A Model of Heteroassociative Memory: Deciphering Surprising Features and Locations

Shashank Bhatia and **Stephan K. Chalup**
School of Electrical Engineering and Computer Science
The University of Newcastle
Callaghan, NSW 2308 Australia
shashank.bhatia@uon.edu.au, stephan.chalup@newcastle.edu.au

Abstract

The identification of surprising or interesting locations in an environment is an important problem in the fields of robotics (localisation, mapping and exploration), architecture (wayfinding, design), navigation (landmark identification) and computational creativity. Despite this familiarity, existing studies are known to rely either on human studies (in architecture and navigation) or complex feature intensive methods (in robotics) to evaluate surprise. In this paper, we propose a novel heteroassociative memory architecture that remembers input patterns along with features associated with them. The model mimics human memory by comparing and associating new patterns with existing patterns and features, and provides an account of surprise experienced. The application of the proposed memory architecture is demonstrated by identifying monotonous and surprising locations present in a Google Sketchup model of an environment. An inter-disciplinary approach combining the proposed memory model and isovists (from architecture) is used to perceive and remember the structure of different locations of the model environment. The experimental results reported describe the behaviour of the proposed surprise identification technique, and illustrate the universal applicability of the method. Finally, we also describe how the memory model can be modified to mimic forgetfulness.

Introduction

Within the context of evaluating computational creativity, measures of accounting surprise and identifying salient patterns have received great interest in the recent past. Known by different names, the problem of accounting surprise has been applied in various research areas. Specifically, the problem of identifying locations that stimulate surprise has important applications in areas such as robotics, architecture, data mining and navigation. Robotics researchers, while aiming towards robot autonomy, intend to identify locations that can potentially serve as landmarks for the localisation of a mobile robot (Cole and Harrison 2005; Siagian and Itti 2009). Architects, on the other hand, intend to design building plans that comprise sufficient salient/surprising locations in order to support way-finding by humans (Carlson et al. 2010). Lastly, navigation experts mine existing maps to identify regions/locations that can serve to better communicate a route to the users (Xia et al. 2008; Perttula, Carter, and Denoue 2009). Common to all these applications is the underlying question, the problem of identifying patterns from raw data that appeal or stimulate human attention. While the aim of these applications is same, the underlying measure of accounting surprise that each one follows has been designed to suit only the respective application. There are no domain-independent methods available that are flexible enough to be adaptable universally. Itti (2009) and Baldi (2010) rely on Bayesian statistics, and their method would require considerable domain-specific alteration, as can be seen in (Ranganathan and Dellaert 2009; Zhang, Tong, and Cottrell 2009). On one hand, designing methods that are domain-independent having capacity of comparing multi-dimensional data is a challenging task. On other hand, the use of dimensionality reduction techniques to limit or reduce dimensionality are known to cause bias. The reduction of dimensions would depend on methods employed, and different methods may assign varying weights to each dimension (Brown 2012). This makes surprise measurement, which includes comparing multi-dimensional patterns, a challenging problem.

Commonly known as outlier detection, novelty detection, saliency detection etc., the question of detecting a "surprising event" has been raised in the past (Baldi and Ittii 2010). Specifically, the methods that provide a domain-independent approach for discovering inherent surprise in perceived patterns aim for information maximisation. In an information-theoretic sense, patterns that are rare are known to contain maximum information (Zhang, Tong, and Cottrell 2009). In a more formal sense, patterns that lead to an increase in entropy are deemed as unique, and are known to cause surprise (Shannon 2001). Another argument in the literature is about the frequency of occurrence of such patterns. An event/pattern that has a lower probability of occurrence/appearance, is deemed rare. Therefore, various proposals have been made that compare probabilities (Bartlett 1952; Weaver 1966) and identify the pattern with the lowest probability value. These techniques were further refined to consider the probabilities of all other patterns as well (Weaver 1966; Good 1956; Redheffer 1951). Most recent developments use Bayesian statistics to compare the probabilities of the occurrence of patterns or features extracted from them. Baldi and Ittii (2010) proposed to employ a distance metric to measure the differences between prior and posterior beliefs of a computational observer, and argued its interpretation to be that of an account of surprise. The authors proposed the use of Kullback-Leibler divergence (Kullback 1997) as the distance metric, and discussed its advantages over Shannon's entropy (Shannon 2001). They demonstrated the use of their proposed method by identifying surprising pixels from an input image. The complex mathematical constructs of modelling surprise that exist in the literature are difficult to adapt, and therefore have not found their applications across different domains.

The concept of surprise can also be understood through its relationship to memory. Something that has not been observed stimulates surprise. In this setting, if a computational agent remembers the percepts presented to it, a measure of surprise can be derived. Baldi and Ittii (2010) follow this idea, but their perceptual memory is in the form of a probabilistic model. The patterns that are already observed compose the prior model, and the model obtained after adding new percepts is the posterior. As noted previously, most often the patterns/features to be evaluated are available in the form of a vector quantity (Brown 2012). Conversion of this multi-dimensional quantity into a probabilistic model not only requires specific expertise, but is also sensitive to the method employed to update the model's parameters. Even after substantial effort in design, the memory is sensitive to the parameters employed for the model. These shortcomings of the state-of-the-art methods form one part of motivation behind the current paper.

Another aspect that is ignored in most contemporary methods is the associative nature of memory. Human memory has a natural tendency to relate/associate newly perceived objects/patterns with those perceived in the past. Recent research in cognitive science supports the influence perceptual inference has on previous memory (Albright 2012). A classical example is the problem of handwritten digit recognition. Multiple handwriting patterns corresponding to the same digit are labelled and associated via the same label. Since the memory is always trying to associate new patterns with previous experience, it is obvious that a strong association will lead to lower surprise and vice versa. This property of association, though well-recognised, has not been incorporated in the state-of-the-art methods of measuring surprise. This forms the second motivation of the current paper.

Inspired by the discussed shortcomings of existing methods, this paper presents a computational memory framework that can memorise multi-dimensional patterns (or features derived from them) and account for inherent surprise after attempting to associate and recall a new pattern with those already stored in the memory. The uniqueness of the memory model is two-fold. Firstly, it can be employed without converting the perceived patterns into complex probabilistic models. Secondly, for the purpose of accounting surprise, the memory model not only aims to match and recall the new pattern, but also attempts to associate its characteristics/features before deeming it surprising. To illustrate these advantages and their usage, the proposed method is employed to identify monotonous and surprising structural features/locations present in an environment. Noted previously, this is an important problem in the field of robotics as well as architecture, and therefore we use a Google Sketchup (Trimble 2013) based architectural model for the demonstration. An isovist - a way of representing visible space from a particular location (Benedikt 1979) - is used for the purpose of perceiving a location in the form of a multi-dimensional pattern. This paper points towards the methods of extracting isovists from respective environments (section: Spatial Perception), and provides details of the neural network based memory architecture (section: Associative memory). Experimental results compare the degree to which identified monotonous locations associate with each other, and illustrate the isovist shape of those that stimulate computational surprise (section: Experiments & Results). Additionally, we describe how the proposed memory model can be modified to mimic forgetfulness, thereby forgetting patterns that have not been seen in a given length of time. To conclude, the paper provides a discussion on prospective applications of the proposed framework, and demonstrates its universality by evaluating its performance in a classification task, on various pattern classification datasets (section: Conclusions & Discussoins).

Spatial Perception

This work utilises multi-dimensional Isovist patterns to perceive/represent a location. Conceptually, an isovist is a geometric representation of the space visible from a point in an environment. If a human were to stand at a point and take a complete $360°$ rotation, all that was visible forms an isovist. In practice, however, this 3D visible space is sliced horizontally to obtain a vector that describes the surrounding structure from the point of observation, also known as the vantage point. This 2D slice is essentially a vector composed of lengths of rays projected from the vantage point, incident on the structure surrounding the point. Therefore, if a $1°$ resolution was utilised, an isovist would be a 360-dimensional vector, $\vec{I} = [r_1, r_2, \ldots, r_{360}]$ where r_θ represents the length of the ray starting from the vantage point, and incident on the first object intersected in the direction θ. This way, an isovist records a profile of the surrounding structure (illustrated in figure 1). In an environment, multiple isovist can be generated from different vantage points. Each isovist can be represented as a 360-dimensional pattern describing the structure visible from the vantage point. In this paper, an indexed collection of isovist patterns extracted from an existing model of the environment is used.

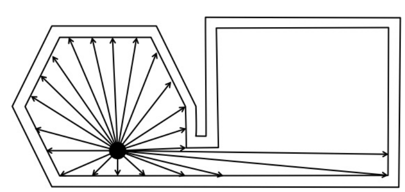


Figure 1: A hypothetical 2D plan of an environment, showing a vantage point (black dot) and the corresponding isovist generated from the vantage point.

Isovist Extraction

The method of extraction of isovists employed in this paper is derived from our previous work (Bhatia, Chalup, and Ostwald 2012), where we employed a Ruby script that executes on the Google Sketchup platform and extracts 3D isovists from a Google Sketchup model. This records the isovists while using the "walk through" tool provided in Google Sketchup. The "walk through" tool allows a user to walk through a 3D model of an architectural building plan. However, in this work, we utilise modified version of the Ruby script that extracts a 2D slice of the perceived 3D isovist. The model of a famous architectural building, Villa Savoye, is used to extract the isovist and identify the surprising locations present. The building is known for uniqueness of its structure, and therefore provides good examples for the evaluation of surprising locations.

Inputs and association patterns

An isovist records a spatial profile, and can be used to memorise a location by a computational memory. This is an advantage while trying to recognise/identify a location by its isovist; however, becomes a drawback when the aim is to infer surprise through association. A simple example is the case of two rectangular rooms that are similar in shape, but have different side lengths. While the isovists recorded at the central point of these rooms would have a large difference, the number of straight edges, and the angles they make, remain the same (90°). Therefore, for the purpose of associating and finding similarities between two locations, in this paper we employed a 3-dimensional feature vector derived from isovist pattern. We compute *(i)* Area of the isovist, *(ii)* Eccentricity value, and *(iii)* Circularity value to form the elements of the 3-dimensional associated feature pattern. This feature pattern is used to associate two isovist patterns. The perceived isovist pattern, therefore, comprises a 360-dimensional vector, and the derived associated pattern is a 3-dimensional feature vector. The isovist of a location and the feature vector are presented as a pair to the memory model proposed in this paper. The memory model remembers essential patterns and computes surprise after associating new patterns and comparing existing ones. Due to the association task that the memory performs, such memories are known as Associative Memories (Palm 2013).

Associative Memory

Associative memories are computational techniques, capable of storing a limited set of input and associated pattern pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$. Depending on the size of the input vector x_i, its associated pattern y_i, and methods of association, various types of such memories are proposed. Kosko (1988) was the first to introduce Bidirectional Associative Memories (BAM), which provides a two way association search mechanism termed Heteroassociation. A BAM can take either the input or associated pattern as its input and has the capacity to recall the respective association. Despite the utility BAM can offer, its usage has been limited due to many existing challenges, such as limited capacity and conditions of instability. Importantly, existing variations of BAM can only memorise binary patterns. Many other variations of BAM have been offered, however, and the present note is provided only as a basis for the following discussion and is by no means an exhaustive account of the developments on this topic. A detailed review can instead be found in (Palm 2013). The proposed memory model offers similar functionality without requirement for input patterns to be binary in nature.

Overview of the architecture

The architecture of the proposed memory model consists of two memory blocks, and can be divided into three parts. *(a)* **Input Memory Block (IMB):** block that stores input patterns, *(b)* **Associated Memory Block (AMB):** block that stores associated feature vectors/patterns, *(c)* **Association Weights:** a matrix that maintains a mapping between the two memory blocks. Complete architecture of the memory is represented in figure 2.

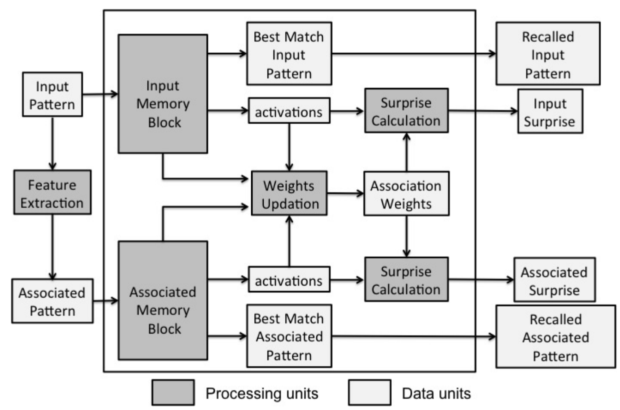


Figure 2: Memory Architecture: Comprise two memory blocks and association weights, all linked through one or more data/processing units presented in white and grey colour respectively.

The memory blocks are the storage units responsible for memorising input and associated patterns. This memory model in concept works similar to traditional BAMs except that it provides additional many-to-many mapping functionality on real-valued vectors. Input patterns (which in the case of this application are isovist vectors) when presented to the memory model are compared in two respects: *(a)* similarity of shape, and *(b)* similarity of the features derived from them. The detailed construction and working of each block and the overall memory model is provided in the following subsections.

Memory Blocks

The smallest unit of storage in this memory model is a Radial Basis activation unit, also known as a Radial Basis Function (RBF). Typically, a RBF is a real valued function with its response monotonically decreasing/increasing with distance from a central point. The parameters that describe a RBF include the central point c, distance metric $\| \cdot \|_d$ and the shape of the radial function. A Gaussian RBF with Euclidean distance metric and centre c_i is defined in equation 1. The parameters c_i and radius σ_i decide the activation level of the RBF unit. Any input x lying inside the circle centred at c_i having a radius less than or equal to σ_i will result in a positive activation, with the level of activation decreasing monotonically as the distance between the input and the centre increases.

$$\phi_i(x) = \exp\left(-\frac{(x-c_i)^2}{\sigma_i^2}\right) \quad (1)$$

The realisation of a memory element in our approach is done by saving the input as the centre c_i, and adjusting the value of the radius σ_i to incorporate values that lie close to each other. Mathematically, this memory element will have $\phi_i(x) > 0$ activation for all values of x that fall in a σ_i neighbourhood of the point c_i defined in equation 2. Further, $\lim_{x\to c_i} \phi_i(x) = 1$. This condition ensures that the activation unit with the centre c_i closest to the current input x activates the most.

$$B(c_i;\sigma_i) = \{x \in X \mid d(x,c_i) < \sigma_i\} \quad (2)$$

In a collection of multiple RBF units, with each having a different centre c_i and radius σ_i, multiple values can be remembered. If an input x is presented to this collection, the unit with highest activation will be the one that has the best matching centre c_i. Or in other words, for the presented input value, the memory block can be said to recall the nearest possible value c_i. For one input pattern, there will be one corresponding recall value. This setting of multiple RBF units can thus work as a memory unit. The Memory Blocks described previously comprise multiple RBF units. As an example, a memory block comprising n RBF units can be represented with figure 3.

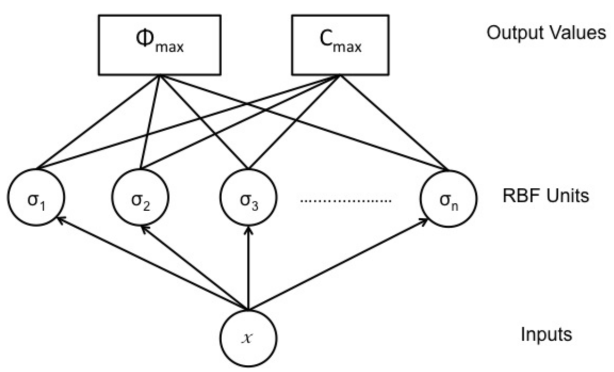


Figure 3: RBF Memory Block: Each RBF unit stores one data value in the form of centre c_i; the range of values for which the unit has positive activation are defined by the values of σ_i according to equation 2. c_{max} is the value that the memory recalls as the best match to the input, and ϕ_{max} represents the confidence in the match.

So far we have described the use of the RBF unit as a memory block having a scalar valued centre c_i. In order to memorise a multi-dimensional pattern (in this application an isovist pattern, comprising 360 ray-lenghts), we modify the traditional RBFs to handle a multi-dimensional input isovist vector $\vec{x}$ by replacing its scalar valued centre with a 360-dimensional vector $\vec{c_i}$. While Euclidean distance and dot product of two multi-dimensional vectors are also scalar and do not disrupt the working of standard RBFs, their capacity to capture the difference in shape between two isovist patterns is minimal. Therefore, in order to account for difference in shape, we replace the Euclidean distance metric by Procrustes Distance (Kendall 1989). The procrustes distance is a statistical measure of shape similarity that accounts for dissimilarity between two shapes while ignoring factors of scaling and transformation. For two isovist vectors $\vec{x}_m$ and $\vec{x}_n$, the procrustes distance $\langle \vec{x}_m, \vec{x}_n \rangle_p$ first identifies the optimum translation, rotation, reflection and scaling required to align the two shapes, and finally provides a minimised scaled value of the dissimilarity between them. An example of two similar and non-similar isovists with their procrustes-aligned isovists is shown in figure 4. Utilising procrustes distance with the multidimensional centre $\vec{c_i}$, we term this Multidimensional Procrustes RBF, which is defined as:

$$\phi_i(\vec{x}) = \exp\left(-\frac{\langle \vec{x}, \vec{c_i} \rangle_p^2}{\sigma_i^2}\right) \quad (3)$$

Procrustes distance provides a dissimilarity measure ranging between 0 and 1. A zero procrustes distance therefore leads to maximum activation and vice versa. A multidimensional procrustes RBF has the capacity to store a multi-dimensional vector in the form of its centre. It is important to note that for the application described in this paper, the difference between two multi-dimensional vectors, viz. the isovists, was recorded using procrustes distance. However, in general the memory model can be adapted for any suitable distance metric, or used with the simple Euclidean distance. The use of procrustes distance as a distance metric was adapted specifically for the purpose of the application of identifying surprising locations in an environment.

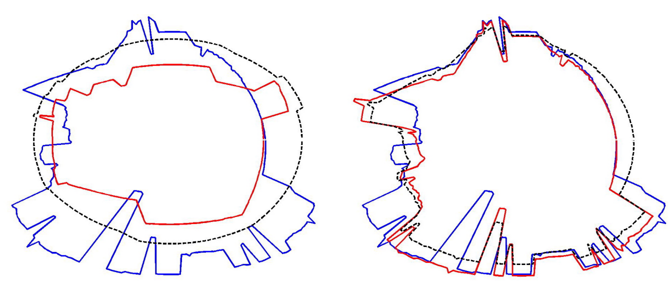


Figure 4: Two isovist pairs (illustrated in red and blue) and corresponding aligned isovists (black dashed), one with a high procrustes distance (left) and other with a low procrustes distance (right).

IMB and AMB IMB and AMB are in principle collections of one or more multidimensional-procrustes RBF and multidimensional RBF units respectively, grouped together as a block (such as the one represented in figure 3). Each block is initialised with a single unit that stores the first input vector (for IMB) and derived features (for AMB). The feature vector employed to associate two input patterns (in this application isovists) comprise *(i)* area, *(ii)* circularity, *(iii)* eccentricity, together making up a 3-dimensional vector. Initially, each block is created with a single memory unit having a default radius 0.1. Thereafter, the memory block adapts one of the two behaviours. For new patterns that lie far from the centre, the memory block grows by incorporating a new RBF unit having its centre same as the presented pattern. On the other hand, for patterns that lie close to existing patterns, the radii of the RBF units are adjusted in order to obtain positive activation. Adjustment of the radii is analogous to adjustments of weights performed during the training of a Neural Network. The procedure followed to expand or adjust the radii can be understood by following algorithms 1 & 2. Consider a memory block comprising k neural units, with their centres $\vec{c_1}, \vec{c_2}, \ldots, \vec{c_k}$ and radii $\sigma_1, \sigma_2, \ldots, \sigma_k$ and the distance metric $\langle\cdot\rangle_d$. Let the model be presented with a new input vector $\vec{x}$. The algorithm 1 first computes $\langle\cdot\rangle_d$ distance (procrustes distance in the case of an isovist block) between each central vector and the presented pattern, and compares the distance with pre-specified best and average match threshold values Θ_{best} and $\Theta_{average}$. If the distance value is found as $d \leq \Theta_{best}$, the corresponding central vector is returned - as this signifies that a similar pattern already exists in memory. However, in the case where $\Theta_{avg} \leq d < \Theta_{best}$, the radius of the corresponding best match unit is updated. This updating ensures that the memory responds with a positive activation when next presented with a similar pattern.

Algorithm 1 Memory Block Updation

Require: $\vec{x}$, $[c_1, c_2, \ldots, c_k]$, Θ_{best}, Θ_{avg} , Σ
- **for all** center vectors c_i **do**
  - $d_i(\vec{x}) \Leftarrow \langle \vec{x}, \vec{c_i} \rangle_d$
- **end for**
- $bestScore \Leftarrow \min\limits_i (d_i)$
- $bestIndex \Leftarrow argmin\,(d_i)$
- $blockUpdated \Leftarrow$ false
- **if** $(\Theta_{best} \leq bestScore)$ **then**
  - $\vec{r} \Leftarrow \vec{c}_{bestIndex}$
  - $blockUpdated \Leftarrow$ true
- **else if** $(\Theta_{avg} \leq bestScore < \Theta_{best})$ **then**
  - **if** $(\sigma_{bestIndex} < \Sigma)$ **then**
    - $[\vec{c}_{bestIndex}, \sigma_{bestIndex}] \Leftarrow$ computeCenter()
    - $blockUpdated \Leftarrow$ true
  - **end if**
- **end if**
- **if** $(blockUpdated ==$ false$)$ **then**
  - add new neural unit center with
  - $\vec{c}_{k+1} = \vec{x}$
  - $\sigma_{k+1} = 0.1$
- **end if**

The network expands on the presentation of patterns that cannot be incorporated by adjusting the weights/radii of the RBF units. This feature provides three advantages over the traditional BAMs. The first is that there is no a-priori training required by the memory block. The memory is updated as new patterns are presented, and the training is online. Secondly, adjustment of weights ensures that similar patterns are remembered through a common central vector, thereby reducing the number of neural units required to remember multiple patterns. Despite the averaging process, a high level of recall accuracy is guaranteed by maintaining all radii $\sigma_i \leq \Sigma$. The values of Θ_{best}, Θ_{avg} and Σ are application specific parameters that require adjustment. However, for the purpose of associating and remembering isovists, in our application we determined these using equations $4, 5, 6$. Here, D_{ij} is a $n \times n$ matrix containing $\langle\cdot\rangle_p$ distances between all central vectors; $std(D_{ij})$ stands for standard deviation.

Algorithm 2 Center vector and radius calculation

Require: $\vec{c}_{bestIndex}, \Theta_{best}, \vec{x}$

$\vec{c}_{old} \Leftarrow \vec{c}_{bestIndex}$

$\vec{c}_{bestIndex} \Leftarrow (\vec{c}_{bestIndex} + \vec{x})/2$

$d_{new} \Leftarrow \frac{\left(\langle \vec{x}, c_{best\vec{I}ndex}\rangle_d\right)^2}{-2\cdot\log(\Theta_{best})}$

$d_{old} \Leftarrow \frac{\left(\langle \vec{c}_{old}, c_{best\vec{I}ndex}\rangle_d\right)^2}{-2\cdot\log(\Theta_{best})}$

$\sigma_{bestIndex} \Leftarrow \max(d_{new}, d_{old})$

$$
\begin{aligned}
D_{ij} &= \langle \vec{c_i}, \vec{c_j} \rangle_p \\
S_d &= \sum_{i \neq j}^{n} D_{ij} \\
\Theta_{best} &= \frac{percentile(S_d, 95)}{S_d} \quad (4) \\
\Theta_{avg} &= \frac{percentile(S_d, 50)}{S_d} \quad (5) \\
\Sigma &= \frac{\min(std(D_{ij}))}{\max(std(D_{ij}))} \quad (6)
\end{aligned}
$$

Association Weights Association weights act as a separate layer of the network architecture, and play the role of mapping the input patterns with their associated features. For a case of m isovist patterns and n associated feature vectors stored in IMB and AMB respectively, the association weights would comprise a $(m \times (n+1))$ matrix. The first column of the matrix contains the indices of each central vector $\vec{c_i}$ and the remaining columns contain mapping weights. On initialisation, the mapping weights are set to zero. Once each memory block is updated, the corresponding best match index obtained as an output of the memory block is used to configure the values of the matrix. Let q be the index returned from IMB, and r be the index obtained from AMB. The weight updation simply increments the value at the q^{th} row and $r + 1^{th}$ column of the weight matrix. If such a row or column does not exist (signifying a new addition to the memory block), a new row/column is added. During the use of the memory model to recall the associated vector from the presented input vector, assuming an index p was returned, the p^{th} row is selected, and the index of the column containing the highest score is obtained.

Let this index be k. If the highest score in k^{th} column this implies that for AMB, the centre of the k^{th} activation unit is most strongly associated with the current input. This kind of mapping look-up can be performed vice versa as well and provides an efficient bi-directional many-to-many mapping functionality, which is hard to implement in traditional memory models.

Surprise Calculation The Kullback-Leibler (KL) divergence (Kullback 1997) is a measure of difference between two probabilistic models of current observations. To estimate KL divergence, an application specific probabilistic model of the current data is required, and in most cases the design of such a model requires specific expertise. In our approach, each memory model computes the surprise without having the need to train/estimate or design any probabilistic model. This is achieved by using activation scores that each memory unit outputs on presentation of a pattern. These scores are obtained through RBF activation units. Each score in principle is therefore a probabilistic estimate of the similarity between the input vector and the centre of the corresponding memory unit. Exploiting this property, we measure the KL divergence on activation scores. On presentation of a new input vector $\vec{x}$ to a memory block, the activation scores are first computed. Since these scores are calculated before the block updates (using algorithm 1 & 2), they are termed a-priors, $A = [a_1, a_2, \ldots, a_n]$. Post the execution of algorithm 1, the memory block would either remain the same (in the case of best match), or change one of its radius values (for average match), or lastly may have an additional neural unit (no match). Accordingly, the activation scores obtained after the updating might be different from the a-priors. Scores obtained after the updating of memory are termed posteriors, $P = [p_1, p_2, \ldots, p_m]$. If $n < m$, the a-priors are extrapolated with the mean value of A to ensure $m = n$, and finally the KL-divergence or the surprise encountered is computed as:

$$S = \sum_{i=1}^{m} \ln\left(\frac{p_i}{a_i}\right) \cdot p_i \tag{7}$$

Here a_i and p_i are a-prior and posterior activation scores respectively. IMB and AMB each provides an estimate of the surprise encountered by each block. Surprise value from IMB indicates the surprise in terms of shape of the isovist (in the current application), and one from AMB indicate the surprise encountered in terms of associated features. Overall surprise is an average of the two surprise values. Illustration of the surprise values returned from AMB along with the values in the input vector are presented in figure 5a. Calculation of surprise in the memory model has two advantages, one that the user does not need to meticulously design of a probabilistic model and second that the surprise calculation is independent of the number of dimensions of the input vector.

Forgetfulness in memory

In order to imitate human memory more closely, one additional functionality that can be added in the presented memory model is the property of forgetting. The principal of "out of sight is out of mind" can be implemented in the presented memory model by the use of a bias value for each memory unit. Diverting from the traditional use of bias values, in our approach a bias value is used to adjust the activation score in such a way that the most recently perceived or activated memory unit attains a tendency to have higher activation score, and vice versa. This is achieved by decrementing the bias values of the units that were not recalled. In this way, if a pattern is presented once to the memory and is never recalled, that pattern will have the lowest bias. The effect of low bias will be low levels of activation, and therefore a low recall rate. This feature is an important consideration when evaluating "what causes surprise" and is therefore programmed as an optional configuration that can be used in the current memory model. However, for the current evaluation of surprising locations, it is assumed that the perceiver will not forget any location that was presented earlier.

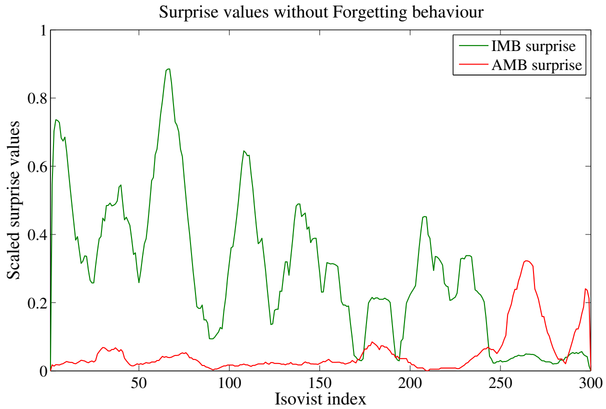
Experiments & Results

Deciphering surprising structures

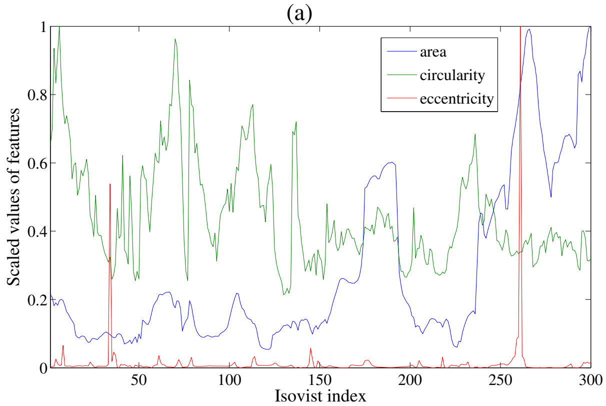
The isovist patterns extracted from the Google Sketchup models along with the feature vector (described earlier) were presented one at a time to the memory model. For the present application, the values of Θ_{best} and Θ_{avg} were appropriately selected to ensure that the change in size of the location, viz. the value of area, does not contribute to the value of AMB surprise. This was deliberately designed to serve the purpose of the present application, viz. deciphering surprising locations. The aim in our application was to consider a location surprising largely based on the surprise caused by its shape (isovist) and, to a limited extent, by the associated features. Hence only regions that differ in shape as well as in the values of derived features tend to be most surprising. The plot in figure 5(a) illustrates the values of surprise (ordinate) obtained from IMB and AMB for each isovist index (abscissa). As evident, the values of IMB surprise are initially very high, since the memory model has not been exposed to any isovist patterns. As the memory is presented with more isovist patterns (represented by increasing index of isovists), the surprise initially fluctuates, and then gradually decreases. On the other hand, AMB surprise always retains low values due to the low value of match thresholds Θ_{best} and Θ_{avg} chosen for AMB. However, despite low match thresholds, the AMB surprise was highest at two locations where the associated feature values peaked (illustrated in figure 5(b)). Again, this sudden drift was surprising and was very well captured by the computed surprise shown in the same plot. The view of the location corresponding to locations with highest and lowest surprise values are presented in figure 5(c). The views are recorded from the Google Sketchup model.

Forgfullness demonstration

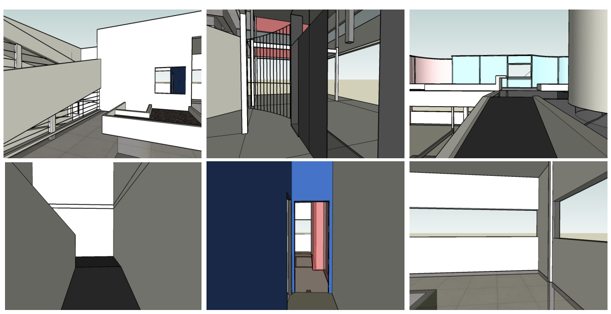
The behaviour of IMB and AMB surprise - while having the forgetting behaviour enabled - can be very well verified from figure 6(a) and (b). Figure 6(a) presented the IMB and AMB surprise values obtained with the same experiment comprising 300 isovists. Unlike figure 5(a), this time the gradual de-



(a)



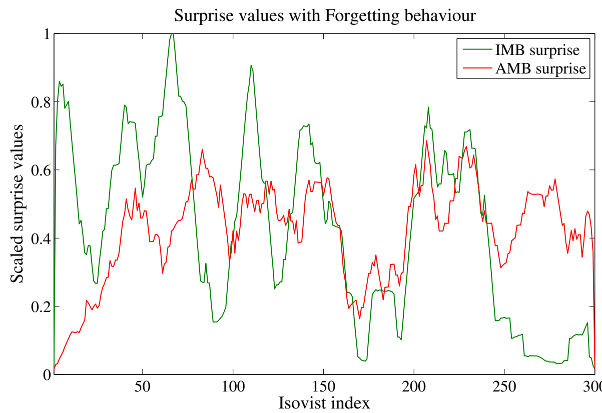
(b)

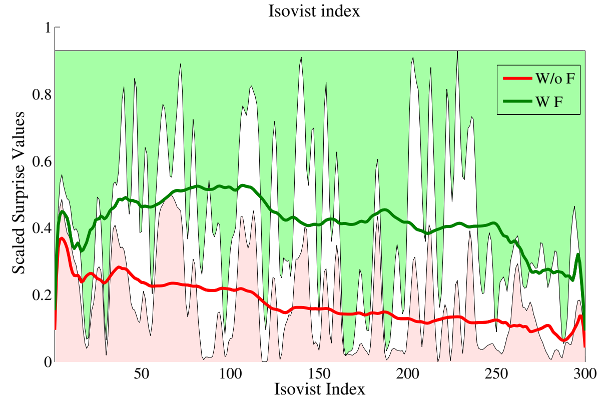


(c)

Figure 5: The figure illustrates the results of surprise evaluation of IMB and AMB, without forgetfulness behaviour. 5(a) presents scaled values of IMB and AMB surprise, and 5(b) presents scaled values of associated features. 5(c) illustrates the view from identified high surprise (top row), and low surprise (bottom row) locations. It was discovered that surprise values were high at transitions between two locations, and low surprise was identified at locations with monotonous passages and rooms.

crease in the values of IMB surprise is not noticed. Regular peaks demonstrate that despite prior exposure to similar isovists or features, both IMB and AMB evaluate high surprise. This is because each memory block is implementing the forgetting behaviour (described earlier). As a result, they forget what was previously remembered, and hence cause higher values of surprise. The general trend in the difference of surprise values with forgetting and without forgetting behaviour is illustrated in figure 6(b). The white region between the two curves is the difference between overall surprise values. Remembering all patterns without forgetting causes the surprise values to gradually reduce. In comparison to the values of surprise with forgetting behaviour, these cause fewer peaks. Additionally, the thick red and green curves present smoothened values of overall surprise with forgetting and without forgetting behaviour respectively. These again provide the reader with the general trend each one follows.





(b)

Figure 6: Comparison of the results of surprise evaluation with forgetfulness either enabled or disabled. 6(a) presents individual IMB and AMB surprise values, and 6(b) presents the difference between overall surprise experienced in the two cases. This is shown by the two shaded regions. Additionally, 6(b) also represents smoothened values of overall surprise in case of forgetting enabled (WF) and disabled (W/o F). Surprise values of IMB and AMB were found to attain more frequent peaks in the memory with forgetfulness, as it tends to "forget" previously presented patterns.

Conclusions & Discussion

In this paper, we presented a computational model of associative memory that is capable of remembering multi-dimensional real-valued patterns, performing bi-directional association, and importantly, mimicking human memory by providing an account of surprise stimulated. The memory model is constructed using collections of multi-dimensional RBF units with procrustes distance as the metric for comparison between input and centre. The unique feature of the presented memory model is that it masks the complex requirement of probabilistic modelling required otherwise in the current literature for computing surprise. Additionally, the presented memory model, while providing similar functionality to BAM has capacity to remember real-valued patterns without issues concerning stability. Furthermore, similar to the working of human memory, the presented memory model can be configured to forget patterns that are not recalled over long periods of time, thereby implementing the rule, "out of sight is out of mind".

The use of the memory model is demonstrated by identifying locations within an architectural building model that has variations in structure, which stimulates surprise. An isovist - a way of representing the structural features of a location - is used to represent the shape of a surrounding environment. Experimental results reveal and confirm the expected behaviour of surprise computation in two ways. First, from the application point of view, the identified high surprise locations were found to exist near transitions between two smaller parts of the "Villa-Savoye" house. This would be expected when the shape of the region where a person/agent enters changes its shape drastically. Second, the expected difference between the surprise values obtained from two experiments with forgetfulness behaviour enabled and disabled was verified (figure 6(b)). While the values of overall surprise continued to spike in the memory with forgetfulness, a gradual decrease was observed in the memory without forgetfulness. These two results verify the behaviour of surprise computation and the forgetfulness behaviour of the proposed memory model, and the technique employed for surprise computation.

Acknowledgments

The 3D model used for the test was obtained from the Google Sketchup Warehouse (Villa Savoye by Keka 3D Warehouse 2007). The authors are grateful to R. Linich for language review.

References

Albright, T. D. 2012. On the Perception of Probable Things: Neural Substrates of Associative Memory, Imagery, and Perception. *Neuron* 74(2):227–245.

Baldi, P., and Ittii, L. 2010. Of bits and wows: A Bayesian theory of surprise with applications to attention. *Neural Networks* 23(5):649–666.

Bartlett, M. S. 1952. The statistical significance of odd bits of information. *Biometrika* 39:228–237.

Benedikt, M. 1979. To take hold of space: isovists and isovist fields. *Environment and Planning B: Planning and Design* 6(1):47–65.

Bhatia, S.; Chalup, S. K.; and Ostwald, M. J. 2012. Analyzing Architectural Space: Identifying Salient Regions by Computing 3D Isovists. *In Proceedings of 46th Annual Conference of the Architectural Science Association, ASA 2012*.

Brown, D. C. 2012. Creativity, Surprise & Design: An Introduction and Investigation. *The 2nd International Conference on Design Creativity (ICDC2012)* 1:75–86.

Carlson, L. A.; Hölscher, C.; Shipley, T. F.; and Conroy-Dalton, R. 2010. Getting Lost in Buildings. *Current Directions in Psychological Science* 19(5):284–289.

Cole, D., and Harrison, A. 2005. Using Naturally Salient Regions for SLAM with 3D Laser Data. In *In Proceedings of International Conference on Robotics and Automation, Workshop on SLAM*.

Good, I. J. 1956. The surprise index for the multivariate normal distribution. *The Annals of Mathematical Statistics* 27(4):1130–1135.

Itti, L., and Baldi, P. 2009. Bayesian surprise attracts human attention. *Vision Research* 49(10):1295–1306.

Kendall, D. G. 1989. A survey of the statistical theory of shape. *Statistical Science* 4(2):87–89.

Kosko, B. 1988. Bidirectional associative memories. *IEEE Transactions on Systems, Man, and Cybernetics* 18(1):49–60.

Kullback, S. 1997. *Information theory and statistics*. Dover Publications.

Palm, G. 2013. Neural associative memories and sparse coding. *Neural Networks* 37:165–171.

Perttula, A.; Carter, S.; and Denoue, L. 2009. Kartta: extracting landmarks near personalized points-of-interest from user generated content. *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* 72.

Ranganathan, A., and Dellaert, F. 2009. Bayesian surprise and landmark detection. In *2009 IEEE International Conference on Robotics and Automation (ICRA)*, 2017–2023. IEEE.

Redheffer, R. M. 1951. A note on the surprise index. *The Annals of Mathematical Statistics* 22(1):128–130.

Shannon, C. E. 2001. A Mathematical Theory of Communication. *ACM SIGMOBILE Mobile Computing and Communications Review* 5(1):3–55.

Siagian, C., and Itti, L. 2009. Biologically Inspired Mobile Robot Vision Localization. *IEEE Transactions on Robotics* 25(4):861–873.

Trimble. 2013. Google Sketchup. *Retrieved from http://sketchup.google.com/intl/en/*.

Weaver, W. 1966. Probability, rarity, interest, and surprise. *Pediatrics* 38(4):667–670.

Xia, J. C.; Arrowsmith, C.; Jackson, M.; and Cartwright, W. 2008. The wayfinding process relationships between decision-making and landmark utility. *Tourism Management* 29(3):445–457.

Zhang, L.; Tong, M. H.; and Cottrell, G. W. 2009. SUNDAy: Saliency using natural statistics for dynamic analysis of scenes. In *Proceedings of the 31st Annual Cognitive Science Conference*, 2944–2949.