figure 2, the *Behavior Selection* function chooses the next OAC sequence based on the current OAC, the current states of the *Short-Term Perception Memory* and the *Short-Term Motor Memory*, as well as the available behaviors in the *Long-Term Memory*. As an OAC executes, it provides context to both the perception and motor pipelines. Data arrives continuously from the sensors and first undergoes specific packaging and transformation procedures, then is formed into percepts (symmetry characterizations), and finally, results are stored in the *Short-Term Memory*. Similarly, motor commands are moved to the *Short-Term Motor memory* where they are then interpreted according to modality and the specific qualities desired, and finally, these more symbolic representations are decoded into specific motor commands for the actuators.

As a simple example of the perceptual-motor duality of the architecture, consider a square shape. As described in more detail below, the boundary of the shape can be represented as a point which undergoes a specific sequence of symmetry transforms: translation (half the side length), reflection (about the line perpendicular to the endpoint), and rotation (4-fold about the z-axis). This same representation can be used to issue motor commands to trace out the shape or to circumnavigate it (e.g., go around a table).

2 Symmetry Detection

Symmetry detection has played a large role in 2D and 3D image and shape analysis and computer graphics; see [13, 16, 29, 28, 33, 34, 39, 47, 51]. In robotics, we have previously shown how to use symmetry detection in range data analysis for grasping [22]. Popplestone and Liu showed the value of this approach in assembly planning [38]. More recently, Popplestone and Grupen [53] gave a formal description of general transfer functions (GTF's) and their symmetries. Finally, Selig has provided a geometric basis for many aspects of advanced robotics using Lie algebras [61, 62].

A symmetry defines an invariant; according to Weyl [73]:

> An object is symmetrical if one can subject it to a certain operation and it appears exactly the same after the operation. The object is then said to be invariant with respect to the given operation.

The simplest invariant is identity. This can apply to an individual item, i.e., a thing is itself, or to a set of similar objects where the operation is some geometric (or other feature like texture) transform. In general, an invariant is defined by a transformation under which one object (or a feature of the object) is mapped to another object (or its features). We propose that sensoriomotor reconstruction can be more effectively achieved by finding such symmetry operators (invariants) on the sensor and actuator data (see also [10, 30]).

2.1 Symmetry Detection in 1-D Signals

Here we are looking for patterns in finite 1-D sample sets. Let t_i be the independent time (sample index) variable and $y_i = f(t_i)$ be the sample values (e.g., real numbers). Assume a set of sensors, $\mathcal{Y} = \{Y_i, i = 1 \ldots n_{\mathcal{Y}}\}$ each of which produces a finite sequence of indexed sense data values, Y_{ij} where i gives the sensor index and j gives an ordinal temporal index, and a set of actuators, $\mathcal{A} = \{A_i, i = 1 \ldots n_{\mathcal{A}}\}$ each of which has a finite length associated control signal, A_{ij}, where i is the actuator index and j is a temporal ordinal index of the control values. Symmetries are defined in terms of permutations of the sample indexes and values. Given $Y_{i,j}, j = 1 \ldots 2k+1$, a set of samples from a sensor, then symmetries are detected as follows. The $2k+1$ sample points comprise a moving window on the data from this sensor, and analysis takes place at the center point (Y_{k+1}). The possible symmetries are:

Constant Signal: (Any point maps to any other.) Under the map $x \rightarrow x + a, \forall a \in [-k, k]$, and mapping the corresponding y values as well, the sample signal does not change. All possible permutations of time-sample pairs leave the signal invariant (i.e., S_n, the symmetry group characterizes a constant signal).

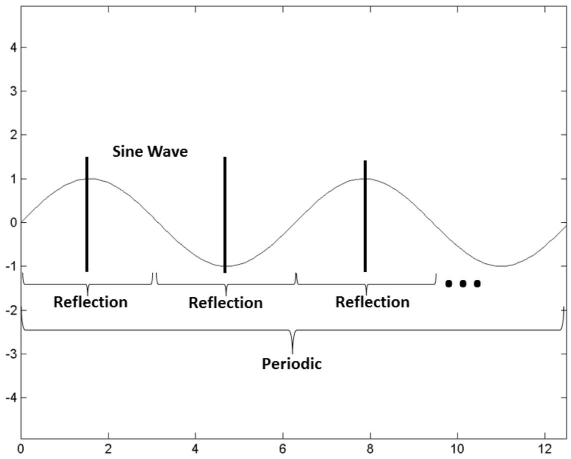


Figure 3: Various Symmetries in a Sin Wave.

Periodic Signal: (No point maps to itself.) Under the map $x \to x + a$, for some fixed, non-zero value a, and mapping the corresponding y values as well, the sample signal does not change.

Reflection Signal: (Only one point maps to itself.) Under the map $x \to a - x, \forall x\, in[-k, k]$, and mapping the corresponding y values as well, the sample signal does not change.

Asymmetric Signal: (Each point maps only to itself.) The only map for which the signal remains unchanged is the identity map: $x \to x$. Note that most functions are like this, as are pure noise signals.

Linear signal In order to detect a linear (non-constant) relation in the data, we take the derivative of the sample data (i.e., $Y_j' = \frac{Y_{j+1} - Y_j}{t_{j+1} - t_j}$) and look for the constant signal.

Gaussian Noise Signal: Any signal for which the autocorrelation of the sample set results in a low amplitude signal everywhere except zero.

Note that the above analysis could be performed on 1-D point sets on the real line by quantizing the sample values, and then looking for specific patterns in symmetries existing on those point sets. E.g., for a periodic pattern, all point sets would have the same translation symmetry. Moreover, the analysis can also be done by parsing the samples in terms of grammars defining these symmetry types (see Section 3).

A first level symmetry is one that characterizes a single signal as belonging to one of these categories. Of course, composite signals can be constructed from these as well, e.g., the sine function has a hierarchy of symmetries (see Figure 3). As seen in the figure, a sine wave gives rise to several symmetries: there is a reflective symmetry about the vertical axis for points between $[0, \pi]$, $[\pi, 2\pi]$, etc., and the predominant symmetry is the discrete translational symmetry of period 2π, i.e., $sin(x) = sin(x+2\pi)$. Such a signal at the highest level is then represented by the token $P_{b=sin([0,2\pi]),T=2\pi}$. Note that symmetry analysis may be applied to transformed signals (e.g., to the histogram of a signal; e.g., where a Gaussian sample is of type D_1). Asymmetric signals will also be represented as a symbolic sequence, i.e., using the Symbolic Aggregate Approximation method of Lin et al. [37].

Next, pairwise signal symmetries can exist between signals in the same class:

- *linear*
  - same line: $a_1 = a_2$, $b_1 = b_2$
  - parallel: $a_1 = a_2$, $b_1 \neq b_2$
  - intersect in point: rotation symmetry about intersection point
- *periodic*
  - same period: $P_1 = P_2$
  - same Fourier coefficients: $C_1 = C_2$
- *Gaussian*
  - same mean: $\mu_1 = \mu_2$
  - same variance: $\sigma_1^2 = \sigma_2^2$

We have developed algorithms to detect these symmetries and have used them to classify sensor and actuator types in the sensorimotor reconstruction problem (see [26] and below). This allows sensor classification without any actuation (i.e., much lower energy expenditure), and achieves much greater classification correctness compared to previous methods. The symbolic output of the 1-D symmetry analysis is one of:

- C_1: an asymmetric signal b.
- T: a continuous translational signal; i.e., a line (segment) $at + by + c = 0$.
- D_1: a signal with reflective symmetry.

- $\mathbb{P}$: a periodic signal with base shape b and period T.

The extended analysis produces results for:

- $GN(X)$: Gaussian noise $\mathcal{N}(\mu, \sigma^2)$.

2.2 Symmetry Detection in 2-D Signals

Symmetries must also be found in 2-D and 3-D spatial data, like camera and range data (note that the spatial layout of 1-D sensors - pixels - can be learned from the correlations in the neighboring streams of 1-D signals [46, 49, 48]). Our view is that much like in the case of 1-D data where a central signal value is chosen as the origin about which to find 1-D symmetries, pixel-centric image transforms (e.g., the log-polar transform) can be used to help bring out symmetries in 2-D shapes. Moreover, such an analysis is performed in terms of a sensorimotor combination which is intrinsic to that object. For example, saccadic movement of the eye relates motor control information coordinated with the simultaneous image percepts. This issue is further explored below in symmetry exploitation.

The 2-D symmetries to be detected are:

- *cyclic symmetry* (denoted C_n): rotational symmetry of $\frac{2\pi}{n}$ radians (e.g., yin-yang symbol) with no reflection symmetry.
- *dihedral symmetry* (denoted D_n): rotational and reflective symmetry (e..g., polygon). [Note that D_1 has one reflection symmetry axis and no rotational symmetry; e.g., a maple leaf with bilateral symmetry.]
- *continuous rotational symmetry* (denoted $O(2)$): can be rotated about the center by any angle; also has an infinite number of symmetry reflection axes (e.g., circle).

These symmetries may be found on any point set and are not restricted to closed boundaries or figures. Thus, a pair of aligned parallel line segments have a D_2 symmetry.

In terms of 2-D image symmetry analysis, we have implemented and investigated a number of existing symmetry detection algorithms, including rotation symmetry group detection [34]. Figure 4 shows a chaos image with several types of symmetry. The four symmetry sets are shown in Figure 5.

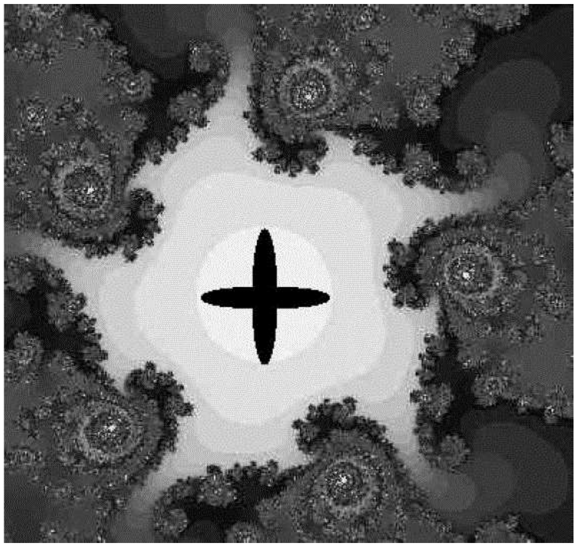
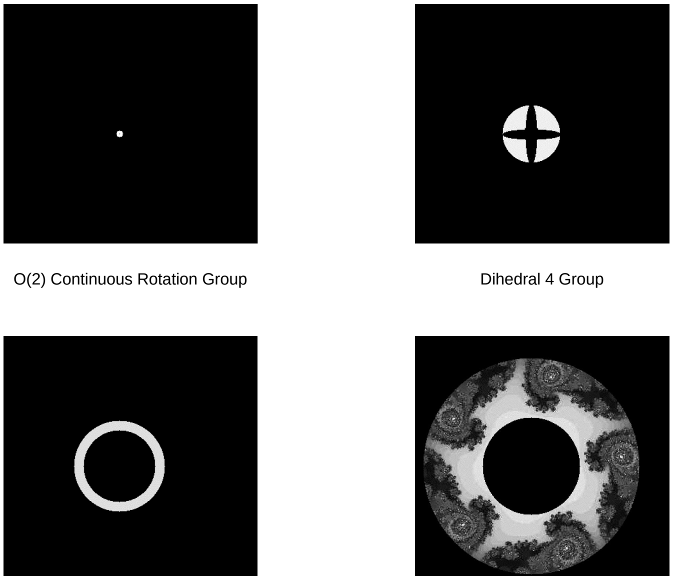


Figure 4: Chaos Image from Lee [34].

However, our contribution to the detection of symmetry in 2-D shapes extends Podolak's planar reflective symmetry transform (PRST) [51]); this method computes a measure of symmetry about each line through every point in a shape. This is a computationally expensive method, and we propose to reduce this cost by choosing a subset of points at which to apply the PRST, as well as the possible orientations. This can be achieved by using the Frieze Expansion Pattern (FEP) [34] which is computed as follows. Pick a point, P, in the shape; for a selected set of orientations, e.g., 1 degree increments from 0 to 360, take that slice of the image and make it a column in the FEP. Figure 6(a,b) shows how the FEP is formed, and the FEP of a square shape. If the FEP is formed at the center of mass of the shape, then the following hold:

- A rotational symmetry exists if the FEP has a reflective axis through the middle row, and the upper half of the FEP image has a translational symmetry (either continuous or discrete).
- For a reflective symmetry axis to exist, it must occur at a maximum or minimum on the upper half shape boundary curve of the FEP and have a max or min at the corresponding location on the lower half shape boundary curve of the FEP.
- Certain features in the 1-D curves found in an FEP can be used to identify the shape basis for a *G-rep*.

These can be robustly determined (see Figure 6(c)); i.e., the point sets do not need to be perfectly symmetric.



O(2) Continuous Rotation Group

Dihedral 4 Group

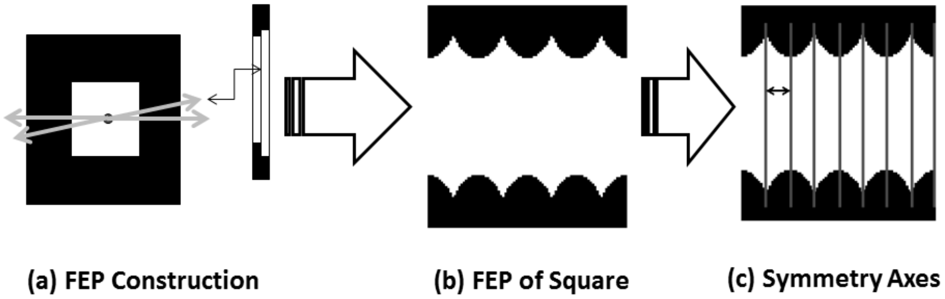
O(2) Continuous Rotation Group

Cyclic 5 Group

Figure 5: Symmetries Found in Chaos Image.

A 2-D reflective symmetry is a set of points in the plane that are invariant under reflection across a symmetry axis line through the set. Podolak's method considers every orientation at every pixel. However, reflective axes can be found as follows: For every segmented object with center of mass CM and FEP F at cm, then if F_1 is the top half shape boundary of F and F_2 is the bottom half shape boundary of F, then let $F\prime$ be F_2 flipped left right and then flipped up-down; next check for translational similarity between F_1 and $F\prime$, and where the similarity is high, there is a reflective axis. Figure 6 (c) shows the detected symmetry axes. Given an FEP, if there are reflective axes, then the shape basis for the full figure must be found between two successive reflective axes. This is shown in Figure 6 (c). In this case for the square, this is any of the half side segments. It is also possible to use the polar image for this analysis.

In addition to 2-D symmetries, shape boundaries may be represented as 1-D signals (e.g., in the FEP), and then analyzed in terms of 1-D symmetries. An example of this is the periodic symmetry in the FEP boundary of a square (see Figure 6 (b)).



(a) FEP Construction (b) FEP of Square (c) Symmetry Axes

Figure 6: Frieze Expansion Pattern (a) Formation, (b) for Square, (c) Symmetry Axes.

2.3 3-D Signals

3-D surface points, homogeneous 2-D surfaces (e.g., planes), and 3-D surface normals may all serve as basic symmetry elements for affordance learning. For example, a flat surface with normal opposite the gravity vector allows platform locomotion. Data from a Kinect or other range sensors allow easy acquisition of such data. We have developed the 3D FEP to detect symmetries in 3D data. For example, Figure 7 shows the FEP for synthetic cube data (expanded at the center), as well as an abstraction of the peaks and pits which are used (in much the same way as maxima and minima in the 2-D FEP) to determine the symmetry planes cutting through the cube (6 diagonal and 3 parallel). That is, any symmetry plane must pass through maxima or minima of the 3-D FEP. Figure 8 shows the FEP for Kinect data of a scene comprised of two corners of a cube viewed from inside the cube. As can be seen,this method works well on real data.

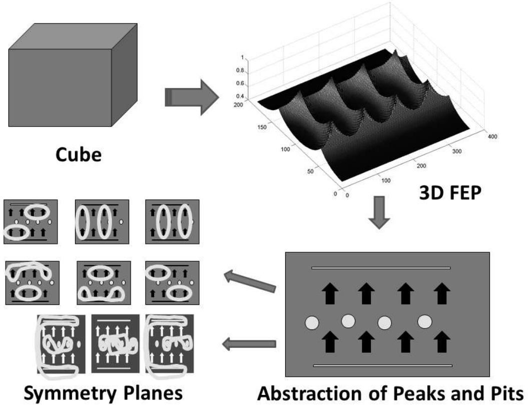
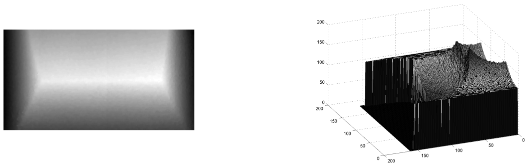


Figure 7: Cube, FEP, and Symmetries.

The 3-D symmetries to be detected are:



Two Corner Kinect Data

Two Corner Peaks in FEP Data

Figure 8: 3D FEP on Real Data.

- *direct isometries* (denoted $SE(n)$): rigid motions; also called the special Euclidean group.
- *indirect isometries* (denoted DR): includes reflections; i.e., D is a direct isometry and R is a reflection.

3 Symmetry Parsing

As a simple example of concept representation, Leyton shows how symmetries can be expressed as symbolic structures which capture not only the perceived layout of a shape, but also to encode how the shape is produced (e.g., put a pen at a point; translate the pen, rotate the pen, translate the pen, etc. to get a generative representation of a square shape). That is, the sensorimotor data is converted into a sequence of symmetry symbols which constitute a string in a language for which syntax and semantics exist. Note that there is evidence that some such form of parsing takes place in the visual system [52]; Poggio et al. describe: ”a class of simple and biologically plausible memory based modules that learn transformations from unsupervised visual experience. The main theorems show that these modules provide (for every object) a signature which is invariant to local affine transformations and approximately invariant for other transformations. [They] also prove that, in a broad class of hierarchical architectures, signatures remain invariant from layer to layer. The identification of these memory-based modules with complex (and simple) cells in visual areas leads to a theory of invariant recognition for the ventral stream.” Bressloff et al. describe symmetry and the striate cortex [8, 9]. Also see [17, 66] as well as early work by Foeldiak [19]. Symmetry is also exploited in various learning paradigms: Ravindran and Barto [55, 56, 57] exploit symmetry in reinforcement learning (see our work also [24]).

Group Representations (G-reps) Given a set of symmetry elements and axes produced by the symmetry detection stage, it is necessary to determine how they are best represented by sequences of symbols in a language. Little detail on this process has been given in the literature. Consider, for example, Leyton's favorite example, the square. While it is true that $Mod \wr T \wr \mathcal{C}_4$ (i.e., a line segment rotated 0,90,180, and 270 degrees) captures the symmetries of the square, it also characterizes a '+' sign. Moreover, there is no association of actuation events required to obtain sensor data for this object. For example, there is control data associated with following the contour of the square either using (actuated) sensors, or by tracing the path with the end effector, and symmetries in these control signals must be parsed and paired with the discovered perceptual symmetries. The resulting *G-rep* will be a description like:

$$T(d = 6cm;\ actuators_{1-3} : [a_{1i}; a_{2i}; a_{3i}], i = 1 \ldots n);$$

$$C_4(\text{90 degrees} : [a_{1i}; a_{2i}; a_{3i}], i = 1 \ldots p)$$

This annotated group sequence gives basic shape information relating length of a side as well as sensorimotor traces in terms of 3 actuators (a_i). Figure 9 shows this schematically. We introduce reflection into the representation since this mirrors the actuation trace required to move along an edge in which velocity starts at 0, increases to a max at the middle, then slows to a stop at the end of the edge segment. Note that it may be more appropriate to specify lengths and angles in terms of sensorimotor sequences of the specific robot in case human defined units are not known. As opposed to a square, a '+' sign will be constructed as two separate strokes: start at a point, make a straight line motion (accelerating and decelerating in reflective symmetry), lifting to the other line segment start point, and again making a linear motion. Although the square and '+' sign share the same symmetries, they are distinguished by their motor sequences. In fact, the square will be more like any other polygon than like a '+' in terms of the actuation sequence. However, symmetry information (including asymmetry) provides a more abstract representation which allows for discrete types of reasoning while retaining a tight grounding to the sensorimotor traces. Thus, a robot can know that two squares are similar in structure, but of different sizes. *G-reps* include information about other physical features like color, weight, material type, etc. One issue not addressed here is the hierarchical nature of complex objects (e.g.,

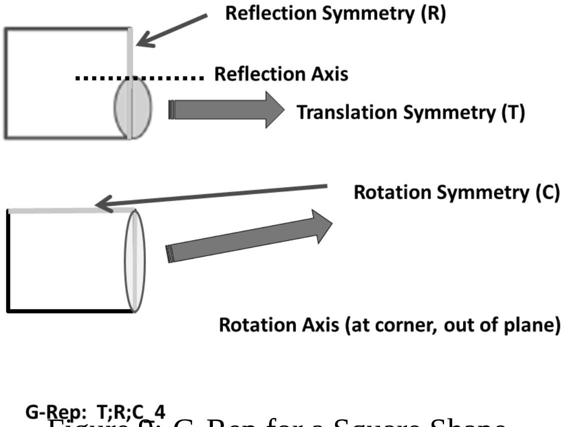


Figure 9: G-Rep for a Square Shape.

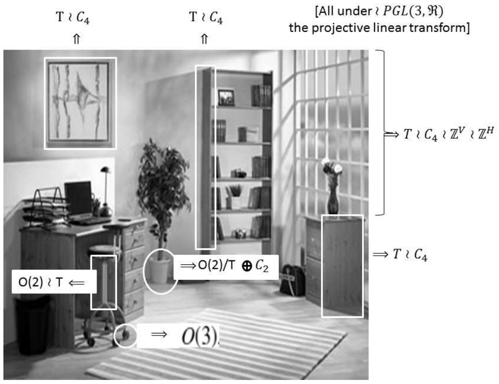

T ≀ C_4
⇑
T ≀ C_4
⇑
[All under ≀ $PGL(3,\Re)$
the projective linear transform]
⟹ $T \wr C_4 \wr \mathbb{Z}^V \wr \mathbb{Z}^H$
⟹ $T \wr C_4$
⟹O(2)/T ⊕ C_2
O(2) ≀ T ⟸
⟹ $O(3)$.

Figure 10: G-Reps Produced for a Scene.

a body has head, torso, legs and arms); however, this is addressed to some extent through the use of the medial axis which provides a characterization of an integral entity (e.g., a human has arms, legs, head, and torso, all related by the various connected segments of the 3D medial axis). The *G-rep* includes the following information:

- Group sequence representation of shape and process entities.
- Sensorimotor sequences of symbols (symmetries or SAX string) associated with entities.
- Medial axis (along with classified characteristic points).
- Properties associated with symmetry elements. This includes not only geometric information, but also semantic information like color, scale parameters, etc. As for the shape itself, the essential characterization can be given in terms of what we call the *shape basis*; this is the smallest part of the shape that informs the reflection and/or rotation symmetries. Figure 11 shows the shape basis circled in the given shapes (the second two shapes share the D_4 symmetry of the first two, but their shape bases are not a simple translation).

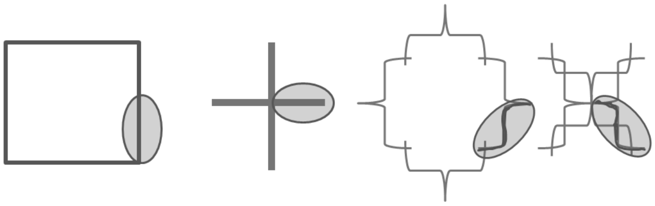


Figure 11: *Shape Basis* for Each of Four Shapes.

The output of the interaction with an environment such as that shown in Figure 10 is to produce symbol sequences (which encode both percept and motor symmetries) for the various entities in the environment.

***G-rep* Grammar (1D)** We next develop an attribute grammar [59] to define the translation semantics from signal values to symmetry symbols. Although currently restricted to 1D signals, this still allows analysis of 2D shapes by encoding their shape boundaries as described from the FEP. The symmetry grammar, $\mathcal{G}_S$, is given as:

1. *context-free syntax*: standard grammar, G, for syntax
2. *semantic attributes*: symbols associated with vocabulary of G
3. *attribute domains*: attribute value sets
4. *semantic functions*: describe how values are produced.

The productions are:
[This is a simplified description of the grammar].

(1) $F \rightarrow S^1S^2\{\mu_1 == \mu_2\}$

(2) $U \rightarrow S^1S^2\{\mu_1 < \mu_2\}$

(3) $D \rightarrow S^1S^2\{\mu_1 > \mu_2\}$

(4) $C \rightarrow F$

(5) $C \rightarrow CF\{constant(C) == constant(F)\}$

(6) $B \rightarrow UD\{slope(U) \approx -slope(D)\}$

(7) $B \rightarrow DU\{slope(D) \approx -slope(U)\}$

(8) $W \rightarrow$ any permutation

(9) $P \rightarrow W^+W^+\{attributes(W^1 \approx attributes(W^2)\}$

(10) $Z \rightarrow C \mid B \mid P$

(11) $R \rightarrow UZD\{slope(U) \approx -slope(D)\}$

(12) $R \rightarrow DZU\{slope(D) \approx -slope(U)\}$

(13) $R \rightarrow FZF\{constant(F^1) \approx constant(F^2)\}$

(14) $R \rightarrow URD\{slope(U) \approx -slope(D)\}$

(15) $R \rightarrow DRU\{slope(D) \approx -slope(U)\}$

(16) $R \rightarrow FRF\{constant(F^1) \approx constant(F^2)\}$

(17) $S \rightarrow R \mid C \mid P$

(18) $A \rightarrow U$

(19) $A \rightarrow AU$

(20) $E \rightarrow D$

(21) $E \to ED$

Note that languages of repeated strings, as in production (11), generally require a context sensitive grammar, but we strict the repeated string's length, and can thus implement an efficient parser. Constant strings can be recognized by FSAs and context free permutation grammars exist (see [3, 11]).

In terms of our implementation, the 1D input string is first processed as described in [37]; i.e., a Piecewise Average ApproXimation (PAA) is found, and from this a Symbolic Aggregate approXimation (SAX). $\mathcal{G}_S$ then parses the SAX string to produce the basic *G-rep*. Figure 12 shows the results of parsing a sine wave $[0, 2 * \pi]$, and Figure 13 shows the parse for a square shape. The symmetry analysis produces the following symmetries for the sine wave:

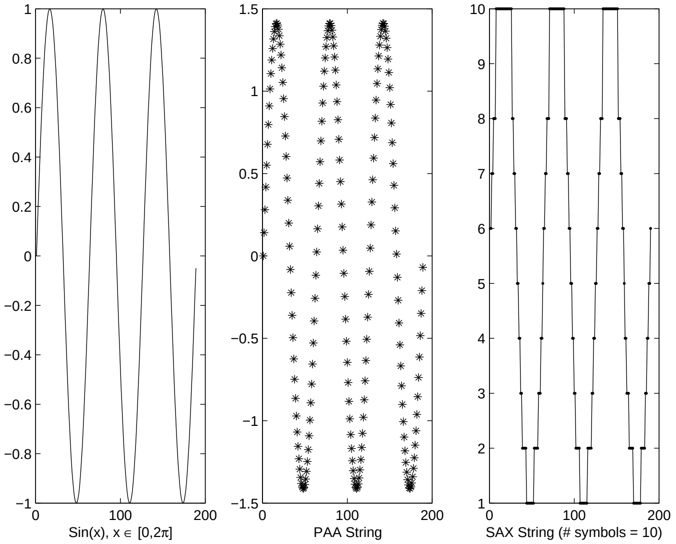


Figure 12: Parse of a Sine Wave.

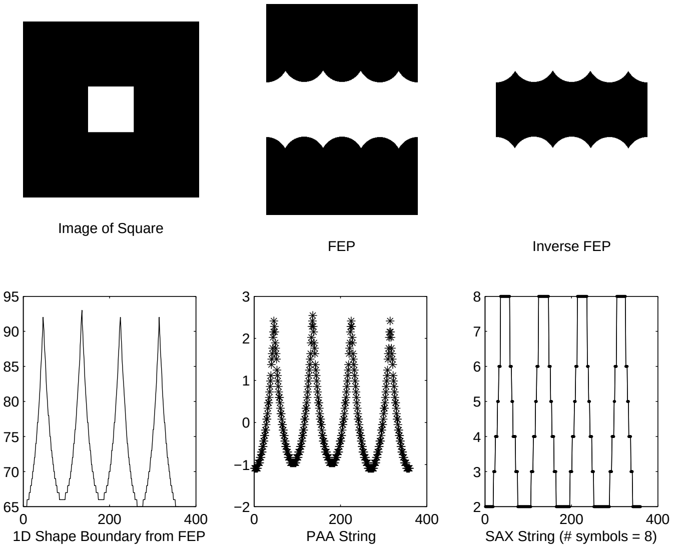


Figure 13: Parse of a Square Shape.

Symmetry Type	*Start Index*	*End Index*	*Basic Length*	*Symmetry Measure*	*Symmetry Index*
periodic	1	189	63	0.7663	
reflective	90	132	21	1.0000	111
reflective	27	69	21	0.9283	48
reflective	2	32	15	0.9239	17
reflective	159	189	15	0.9239	174
reflective	59	101	21	0.7334	80
reflective	121	163	21	0.7334	142

and these for the image of a square:

Symmetry Type	*Start Index*	*End Index*	*Basic Length*	*Symmetry Measure*	*Symmetry Index*
periodic	1	356	89	0.9775	
reflective	2	98	48	0.4063	50
reflective	2	186	92	0.4168	94
reflective	2	276	137	0.3895	139
reflective	3	359	178	0.7410	181
reflective	87	359	136	0.4554	223
reflective	177	359	91	0.4673	268
reflective	265	359	47	0.4072	312

Note that it is the symmetry axes which are important.

4 Symmetry Exploitation

Next we demonstrate two powerful ways to exploit symmetry analysis: (1) sensorimotor reconstruction, and (2) symmetry bundles as robot affordances. (1) is the semantic compilation of 1-D sensor signals into equivalence classes (i.e., determine similar sets of sensors). This allows further analysis to determine spatial layout of sensors, etc. (2) aims to detect simultaneous sensor actuator symmetry sequences that lead to a useful behavior. For example, pure translation for a two-wheeled robot results from constant (actuation) signals to the wheels and results in a vertical translation symmetry in the FEP. These can be grouped to capture the notion of *move forward* and *move backward*. The experiments described here have been performed on a Turtlebot based on an I-Create platform (see Figure 14) equipped with cameras, IR, and a Kinect sensor.

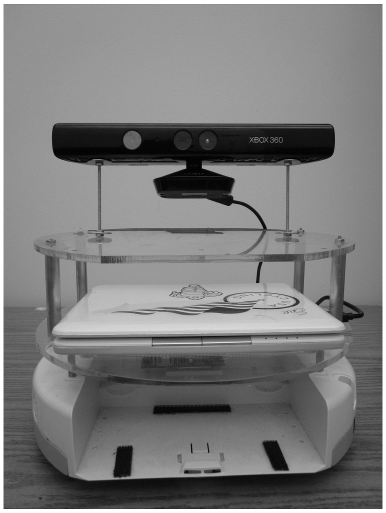


Figure 14: Turtle Robot Platform.

4.1 Sensorimotor Reconstruction

The sensorimotor reconstruction process consists of the following steps: (1) perform actuation command sequences, (2) record sensor data, (3) determine sensor equivalence classes, and (4) determine sensor-actuator relations. An additional criterion is to make this process as efficient as possible.

In their sensorimotor reconstruction process, Olsson, Pierce [45, 50] and others produce sensor data by applying random values to the actuators for some preset amount of time, and record the sensor sequences, and then look for similarities in those sequences. This has several problems: (1) there is no guarantee that random movements will result in sensor data that characterizes similar sensors, (2) there is no known (predictable) relation between the actuation sequence and the sensor values, and (3) the simultaneous actuation of multiple actuators confuses the relationship between them and the sensors.

To better understand sensorimotor effects, a systematic approach is helpful. That is, rather than giving random control sequences and trying to decipher what happens, it is more effective to hypothesize what the actuator is (given limited choices) and then provide control inputs for which the effects are known. Such hypotheses can be tested as part of the developmental process. The basic types of control that can be applied include: none, impulse, constant, step, linear, periodic, or other (e.g., random).

Next, consider sensors. Some may be time-dependent (e.g., energy level), while others may depend on the environment (e.g., range sensors). Thus, it may be possible to classify ideal (noiseless) sensors into time-dependent and time-independent by applying no actuation and looking to see which sensor signals are not constant (this assumes the spatial environment does not change). Therefore, it may be more useful to not actuate the system, and then classify sensors based on their variance properties. That is, in realistic (with noise) scenarios, it may be possible to group sensors without applying actuation at all. The general symmetry transform discovery problem for sensorimotor reconstruction is: Given two sensors, S_1 and S_2, with data sequences T_1 and T_2, find a symmetry operator σ such that $T_2 = \sigma(T_1)$.

Using the symmetries described above, we propose the following algorithms.

Algorithm SBSG: Symmetry-based Sensor Grouping

```
1. Collect sensor data for given period
2. Classify Sensors as Basic Types
3. For all linear sensors
   a. Group if similar regression error
4. For all periodic sensors
  a. Group if similar P and C
5. For all Gaussian sensors
  a. Group if similar signals
```

This algorithm assumes that sensors have an associated noise. Note that this requires no actuation and assumes the environment does not change. Finally, the similarity test for the above algorithm depends on the agent embodiment.

Algorithm SBSR: Symmetry-based Sensorimotor Reconstruction

```
1. Run single actuator and
   collect sensor data for given period
2. For each set of sensors of same type
   a. For each pair
      i. If translation symmetry holds
         Determine shift value
         (in actuation units)
```

This determines the relative distance (in actuation units) between sensors. E.g., for a set of equi-spaced range sensors, this is the angular offset. We have demonstrated this algorithm elsewhere [25].

Any experiment should carefully state the questions to be answered by the experiment and attempt to set up a valid statistical framework. In addition, the sensitivity of the answer to essential parameters needs to be examined. We propose to address grouping correctness: What is the correctness performance of the proposed grouping generator? This requires a definition of correctness for performance and we propose the following (for more details, see [23]):

Correctness Measure: Given (1) a set of sensors, $\{S_i, i = 1 : n\}$ (2) a correct grouping matrix, G, where G is an n by n binary valued matrix with $G(i, j) = 1$ if sensors S_i and S_j are in the same group and $G(i, j) = 0$ otherwise, and (3) H an n by n binary matrix which is the result of the grouping generator, then the grouping correctness measure is:

$$\mu_G(G, H) = \sum_{i=1}^{n} \sum_{j=1}^{n} [(\delta_{i,j})/n^2]$$

$\delta_{i,j} = 1$ if G()==H(); 0 otherwise

We performed experiments with four types of physical sensors: microphone, IR, camera and range (the latter two from a Kinect) to validate the proposed approach. Data was taken for the static case (no actuation). The microphone provided one data stream, the IR was used to make 12 data streams, while the camera and range data were taken from 625 pixel subset in the images. Thus, a total of 1,263 1D sensor data streams were analyzed. Figure 15 shows sample data from a camera and the microphone, as well as their histograms.

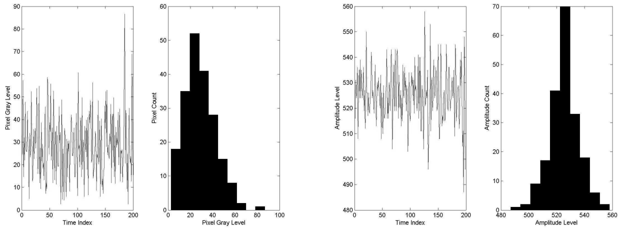


Figure 15: Trace and Histogram of a Pixel Data Stream (left); Trace and Histogram of the Microphone Data Stream (right).

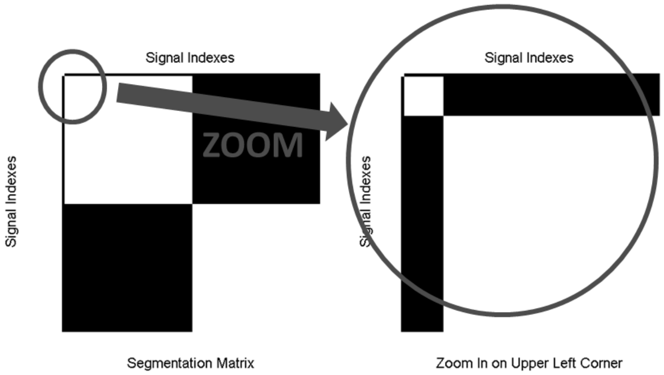


Figure 16: Grouping Matrix (White indicates Similar Sensor).

Figure 16 shows the grouping matrix for similar sensors, $G(i, j) == 1$ means sensors i and j are similar. The left side of the figure shows the 12x12 group of IR sensors (upper left) and the two 625x625 groups of camera and range sensors. The right side of the figure zooms in to show the 1x1 group (upper left) of the microphone sensor. The performance of this grouping depends on a threshold, and we looked at the impact on the correctness measure for a wide range of threshold value. The result is that the grouping correctness measure was above 97% for all threshold values except at the very low and very high end.

4.2 Concept Formation

A low-level concept is formed with the discovery of a coherent set of sensor data exhibiting symmetry. We demonstrate this on real 2D camera data from the robot's vision sensor (see

Figure 17). The first step in this process is to segment the image to obtain object shapes (boundaries) which can then be converted to 1D signals (in polar form) and parsed for symmetries. A simple k-means clustering algorithm on the HSV transform of the original image segments objects based on color, and object boundaries in different clusters are obtained by using the gradient (edge) map of the original image. These 2D boundaries are then converted to polar images from which 1D signals can be extracted to obtain the SAX representation which is the input to the symmetry detector. The symmetry detector successfully finds the periodic and reflective symmetries in 3 out of 4 objects in the image which are shown as vertical red lines in Figure 17. (Due to the simplicity of our image segmentation code, boundaries may not be detected well enough for some objects to obtain good enough 1D SAX signals for symmetry detection; the book that is missed presents two surfaces and this causes a poor segmentation. Implementing robust image segmentation techniques using multisensor data (e.g., range) would help solve this problem, and that is a part of our future work.)

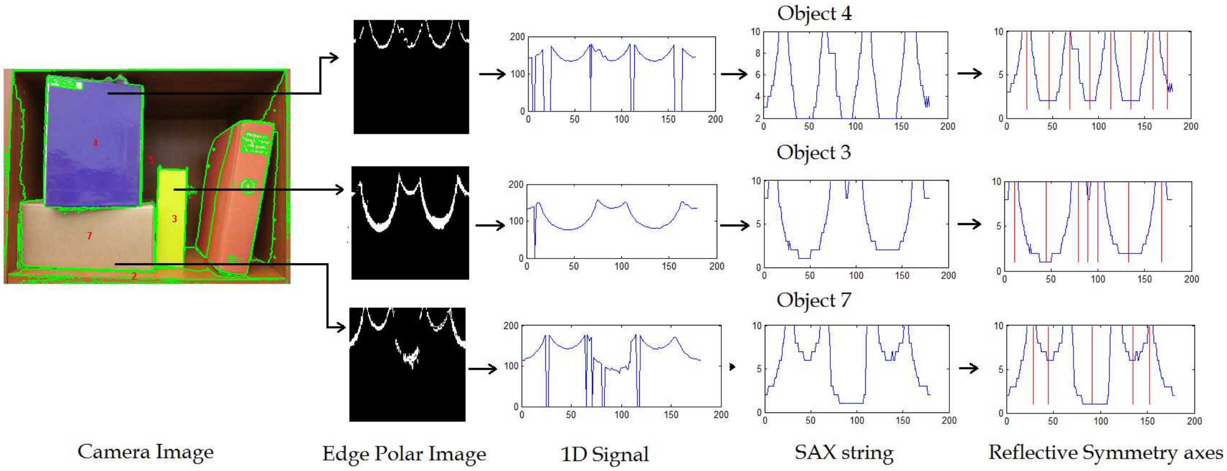


Figure 17: Symmetry Detection in Bookshelf Objects.

The following symmetries were detected in the bookshelf objects corresponding to the numbered segments.

Object 3 *Symmetry Type*	*Start* *Index*	*End* *Index*	*Basic* *Length*	*Symmetry* *Measure*
periodic	1	270	90	0.5000
reflective	2	20	9	0.3497
reflective	2	88	43	0.4241
reflective	68	90	11	0.4152
reflective	2	176	87	0.4627
reflective	90	110	10	0.6862
reflective	87	179	46	0.4759
reflective	157	179	11	0.6412

Object 4 *Symmetry Type*	*Start* *Index*	*End* *Index*	*Basic* *Length*	*Symmetry* *Measure*
periodic	1	270	90	0.6000
reflective	2.0000	42.0000	20.0000	0.5902
reflective	2.0000	90.0000	44.0000	0.6079
reflective	36.0000	102.0000	33.0000	0.4067
reflective	3.0000	179.0000	88.0000	0.6802
reflective	62.0000	164.0000	51.0000	0.3894
reflective	91.0000	179.0000	44.0000	0.4098
reflective	139.0000	179.0000	20.0000	0.5495
reflective	171.0000	179.0000	4.0000	0.5698

Object 7 *Symmetry Type*	*Start* *Index*	*End* *Index*	*Basic* *Length*	*Symmetry* *Measure*
periodic	1	180.0000	90.0000	0.4333
reflective	9.0000	49.0000	20.0000	0.3173
reflective	2.0000	88.0000	43.0000	0.3473
reflective	74.0000	108.0000	17.0000	0.8864
reflective	111.0000	159.0000	24.0000	0.6126
reflective	136.0000	168.0000	16.0000	0.3159

We are able to detect the reflective axes of objects close to the expected angles, as is evident from above results. The reflective symmetry axes detected in the previous step characterize the symmetry of objects; Object 4 is characterized as D_4 while Objects 3 and 7 are determined to be D_2 (note that each object has its own shape basis set, etc.). Although we propose such *G-reps* as direct representations for cognition, such results can also provide advantages to other approaches. For example, the T function of an OAC would benefit from an attribute space augmented with such symmetry descriptors, particularly, during execution of a particular action on an object. Interactions between robot end effectors and world objects can be well defined in terms of actions (*E*) over *T*, and expected outcome (*M*), if certain attributes of those objects (e.g., symmetry groups) are known. Consider the action *"Push a cube in a straight line without effecting a rotation on it"* which requires the robot to push a cube to move it straight without rotating it. Knowing the symmetry axes of the cube allows this whereas pushing at any point away from these axes induces a torque and hence a rotation. Symmetries also provide the basis for structural bootstrapping: if a robot has formed the concept that a Dihedral Group 4 symmetry (D_4) object, like a square, stays invariant under any multiples of 90° rotation about its center of mass, or being flipped about its symmetry axes, the robot can then predict that the result of a similar action carried out on any other object having a D_4 symmetry would be the same. Concepts like these help identify similar or dissimilar objects, and can be used directly as a representation or to augment other approaches (like OACs) and therefore lends itself to learning interactions between robot and the world.

4.3 Symmetry Bundles as Affordances

Once sensorimotor data is converted to symmetry symbol sequences, they must be filtered by the effects that they afford. This may also be keyed to 3D space group symmetry (affine) operations (translation, rotation), and grounded in the particulars of the objects involved. As an example of some simple affordances, consider the following two.

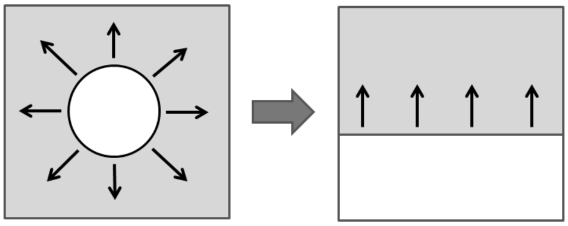


Figure 18: Polar Image Optical Flow Method to Detect Pure Translation.

Translation For our two-wheeled Turtle robot, if constant and equal torques are applied to the two wheel motors, then the motion gives rise to a focus of expansion image in the direction of motion. Figure 18 shows how this produces a columnar translation symmetry in the polar image in that all motion is upward. To determine this, motion direction similarity is used.

Rotation Constant but opposite torques on the two wheels results in a periodic translation symmetry in image or range data. That is, a pixel produces a periodic sequence of values, where the period is related to the rotational speed. Rotation shows up as a translation (periodic) in the polar image.

Thus, by setting up innate mechanisms to look for combinations of symmetric (e.g., constant, linear, periodic, etc.) actuator sequences that result in specific symmetries in the sensor data, the robot will be able to find useful behaviors.

A robot learns affordances as follows: it sends control signals to its actuators which immediately start receiving a stream of sensory signals (e.g., from cameras, odometers, microphones, etc.). It is useful to find relations between these sensory and control signals, and to characterize how one varies with respect to the other when interesting patterns or invariants occur. The sensor-actuator signal sets are processed through the Symmetry Engine (SE) architecture to find invariances, if any, and store them as *symmetry bundles* (see below); the robot can re-use that knowledge should it encounter a similar situation again; this may form the basis for structural bootstrapping [27]. Also, certain sensor signals can be better analyzed if they are first transformed to another representation in which it is easier and more efficient to identify certain forms of invariance.

4.3.1 Symmetry Bundles

A *Symmetry Bundle* is a combination of

1. The sensorimotor or transformed signals of a robot's sensors and actuators.
2. The operator which transforms the signal into a representation where the symmetry exists.
3. The corresponding symmetries observed in the resulting signals.

Sensor/Actuator Signals (S_{ij}) These consist of the 1-D, 2-D or 3-D actuator and sensor signals (samples) produced or received during a specific behavior. Symmetry bundles with no actuation signals are called *actuation-free symmetry bundles*.

Transform Operator (T) In the simple translation behavior described above, T is the transformation of a camera image to the polar image followed by a histogram of the motion direction angles.

Symmetry (Ψ) A symmetry is one of the 1-D, 2-D or 3-D signal symmetries defined above. In the case of the translation behavior, the symmetry for the both left and right wheel actuation would be the same 1-D constant signal, while the symmetry in the polar image would be the constant angle of motion direction (upward in each column).

We now describe in more detail the theory behind the translation behavior scenario. Assume a perspective projection camera model and a differential drive (two-wheeled) robot (see Figure 14) that undergoes various motions (actuations) which cause a change in its video (sensor) signal. We use the perspective projection theory given in [20] and [67]. We now describe in more detail the theory behind the translation behavior scenario. Assume a perspective projection camera model and a differential drive (two-wheeled) robot (see Figure 14) that undergoes various motions (actuations) which cause a change in its video (sensor) signal. We use the perspective projection theory given in [20] and [67]. Figure 19 depicts the perspective projection model and the simplified derivation can be stated as follows.

A point in the world P_w is mapped to a point in the image (x_{im}, y_{im}) as:

$$u - o_x = -f_x \frac{r_{11}X^w + r_{12}Y^w + r_{13}Z^w + t_x}{r_{31}X^w + r_{032}Y^w + r_{33}Z^w + t_z} \tag{1}$$

$$v - o_y = -f_y \frac{r_{21}X^w + r_{22}Y^w + r_{23}Z^w + t_y}{r_{31}X^w + r_{32}Y^w + r_{33}Z^w + t_z} \tag{2}$$

where, s_x, s_y is the pixel size in the horizontal and vertical direction, respectively, $f_x = \frac{f}{s_x}$ is the length in horizontal pixel units, $f_y = \frac{f}{s_y}$ is the length in horizontal pixel units, f is the focal length, and (o_x, o_y) is the image center.

The intrinsic parameters are embedded in (2). Neglecting the radial distortion caused by the lens, we can define the intrinsic and extrinsic transformation matrices M_{int} and M_{ext}, respectively, as

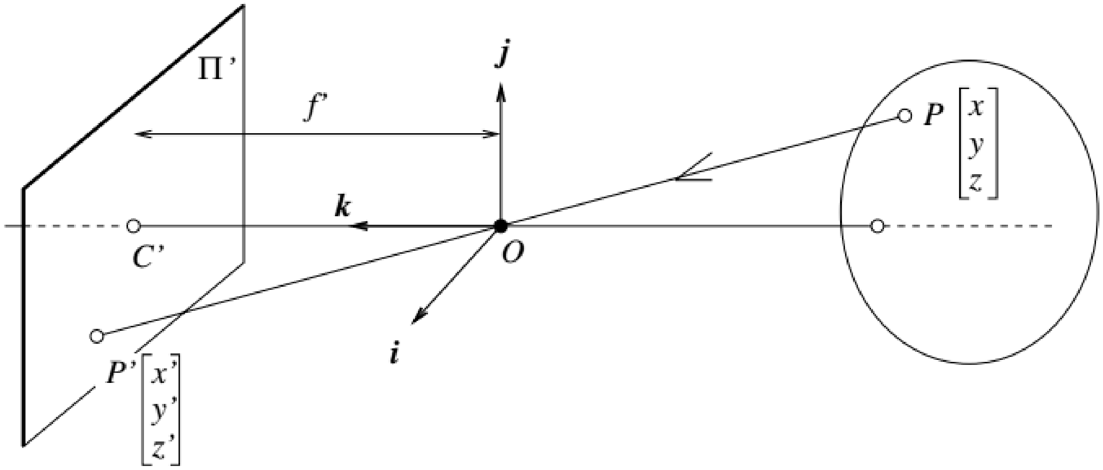

Π'
f'
j
P
$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$
k
C'
O
i
P'
$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$

Figure 19: Camera Model.

$$M_{int} = \begin{pmatrix} -f_x & 0 & o_x \\ 0 & -f_y & o_y \\ 0 & 0 & 1 \end{pmatrix}$$

and

$$M_{ext} = \begin{pmatrix} r11 & r12 & r13 & t_x \\ r21 & r22 & r23 & t_y \\ r31 & r32 & r33 & t_z \end{pmatrix}$$

Perspective projection can now be defined as:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = M_{int} M_{ext} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{3}$$

If $P' = [x,\ y,\ z]^T$ is the image point and $P = [X,\ Y,\ Z]^T$ is the world point, then:

$$x = \frac{f}{Z}X,\ y = \frac{f}{Z}Y \tag{4}$$

4.3.2 A Symmetry Bundle Example: Pure Translational Motion

Here we assume the camera is moving forward in the direction of its optical axis. A symmetry bundle for this motion can be produced as follows.

Actuation and Sensor Signals The actuation signals are constant and small values so that the robot moves forward slowly. The sensor signal is histogram of motion directions in the polar images derived from the sequence of camera images acquired during the motion.

Transform Operator The transform operator is the polar transform (defined above) followed by the angle histogram operation.

Symmetries Three symmetries are found: (1) 1-D constant actuation signal for left wheel, (2) 1-D constant actuation signal for right wheel, and (3) vertical translation for all pixels in the polar image (i.e., similar motion direction angle of $\frac{\pi}{2}$ radians.

We now show that the vertical motion symmetry holds in the polar image for pure translational motion of the camera on the robot. We have from (4),

$x = \frac{f}{Z}X,\ y = \frac{f}{Z}Y$

Assuming that the focal length of the camera f is 1, we have
$x = \frac{X}{Z},\ y = \frac{Y}{Z}$

Therefore, if we move the camera forward by δz, the image points in the new image will be $x' = \frac{x}{z-\delta z},\ y' = \frac{y}{z-\delta z}$.

Note that here we are assuming the camera frame and world frame to be the same, hence the z axis from the optical center to the world reference (and the world point) is positive. Therefore a shift of δz in the direction of the world point should decrease the z value by that amount, given by $z - \delta z$, assuming $\delta z > 0$. The vector representing the movement of the image pixels can be defined as,

$$\begin{pmatrix} \frac{x}{z-\delta z} - \frac{x}{z} \\ \\ \frac{y}{z-\delta z} - \frac{y}{z} \end{pmatrix} = \begin{pmatrix} \frac{x\delta z}{z(z-\delta z)} \\ \\ \frac{y\delta z}{z(z-\delta z)} \end{pmatrix}$$

Its direction is then given by
$\arctan[(\frac{y\delta z}{z(z-\delta z)})/(\frac{x\delta z}{z(z-\delta z)})] \; = \; \arctan\frac{y}{x}$

This means that the motion of each point in the polar image is along the column corresponding to the angle that point makes with the epipole. The amount of movement of the pixel is given by the magnitude of this vector as

$$r = (\sqrt{x^2+y^2})\;(\frac{\delta z}{z(z+\delta z)}) \tag{5}$$

Notice that the movement of each point in the original camera image is along a vector which projects out from the focus of expansion in the image (also called as the *epipole*).

Figure 20 (a,b) shows two images from a translation sequence, while (c) shows the motion vector angle histogram for this pair. As can be seen, the majority of motion vectors are around $\frac{\pi}{2}$ radians.

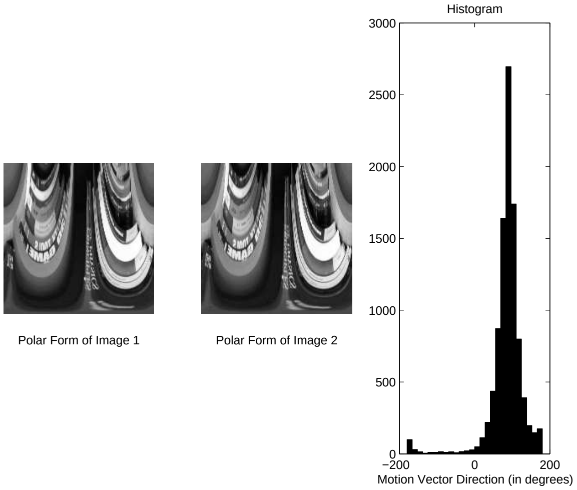


Figure 20: Pure Translation Sequence (a) and (b) and Resulting Motion Vector Angle Histogram (c).

Although we do not exploit it here, note that range segmentation is possible for pure translational motion when the image is converted to log-polar form. The transform from Cartesian coordinates, (x, y), to log-polar, (ρ, θ), can be given as [74]:

$\rho \; = \; \log\sqrt{(x-x_c)^2+(y-y_c)^2}$ is the distance of that point from the center of expansion (x_c, y_c), and

$\theta = tan^{-1}\frac{y-y_c}{x-x_c}$ is the angle.

In case of a forward translating camera, the image points in consecutive images move radially outwards from the center.

We know the distance ρ for this movement. This difference in the radial movement - assuming the epipole is chosen as the center for log-polar transformation - can be derived as follows.

$\rho_1 = \log\sqrt{(\frac{x}{z})^2 + (\frac{y}{z})^2}$ where $\frac{x}{z}$ and $\frac{y}{z}$ is the world point projected onto the image and $\rho_2 = \log\sqrt{(\frac{x}{z-\delta z})^2 + (\frac{y}{z-\delta z})^2}$ where $\frac{x}{z-\delta z}$ and $\frac{y}{z-\delta z}$ is the world point projected onto the image after moving δz distance towards the world point.

The upward shift of this point in the log-polar image can be given as

$\rho_2 - \rho_1 = \log\sqrt{(\frac{x}{z-\delta z})^2 + (\frac{y}{z-\delta z})^2} - \log\sqrt{(\frac{x}{z})^2 + (\frac{y}{z})^2}$

$= \log\sqrt{x^2 + y^2} - \log(z - \delta z) - \log\sqrt{x^2 + y^2} + \log z$

$= \log z - \log(z - \delta z)$

This final value is a constant for all world points having the same z coordinate, and can thus be used to perform range segmentation. This could, in fact, provide a motivation for a robot to select pure translation behavior.

Perspective projection is demonstrated in Figure 21 where the paths followed by different image points are given by the red and green curves. In this experiment, the camera is rotated about its optical axis in the horizontal plane. A point with a greater z-distance (and constant x and y distance) from the optical center will be projected closer to the image center than another non-collinear point which has a smaller z-distance, since z is in the denominator.

4.3.3 Rotational Motion - Y Axis

Using equation (3) we can represent any world point visible to the camera, on the image plane. Assume that initially the camera reference frame (C) and the world reference frame

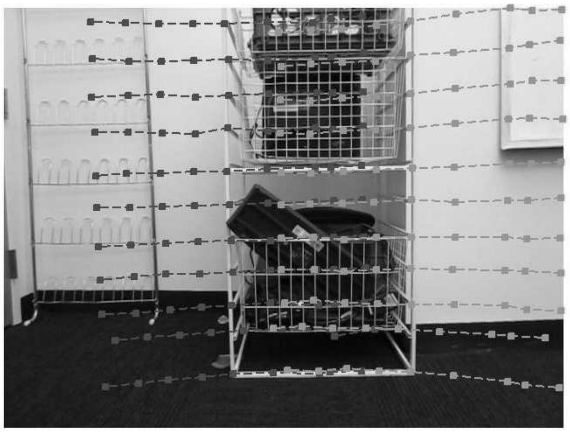


Figure 21: Lab Experiment.

(W) are coincident and aligned, i.e. $W \equiv C$. However, if the camera rotates about the Y axis of W (or C), we have a rotation, say R, applied to the camera frame (Note that there is no translation involved if the camera is rotated about its optical center, since the origin of both the camera frame and the world frame stay at the same position). This gives us

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}_{imagepoint} = R\ M_{int} M_{ext} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}_{worldpoint} \tag{6}$$

We use Euler angles [15] which differ from rotations in the Euclidean space in the way they express the rotations in terms of the moving frame; To rotate frame A to B we can use Euler angles rotation sequence as ${}^{A}_{B}R_{Z'Y'X'}(\alpha, \beta, \gamma)$ also denoted as,

${}^{A}_{B}R \ = \ {}^{A}_{B'}R\, {}^{B'}_{B''}R\, {}^{B''}_{B}R$,

or $R_Z(\alpha)\ R_Y(\beta)\ R_X(\gamma)$

$$= \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ -\sin\alpha & 0 & \cos\alpha \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & \cos\alpha & -\sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \\ 0 & 0 & 1 \end{bmatrix}$$

Since our camera rotates only about the Y axis, we can set $R_Z(\alpha)$ and $R_X(\gamma)$ to the identity which yields
${}^{A}_{B}R_{Z'Y'X'}(\alpha, \beta, \gamma) \ = \ R_Y(\beta)$

Therefore we have, from (6),

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}_{image} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} M_{int} M_{ext} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}_{world} \tag{7}$$

Equation (7) gives the projection of a world point to an image point given that the camera has rotated by an angle α counterclockwise about the Y axis.

An Euler angle rotation about $\mathbf{Z} - \mathbf{Y} - \mathbf{X}$ is equivalent to a Euclidean rotation about the fixed axes taken in opposite order (viz. $\mathbf{X} - \mathbf{Y} - \mathbf{Z}$), so this method can be used to rotate the camera instead of the standard rotation method.

Assuming for the moment that the camera reference frame and world reference frame are aligned, we have $M_{int} = I$ and $M_{ext} = I$ and therefore

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}_{image} = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix}_{world} \tag{8}$$

$$= \begin{pmatrix} X_w \cos\beta - Z_w \sin\beta \\ Y_w \\ X_w \sin\beta + Z_w \cos\beta \end{pmatrix}_{world} \tag{9}$$

Assume $u = \frac{x_1}{x_3}$ and $v = \frac{x_2}{x_3}$, therefore from (9) we have

$$u = \frac{X_w \cos\beta - Z_w \sin\beta}{X_w \sin\beta + Z_w \cos\beta} \tag{10}$$

$$v = \frac{Y_w}{X_w \sin\beta + Z_w \cos\beta} \tag{11}$$

From (10) and (11) we can see that the points that follow a straight line in the image plane (i.e., v = constant) are the points with $Y_w = 0$ assuming the image center as the origin. For these points the fact that Z_w changes does not matter. For all other points the paths - followed by points as the camera rotates - change as shown in Fig. 4, which is generated using simulation and (10) and (11).

The path followed by a point in the image corresponding to a world point ($Y = 0, X = \pm x$), is a straight line, and for all other world points ($Y = \pm y, X = \pm x$), the path becomes

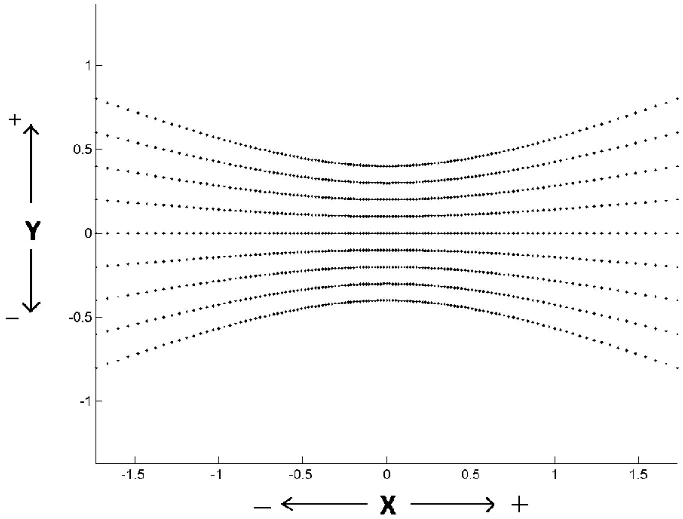


Figure 22: Point motion: As Y values changes from 0 towards positive or negative, the path followed by the point tends towards a parabola. For Y=0 the path is a straight line.

a parabola. This simple illustration allows us to see the invariance for this type of camera rotation; a point ($Y = 0, X = \pm x$) will maintain a Y-axis invariance for a camera rotating about the Y-axis in the XZ-plane.

4.3.4 Rotational Motion - X Axis

For rotation about the X-axis we have the rotation matrix

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

Therefore, from (6) we have the transformation (assuming again that camera reference frame and world reference frame are coincident and aligned)

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}_{image} = \begin{pmatrix} X_w \\ Y_w \cos\alpha - Z_w \sin\alpha \\ Y_w \sin\alpha + Z_w \cos\alpha \end{pmatrix}_{world} \tag{12}$$

and

$$u = \frac{x_1}{x_3} = \frac{X_w}{Y_w \sin\alpha + Z_w \cos\alpha} \tag{13}$$

$$v = \frac{x_2}{x_3} = \frac{Y_w \cos\alpha - Z_w \sin\alpha}{Y_w \sin\alpha + Z_w \cos\alpha} \tag{14}$$

From (13) and (14) we can see that the points that follow a straight line in the image plane (i.e., u = constant) are the points with $X_w = 0$ assuming the image center as the origin.

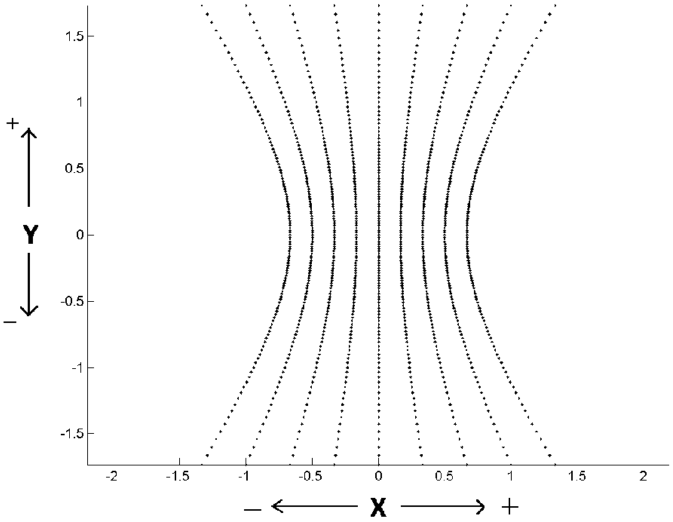


Figure 23: Point motion: As X values changes from 0 towards positive or negative, the path followed by the point tends towards a parabola. For X=0 the path is a straight line.

4.3.5 Rotational Motion - Z Axis

For rotation about the Z-axis we have the rotation matrix

$$R = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore, from (6) we have the transformation (assuming again that camera reference frame and world reference frame are coincident and aligned)

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}_{image} = \begin{pmatrix} X_w \cos\gamma - Y_w \sin\gamma \\ X_w \sin\gamma + Y_w \cos\gamma \\ Z_w \end{pmatrix}_{world} \tag{15}$$

and

$$u = \frac{x_1}{x_3} = \frac{X_w \cos\gamma - Y_w \sin\gamma}{Z_w} \tag{16}$$

$$v = \frac{x_2}{x_3} = \frac{X_w \sin\gamma + Y_w \cos\gamma}{Z_w} \tag{17}$$

From (16) and (17) we can see that Z_w distance remains constant. Assuming $Z_w = c$, the equations for u and v are reduced to the 2-D rotation of a point in the $X - Y$ plane, which effectively rotates a vector about the origin (therefore tracing a circle).

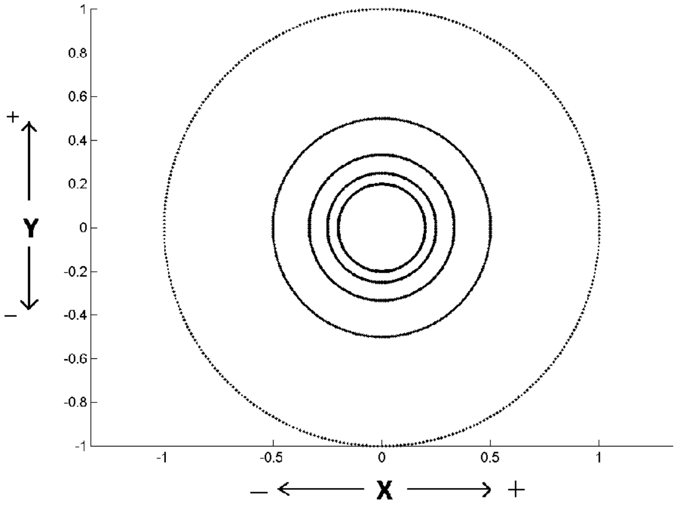
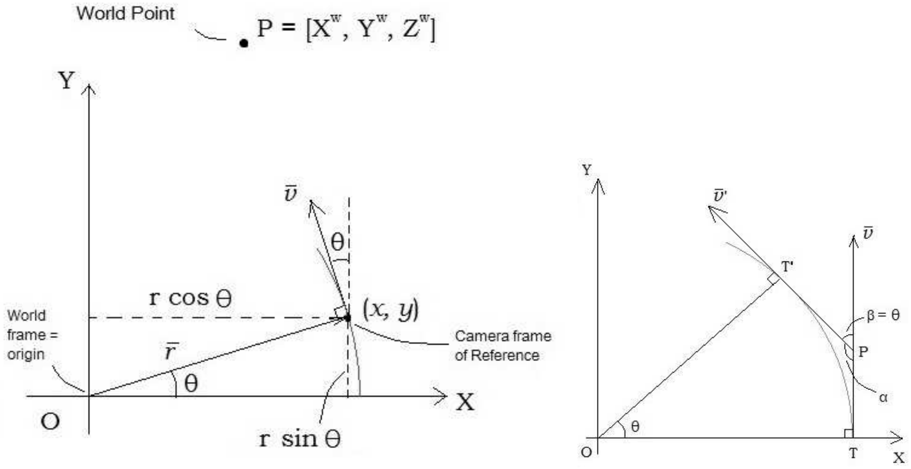


Figure 24: Point motion: As Z increases along with camera rotation about Z-axis, the world point which follows a circular path in the image plane converges to a point.

Thus, rotation of the camera about the Z-axis will result in the points in the image moving in a circle as showin in Fig. 6. As Z increases, because of perspective projection the circle will become smaller and smaller and finally converge to a point.

4.3.6 Circular Motion - Single Time Step (Continuous)

Dudek et al. have explained the theory of a differential drive robot in [18]. Continuous circular motion can be defined using Fig. 7 as follows.



(a) Uniform circular motion.

(b) Angle of rotation.

Figure 25: Circular Motion

Consider the following scenrio: A robot moves along a circular path (shown by the arc). Its direction of velocity at (x, y) is given by $\vec{v}$. Every point (x, y) on this arc is at a distance of r from the center, which is also the origin and coincides with the world reference frame. If $\vec{r}$ is defined as the vector from the origin to any point on the circle, at any angle θ, the coordinates of that point can be given as

$$x = r\ \cos\theta \tag{18}$$

$$y = r\ \sin\theta \tag{19}$$

Therefor $\vec{r}$ can be given by

$$\begin{bmatrix} r\ \cos\theta \\ r\ \sin\theta \end{bmatrix}$$

Note that the point (x, y) happens to be the origin $(0, 0)$ in the camera reference frame.

Fig. 8 shows the camera, initially at T in the tangential direction $\vec{v}$, move on the circumference of a circle to point T' in the tangential direction $\vec{v'}$. It can be easily proven that it has rotated anti-clockwise by an angle θ; Consider the polygon $OTPT'$. We know that the angles of this polygon sum up to $360°$, therefore $\angle TPT' = 180 - \theta$. We also know that $\alpha + \beta = 180$ and that $\alpha = \angle TPT'$. Therefore $\beta = 180 - \alpha = 180 - (180 - \theta) = \theta$

Fig. 9 gives a 3D illustration of the circular motion as one coordinate frame (the camera frame O^C) moving with respect to another (the world frame O^W). Initially O^C is at a distance of r with respect to O^W on the X^W axis. O^C the moves along the circumference of a circle with radius r and O^W as its center. [Since the O^C frame is the camera frame, conventionally, the $-Z^C$ axis points towards the world points.]

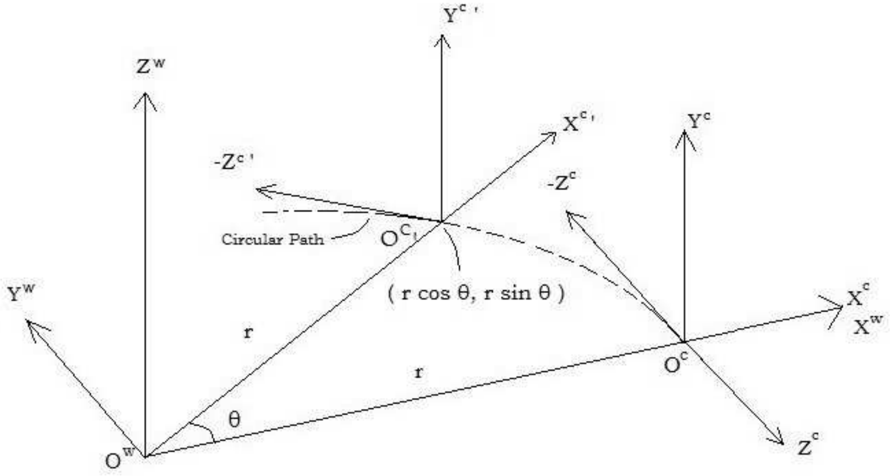


Figure 26: Uniform circular motion.

Since we have the world point in the fixed frame O^W, we can use the perspective projection equation after aligning O^W with the camera frame O^{C}’. To achieve this we need the folllowing set of rotations and a translation

1) Anti-clockwise rotation of O^W about the X^W axis (giving us $O^{W'}$ which aligns with O^C) by an angle $\frac{\pi}{2}$ and translation by $\vec{r}$ to coincide with the origin $O^{C'}$, so that the Z^W axis coincides with Z^C.

2) Anti-clockwise rotation of $O^{W'}$ by and angle θ about the $Y^{W'}$ axis to give us $O^{W''}$.

The transform matrix for the rotation from $O^W \rightarrow O^C$ can be given by

$$R_{X^W}(-\tfrac{\pi}{2}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\frac{\pi}{2}) & \sin(\frac{\pi}{2}) \\ 0 & -\sin(\frac{\pi}{2}) & \cos(\frac{\pi}{2}) \end{bmatrix}$$

and the translation can be given as

$T_x = -r\cos(\theta)$
$T_y = -r\sin(\theta)$
$T_z = 0$

So the combined transform matrix for this transformation would be

$$[R\,T]_X = \begin{bmatrix} 1 & 0 & 0 & -r\cos(\theta) \\ 0 & \cos(\frac{\pi}{2}) & \sin(\frac{\pi}{2}) & -r\sin(\theta) \\ 0 & -\sin(\frac{\pi}{2}) & \cos(\frac{\pi}{2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

The second rotation giving O^{W}' $\rightarrow O^{W}$" can be given as

$$R_{YW'}(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

And since we are not translating in step 2), the translation can be given as
$T_x = 0$
$T_y = 0$
$T_z = 0$

Hence the combined transform matrix can be given as

$$[R\,T]_Y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \end{bmatrix} \tag{21}$$

A world point P can then be transformed using

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = [R\,T]_Y\ [R\,T]_X \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{22}$$

and the image coordinates $u,\ v$ can be given (by perspective projection) as

$$u = \frac{x_1}{x_3}, \qquad v = \frac{x_2}{x_3} \tag{23}$$

Equation (22) and (23) gives us the following u and v:

$$
\begin{aligned}
u &= \frac{(X - r\cos\theta)\cos\theta - Y\sin\theta}{-(X - r\cos\theta)\sin\theta - Y\cos\theta} \\
v &= \frac{Z - r\sin\theta}{-(X - r\cos\theta)\sin\theta - Y\cos\theta}
\end{aligned}
\tag{24}
$$

and

$$
\frac{u}{v} = \frac{(X - r\cos\theta)\cos\theta - Y\sin\theta}{Z - r\sin\theta} \tag{25}
$$

Consider a point directly in front of the camera (i.e., on the optical axis at some finite distance from the optical center). This point - $P^W = [X^W, Y^W, Z^W]^T$ - will have its world coordinates as $P^W = [r, Y^W, 0]^T$, where

$1)$ $X^W = r$, since the point is at the same distance along X-axis as the camera
$2)$ Y^W is at some finite distance from the optical center of the camera
$3)$ $Z^W = 0$ since the point is directly in front of the camera
$4)$ $-Z^C$ is the Z coordinate of the point in camera frame

As the camera moves in a circular fashion, a world point $P^W = [r, Y^W, 0]^T$ traces the following path in the image plane (Fig. 7).

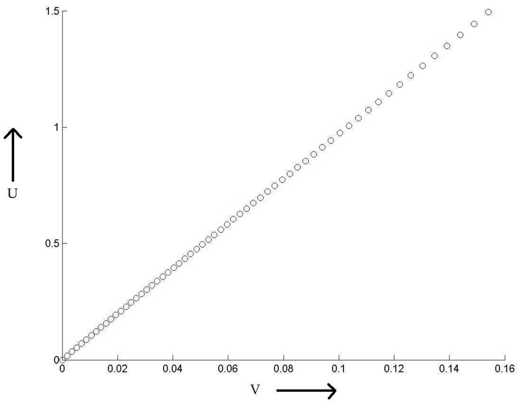


Figure 27: Path traced by a world point in the image plane (V axis is scaled).

In this case, equation (25) becomes

$$\frac{u}{v} = \frac{(r - r\cos\theta)\cos\theta - Y^W \sin\theta}{-r\sin\theta} \tag{26}$$

When $\theta \to 0$, $\cos\theta \to 1$ and we can write (26) as

$\lim_{\theta\to 0} \frac{(r-r\cos\theta)\cos\theta - Y\sin\theta}{-r\sin\theta}$

$= \lim_{\theta\to 0} \frac{-Y^W \sin\theta}{-r\sin\theta} = \frac{Y^W}{r}$

When $\theta \to \frac{\pi}{2}$, $\cos\theta \to 0$, $\sin\theta \to 1$ and we can write (26) as

$\lim_{\theta\to\frac{\pi}{2}} \frac{(r-r\cos\theta)\cos\theta - Y^W\sin\theta}{-r\sin\theta}$

$= \lim_{\theta\to\frac{\pi}{2}} \frac{-Y^W}{-r} = \frac{Y^W}{r}$

When $\theta \to \frac{\pi}{4}$, $\cos\theta = \sin\theta = 0.7071 = \alpha\ (constant)$ and we can write (26) as

$\lim_{\theta\to\frac{\pi}{4}} \frac{(r-r\alpha)\alpha - Y^W\alpha}{-r\alpha} = \lim_{\theta\to\frac{\pi}{4}} \frac{(r-r\alpha)-Y^W}{-r}$

Substituting the value of α we get

$\frac{u}{v} = \frac{(r-r(0.7071))-Y^W}{-r}$

If we assume $r << Y^W$ and $0.7071 << Y^W$, we have $\frac{u}{v} \approx \frac{Y^W}{r}$

We can see that given a point on an optical axis of a camera directly in front of it, the line along which the point moves as the camera rotates in circular fashion has a slope $(\frac{v}{u})$ that can be given by the ratio of the radius of the circle to the world Y distance, $\frac{Y^W}{r}$.

The following experiment (Fig. 8) shows how the said point behaves when the camera is moved along the circumference of a circle.

5 Conclusions and Future Work

We propose symmetry theory as a basis for sensorimotor reconstruction in embodied cognitive agents and have shown that this allows the identification of structure with simple and

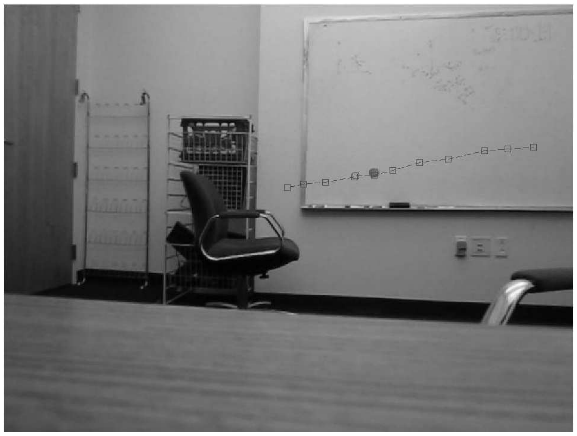


Figure 28: Path traced by a world point in the image plane.

elegant algorithms which are very efficient. The exploitation of noise structure in the sensors allows unactuated grouping of the sensors, and this method works robustly for physical sensor data. Symmetry bundles are also proposed as an approach for affordance discovery.

Several directions remain to be explored:

Structural Bootstrapping Once *G-reps* can be synthesized for affordance, then bootstrapping can be accomplished as follows. Given a *G-rep* with group sequence $G_1 \wr G_2 \wr \ldots \wr G_i \wr \ldots \wr G_n$, then it is abstractly the case that any group equivalent entity to G_i may be substituted in its place as a hypothesized new *G-rep*: $G_1 \wr G_2 \wr \ldots \wr G_e \wr \ldots \wr G_n$. Of course, this will need to be checked in the real world. E.g., a young child knows that it can get into a full-sized car; when presented with a toy car, the child may try to get into it, not realizing that there is a problem with scale. We plan to explore these issues in conjunction with colleagues working on OAC's. Moreover, as pointed out earlier, symmetries can serve as strong semantic attributes in learning OAC prediction functions.

Evolving Communication Mechanisms G-reps provide physical grounding for a robot; i.e., a link between internal categories and the external world. In order to achieve social symbol grounding (see Cangelosi [12]), robots must agree to some shared symbols and their meaning. Schulz et al. [60] propose Lingodroids as an approach to this, and describe experiments in which a shared language is developed between robots to describe places (toponyms) and their relationships. Speakers and microphones are used for communication, and good success was achieved. We propose to apply this method to attempt to have robots develop a shared language for G-reps and behaviors. In particular, we will explore a

What is this? game in which robots will exchange G-reps for specific objects or behaviors (e.g., move straight forward) based on their individual G-reps. Measures of success can be based on the ability to perform the requested behaviors, or to trace or circumnavigate specific objects.

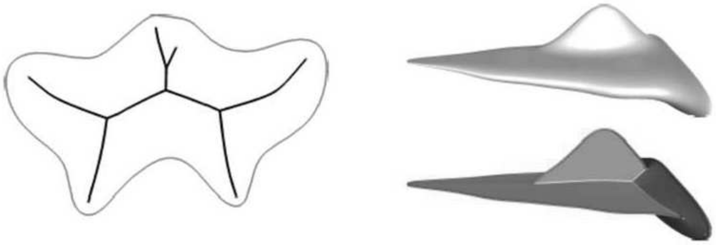


Figure 29: The complete description of the medial axis structure was defined by Giblin[21]. Our algorithm computes all critical points and characterizes them. On the left the creation points are the endpoints of the medial axis, while the junction points are where three curves of the medial axis meet. The curves of the medial axis are traced using an evolution vector field. No offsets or eikonal flows were computed. On the right the visible key points are the where the boundary of the medial axis is closest to the object boundaries, fin points (the ends of the junction curves) and 6-junction points, where junction curves meet. The bounding crest curves are traced, the junction curves and the sheets are traced with the algorithm's evolution vector fields as functions of time. No eikonal offsets are computed.

Symmetry Axes Although generally not explicit in sensor data, symmetry axes are also important cognitive features. Blum introduced the medial axis transform in [4], and much subsequent work has been done in terms of algorithms for its determination (also see Brady [7]). The medial axis gives the morphology of a 3D object and can be used to determine the intrinsic geometry (thickness) of both 2D and 3D shapes. Since it is lower dimensional than the object, it can be used to determine both symmetry and asymmetry of objects. In previous work our colleagues have obtained results on tracking the distance between a moving point and a planar spline shape [14, 64], and computed planar Voronoi diagrams between and within planar NURBS curves[63] (see Figure 29). However, in continuing the search for methods that allow us to characterize the correct topology as well as shape of the planar and 3D medial axis, an approach is developed that used mathematical singularity theory to compute all ridges on B-spline bounded surfaces of sufficient smoothness[42], and then extended the results to spline surfaces of deficient smoothness[41] and also to compute ridges of isosurfaces of volume data[44]. Most recently this approach has been extended to compute the interior medial axis of regions in R^3 bounded by tensor product parametric B-spline surfaces[43]. The generic structure of the 3D medial axis is a set of smooth surfaces along with a singular set consisting of edge curves, branch curves, fin points and six junction points. We plan to exploit these methods to determine topological and metrical symmetries. Although useful for a number of applications, one of high importance here is for grasp planning; colleagues in the European Xperience project team (Przybylski et al. [54] have recently developed a grasp planning method based on the medial axis transform.

6 Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1021038.

References

[1] N. Adermann. Connecting Symbolic Task Planning with Motion Control on ARMAR-III using Object-Action Complexes. Master's thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, August 2010.

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

[3] P.R.J. Asveld. Generating All Permutations by Context-Free Grammars in Chomsky Normal Form. *Theoretical Computer Science*, 354(1):118–130, 2006.

[4] H. Blum. A Transformation for Extracting New Descriptors of Shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, Cambridge, Ma., 1967. MIT Press.

[5] B. Bolder, H. Brandl, M. Heracles, H. Janssen, I. Mikhailova, J. Schuedderich, and C. Goerick. Expectation-driven Autonomous Learning and Interaction System. In *Proceedings of the IEEE Conference on Humanoid Robots*, Daejeon, Korea, December 2008. IEEE.

[6] C. Goerick B. Bolder, H. Janssen, M. Gienger, H. Sugiura, M. Dunn, I. Mikhailova, T. Rodemann, H. Wersing, and S. Kirstein. Towards Incremental Hierarchical Behavior Generation for Humanoids. In *Proceedings of the IEEE Conference on Humanoid Robots*, Pittsburgh, PA, November - January 2007. IEEE.

[7] M. Brady and H. Asada. Smoothed Local Symmetries and their Implementation. Technical Report AIM-757, MIT, Cambridge, MA, 1984.

[8] P.C. Bressloff, J.D. Cowan, M. Golubitsky, P.J. Thomas, and M.C. Wiener. Geometric Visual Hallucinations, Euclidean Symmetry and the Functional Architecture of striate Cortex. *Philosophical Trans. Royal soc London, B Biol Sci*, 356(1407):299–330, 2001.

[9] P.C. Bressloff, J.D. Cown, M. Golubitsky, P.J. Thomas, and M.C. Wiener. What Geometric Visual Hallucinations Tell Us about the Visual Cortex. *Neural Computation*, 14:473–491, 2002.

[10] F. Bullo and R. M. Murray. Proportional Derivative (PD) Control On The Euclidean Group. In *Proceeding of the European Control Conference*, pages 1091–1097, 1995.

[11] R.D. Cameron. Extending Context-Free Grammars with Permutation Phrases. Technical Report CMPT TR 92-07, Simon Fraser University, Burnaby, BC, Canada, 1992.

[12] A. Cangelosi. The Grounding and Sharing of Symbols. *Pragmatics and Cognition*, 14(2):275–285, 2006.

[13] P-C Chen, J. Hays, S. Lee, M. Park, and Y. Liu. A Quantitative Evaluation of Symmetry Detection Algorithms. Technical Report CMU-RI-TR-07-36, Carnegie-Mellon University, Robotics Institute, Pittsburgh, PA, September 2007.

[14] X. Chen, E. Cohen, and R.F. Riesenfeld. Tracking Point-Curve Critical Distances. In *Geometric Modeling and Processing (GMP) 2006, Lecture Notes in Computer Science, V. 4077*, pages 87–100, Berlin, 2006. Springer Verlag.

[15] J.J. Craig. *Introduction to Robotics*. Pearson, NY, NY, 2004.

[16] S. Derrode and F. Ghorbel. Shape Analysis and Symmetry Detection in Gray-level Objects using the Analytical Fourier-Mellin Representation. *Signal Processing*, 84(1):25–39, 2004.

[17] J.J. DiCarlo and D.D. Cox. Untangling Invariant Object Recognition. *Trends in Cognitive Sciences*, 11(8):333–341, 2007.

[18] G. Dudek and M. Jenkin. *Computational Principles of Robotics*. Cambridge University Press, Cambridge, UK, 2000.

[19] P. Földiák. Learning Invariance from Transformation Sequences. *Neural Computation*, 3:194–200, 1991.

[20] D.A. Forsyth and J. Ponce. *Computer Vision*. Prentice Hall, Upper Saddle River, NJ, 2003.

[21] P. Giblin and B. Kimia. A Formal Classification of 3D Medial Axis Points and Their Local Geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):238–251, 2004.

[22] R. Grupen and T.C. Henderson. Apparent Symmetries in Range Data. *Pattern Recognition Letters*, 7:107–111, 1988.

[23] T. Henderson and H. Peng. A study of pierce's group generator. Technical Report UUCS-10-001, The University of Utah, December 1984.

[24] T.C. Henderson, Y. Fan, A. Alford, E. Grant, and E. Cohen. Innate Theories as a Basis for Autonomous Mental Development. Technical Report UUCS-09-004, The University of Utah, October 2009.

[25] T.C. Henderson, Anshul Joshi, and Eddie Grant. From Sensorimotor Data to Concepts: The Role of Symmetry. Technical Report UUCS-12-012, University of Utah, School of Computing, Salt Lake City, UT, October 2012.

[26] T.C. Henderson, Hongchang Peng, Christopher Sikorski, Nikhil Deshpande, and Eddie Grant. Symmetry: A Basis for Sensorimotor Reconstruction. Technical Report UUCS-11-011, University of Utah, School of Computing, Salt Lake City, UT, May 2011.

[27] Thomas C. Henderson, Elaine Cohen, Edward Grant, M.T. Draelos, and Nikhil 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.

[28] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Symmetry Descriptors and 3D Shape Matching. In *Proceedings of Eurographics Symposiuam on Geometry Processing*, pages 115–123, Nice, France, July 2004. ACM.

[29] Michael Kazhdan, Bernard Chazelle, David Dobkin, Thomas Funkhouser, and Szymon Rusinkiewicz. A Reflective Symmetry Descriptor for 3D Models. *Algorithmica*, 38(1):201–225, October 2003.

[30] J.C. Kinsey and L.L. Whitcomb. Adaptive Identification on the Group of Rigid Body Rotations. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3256–3261. IEEE Press, 2005.

[31] J.L. Krichmar and G. Edelman. Principles Underlying the Construction of Brain-Based Devices. In *Proceedings*, pages 37–42, Bristol, UK, 2006. Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

[32] 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.

[33] S. Lee, R. Collins, and Y. Liu. Rotation Symmetry Group Detection via Frequency Analysis of Frieze-Expansions. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[34] S. Lee and Y. Liu. Skewed Rotation Symmetry Group Detection. *IEEE-T Pattern Analysis and Machine Intelligence*, 32:1659–1672, September 2010.

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

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

[37] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In *ACM DMKD*, San Diego, CA, June 2003.

[38] Y. Liu and R. Popplestone. A Group Theoretic Formalization of Surface Contact. *International Journal of Robotics Research*, 13(2):148–161, 1994.

[39] Gareth Loy and Jan olof Eklundh. Detecting Symmetry and Symmetric Constellations of Features. In *ECCV*, pages 508–521, 2006.

[40] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini. Developmental Robotics: A Survey. *Connection Science*, 15(4):151–190, 2003.

[41] S. Musuvathy and E. Cohen. Extracting Principal Curvature Ridges from B-Spline Surfaces with Deficient Smoothness. In *Proc. of International Symposium on Visual Computing, Lecture Notes in Computer Science, v. 5875*, pages 101–110, 2009.

[42] S. Musuvathy, E. Cohen, J. Damon, and J.-K. Seong. Principal Curvature Ridges and Geometrically Salient Regions of Parametric B-Spline Surfaces. *Computer Aided Design (CAD)*, 43:756–770, July 2011.

[43] S. Musuvathy, E. Cohen, and J.N. Damon. Computing Medial Axes of Generic 3D Regions Bounded by B-spline Surfaces. *Computer Aided Design (CAD)*, to appear, 2012.

[44] S. Musuvathy, T. Martin, and E. Cohen. Ridge Extraction from Isosurfaces of Volumetric Data using Implicit B-splines. In *Shape Modeling International Conference*, pages 163–174, Aix-en-Provence, France, June 2010.

[45] L. Olsson, C.L. Nehaniv, and D. Polani. From Unknown Sensors and Actuators to Actions Grounded in Sensorimotor Perceptions. *Connection Science*, 18:121–144, 2006.

[46] J.K. O'Regan and A. Noë. A Sensorimotor Account of Vision and Visual Consciousness. *Behavioral and Brain Sciences*, 24:939–1031, 2001.

[47] M. Park, S. Lee, P-C Chen, S. Kashyap, A.A. Butt, and Y. Liu. Performance Evaluation of State-of-the-Art Discrete Symmetry Detection Algorithms. In *Proceedings of the 6th International Conference on Computer Vision and Pattern Recognition*, June 2008.

[48] D. Philipona, J.K. O'Regan, and J.-P. Nadal. Is There Something Out There? Inferring Space from Sensorimotor Dependencies. *Neural Computation*, 15(9):2029–2049, 2003.

[49] D. Philipona, J.K. O'Regan, J. p. Nadal, O. J. m, and D. Coenen. Perception of the Structure of the Physical World Using Unknown Multimodal Sensors and Effectors. In *Advances in Neural Information Processing Systems*, pages 945–952. MIT Press, 2003.

[50] D.M. Pierce. *Map Learning with Uninterpreted Sensors and Effectors*. PhD thesis, University of Texas, Austin, Austin, Texas, May 1995.

[51] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser. A Planar-Reflective Symmetry Transform for 3D Shape. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3), July 2006.

[52] T. Poggio, J. Mutch, J. Leibo, and L. Rosasco. The Computational Magic of the Ventral Stream. In *Nature Preceedings*, December 2011.

[53] R. Popplestone and R. Grupen. Symmetries in World Geometry and Adaptive Behaviour. In *Proceedings of the Workshop on Algebraic Frames for the Perception Action Cycle*, pages 269–283, Kiel, Germany, 2000.

[54] M. Przybylski, T. Asfour, and R. Dillmann. Planning Grasps for Robotic Hands using a Novel Object Representation based on the Medial Axis Transform. In *Proceedings of the International Conference on Intelligent Robots and Systems*, San Francisco, CA, September 2011. IEEE.

[55] B. Ravindran. *An Algebraic Approach to Abstraction in Reinforcement Learning*. PhD thesis, University of Massachusetts, Amherst, MA, February 2004.

[56] B. Ravindran and A.G. Barto. Symmetries and Model Minimization in Markov Decision Processes. Technical Report CMPSCI TR 01-43, University of Massachusetts, Amherst, MA, September 2001.

[57] B. Ravindran and A.G. Barto. Model Minimization in Hierarchical Reinforcement Learning. In *Proceedings of the 5th International Symposium on Abstraction, Reformulation and Approximation*, pages 196–211, London, UK, 2002. Springer-Verlag.

[58] S. Rebhan, F. Roehrbein, J. Eggert, and E. Koerner. Attention Modulation using Short and Long Term Knowledge. In *Proceedings of the 6th International Conference on Computer Vision Systems*, Lecture Notes on Computer Science, pages 151–160. Springer, 2008.

[59] S.C. Reghizzi. *Formal Languages and Compilation*. Springer Verlag, NY, NY, 2009.

[60] R. Schulz, A. Glover, M.J. Milford, G. Wyeth, and J. Wiles. Ligodroids: Studies in Spatial Cognition and Language. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011. IEEE Press.

[61] J.M. Selig. Lie Groups and Lie Algebras in Robotics. In J. Byrnes, editor, *Proceedings of the NATO Advanced Study Institute on Computational Noncommutative Algebra and Applications*, Il Ciocco, Italy, 2004. Kluwer.

[62] J.M. Selig. *Geometric Fundamentals of Robotics*. Springer, Berlin, 2005.

[63] J-K Seong, E. Cohen, and G. Elber. Voronoi Diagram Computations for Planar NURBS Curves. In *Solid and Physical Modeling Symposium*, pages 67–77, 2008.

[64] J.-K. Seong, D.E. Johnson, and E. Cohen. A Higher Dimensional Formulation for Robust and Interactive Distance Queries. In *Proceedings of ACM Solid and Physical Modeling Conference (SPM) 2006*, pages 197–205, 2006.

[65] J.F. Soechting and M. Flanders. Moving in Three-Dimensional Space: Frames of Reference, Vectors, and Coordinate Systems. *Annual Reviews on Neuroscience*, 15:167–191, 1992.

[66] S.M. Stringer, G. Perry, E.T. Rolls, and J.H. Proske. Learning Invariant Object Recognition in the Visual System with Continuous Transformations. *Biol. Cybernetics*, 94:128–142, 2006.

[67] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Upper Saddle River, NJ, 1998.

[68] D. Vernon. Enaction as a Conceptual Framework for Developmental Cognitive Robotics. *Journal of Behavioral Robotics*, 1(2):89–98, 2010.

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

[70] M.A.G. Viana. *Symmetry Studies*. Cambridge University Press, Cambridge, UK, 2008.

[71] J. Weng and I. Stockman. Autonomous Mental Development: Workshop on Development and Learning. *AI Magazine*, 23(2), 2002.

[72] H. Wersing and E. Koerner. Learning Optimized Features for Hierarchical Models of Invariant Object Recognition. *Neural Computation*, 15(7):1559–1588, 2003.

[73] H. Weyl. *Symmetry*. Princeton University Press, Princeton, NJ, 1952.

[74] G. Wolberg and S. Zokai. Robust Image Registration using Log-Polar Transform. In *Proceedings of International Conference on Image Processing*, pages 493–496, Vancouver, BC, September 2008.