Representing spatial relations with fractional binding

Thomas Lu (tlu@uwaterloo.ca)
Aaron R. Voelker (arvoelke@uwaterloo.ca)
Brent Komer (bjkomer@uwaterloo.ca)
Chris Eliasmith (celiasmith@uwaterloo.ca)
Centre for Theoretical Neuroscience, University of Waterloo
Waterloo, ON, Canada, N2L 3G1

Abstract

We propose a cognitively plausible method for representing and querying spatial relationships in a neural architecture. This technique employs a fractional binding operator that captures continuous spatial information in spatial semantic pointers (SSPs). We propose a model that takes an image with several objects, parses the image into an SSP memory representation, and answers queries about the objects. We demonstrate that our model allows us to not only store and extract objects and their spatial information, but also perform queries based on location and in relation to other objects. We show that we can query images with 2, 3, and 4 objects with relative spatial locations. We also show that the model qualitatively reproduces Kosslyn's famous map experiment.

Keywords: Semantic Pointer Architecture; spatial representation; spatial memory; spatial relations; fractional binding; continuous spaces; cognitively plausible representation

Introduction

Capturing spatial reasoning has been a long-standing and difficult challenge when using artificial neural network models (Haldekar et al., 2017). Nevertheless, spatial cognition has long been studied in cognitive science (Kosslyn, 1980). Often, such research has led to proposals in which mental representations of space are continuous (Kosslyn, 1984). These representations are thus manipulated like physical images: shifting them, scanning over them, extracting spatial relations from them – effectively treating mental representations of images somewhat like physical maps. While there have been vigorous debates on the empirical adequacy of such proposals (Pylyshyn, 1973), here we explore the practicalities of implementing mental manipulations of this variety in compact and efficient representations that lend themselves to implementation in neural networks.

We approach this problem by using an architecture that deploys fractional binding to construct spatial semantic pointers (SSPs). We demonstrate that our binding architecture allows us to not only store and extract objects and their locations, but also perform mental queries to find objects based on location and in relation to other objects. It is, in particular, the ability to query such representations regarding spatial relations that we believe makes this a promising architecture for capturing many human mental image manipulation behaviors. The ability to perform such queries relies on the fact that these representations are continuous, as proposed by Kosslyn and others. The specific goal of this paper is to describe and simulate a cognitively plausible architecture that captures core qualitative features of spatial reasoning.

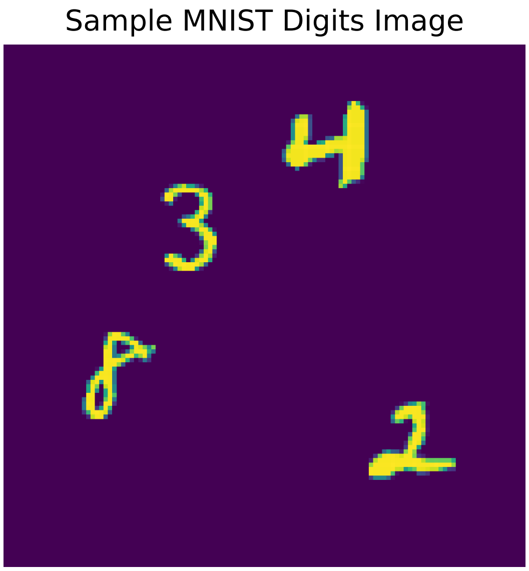


Figure 1: MNIST digits placed randomly in a 120x120 space. Given a query of: "8" and "up and right", the correct response is either: "3" or "4".

We begin by specifying our experimental design, which focuses on asking relational questions about a represented image. We then describe the spatial representation we use, discuss its properties, and note its natural affinity for implementation in spiking neural networks. After this we describe how regions are represented to allow for spatial relation queries. Then, we describe each of the elements of our architecture, as well as how they are integrated in the final system. We subsequently present results showing the accuracy of assessing spatial relations in a spatial working memory task. We also use this same representation to reproduce the main feature of Kosslyn's famous map experiment: reaction time scaling linearly with spatial distance. Finally, we discuss our findings and identify future work.

Experimental design

Our first experiment adopts a task similar to that proposed by Weiss et al. (2016). Specifically, we construct a set of example images to perform queries on by selecting batches of digits from the MNIST database and placing them at random locations on a 120x120 image. We choose between 2 and 4 digits to include in a given image. We then generate queries automatically by randomly selecting a target digit and computing its relative direction from another randomly selected query digit. For this experiment, we limited the query direction to 4 possible quadrants: up and left, up and right, down and left, down and right. Given the query digit and direction, we expect the response to be one of the target digits (if there are multiple such digits, then either one is marked correct). For instance, in Figure 1 we show an example randomly generated image, for which we might query "What is up and to the right of the 8?" A response of either "3" or "4" would be marked correct.

For this experiment, we normalized the coordinates of the digits in the 120×120 pixel image to a continuous 10x10 space, specifically the intervals $x \in [-5,5]$ and $y \in [-5,5]$, before encoding them in a memory through our model visual system. Given our chosen representation, this range was found to provide a good trade-off between accuracy and precision.

We also performed a second experiment, similar to the visual-spatial map experiment by Kosslyn et al. (1978). Kosslyn's map experiment recorded the time that it takes for a subject to scan from one location to another in memory, and demonstrated that closer objects are typically reached faster. For our experiment, we used a memory of several digits placed randomly, and scanned from a queried starting object to the queried ending object.

Methods

Spatial representation

We employ the method for spatial representation proposed by Komer et al. (2019). This method generalizes the notion of binding that is employed by several vector symbolic architectures (VSAs) to continuous spaces. The method defines a "spatial semantic pointer" (SSP) to be the result of a fractional binding. The particular binding used is the circular convolution operator proposed by Plate (1995), which is essentially element-wise multiplication of vectors in Fourier space. The natural extension of this is then element-wise exponentiation in Fourier space. Supposing B is a fixed d-dimensional vector (i.e., semantic pointer), fractional binding is defined by expressing the binding in the complex domain:

$$B^k = \mathcal{F}^{-1}\left\{\mathcal{F}\left\{B\right\}^k\right\}, \quad k \in \mathbb{R}, \tag{1}$$

where $\mathcal{F}\{\cdot\}$ is the Fourier transform, and $\mathcal{F}\{B\}^k$ is an element-wise exponentiation of a complex vector—analogous to exponentiation using fractional powers (e.g., $b^{2.5}$)—permitting k to be real. This representation can thus map from a continuous space, $\mathbb{R}$, to a high-dimensional vector space, $\mathbb{R}^d$. Because the high-dimensional space of semantic pointers can support construction of cognitive structures, various kinds of syntactic inference, and so on (Eliasmith, 2013), this proposed representation provides a novel link between such cognitive operations and continuous spaces.

To explore this link, in this work we use a generalization of the representation to multiple dimensions (Komer et al., 2019). We can represent points in $\mathbb{R}^n$ by repeating equation 1, n times, using a different semantic pointer for each represented dimension (i.e., for each axis), and then binding all of the resulting vectors together. For $n = 2$ (i.e., for a 2-D spatial map), the SSP that represents the point (x, y) is defined as the vector resulting from the function:

$$S(x,y) = X^x \circledast Y^y, \tag{2}$$

where X and Y are fixed semantic pointers, x and y are reals, and we are using fractional binding as defined by equation 1.

In this work we explore querying spatial relations between multiple objects in memory – for instance, asking "What is below and left of the 3?" To specify the spatial query, we represent the region of space being queried (e.g., below and left) as another SSP. The SSP that represents a continuous region (e.g., a solid rectangle), specified by some infinite set of points R, is defined as:

$$S(R) = \int_{(x,y)\in R} X^x \circledast Y^y \, dx\, dy. \tag{3}$$

To move this region to be relative to a given starting point, we exploit the shift property of SSPs. In particular,

$$B^{k_1} \circledast B^{k_2} = B^{k_1+k_2}, \quad k_1, k_2 \in \mathbb{R}. \tag{4}$$

This means that to shift any SSP, we only need to convolve the spatial representation of a region or objects with the SSP representing the coordinates of the shift direction. For example, we can shift a region representing a direction, (e.g., "up and right") to the location of an object to generate a region representing a query (e.g., the "8" in the previous example).

Conversely, we can also leverage this property to shift the entire spatial memory relative to the origin. This gives rise to a notion of movement through the space and an egocentric interpretation of the space rather than the previous allocentric interpretation. Thus this method of semantic pointer supports both egocentric and allocentric coding of space.

To represent a single object occupying some location, we bind its tag (OBJ) with the SSP from equation 2:

$$M = OBJ \circledast S(x,y). \tag{5}$$

In general, to represent a set of m labelled objects together in the same memory, we can use superposition:

$$M = \sum_{i=1}^{m} OBJ_i \circledast S(x_i, y_i), \tag{6}$$

with a distinct semantic pointer OBJ_i tagging each object.

Furthermore, rather than placing objects at singular points in memory, it may be more intuitive to bind objects to regions in memory. This can be done similarly:

$$M = \sum_{i=1}^{m} OBJ_i \circledast S(R_i), \tag{7}$$

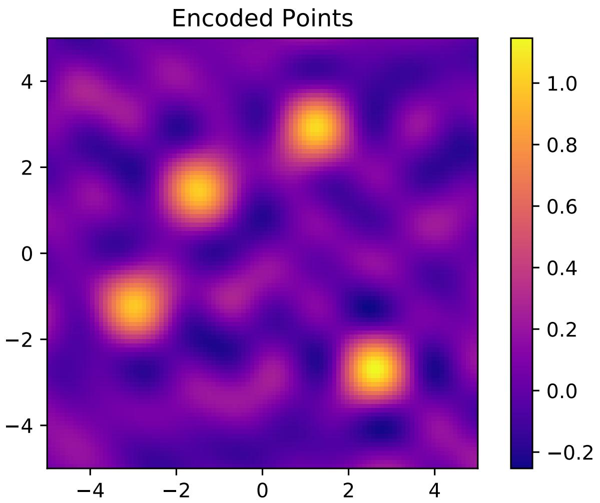


Figure 2: Heatmap of the four locations from Figure 1, represented by a single spatial semantic pointer (equation 6).

with R_i representing the region that a particular object occupies. This representation allows us to represent notions of size and shape in memory as well.

Given a representation like that in equation 6 or 7, we can query it in a number of ways. For example, to determine what object(s) are within some region R, we can compute:

$$M \circledast S(R)^{-1}, \tag{8}$$

where $(\cdot)^{-1}$ corresponds to the approximate inverse vector used to unbind using circular convolution. By the properties of binding and superposition, the resulting vector will have the highest cosine similarity (i.e., dot product) with the object(s) within R.

While only part of our architecture is currently implemented in a neural network (see below), all of the operations, except fractional binding, needed for the architecture have previously been implemented in spiking neural network models (Eliasmith, 2013). The fractional binding itself is implemented in spiking neurons by Komer et al. (2019). These implementations use the methods of the Neural Engineering Framework (Eliasmith & Anderson, 2003).

Using the spatial representation

In this section we briefly demonstrate the use of equations 3, 4, and 6. All of the SSP representations in the model are 512-dimensional. We begin by encoding multiple objects into the memory, as per equation 6, is demonstrated in Figure 2. Here we can see an example of the four objects from Figure 1 being encoded into the represented space. While we are showing a decoding of this representation mapped into the continuous space, the full representation is a single 512-dimensional vector. The number of objects in memory does not change the size of the representing vector, although there are effective limits on capacity (Komer et al., 2019). We also tested a region based system by binding every digit to the square region occupied by the digit rather than just a single point.

To query such a representation, we can construct a region vector. A region vector, as defined by equation 3, is also a 512-dimensional SSP, but it represents an entire region instead of a specific point. Region vectors can be used just like a regular location vector. We can bind objects to it, add it to a memory, and we can also use it to extract objects that are located within a region. Furthermore, as regions are integrals over pointers raised to coordinate exponents, binding a region to a point vector shifts the exponents in the integral by the coordinates of the point (see equation 4), which in turn shifts the entire region represented by the integral (see Figure 3-Top) in the direction of the point relative to the origin. In our experiment, this allows us to pre-compute four regions at the origin and then use binding to shift them to generate any specific query vector (see Figure 3-Bottom). Notably, when region vectors are used to query memories encoding objects at those locations, there is no need to extract the coordinates of the objects being searched over; all computations are performed within the space of our SSPs, without multiple encoding and decoding steps.

Model architecture

In this section we briefly describe each of the components in our model that perform the tasks described in the experimental design section. We also describe the integration of the components and overall flow of information through the model.

Image generation

The images processed by the system are generated by using batches of 28x28 pixel images from the MNIST database and placing them randomly on a 120x120 image (see Figure 1). Because queries are limited to the 4 diagonal directions, we ensured the digits are not too close in the vertical or horizontal direction. We also ensured the digits do not overlap. We generated sets of 5,000 samples for images containing each of 2, 3, and 4 digits.

MNIST Network

In order to generate the SSP representation from the experiment images, we use a straightforward convolutional deep neural network as a perceptual module. It consists of two 3-by-3 convolution layers with 32 and 64 filters respectively. These were followed by a 128 unit fully connected dense layer and a 10 unit fully-connected dense layer for classification. This network was trained on the MNIST dataset achieving 99% validation accuracy.

Since our work focuses on representing spatial relationships rather than classifying multi-digit MNIST images, we use the actual coordinates of the digits to generate a saccade-like cropping of the full image to 28x28 sub-images before providing them to the convolutional network. The identified images are then mapped to random 512-dimensional semantic pointers, which are bound to SSP encoded locations, and

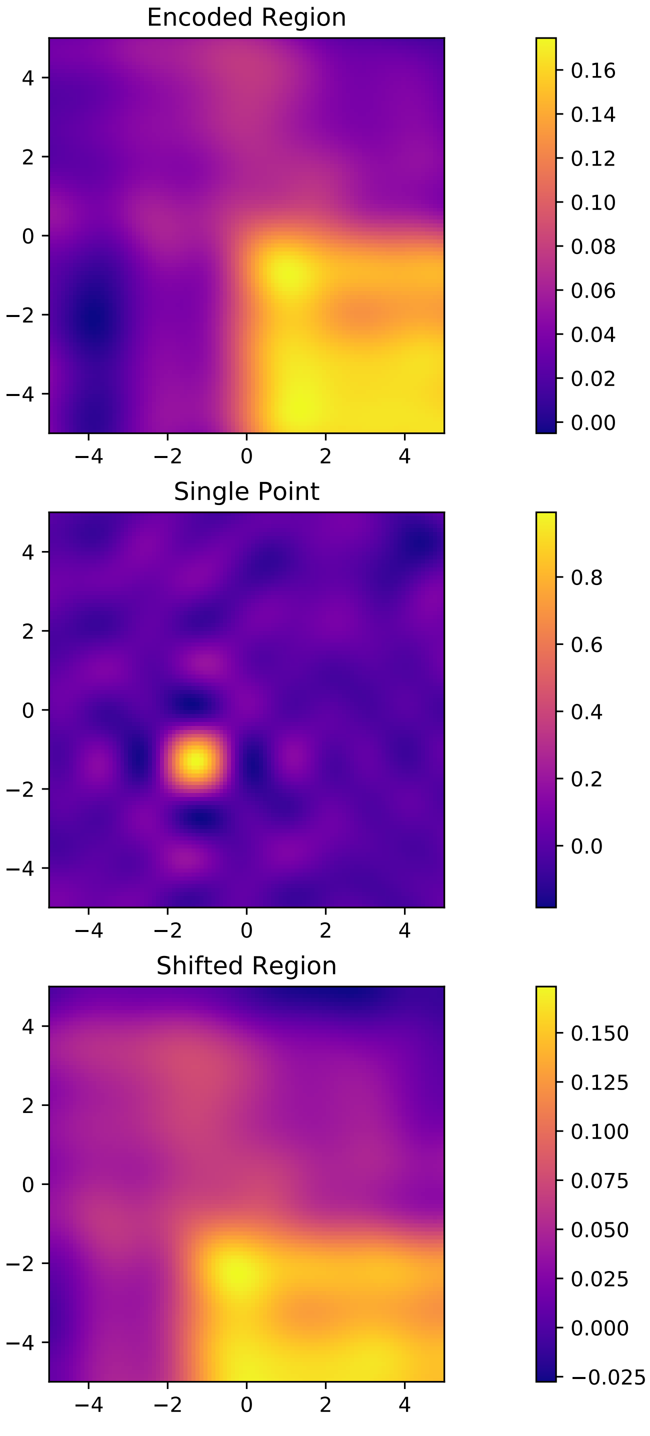
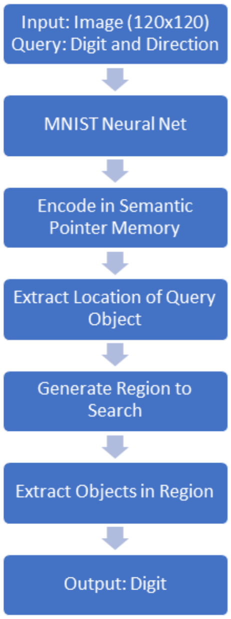


Figure 3: Demonstration of shifting the region representation for a "down and right" query (top panel), to a point encoded as an SSP (middle panel), resulting in a region vector for querying "down and right" with respect to the point (bottom panel).



Input: Image (120x120)
Query: Digit and Direction

MNIST Neural Net

Encode in Semantic Pointer Memory

Extract Location of Query Object

Generate Region to Search

Extract Objects in Region

Output: Digit

Figure 4: Flowchart of entire process

summed across all objects. This process results in a memory representation in the 512-dimensional space, which is subsequently queried using a region representation as described above.

Cleanup memory

When using SSP representations, as with any compressed VSA, the extracted location vector of an object includes noise. When there are multiple objects in the memory, the amount of noise grows. As a result, VSAs of this sort typically include a cleanup memory that maps a noisy vector onto the nearest known vector in the space. In the case of a continuous space, to extract the (x,y) location of the query digit, we generate the known vectors by sampling the continuous space on a 100x100 grid. This grid covers the two $[-5,5]$ axes of the image. To implement the memory, we perform a simple dot product similarity check between the extracted noisy vector and the set of known vectors to find the closest matching vector within the resolution of our grid. Dot products are easily computed in parallel, making this a quick and effective way to reduce noise and improve performance. This kind of memory can be efficiently implemented in spiking neurons (Stewart et al., 2011).

Full system

Before running an experiment, we set up the model by randomly selecting two 512-dimensional unitary semantic pointers to use as axis vectors (i.e., X and Y in 2). We also create a vocabulary of ten 512-dimensional semantic pointers, one for each digit. We then pre-compute 10x10 region vectors for each query, as well as a 100x100 resolution cleanup memory.

We then feed an image into the model architecture, the full pipeline of which is depicted in Figure 4. The image is classifed by the MNIST network, and an object memory is created by summing the SSP representations for each digit in the image, as described above. Extraction of the location of the query object (i.e., the object mentioned in the query) proceeds by performing an inverse convolution on the memory with the query object to find its location, and the cleanup memory is used to reduce noise on the found location. Generating the region to search is accomplished by convolving the identifed location of the query object with the region vector corresponding to the query direction to find the region where the target object might be. Extracting objects in the region occurs by performing an inverse convolution between the shifted region and the original memory. Finally, the similarity between the results of this query and each object in the vocabulary is calculated as a dot product. The object with the highest similarity determines the model's response to the query.

Scanning System

The scanning system involves similar steps. We reuse the axis vectors as well as the pre-computed cleanup memory tables from the previous system. The map image is converted to a memory vector as above. Given a starting and ending object, the locations are extracted by performing inverse convolution with the objects in question on the memory. These locations are cleaned with the cleanup memory and used to determine the direction vector of the scanning using SSPs:

$$V = (X^{x_5} \circledast Y^{y_5}) \circledast (X^{x_2} \circledast Y^{y_2})^{-1} \tag{9}$$

where x_5 is the x position of the "5", and so on. We then normalize this vector, shrinking it to a 0.05 unit step, and repeatedly apply it to the starting vector ($V_{t+1} = V_t \circledast V$ where $V_0 = X^{x_5} \circledast Y^{y_5}$).

To scan the memory, we started at the starting location from above, and extracted the objects in that location with inverse convolution. The scan location is then updated by convolution with the step vector generated above, shifting the location towards the target object, and the above steps are repeated. A dot product similarity comparison is used at each step to determine what objects were extracted or "seen" by the scan. A 0.8 similarity threshold is used to determine when the target object has been reached.

Results

Relational Query Experiment

For the query experiment, we ran 5,000 randomly generated experiments for each of 2, 3, and 4 digit images. For the experiment, we tested the accuracy of the output by simply marking the response as correct if the model response matched an object in the queried region.

Table 1 shows the results from the experiment involving identifying a target digit given an image, a query object, and a query. Correctness is calculated by comparing the output to all digits in the correct direction. Baseline performance is the probability of answering a query correctly by randomly selecting one of the remaining digits in the image. This is calculated by dividing the average number of correct answers in each image by one less than the total number of digits in the image. Naturally for the 2 digit case, there is only one possible answer other than the digit used to query so the probability would be 100%. The baseline probability is very high due to the broadness of our query.

	2 Digits	3 Digits	4 Digits
Point Representation Accuracy	92.18	84.40	72.90
Region Representation Accuracy	95.98	87.22	81.24
Baseline probability	100.00	71.76	62.60

Table 1: Experimental results for spatial relation queries.

The results from the 2 object query indicate that using a region vector decreases accuracy compared to a simple location query. A location query with two objects in memory (e.g., what is at location (x,y)) has 100% accuracy (results not shown). In this experiment, the 2 digit case is similar to a location query, but for a region. The drop in accuracy is likely because as the region representation becomes larger, a single vector is being used to represent the effective superposition (integral) of many vectors (all those defining the region). This result suggests that region size will determine decoding accuracy, a hypothesis to test in future work.

The 3 and 4 digit experiments showed that extracting object information from an object-location memory improved performance by about 13% and 10% for point based memory and 15% and 19% for region based memory compared to the baseline guessing probability. Representing object locations as regions in memory rather than singular points provided dramatic improvements in accuracy, particularly in the 4 digit case. This is likely due to the fact that a query region is more similar to a square within the region than a single point, leading to higher accuracy extractions with inverse convolution. This suggests that more specific queries involving smaller or tighter regions would yield higher accuracy as their shapes would more closely resemble the regions the objects are bound to compared to the large query regions used in our experiment. Comparing the differences in accuracy for queries of different shapes and sizes is a topic for future study.

The decrease in accuracy as number of digits in the image increases is expected, as a higher number of digits adds difficulty in selecting the correct output since the memory encodes all of the digits. It is a standard property of VSAs for decodability to decrease as a function of the number of objects represented in a structure. While we have not determined the maximum capacity of the proposed representation, being able to store and reasonably accurately recall the relations between four numbers is consistent with standard estimates of working memory capacity at 4 items (Buschman et al., 2011).

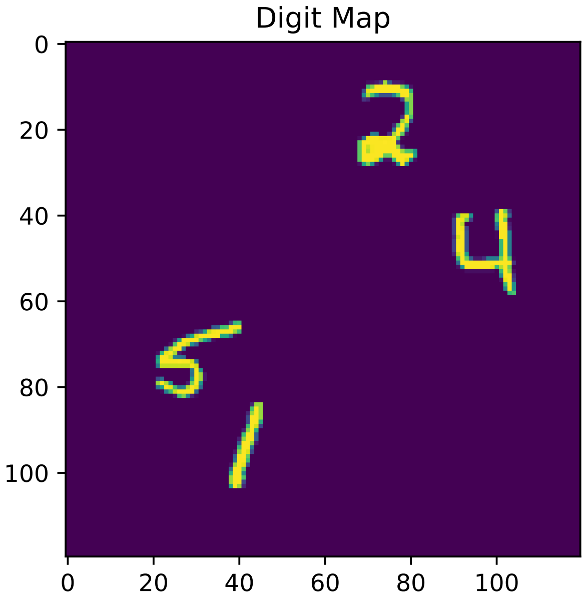


Figure 5: Digits placed randomly in a 120x120 space to represent a map of objects. The memory is scanned from “5” to “1” and “5” to “2”.

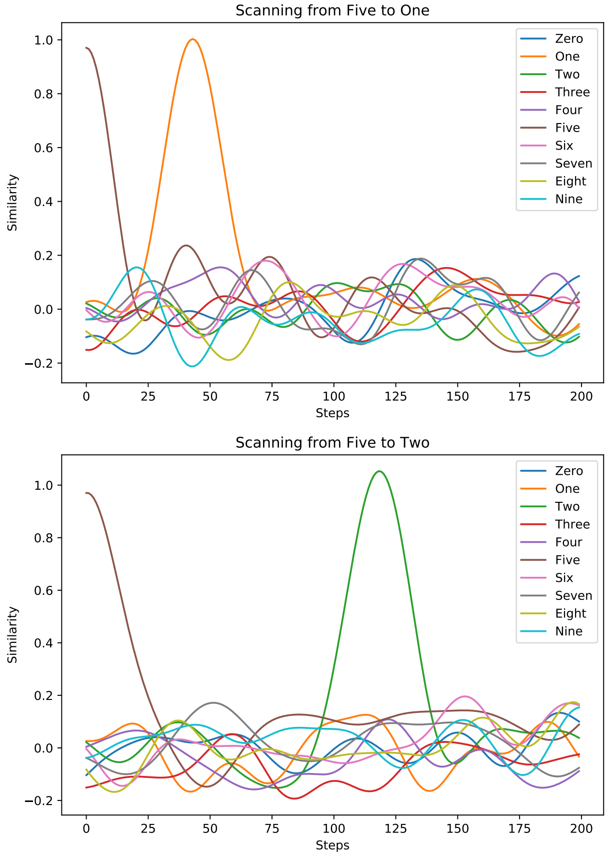


Figure 6: Similarity outputs over time for scanning from the “5” to “1” (top) or “5” to “2” (bottom).

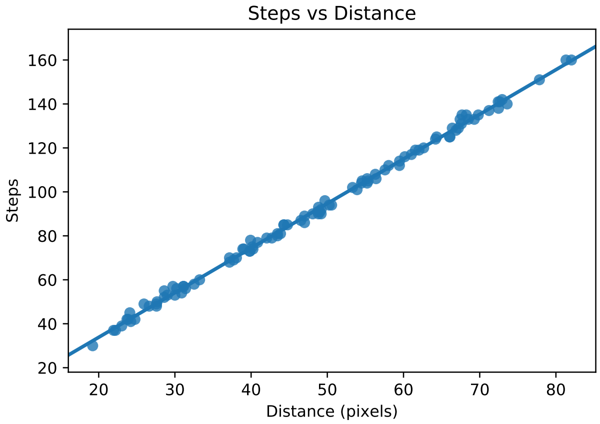


Figure 7: Plot of steps to reach target object vs the distance between starting and target objects over 100 trials (Pearson’s $R > 0.99$, p-value $< 10^{-6}$).

Image Scanning

An image is generated with the same method as in the first experiment to represent a map of objects, with each digit representing an arbitrary object in the map (Figure 5). For the experiment shown in Figure 6 we chose the object “5” as the starting location and the two objects “1” and “2” to be the near and far target objects respectively. From the two plots, we can see a peak at the 0 mark for the starting object “5” which falls away, and a peak at the target object, “1” and “2”, when the scan reaches it.

This experiment was repeated 100 times, with the number of steps required to reach a similarity threshold of at least 0.8 recorded for each trial (Figure 7).

Kosslyn et al. (1978) showed that human spatial memory is represented in a metric space by demonstrating that further objects take longer to scan to in memory, with time linearly related to distance. This experiment shows that the qualitative cognitive behaviour demonstrated by Kosslyn’s map scanning experiment is naturally captured by our SSP memory representation.

Discussion

Our proposed architecture is able to receive an image of multiple objects and generate an SSP representation. Subsequently given a spatial relation query the model can successfully answer with reasonably high accuracy. This provides evidence that the SSP representation can be used to encode continuous spaces in a kind of mental map using representations easily implementable in neural networks. In short, our results show that such representations can be used to reproduce qualitative cognitive behavior relying on spatial manipulation of information encoded in this manner.

A critical next step is to compare human performance on this same task with the proposed model. Preliminary results suggest that accuracy can be manipulated by appropriately choosing the base vectors (i.e., X and Y), and manipulating the dimensionality of the vector space being used. The range of these parameters that match human performance remains to be determined.

There are many possibilities for extending this model. Our particular focus was on two kinds of spatial relation query. However, the direction queries could be generalized to be in any direction (e.g., specifying a vector direction and generating a cone region in that direction). As well, other manipulations, such as spatial rotations, shifts, and so on, can be performed without decoding the SSP. There are a wide variety of psychological results that can provide points of comparison for such manipulations.

Furthermore, the representation itself could be made more complex. For instance, introducing the color of the object (encodable as another 3D continuous space for RGB values), or additional features is natural in this framework. We expect additional information encoded in the memory will adversely affect performance, as seen in human memory tasks.

Finally, the full model can be implemented in a spiking neural network to determine if the proposed representations are robust to biologically plausible implementation. While we expect that this will be successful, given past work that has implemented each of the components, it remains to be seen what effect such implementation has on the accuracy of responding to spatial queries.

Conclusions

We have demonstrated that spatial semantic pointers (SSPs) using fractional binding provide a viable method of representing spatial relationships in a simple model supporting two kinds of visual spatial reasoning. This method lends itself well to implementation in neural networks, and is consistent with cognitive work suggesting that internal representations used in mental imagery represent continuous mental spaces. We believe this is one of few available suggestions for how complex object representations (i.e., high-dimensional feature vectors for digits) can be encoded in a continuous space, and manipulated to answer questions about relations in that space.

Acknowledgments

We would like to thank Terry Stewart for his early work exploring SSPs and personal discussions, and Jan Gosmann for his work on the mathematics of fractional binding for semantic pointers. This work was supported by CFI and OIT infrastructure funding, the Canada Research Chairs program, NSERC Discovery grant 261453, ONR grant N000141310419, AFOSR grant FA8655-13-1-3084, OGS, and NSERC CGS-D.

References

Buschman, T. J., Siegel, M., Roy, J. E., & Miller, E. K. (2011). Neural substrates of cognitive capacity limitations. *Proceedings of the National Academy of Sciences*. Retrieved from `https://www.pnas.org/content/early/2011/06/13/1104666108` doi: 10.1073/pnas.1104666108

Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. Oxford University Press.

Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press.

Haldekar, M., Ganesan, A., & Oates, T. (2017, June). Identifying Spatial Relations in Images using Convolutional Neural Networks. *arXiv e-prints*, arXiv:1706.04215.

Komer, B., Stewart, T. C., Voelker, A. R., & Eliasmith, C. (2019). A neural representation of continuous space using fractional binding. *Proceedings of the 41st Annual Meeting of the Cognitive Science Society*.

Kosslyn, S. M. (1980). *Image and mind*. Harvard University Press.

Kosslyn, S. M. (1984). *Image and brain*. MIT Press.

Kosslyn, S. M., Ball, T. M., & Reiser, B. J. (1978, 2 1). Visual images preserve metric spatial information: Evidence from studies of image scanning. *Journal of Experimental Psychology: Human Perception and Performance*, *4*(1), 47–60. doi: 10.1037/0096-1523.4.1.47

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural networks*, *6*(3), 623–641.

Pylyshyn, Z. W. (1973). What the mind's eye tells the mind's brain: A critique of mental imagery. *Psychology Bulletin*, *80*, 1–24.

Stewart, T. C., Tang, Y., & Eliasmith, C. (2011). A biologically realistic cleanup memory: Autoassociation in spiking neurons. *Cognitive Systems Research*, *12*, 84-92. Retrieved from `http://dx.doi.org/10.1016/j.cogsys.2010.06.006` doi: 10.1016/j.cogsys.2010.06.006

Weiss, E., Cheung, B., & Olshausen, B. (2016). A neural architecture for representing and reasoning about spatial relationships. *Proceedings of the International Conference on Learning Representations (ICLR), Workshop Trak,*.