# A Sensorimotor Approach to Concept Formation using Neural Networks

Tanya Beall and Thomas C. Henderson

***Abstract*— We propose an active perception paradigm which combines actuation (control signals) and perception (sensor signals) to form concepts of shape using recurrent neural networks; this representation characterizes not only what the shape is, but also how it is created. The approach is based on the group theoretic wreath product which specifies a sequence of actions on a set of points which when completed comprise the shape. Leyton originally proposed the use of wreath products for concept representation (see [6]). Wreath product descriptions provide an abstract generative representation of shape, but can be annotated for specific actuation systems; this provides a mechanism for knowledge transfer across different motor systems (e.g., visual vs. arm control). We describe how wreath products can be implemented as recurrent neural networks, and demonstrate their application to shape recognition.**

## I. INTRODUCTION

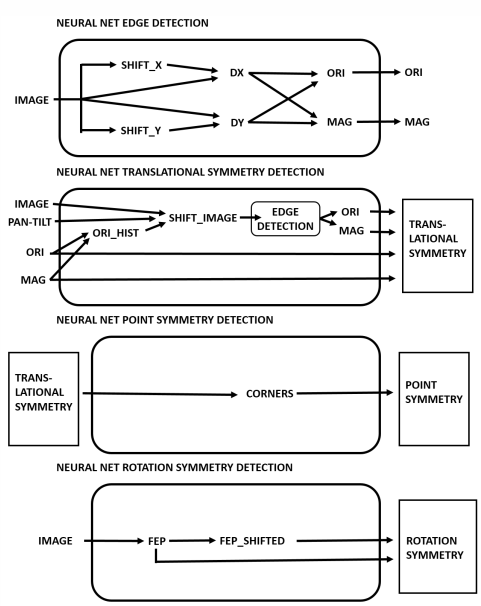


Fig. 1. The Analysis of a Shape from Input Image to Point, Translation and Rotation Symmetries. *Edge Detection*: Neural Networks provide Magnitude and Orientation of Edges in Image. *Translation Symmetries*: These are found by marking pixels that have similar locale after short translation in major edge directions; actuation model is used for this. *Point Symmetries*: find end points of line segments. *Rotation Symmetries*: translate Frieze Expansion Pattern (FEP) from 0 to 359 in 1 degree steps and if similar to original FEP, then rotation symmetry exists.

The goal is to extract abstract representations of 2D shapes based on both sensor (image) data and actuation (pan-tilt or robot arm control) data. This sensorimotor representation is encoded as a wreath product based on shape symmetries as defined by group actions on point sets. Shapes can then be classified according to the resulting wreath product. In particular, we propose a set of recurrent neural networks for wreath product discovery and measure their performance in terms of the robustness of the method on a variety of shape deformations. In addition, we measure the effectiveness of knowledge transfer between distinct motor system representations. A high-level flow of the analysis is given in Figure 1. As shown in the figure, edges and corners are detected in the input image. The other initial required information is an actuation model for whatever actuation systems are available in the robot; these models describe the kinematic relations of the controls to the scene. For example, a pan-tilt model relates the commanded pan-tilt angles to the orientation of the camera. The translation symmetry is found by using the actuation model to move the sensor in the directions of the highest response edges orientations. The edges are recomputed, and any edge in the original image which matches an edge in the shifted image gives rise to a translation symmetry response. Neighbors with similar responses are grouped into linear segments (represented as $\{e\} \wr \Re$ which signifies a point ($\{e\}$) acted on ($\wr$) by a translation ($\Re$)). Corners are found as the closest non-responsive locations to line segments (this effectively allows for the full line segment wreath product $\{e\} \wr \mathcal{Z}_2 \wr \Re$ where $\mathcal{Z}_2$ acts as a characteristic function to select points on the finite line segment). In addition, the image is also transformed to a Frieze Expansion Pattern (a log-polar representation [5]). The line segments are used with the polar coordinate image and the actuator model to discover rotational and reflection symmetries (using 1D operators in the polar representation). We have described elsewhere the efficacy of these methods for document analysis [2]. The major contribution here is that we develop these algorithms in a form more suited to neural network expression. Once these group actions have been recovered, they are combined into the wreath product as a final output.

## II. BACKGROUND

Leyton [6] proposed the wreath product as the basis of cognition, and we have recently provided an implementation for 2D and 3D shape analysis in an effort to validate the theory (see [2]). There we demonstrated the effectiveness of actuation-based shape analysis in terms of virtual sensors and actuators and applied the method to character recognition in engineering drawing document analysis (see [4] for a detailed survey of this research area). We give here a brief technical description of wreath products; for more information, see Leyton.

In order to understand a wreath product we briefly explain the concept of a *semidirect product* in group theory, that underlies the concept of a wreath product. Consider a homomorphism $\phi$ given by $\phi_h(n) = hnh^{-1}$ for all $n \in N$ and $h \in H$, where H and N are groups, and H is a group that acts on N by conjugation. For each $h \in H$, conjugation by h is an element of $Aut(N)$ (automorphism group of N).

Given two groups $N$ and $H$, and a group homomorphism $\phi$ from $H$ into $Aut(N)$, $N \rtimes_\phi H$ denotes the *semidirect* product of $N$ and $H$ with respect to $\phi$ and satisfies the following:

1) $N \rtimes_\phi H$ contains elements from $N \times H$
2) Group operation $\star$ of $N \rtimes_\phi H$ is defined as: $(n_1, h_1) \star (n_2, h_2) = (n_1 \phi_{h_1}(n_2), h_1 h_2)$, where $n_1, n_2 \in N$ and $h_1, h_2 \in H$.

Now consider a group $L$ where $L$ consists of the direct product of $k = \mid H \mid$ copies of N, i.e., $L = N_1 \times N_2 \times ... N_k$. The *wreath product*, $\mathbf{G} = \mathbf{N} \wr \mathbf{H}$, is formed by the *semidirect product* of L and H. Thus $\mathbf{G} = \mathbf{N} \wr \mathbf{H} \equiv \mathbf{G} = \mathbf{L} \rtimes \mathbf{H}$.

## III. Related Work

Another approach which includes actuation in the representation is that of Plamondon [10], [11], [12], [13], but that work is based on some aspects of human rapid movements. From a psychological perspective, Noë [8] argues that "sight and touch can share a common spatial content" and that some higher-level abstraction exists which links movement and sensory perception; we propose that the wreath product is one such an abstraction.

Of course, neural networks have been shown capable of detecting image features (e.g., corners, gradients, edges, blobs, etc.), and more recently, some work has been done on implementing graphics like geometric transforms on image data. In particular, Hinton et al. [3], have proposed transforming auto-encoders to produce a vector of instantiation parameters to deal with variations in position, orientation and scale in images; also see their work on factored higher-order Boltzmann machines [7]. An alternative approach is given by Berkes [1] for transformation learning, but only learn parameters that are linear functions of the input; however, they argue that their "model corresponded closely to functional and anatomical properties of simple and complex cells in the primary visual cortex." None of this previous work formulates a unified group theoretic sensorimotor analysis as proposed here.

## IV. Method

Here we assume that networks exist for all the processing shown in Figure 1, except for the discovery of translation ($\Re$) and rotation ($\mathcal{Z}_n$) symmetries which are newly proposed here. The discovery of translation symmetry is achieved by moving the virtual 'eye' and finding invariant pixels (i.e., there is an edge of the same orientation where they wind up). Rotational symmetry can be found in various ways, e.g., (1) rotate the image a small amount and check similarity to the original image, or (2) use the Frieze Expansion Pattern (FEP) in which rotational symmetry is found as a translation symmetry along the theta (horizontal) axis. Implementation details are given in the following subsections.

### A. Translational Symmetry

The intrinsic meaning of a translation symmetry in a specific direction is that a point set moved in that direction is isomorphic to the original set. Thus, sliding all the points on the x-axis one unit in the positive direction results in the same set of points (the x-axis); i.e., $T(S) = S$. When dealing with finite objects (e.g., a line segment with two end points), the translation semantics is retained by adding a characteristic function to each point which is 1 if the point is in the set and 0 otherwise. When points move out of the finite line segment, they are no longer 'on' and when they move onto the line segment, they are 'on.'

This notion can be achieved quite simply by combining information about edge directions of interest (say directions with large values in the orientation histogram), an actuation model which allows simulating a small displacement, and a similarity function between two images. Algorithm $\Re$-Symm describes how translation symmetry is detected at each pixel. Most of the functions used in the computation

**Algorithm R-Symm**
**Data**: image, orientation, histogram, actuation model
**Result**: line segments

```
for each orientation above threshold
  move image small amount in
    orientation direction and record
    invariant pixels
  move edge image small amount in
    negative orientation direction
    and record invariant pixels
  every invariant pixel with similar
    neighbors put in linear segments
```

can be implemented as hard-wired neural networks. An image shift of any amount can be achieved by repeated shifts in $x$ and $y$ using a recurrent neural network. Figure 2 shows the neural net connections for a 1-pixel translation in the positive $x$ direction; a similar network exists for shift in the $y$ axis.

$dx$ is computed by shifting the image in the $x$ direction and then subtracting the two images; $dy$ is formed in a similar manner. The orientation ($atan2(dy, dx)$) and magnitude ($\sqrt{dx^2 + dy^2}$) of the gradient $\frac{\partial f}{\partial x}$ are then computed as learned neural network functions. We have trained neural networks to compute both the magnitude of the gradient as well as the orientation. Figure 3 (upper) shows the results of learning the *atan2* function, and Figure 3 (lower) shows the results for the magnitude function.

A comparison of two corresponding pixel values is determined as the absolute value of their difference. Finally, the histogram function is represented as a neural network.

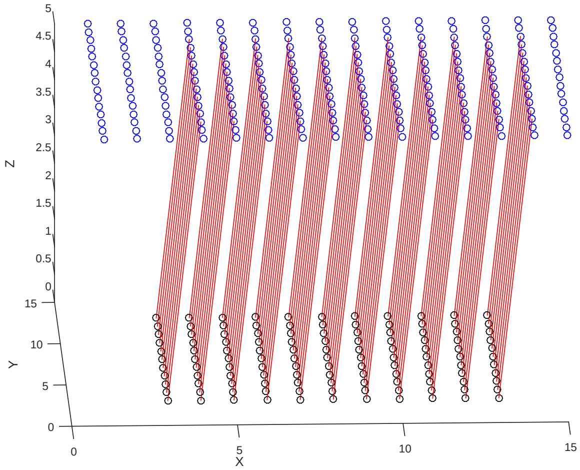


Fig. 2. Neural Network Translation (0 degrees) Mapping.

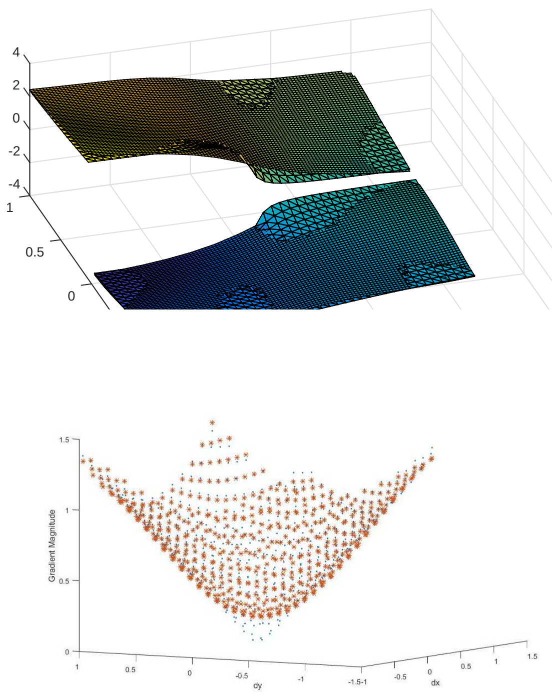


Fig. 3. The Learned atan2 Function (upper) and magnitude (lower) overlayed on the Actual Values.

## B. Rotational Symmetry

A major cost in detecting rotational symmetries arises in finding the location of the rotational axis (typically at the center of a shape). This is handled here by finding the center of mass (CoM) location for each shape of interest, and then transforming to a polar representation expanded about the center of mass of the shape. The FEP is easily achieved by a simple mapping of Cartesian layout pixels to a polar form; Figure 4 shows the wiring diagram of such a mapping as a neural network. Figure 5 shows just the middle row of the FEP which consists of just the point of expansion pixel for each $\theta$. For example, a circle is transformed into a rectangle (see left side of Figure 6), while a square is transformed as shown on the right side.

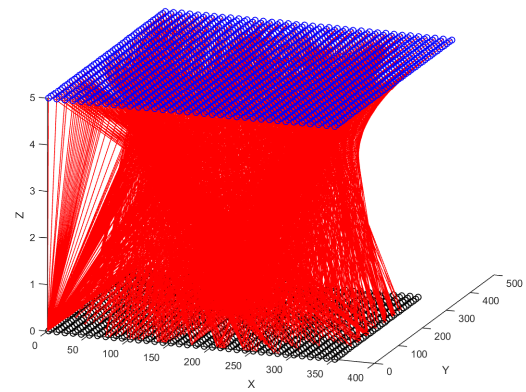


Fig. 4. Wiring Diagram of Neural Network Mapping from Regular Image to FEP.

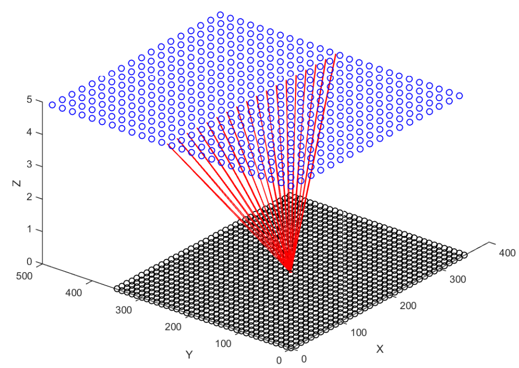


Fig. 5. Wiring Diagram of Middle Row of FEP (which Consists of Just the Point of Expansion).

Note that rotational symmetry is found by translating the image horizontally (with wraparound); we have shown previously that the only translations that need be considered are those that align max (or min) points along the upper boundary of the shape in the FEP since symmetry requires that such a max (or min) maps to a similar max (or min), but we do not exploit this fact here. Algorithm $\mathcal{Z}_k$-Symm describes how rotational symmetries are found. The FEP is produced by a hard-wired neural network as shown in the previous figures. A recurrent neural network is used to shift the FEP in the $x$ axis in steps of one degree, and then the result is compared to the original FEP; if they are similar, then the angle is output as a symmetry.

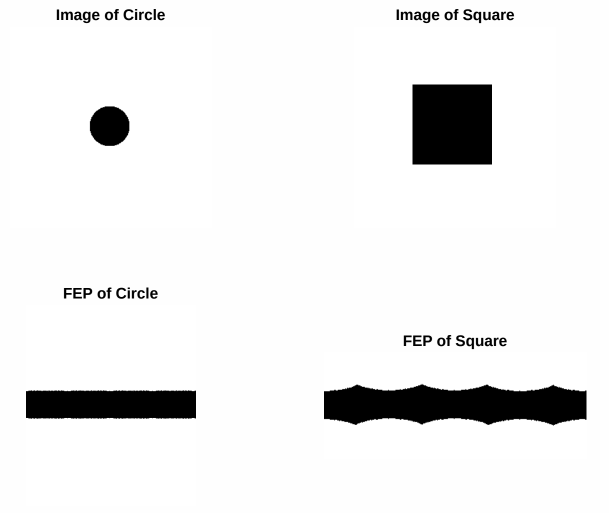


Fig. 6. Image and FEP of Circle (left); Image and FEP of Square (right).

**Algorithm Z_k-symm**
**Data**: FEP (one for each CoM)
**Result**: $\theta$ vector (symmetry angles)

```
forall shifts t
  if similar to untranslated FEP
    add t to theta vector
```

## V. Results

The method was tested on a set of standard shapes (square, rectangle, circle, tri-lobed shape) as well as distorted versions of these in which noise was added by deleting and adding shape boundary points. Correct edge elements, as well as translational and rotational symmetries were robustly recovered. Figures 7– 14 show the results of the rotational translational analysis on several shapes and noisy versions of those shapes as well. As can be seen the analysis is very robust to noise added to the shape. Note that in Figure 7, there are four symmetries found (with near 100% similarity measure), and these are at theta equal to 0, 90, 180, and 270 degrees; in the noisy square data analysis, the same result holds, albeit with a lower similarity measure. Noise is introduced by means of disturbing the boundary pixels of the shape, and this causes a perturbation of the FEP as well. The rectangular shape yields two rotational symmetries, 0 and 180 degrees, respectively, and these are also discovered in the noisy data. The circle shape results in a high similarity measure at all angles (note that the range of the similarity measure in the plot is from 0.996 to 1 in both the perfect data and the noisy circle shape. By way of a more detailed comprehensive analysis of these results, consider the three parts of Figure 14. The upper part of the figure shows the rotational symmetry measures as the continuous curve, and plotted on top of that are the discrete locations (i.e., displayed as red circles) where the similarity measure had a value of 0.90 or greater. Thus, it can be seen that these maximal matches occur at just the three rotational symmetries of the shape. What is interesting about this is that the method is extremely robust to the amount of noise introduced into the shape. The noise mechanism works as follows: (1) some percentage, $p$, of edge pixels are selected for distortion, (2) a maximal distortion neighborhood size,$d$, is selected, and (3) the edge pixels are chosen randomly according to $p$, and then the edge pixel is displaced to another pixel location in the $d$ by $d$ window surrounding the selected pixel (note that connectivity of the figure is maintained during the process). For the specific shape used here, $p$ was set to 0.9, and $d$ was set to 1. Thus, Figure 14, middle, shows the resulting noisy shape. The lower part of the figure shows the FEP produced from the noisy shape. We also studied rotated versions of the shapes and those worked well, too. The wreath product outputs are just the catenation of the translation and rotational symmetries.

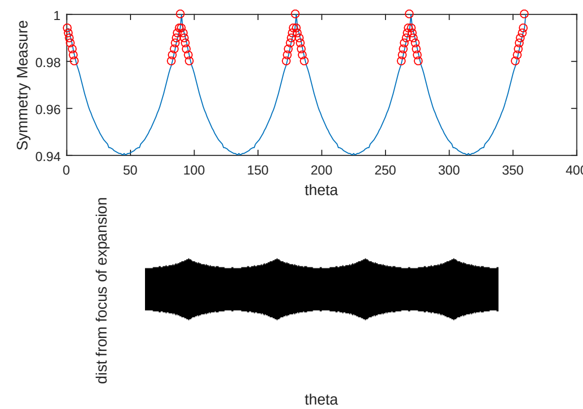


Fig. 7. Square Analysis.

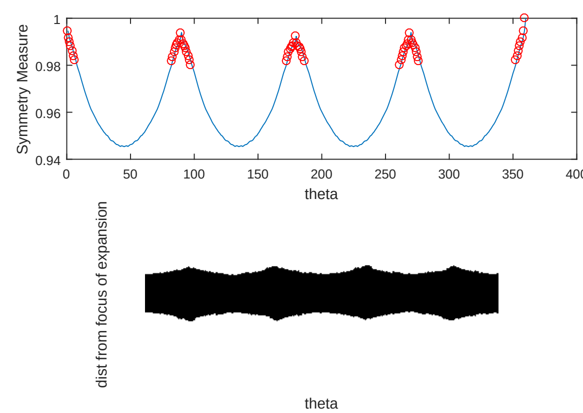


Fig. 8. Noisy Square Analysis.

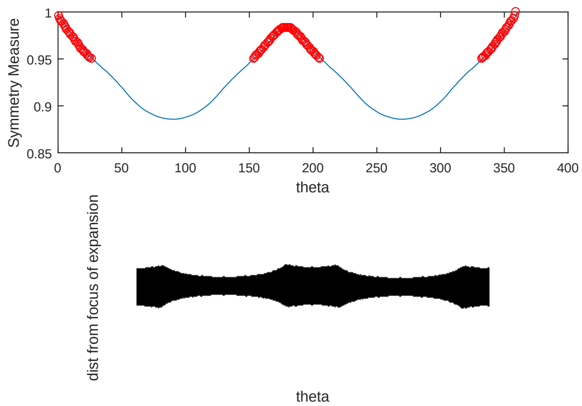


Fig. 9. Rectangle Analysis.

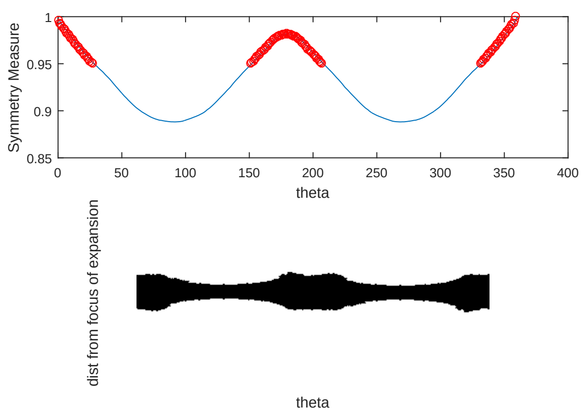


Fig. 10. Noisy Rectangle Analysis.

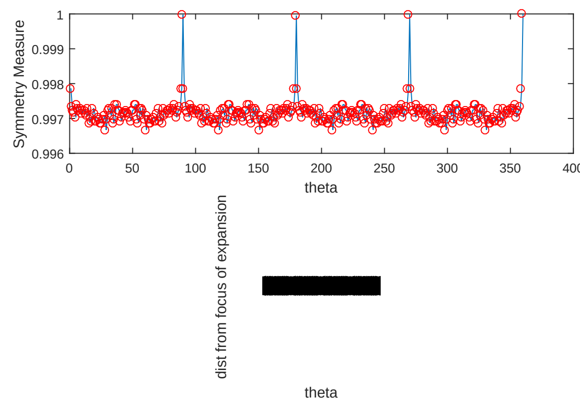


Fig. 11. Circle Analysis.

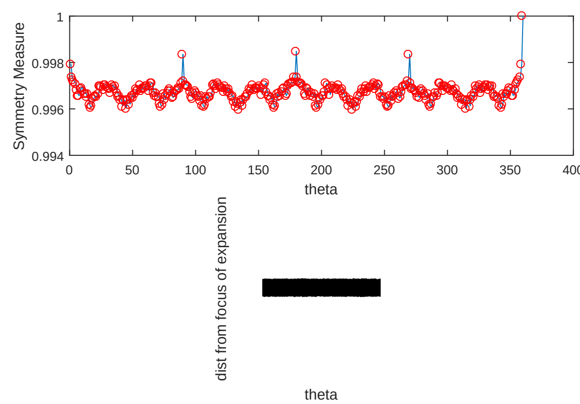


Fig. 12. Noisy Circle Analysis.

In addition to discovering the wreath product representation of the shape as a set of actuation (pan-tilt angles) and perception signals, a neural network was learned for the transformation from the pan-tilt actuation system to a two-link robot arm (see Figure 15).

E.g., once the square shape is represented as a sequence of rotations of a line segment, then the two-link manipulator end effector can be driven through the same point set using the wreath product description. The mean error of the neural network produced (x,y) locations was 0.13 units.

The main conclusion to be drawn from these experiments is that the neural network representation successfully and robustly computes the wreath product representations (i.e., produces a combined perception and actuation formulation of sensorimotor data), and has the potential to be much lower computational complexity when fully implemented on a native neural network hardware architecture. The results can then be used in some form of cognitive architecture for an embodied agent.

## VI. Conclusions and Future Work

The major contribution here is that we have shown how neural networks can be used to derive an abstract representation (wreath products) of 2D shape that ties actuation to sensory perception. Moreover, multiple actuation systems can be mediated through the abstraction to translate between distinct generative actions. Another target is to implement the system in some hardware-based neural net system, e.g., SpiNNaker [9]; although execution on a system allowing modeling with spiking neuron systems is not necessary theoretically, it would be interesting to explore this avenue in brain emulation. Furthermore, the work needs to be extended to temporal sequences to actually generate the shapes. Finally, in future work, we intend to embed this capability in a mobile robot and extend the representation to 3D.

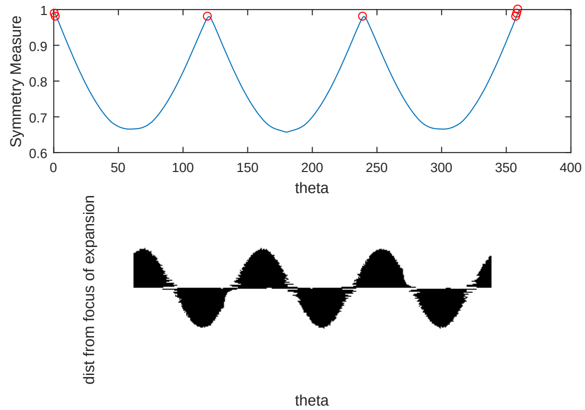


Fig. 13. Three-Figure Analysis.

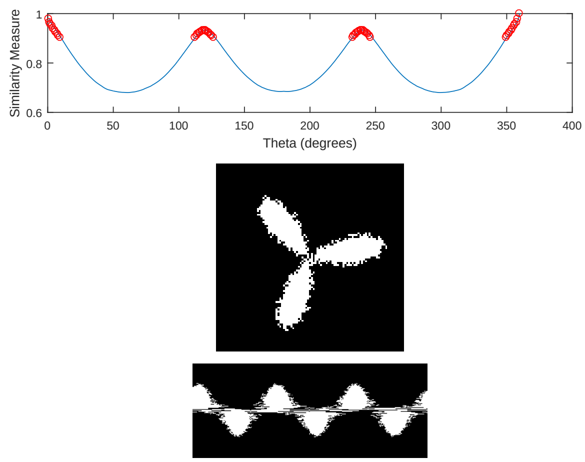


Fig. 14. Noisy Three-Figure Analysis.

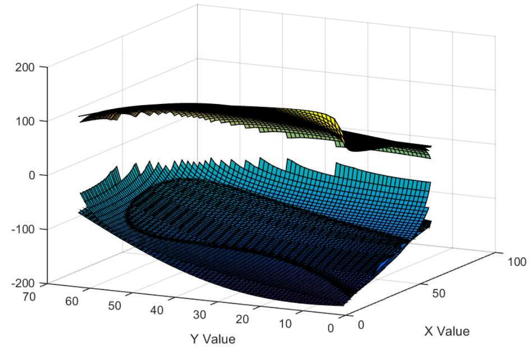


Fig. 15. Neural Network Output for Map from $(x, y)$ to $(\theta_1, \theta_2)$.

## References


[1] P. Berkes, R.E. Turner, and M. Sahani. A Structured Model of Video Reproduces Primary Visual Cortical Organisation. *PLoS Computational Biology*, 5(9):1–16, 2009.

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

[3] G.E. Hinton, A. Krizhevsky, and S.D. Wang. Transforming Auto-Encoders. In *International Conference on Artificial Neural Networks and Machine Learning*, Espoo, Finland, June 2011.

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

[5] 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.

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

[7] R. Memisevic and G. Hinton. Learning to Represent spatial Transformations with Factored Higher-Order Boltzmann Machines. *Neural Computation*, 22:1473–1492, 2010.

[8] A. Noë. *Action in Perception*. MIT Press, Cambrideg, MA, 2004.

[9] E. Painkras, L.A. Plana, J. Garside, S. Temple, S. Davidson, J. Pepper, D. Clark, C. Patterson, and S. Furber. SpiNNaker: A Multi-Core System-on-Chip for Massively Parallel Neural Net Simulation. In *IEEE Conference on Custom Integrated Circuits*, San Jose, CA, September 2012.

[10] R. Plamondon. A Kinematic Theory of Rapid Human Movements. I: Movement Representation and Generation. *Bio. Cybernetics*, 72(4):295–307, 1995.

[11] R. Plamondon. A Kinematic Theory of Rapid Human Movements. II: Movement Time and Control. *Bio. Cybernetics*, 72(4):309–320, 1995.

[12] R. Plamondon. A Kinematic Theory of Rapid Human Movements. III: Kinematic Outcomes. *Bio. Cybernetics*, 78(2):133–145, 1998.

[13] R. Plamondon, C. O'Reilly, J. Galbslly, A. Almaksour, and E. Anquetil. Recent Developments in the Study of Rapid Human Movements with the Kinematic Theory: Applications to Handwriting and Signature Synthesis. *Pattern Recognition Letters*, 35(1):225–235, 2014.