Information Technology Course
Module Individual Project
by Prof. Dr. Andreas Pech

Grid Cell Encoder

Sabin Bir Bajracharya
sabin.bajracharya@stud.fra-uas.de

Abstract—**Hierarchical Temporal Memory (HTM) provides a flexible and biologically accurate framework for solving prediction, classification, and anomaly detection problems for a broad range of data types. HTM systems require data input in the form of Sparse Distributed Representations (SDRs). An SDR consists of a large array of bits (0's and 1's) where each bit carries some semantic meaning. The first step of using an HTM system is to convert a data source into an SDR with the help of an encoder. The encoder converts the native format of the data into an SDR that can be fed into an HTM system. One of such encoders proposed in this paper is called Grid Cell Encoder. The grid cell encoder converts a 2-D coordinate into plausible grid cell activity. Grid cells are thought to form an essential part of the brain's coordinate system for metric navigation. It is a place-modulated neuron whose multiple firing locations define a periodic triangular array covering the entire available surface of an open two-dimensional environment. The output SDR from this encoder is divided into modules where each module is a distinct group of cells with a common grid spacing and orientation. Different modules have different spacings and orientations. There are different models as to how exactly grid cells work. After a thorough discussion on these, one way of implementing grid cells has been followed to generate SDRs in this paper**

Keywords— Hierarchical Temporal Memory, SDR, Encoder, Grid Cell, Grid Modules

I. Introduction

One of the most fundamental and important abilities in the animal kingdom, and also for humans, is navigation in the environment, or traveling from one location to another. The navigation system in the brain is in charge of representing the location in the environment and orienting the body so that movement from one place to another is successful. This mental image of the surroundings is frequently referred to as a cognitive map. An animal needs to construct an internal "cognitive map" of the surrounding world in order to successfully navigate. This is carried out through a particular system in the brain that consists of multiple different brain areas and different cell types, each with a special function in navigation. One of such special systems of nerve cells in the brain is called grid cells.

Grid cells are neurons that fire according to the position of an animal in its environment. A single grid cell activates at multiple spatial locations with firing fields that are arranged in a strikingly-regular triangular pattern. Grid cells are thought to support animal's navigation and spatial memory, but the cellular mechanisms that generate their patterns are still unknown.

1.1 *Grid Cell properties*

The spatial firing pattern of a grid cell is characterized by three fundamental properties: 1) the grid scale, i.e., the distance between two neighboring firing fields; 2) the grid orientation, i.e, the angle between one of the grid axis and a reference direction; and 3) the grid spatial phase, i.e., the two-dimensional spatial offset between the firing fields and a reference point (Figure 1.1) [Edvard Moser and May-Britt Moser (2007)].

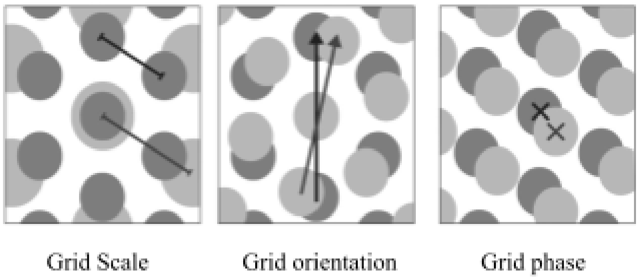

Figure 1.1 Three grid cell properties (Scale, Orientation and Phase) [figa]

1.2 *Modular Organization of grid cell activity*

Two years after grid-cell activity was first reported, with the help of sampling several dorso-ventral locations in the medial entorhinal cortex (MEC), it was observed that grid scale did not vary continuously in the tissue, but rather in discrete steps. The ratio between two subsequent grid scales in a given animal was a constant number, about 1.7, which roughly matched theoretical predictions for the optimal coding of space with periodic signals.[3] A similar scale ratio (in the range 1.4–1.7) was later reported by two other independent studies.[9]

Grid cell activity is generally organized in a small number of discrete functional modules. Within a module, grid patterns had a similar scale, orientation, elliptic distortion, and theta-frequency modulation, but the same properties varied substantially across modules.

1.3 *Role of sensory inputs in grid cell firing*

It is believed that grid patterns may initially emerge from the integration of self-motion cues (path integration) and then get anchored to external landmarks with experience.[10] The external sensory input exerts a strong control on grid cell activity initially but self motion cues may become important with experience to that environment.

Grid patterns are typically unaffected by the size or the aspect ratio of a familiar enclosure, however, for example, when a square arena is suddenly rescaled along one or two directions, grid patterns either stretch accordingly or rearrange their firing fields completely. Also, grid patterns deform coherently within but not across modules suggesting that functionally independent grid cell networks may coexist in the MEC.[3]

The arena boundaries of an environment are believed to control the grid cell activity.[11][12] The grid alignment is known to be preserved after the arena is rotated within the room despite prominent visual cues remaining stationary.[13]

1.4 *Grid pattern formation theories*

This section briefly discusses the three most important computation models for the origin of grid cell patterns. The three classes of models are continuous attractor models, which emphasize the role of recurrent connectivity, oscillatory interference models, which emphasize the role of rhythmic oscillations, and single cell plasticity models, which emphasize the role of external sensory cues and learning.

1.4.1 *Continuous attractor models*

The idea behind the model is that a properly wired recurrent network can sustain a continuum of stable states in the activity of its neurons. As all states are neutrally stable, small external perturbations can push the network to a nearby configuration so that the network activity at any time reflects the cumulative effects of the external inputs onto the system.[3]

The attractor states can be represented by continuous attractor networks (CAN) in two different ways: with either a “rate code” or a “location code”. In a rate coding scenario, the network maintains a state of graded persistent firing, and the stimulus value is encoded by the firing rate of the neurons. On the other hand, in a location coding scenario, the network generates an activity packet or “bump” at the population level, and the stimulus value is encoded by the bump location on the neural sheet (by the identity of the active neurons).

A fundamental feature of all attractor models of grid cell activity is that spatial firing is driven by self motion cues. The attractor theory is consistent with the modular organization of grid cell activity and its coherent remapping across environments. On the other hand, attractor models require a sophisticated neural architecture whose developmental origin is unclear and they cannot explain how grid fields are distorted by external sensory cures and the geometry of the environment.

CAN models of hippocampal place cells have been adapted to explain the activity of grid cells where the connectivity structure of the network is changed to a twisted torus topology. The “twists” consists of tilting the axes of the neuronal sheet from 90 degrees to 60 degrees (Figure 1.2 A). Such a tilt affects the way the population activity bump moves on the periodic network and generates a triangular grid at the single cell level.

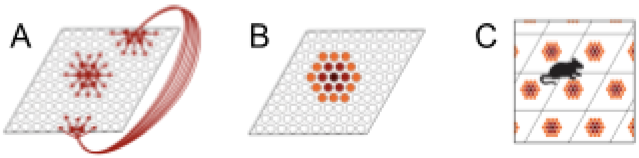

Figure 1.2: Grid cell CAN model [figb]

1.4.2 *Oscillatory interference models*

This model is based on temporally interfering oscillations in single neurons. Due to the dual oscillator model generator of periodic patterns, this model was proposed as a mechanism for grid cell activity.

The original dual oscillator model was extended to explain grid cell activity in two dimensional environments.[14] It is assumed that, within a single cell, a somatic oscillation at theta frequency interferes with multiple dendritic oscillations at slightly higher frequencies. To obtain a planar interference pattern, the frequencies of the dendritic oscillators were modulated by the running speed and direction of the animal (velocity controlled oscillators, VCO). But later on VCos were modeled as separate groups of afferent units (referred as theta cells) as the dendritic and somatic oscillations inevitably lead to phase locking in single cells.[3]

To generate a triangular grid, VCOs must be tuned to the animal's movement directions differing by multiples of 60 degrees. Two VCOs are sufficient to generate grid like patterns but six are required to account for omnidirectional phase precession.

A problem of the OI theory is that grid-like periodicity could be rapidly disrupted by noise and it remains unclear whether such mechanisms could be still effective in biologically realistic networks.

1.4.3 *Single cell plasticity models*

In this model grid patterns originate at the single cell level via spatially tuned feed forward inputs and Hebbian synaptic plasticity.

As shown in figure 1.2, a single cell (black disk) receives spatially selective input from an ensemble of presynaptic cells (white disk). Each input neuron is active in a single location of the environment. Input synapses are plastic and the output neuron is subject to spike rate adaptation. As the animal traverses a place field (dark-red disk), the corresponding presynaptic input is active, the output firing rate increases, and the related synaptic weight is potentiated by Hebbian plasticity.[3] After a short time, however, the output firing rate decreases due to spike rate adaptation, i.e., the neuron is less active in a ring of positions around the current location of the animal (light-blue disk). Inputs with place fields within this 'adaptation ring' are uncorrelated with the output spiking and the corresponding weights are depressed. As the output activity recovers, a new input place-field association can be learned (light-red disk). Such a learning process reaches an equilibrium when the output place fields are roughly equally-distributed in the arena. Assuming isotropic spatial exploration (i.e., roughly circular adaptation rings), an optimal equilibrium in terms of coverage is obtained when the output fields are arranged in a lattice of equilateral triangles — a configuration that is reminiscent of grid-cell patterns.

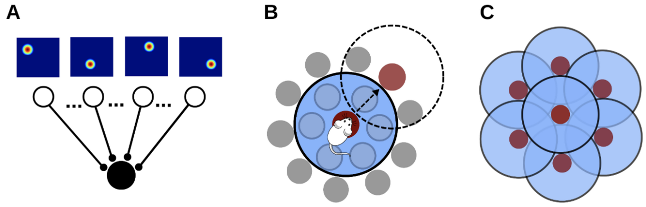

Figure 1.3: Model architecture and pattern formation [figc]

So far, three models have been discussed on the possibilities of how grid cells could work, each with their own advantages and disadvantages. There is no clear winner among them with evident proof as to how a grid cell "should" work. With this in mind, the next section focuses on how grid cells could be implemented for the purpose of this project. The theory basis for this implementation is discussed here

1.5 *Grid cell Encoder*

The grid cell encoder converts a 2-D coordinate into plausible grid cell activity. The proposed grid cell encoder is loosely based on the working principle of Continuous attractor model as explained in the section 1.4.1. To obtain the grid-like pattern when an object is at a given location the axis of the neuronal sheet is titled to a certain degree as shown in Fig 1.4.2 A. Such tilt affects the way the population activity bump moves on the network and generates a triangular grid.

In figure 1.4, the overlay on the bottom left shows all 16 grid cells within the module and the outer image should this module repeat over the region. The blue dot is the cell that is currently active in response to an object's location (red dot) in space. A grid cell module is simply a group of grid cells sharing the same projection properties onto space. Using many grid cells, we can cover a patch of space and use it to tile over a larger space. One grid cell module can tell us a lot more about an object's location in space than a lone grid cell. The grid cell module can show one must be within one of many locations moving across the space. But it cannot decipher an exact location. So one grid cell module is not enough to uniquely map space. But using many grid cell modules together, it is possible to map virtually infinite amount of space.[1]

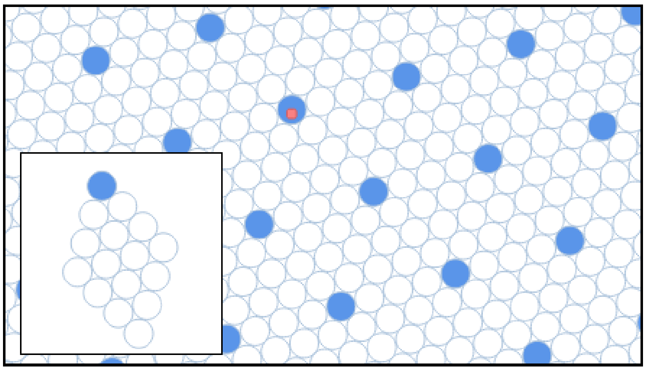

Figure: 1.4 Grid cell modules containing 16 grid cells [figd]

When grid cell modules work together, they can uniquely map space by combining their representations. Each cell in a grid cell module is like a bit in a Sparse Distributed Representation (SDR). As shown in Figure 1.5, this space is being mapped with 6 grid cell modules each with 16 cells (96 grid cells).

Fig 1.5 consists of multiple grid modules stacked on top of each other with each grid module having different offset and size (fig on the left) and a graph (fig on the right) i.e SDR consisting of small square boxes where each small box represents a bit in an SDR and is associated with one of the cells from the left box. When an object (as shown by white square with black outline in the left figure) is at any given location on a space, the cells of all the modules associated with that location are activated. The exact location of the object is determined by looking at the space with the most overlap of the activated cells. The corresponding bits in the SDR of the overlapping cells in the left figure are activated which are represented by the blue box in the left figure.

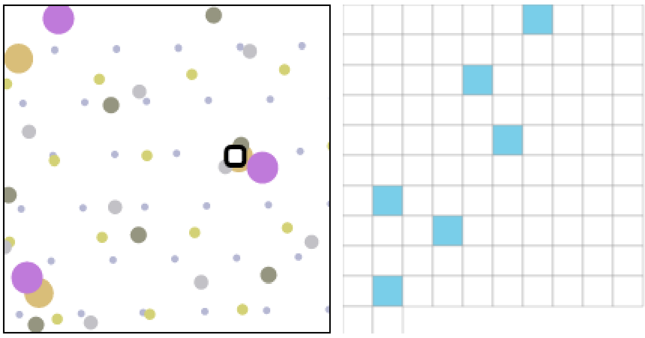

Figure 1.5: Grid cell modules as SDR [fige]

II. Methods

This section describes the methods for the implementation of the grid cell encoder to encode 2D coordinates (x,y). Each sub section explains the process and the mathematical formula required for the encoder implementation.

2.1 *Model of grid cell*

2.1.1 *SDR size and Sparsity*

The grid cell encoder takes two important inputs – the size of the SDR and the sparsity where the size of SDR allows the encoder to define the number of bits in the output SDR and the sparsity specifies the number of bits in a SDR that can be active at a given time. The encoder produces an SDR which consists of the desired number of bits (as per the input) where at any point in time a small percentage of the bits are 1's and the rest are 0's. The percentage of active bits (1's) is determined by the value of the sparsity. If two SDRs have active bits in the same locations, they share the semantic attributes represented by those bits.[5] By determining the overlap between two SDRs (the equivalent bits that are 1 in both SDRs) we can immediately see how two representations are semantically similar and how they are semantically different.

2.1.2 *Virtual rat and environment*

A virtual rat is simulated to explore an arena (a small enclosed space). The movements of the virtual rat alternated randomly at each step with a probability of 0.5 between translation and rotation which is combined with obstacle avoidance at the walls. On each virtual rat movement a coordinate (x, y) of the rat location is generated which is then used as an input to the grid cell encoder and a corresponding SDR is generated.

2.1.3 *Grid Cell Encoder Algorithm*

The Table 1 lists out the steps for the Grid cell encoder which are required to generate SDR from the 2D location coordinate. Each step has the sections listed that describe the step in more detail.

Step 1: Setup Grid cell modules (section 2.1.4)

Step 2: Map each grid modules to the bits on the output SDR (section 2.1.5)

Step 3: Apply offset and orientation to each grid modules (section 2.1.6 & 2.1.7)

Step 4: Simulate grid cell hexagonal grid pattern firing property (section 2.1.8)

Step 5: Activating the closest grid cells in each grid modules (section 2.1.9 & 2.1.10)

Table 1: Steps for the grid cell encoder algorithm

2.1.4 *Grid Cell modules*

The grid cell encoder takes an argument called "periods". The argument "periods" is a list of distances where the length defines the number of distinct grid cell modules. For example, periods of size 5 denote 5 grid cell modules. The period of a module is the distance between the centers of a grid cell's receptive fields (RF) which is a region of the sensory space in which a stimulus will modify the firing of the neuron.

For this paper, a period of length 5 (five grid cell modules) is used which are defined as follows.

$$periods_{(i)} = 6 * (2^{0.5})^{i} \qquad (1)$$

2.1.5 *Associate grid cell modules with output SDR*

Gird cell modules work together to uniquely map the space by combining their representation. Each cell in a grid cell module is like a bit in a SDR. The SDR is partitioned such that each grid cell module is responsible for generating bits for its associated range of cells in the SDR.

The SDR is evenly divided into intervals based on the number of grid cell modules. The partition (p) is derived by the following linspace formula.

$$p(i) = start + (i - 1) * (stop - start)\%(n - 1) \qquad (2)$$

where,

i $\in \{1, 2, \ldots, n\}$

start = 0

stop = size of SDR (100 for this experiment)

n = number of modules (5 for this experiment)

The result of this computation produces the output ([0, 20, 40, 60, 80, 100]) based on the input used in this experiment. As a result, the bits in the SDR from 0 to 20 are generated by the first grid cell module, the next bits from 21 to 40 are generated by the second grid cell module and so on. In this way each grid cell module is responsible for generating the active bits for the defined range of cells in SDR and all the bits are combined together to give a single big SDR.

2.1.6 *Grid cell modules offset and orientation*

Within local regions of the entorhinal cortex all grid cells have the same bump spacing (scale) and orientation (angle of the rows and columns), but differ in bump placement. The cells in the module share a single bump pattern that can be moved horizontally or vertically, with a different x- y-offset for each cell.

A random offset (distance) is generated for each grid cell. For this experiment, the random offset is generated randomly between the following range

$$0 \leq offset_{i}(x, y) \leq max(periods) \qquad (3)$$

where i = 0 .. size of the SDR

For each grid cell module a random orientation is generated. For this paper, the angle is generated between 0 and 1 and multiplied by 2 times the value of π

$$angle_{j} = rand(0, 1) * 2 * \pi \qquad (4)$$

where,

j = 0 .. length of periods (number of grid modules)

rand = generate uniform random value between 0 and 1

The rotation matrix for the grid modules is as follows:

$$rotation_k = \begin{bmatrix} \cos(angle_j) & -\sin(angle_j) \\ \sin(angle_j) & \cos(angle_j) \end{bmatrix} \quad (5)$$

2.1.7 *Applying the offset and rotation*

For a given input coordinate (x, y) which represents the virtual rat location in the given room, the offset is subtracted from the given x,y virtual rat coordinate which gives the change in the position called displacement (d).

$$d_i(x) = x - offset_i(x) \quad (6)$$

$$d_i(y) = y - offset_i(y) \quad (7)$$

After applying the offset, the rotation matrix is applied to the result of the above operation i.e displacement. As the number of rotation matrices is equal to the number of grid cell modules and the length of the calculated displacement is equal to the size of the SDR, each rotation matrix is applied for each partition range(equation 8). The dot multiplication between the rotation matrix and the displacement within the range m and n is as follow:

$$d_{m:n} = [rotation_k]\,.\,[d_{m:n}] \quad (8)$$

where,

k = 0 .. number of grid modules

i = lower limit of the partition range

j = upper limit of the partition range

2.1.8 *Convert into and out of hexagonal coordinates*

Grid cells in the dorsal segment of the medial entorhinal cortex show remarkable hexagonal activity patterns, at multiple spatial scales, during spatial navigation. The firing fields are organized into a hexagonal lattice. To achieve this property the offset coordinate is converted into cube coordinates. The cube coordinates are a reasonable choice for a hex grid coordinate system.

Given the offset coordinate x and y,

$$q_i = (d_i(x) * \sqrt{3} \div 3.0 - d_i(y) \div 3.0) \div Side \quad (9)$$

$$r_i = d_i(y) * 2.0 \div 3.0 \div Side \quad (10)$$

where,

$Side = radius / Cos(\pi/6)$

$radius = periods_k / 2$

Then the (x, y, z) value in the cube coordinate are as follows:

$$x_i = q_i \quad (11)$$

$$y_i = -q_i - r_i \quad (12)$$

$$z_i = r_i \quad (13)$$

The cube coordinates thus obtained are converted back to the regular m, n coordinate in 2d space which as a result rounds to the nearest hexagonal (RF) center.

$$m_i = Side * (\sqrt{3} * x_i + \sqrt{3}/2 * z_i); \quad (14)$$

$$n_i = Side * (3.0/2.0 * z_i); \quad (15)$$

2.1.9 *Distance between the location and the RF center*

The m,n coordinate obtained from the above equation is subtracted from the displacement gives the coordinate from the center of the grid cell.

$$x_i, y_i = d_i(x, y) - (m_i, n_i) \quad (16)$$

The distance (hypotenuse) from the center to the above coordinates can be calculated by the pythagoras theorem.

$$h_i = \sqrt{x_i^2 + y_i^2} \quad (17)$$

2.1.10 *Activate the closest grid cells in each module*

Each grid cell module controls the activation state of a certain range of bits in the SDR. The sparsity parameter is used to determine how many bits should be activated by each grid module.

Within each partition range the distance (h) is sorted by ascending order of its value. The lesser value shows that that grid cell is closer to the center. The number of grid cells closer to the center is activated within the sparsity factor. For example, if the sparsity is 0.15, the top 15 percent cells that are closer to the center are activated.

A sample code showing the selection of active cells (in C#) is shown in Table 2. The partitions variable on line 2 has the start and stop range of the cell controlled by each grid module. The variable ‘z’ on line 6 represents the total number of cells that should be activated and is controlled by the ‘sparsity’ variable. From line 8 to 15, each distance of each grid cell from center is sorted from closest to furthest and the corresponding indexes of the cells are obtained.

```
1. var activatedCells = new List<int>();
2. foreach (var partition in partitions)
3. {
4.     var start = partition[0];
5.     var stop = partition[1];
6.     var z = (int) (Math.Round(sparsity * (stop - start)));
```

```

    var indexes = distances[start..stop]
       .Select((x, i) =>
         new KeyValuePair<double, int>(x, i)
        )
       .OrderBy(x => x.Key)
       .Take(z)
       .Select((x, i) => x.Value + start)
       .ToList();

     activatedCells.AddRange(indexes);
}

activatedCells.Sort();
```

Table: 2. Sample code to find the active cells within sparsity.

2.2 *Parameters*

The value of the parameter used in this experiment is given in Table 3.

Parameter	Value	Description
size	100	Size of SDR
sparsity	0.15	Percentage of active bits in SDR
periods	[6, 8, 12, 16, 24]	distance between the centers of a grid cells receptive fields for each grid module
partitions	[[0,20], [20,40], [40,60], [60,80], [80,100],]	The partition divides the SDR bits to the total grid modules. Each item in the list represents the range of SDR bits controlled by each grid module.

Table: 3. Parameter values used in this experiment

III. Results

3.1 *Generate SDR*

To analyze the SDR produced by the proposed Grid cell encoder, the location obtained from the random navigation by the virtual rat (2.1.2) is used. Out of the many locations navigated by the virtual rat, three coordinates are selected for this paper to present the resulting SDR generated by the Grid Cell Encoder.

x	**y**
100	100
100	100.50
5000	400

Table: 4 Three coordinates obtained from the virtual rat navigation in a room

Each of the above (x, y) coordinates is passed through the grid cell encoder which applies the offset and the orientation as per 2.1.7 to these coordinates which gives the displacement.

The resulting displacement is converted to cube coordinates (2.1.8) and back to the offset coordinate to achieve the hexagonal property of the grid cells which in the process of conversion results in the coordinate nearest to the hexagonal center.

2.1.9 gives the distance between the center of the grid cell and the location. The closest grid cells are then activated (within the sparsity range) for each of the grid cell modules as described in 2.1.10.

Each active bit in the SDR has a value of 1 whereas the inactive bit has a value of 0. The SDR of the 3 coordinates are shown in Figure 3.1, 3.2 and 3.3.

The number of boxes is equal to the size of the SDR. The blue box represents the active bits whereas the white box represents the inactive bits.

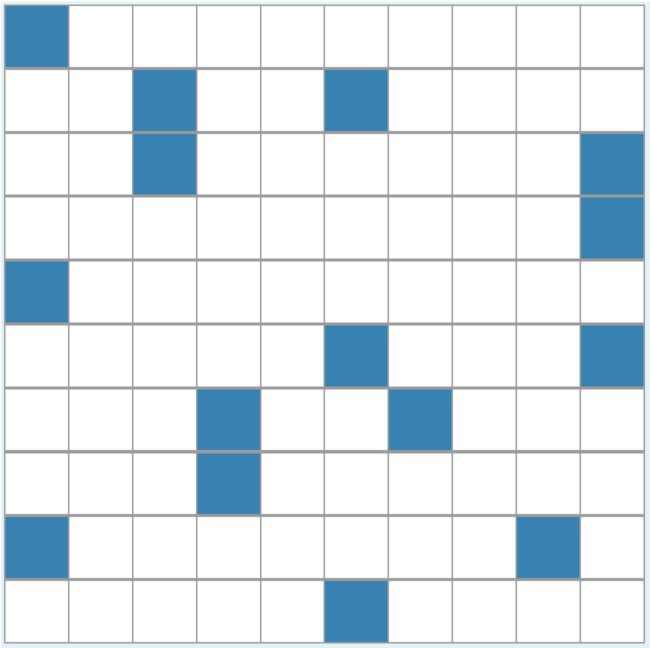

Figure 3.1: SDR for coordinate 100,100

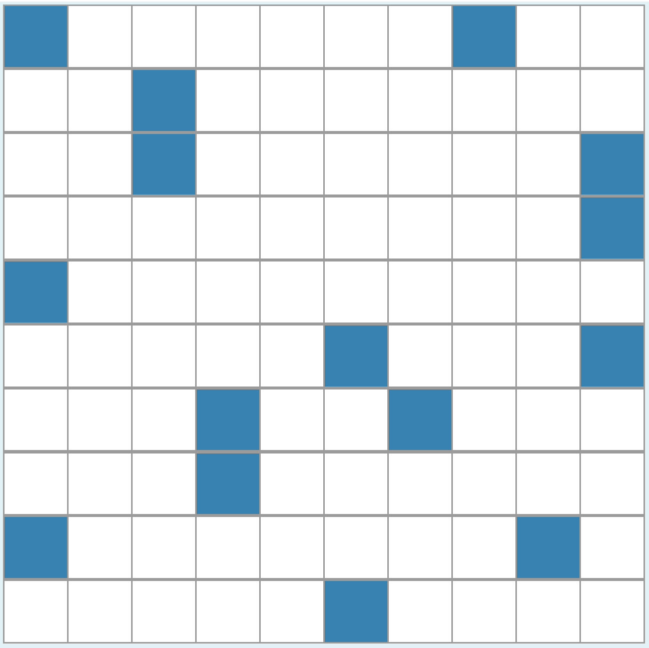

Figure 3.2: SDR for coordinate (100,100.50)

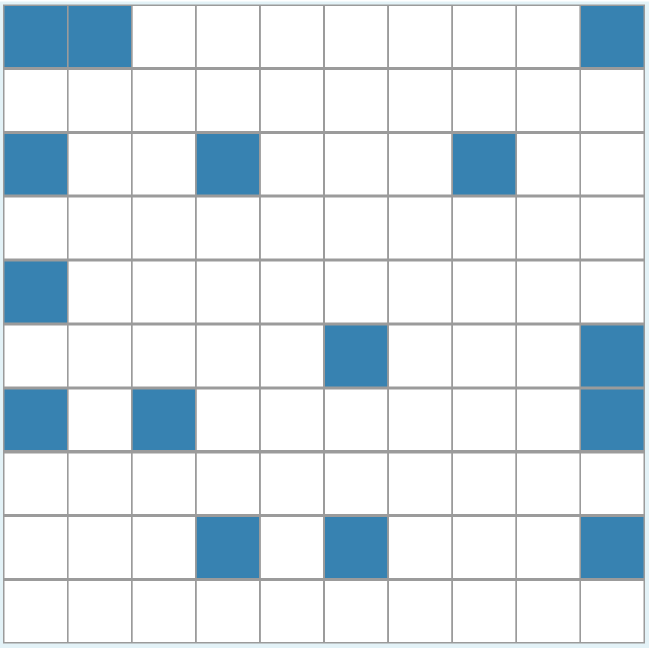

Figure 3.3: SDR for coordinate (5000, 400)

3.1 *Similarity matrix*

The elements of a similarity matrix measure pairwise similarities of objects - the greater similarity of two objects, the greater the value of the measure.

The similarity matrix for the resulting SDR of the three coordinates is calculated and presented in Table 3.

Coordinates (x,y)	100x100	100x100.5	5000x400
100x100	100	93.33	26.67
100x100.5	93.33	100	26.67
5000x400	26.67	26.67	100

Table 5: Similarity matrix for the three coordinates

From the similarity matrix it can be observed that although the coordinates (100,100) and (100, 100.5) are really close, the method to generate SDR proposed in this paper is still able to produce a unique SDR for such similar input coordinates. The coordinates (5000, 400) produce SDR which is only 26.67% similar to the other two coordinates as it is very far from the other two coordinates.

IV. Discussion

Every learned environment is associated with a set of unique locations. Upon entering a learned environment, entorhinal grid cell modules select which grid cells in each module should be active at the current location. This suggests that the current location and all the locations that the animal can move to in that environment will, with high certainty, have representations that are unique to that environment.

Combining these properties, it can be used to broadly describe how grid cells represent an environment such as a room. Each location representation in the set is unique to that environment and will not appear in any other environment. An environment consists of all the locations that the animal can move among, including locations that have not been visited, but could be visited.

This paper gives an overview of one such method to represent the location of such an environment with a unique SDR using grid cells. This generated SDR can be used to encode the set of semantic attributes of the location which can be fed to the HTM system. Similar locations produce highly overlapping SDRs while the same location produces the same SDR as output.

The parameters – size, sparsity and periods – mentioned in this paper may not be the best ideal values. The value depends on the requirement and more deep investigation can help to best understand how the change in such parameters affects the performance and reliability of the encoder.

It has also been proposed that grid cells may be part of a generalized path integration system. This raises the question of how such a system can be implemented in the brain. In addition, since a path integration system is inherently prone to the accumulation of error, it remains to be determined how the grids can be anchored to allothetic information.

Grid cells may be the key elements of a modularly organized network for metric spatial representation. The neural map is thought to represent self-location based on integration of speed and direction signals during movement. While the contours of some possible mechanisms can be apprehended, much work is needed to understand how the grid pattern is generated in single cells and in neural networks of the entorhinal cortex and beyond.

References

[1] Numenta.(2021) How grid cells map space. Available at: https://www.numenta.com/blog/2018/05/25/how-grid-cells-map-space/ (Accessed: November 21, 2022).

[2] Authors Dori Derdikman et al. (no date) How do we find our way? grid cells in the brain, Frontiers for Young Minds. Available at: https://kids.frontiersin.org/articles/10.3389/frym.2021.678725 (Accessed: November 21, 2022).

[3] D'Albis, Tiziano. (2018). Models of spatial representation in the medial entorhinal cortex. 10.18452/19306.

[4] Edvard Moser and May-Britt Moser (2007) Grid cells. Scholarpedia, 2(7):3394.

[5] Purdy, S.. (2016). Encoding Data for HTM Systems.

[6] Rowland, David & Roudi, Yasser & Moser, May-Britt & Moser, Edvard. (2016). Ten Years of Grid Cells. Annual review of neuroscience. 39. 10.1146/annurev-neuro-070815-013824.

[7] Guanella, Alexis & Kiper, Daniel & Verschure, Paul. (2007). A model of grid cells based on a twisted torus topology. International journal of neural systems. 17. 231-40. 10.1142/S0129065707001093.

[8] Hawkins, Jeff & Lewis, Marcus & Klukas, Mirko & Purdy, Scott & Ahmad, Subutai. (2018). A Framework for Intelligence and Cortical Function Based on Grid Cells in the Neocortex. 10.1101/442418.

[9] Wei, X. X., Prentice, J., & Balasubramanian, V. (2015). A principle of economy predicts the functional architecture of grid cells. eLife 2015;4:e08362

[10] McNaughton, B., Chen, L., and Markus, E. (1991) "dead reckoning", landmark learning, and the sense of direction: a neurophysiological and computational hypothesis. J. Cogn. Neurosci., 3:190–202.McNaughton, B., Chen, L., and Markus, E. (1991) "dead reckoning", landmark learning, and the sense of direction: a neurophysiological and computational hypothesis. J. Cogn. Neurosci., 3:190–202.

[11] Barry, C., Hayman, R., Burgess, N., and Jeffery, K. J. (2007) Experience-dependent rescaling of entorhinal grids. Nat. Neurosci., 10:682–684.

[12] Barry, C., Hayman, R., Burgess, N., and Jeffery, K. J. (2007) Experience-dependent rescaling of entorhinal grids. Nat. Neurosci., 10:682–684.

[13] Krupic, J., Bauza, M., Burton, S., Barry, C., and O'Keefe, J. (2015) Grid cell symmetry is shaped by environmental geometry. Nature, 518:232–235.

[14] Burgess, N., Barry, C., and O'Keefe, J. (2007) An oscillatory interference model of grid cell firing. Hippocampus, 17:801–812.

[figa] D'Albis, Tiziano. (2018). Models of spatial representation in the medial entorhinal cortex. 10.18452/19306.

[figb] D'Albis, Tiziano. (2018). Models of spatial representation in the medial entorhinal cortex. 10.18452/19306.

[figc] D'Albis, Tiziano. (2018). Models of spatial representation in the medial entorhinal cortex. 10.18452/19306.

[figd] How grid cells map space (2021) Numenta. Available at: https://www.numenta.com/blog/2018/05/25/how-grid-cells-map-space/ (Accessed: November 21, 2022).

[fige] How grid cells map space (2021) Numenta. Available at: https://www.numenta.com/blog/2018/05/25/how-grid-cells-map-space/ (Accessed: November 21, 2022).