Abstract
The current paper demonstrates how Second Order Neural Networks (SONN) are biologically implementable and are capable of solving visual pose (or transformation) estimation problems. Several transformations and thus architectures are investigated, e.g.: translation, scale, rotation and rigid transformations. The paper structurally optimizes the neural architectures by minimizing their total wiring length, and consequently reveals a number of general design principles, the most recurring of which are topography, mirror-symmetry, overlap, centrality and clustering. These principles inform artificial designs and generate predictions for biological systems.

Keywords: Vision, Neural Networks, Topographic Maps, Structural Optimization, Wiring Length Minimization

Introduction
Second Order Neural Networks for Pose Estimation
The current paper is based on a hypothesis of how biological neural systems implement pose estimation. In particular, the paper delves into the structural optimization of the relevant neural architectures, which among other things generates a set of testable predictions regarding the spatial organization of certain biological neural maps.

How do biological neural systems implement pose estimation? The research stemming from this question is still in its infancy and is mostly confined to modelling, such as what is found within Computational Neuroscience (e.g. [1], [2], [3], [4], [5] and [6]). Most of the models studied so far involve some form of iterative template search. Another approach to pose estimation can be found in the field of Computer Vision, which is based on local feature correspondences. To the best of our knowledge, the approach has yet to be fully explored in the context of Computational Neuroscience.

A correspondence is essentially a vector between two matching local features, one in a source pattern and the other in a target pattern. The approach assumes the existence of two patterns, whereby the target represents a transformed version of the pattern in the source. Each correspondence contains crucial information regarding the transformation that interrelates the two patterns. An analysis of some or all of the inter-pattern correspondences reveals the relevant transformation and thus implements pose estimation. Correspondence based approaches can be divided into two main categories: 1) search based (e.g. [7], [8], [9] and [10]) and 2) vote based (e.g. [11], [12], [13], [14] and [15]). The vote based correspondence approach is the most neurally implementable of the two, and is thus the most biologically plausible and relevant approach in this context. The approach also embodies the following crucial characteristics: parallelizability, computational simplicity, generality, efficiency and robustness.

Figure 1. A simple SONATE architecture is depicted in (a) while a general HONN is depicted in (b). Both types of networks exhibit map nodes (c), multiplicative nodes (d) and summation nodes (e). SONATE architectures are restricted to representing second order information between two distinct maps, while general HONNs represent all higher-order information (e.g. first (f), second (g) and third (h) order).

The main element of the approach (i.e. correspondences) can be represented by simple conjunctions, which can be compactly implemented at the level of dendritic branches [16]. In fact, neural architectures embodying the vote based correspondence approach can be seen as specializations of Higher Order Neural Networks...
Structural Optimization of SONATE

The current paper is concerned with the structural optimization of SONATEs. In the context of neural architectures, structural optimization is concerned with seeking a spatial configuration of neural components which maximizes or minimizes a particular function. In biological systems this is important because the optimized functions tend to have a direct or indirect impact on survival, e.g.: energy efficiency [18]. From a scientific point of view, one of the advantages of conducting structural optimization experiments is that the latter lead to testable predictions regarding real neural systems, thus fuelling the fruitful cycle of modelling/experimentation that is typical of Computational Neuroscience.

One cost function which seems to be significantly prevalent in biological neural systems pertains to the total length of the connections between nodes (e.g. [19] and [20]) and is usually referred to as total wiring length (TWL). One of the reasons behind the apparent ubiquity of this cost function probably stems from the fact that TWL minimization has a favourable impact on the following factors: compactness, energy efficiency, robustness, speed of signal propagation and developmental feasibility. In this paper we deal with structural optimization exclusively from the perspective of TWL minimization.

Regarding optimization, we chose a geometrical approach which treats the neural architecture as a physical system upon which several forces act through time (iteratively) until a stable state has been reached. In brief, all nodes repulse each other, while connected nodes attract each other. The attraction of connected nodes leads to the shortening of their connections, thus indirectly fulfilling the optimization goal. Since the attraction forces can be seen to be caused by virtual coils between connected nodes, we here refer to the approach as CoilNet optimization. In the past, the approach has been successfully employed in the context of Graph Layout Optimization [21] and is somewhat related to the ElasticNet [22]. Seeing that the method is only implicitly minimizing TWL, it is necessary to compare its results with those of an explicit method, for which we chose an Evolutionary approach. The structural equivalence of the architectures resulting from both approaches confirms the intuition that CoilNet implicitly minimizes TWL. Note that one of the main advantages of CoilNet lies in its efficiency.

In the following section the main algorithms, architectures and methods adopted will be outlined. The section after that will describe the resulting design principles. The final section will discuss the relevance of the design principles revealed by the experiments.

Methods

CoilNet Optimization

As already mentioned, the CoilNet approach treats a neural network as a physical system within which attraction and repulsion forces act. Repulsion forces apply to all nodes while attraction forces apply only to interconnected nodes. Since interconnected nodes attract each other along their connections, they tend to draw near each other and thus shorten their connections. This local shortening in turn indirectly leads to TWL minimization. The rationale behind repulsion, on the other hand, is to indirectly enforce an even distribution of nodes, thus avoiding node clumping. Refer to Figure 2(a) for an illustration of the attraction and repulsion forces acting upon a simple network of three nodes and refer to Figure 2(b) for the corresponding resultant forces.

![Figure 2: Two CoilNet iterations. The n^th iteration is denoted by “It. n”. Attraction and repulsion forces are computed in (a), resultants are depicted in (b) (note for example (k)) and new node positions are depicted in (c). The configuration in (d) depicts the final state of the network, when all forces balance each other out. Note that only nodes (f) and (g) are not connected to each other. Attraction forces manifest themselves between connected nodes as in (j), while repulsion forces manifest themselves between all nodes.](image)

The CoilNet algorithm essentially cycles around two main computations: 1) the computation of forces and 2) the computation of node positions. Much like a physical system, nodes are moved (or change position) according to the forces acting upon them. After nodes have changed position, their forces will also change and the cycle will continue, until stability (i.e. equilibrium of forces) has been reached (see Figure 2(d)). Refer to Algorithm 1 for a more detailed description of the computations being carried out. Note that the repulsion force is inversely proportional to the square of the distance between nodes (refer to line 32 of Algorithm 1). Note also the use of proportionality constants (or force ratios) \( \alpha \) and \( \beta \) for attraction and repulsion forces respectively.
Algorithm 1. The basic CoilNet algorithm.

### Underlying Neural Architectures

The networks we are concerned with here have been designed to solve different pose estimation problems, e.g.: translation, scale, rotation and combinations of these. Note that the type of transformation being estimated determines the type of neural architecture adopted (i.e. the connectivity patterns between map and vote nodes), which in turn determines what the optimal configuration will look like. Note that although different architectures are being optimized, leading to different optimal configurations, we are searching for general design principles, the generality of which is advantageous both in their application to new artificial architectures and when searching for their counterparts in biological systems.

Six different types of architectures are reported here: 1) a one-dimensional shift estimator, 2) a two-dimensional shift estimator, 3) a two-dimensional shift estimator with multiple feature types per position, 4) a scale estimator, 5) a rotation estimator and 6) an estimator of rigid transformations (i.e. translation and rotation combinations).

### Results

Figure 3 depicts the progressive optimization of a one-dimensional shift estimator, which results in architectures exhibiting the following design principles: 1) all nodes are organized topographically, 2) source and target nodes run in opposite (or mirror-symmetric) directions, 3) vote nodes are sandwiched in between source and target nodes, 4) the zero-shift vote node is placed at the centre of the architecture and 5) map nodes meander somewhat symmetrically around vote nodes. The figure demonstrates several iterations of CoilNet optimization and the gradual emergence of design principles (e.g. topography) from an initially random configuration.

An optimized 2D translation estimator is illustrated in Figure 4. This architecture, as the one in Figure 3, estimates translations (or shifts) but, unlike the latter, involves more realistic two-dimensional maps. The optimized architecture exhibits the following main design principles: 1) source, target and vote nodes are organized topographically in two dimensions, 2) source and target nodes run in opposite (or mirror-symmetric) directions, 3) vote and map nodes overlap each other evenly, 4) vote map origins are centred in the optimized architectures and 5) source and target nodes representing opposite positions are clustered in pairs.

Figure 5 depicts an architecture which is identical to the two-dimensional translation estimator in Figure 4, except for the fact that each map position allows for multiple feature types. A simple illustration of this scheme would be a...
CoilNet can be used -
odest are often laid -
ventions are the same as those -
ictures from -
ological systems -
é.g. -
The central transformation (e.g. a -
tain design -
se design -
zation, 2) n -
Figure -
Refer to -
Symmetry -
Sandwich -
Overlap -
Centrality -
Meandering -
Clustering -

Probably one of the greatest lessons from the above optimization experiments is that optimal architectures (at least relative to TWL) can be counter-intuitive and unexpected. Several of these unexpected principles keep recurring in different architectures, e.g.: source and target nodes running in opposite directions (i.e. mirror-symmetry). Refer to Table 1 in order to compare architectures regarding the main design principles that emerged from the experiments (and others not illustrated due to lack of space). The design principles, from top to bottom refer to: 1) nodes topographically organized, 2) nodes running in mirror-symmetric directions, 3) vote nodes sandwiched in between map nodes, 4) map and vote nodes evenly overlapped, 5) vote nodes representing central transformations located at the centre of architectures, 6) nodes meandering symmetrically in relation to each other and 7) nodes forming small clusters. The architectures from left to right, refer to: 1) one dimensional shift estimators (T1), 2) two dimensional shift estimators (T2), 3) two dimensional shift estimators with multiple feature types per position (T3), 4) scale estimators (T4), 5) rotation estimators (T5), 6) estimators of translation and rotation combinations (T6).

The fact that many of the identified principles keep recurring, is an indication that they are quite general, and most probably apply to more complex/realistic cases. They provide us with suggestions for hardware implementations and hypotheses for biological investigations. The following list condenses the most common design principles emerging from the above experiments: 1) map and vote nodes tend to exhibit topographical organization, 2) nodes are often laid out in a mirror-symmetric pattern, e.g.: source and target nodes running in opposite directions, 3) map and vote nodes tend to evenly overlap, 4) the central transformation (e.g. a zero shift or a scale factor of one) tends to placed at the centre of the architecture and 5) nodes tend to form clusters, e.g.: map nodes from opposite (or same) positions or multiple feature types for the same map position.

Table 1. Summary of emergent design principles.

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topography</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>●</td>
</tr>
<tr>
<td>Symmetry</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Sandwich</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
</tr>
<tr>
<td>Overlap</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>●</td>
</tr>
<tr>
<td>Centrality</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Meandering</td>
<td>○</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Clustering</td>
<td>○</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>○</td>
</tr>
</tbody>
</table>

Discussion

The current paper has put forth a hypothesis regarding the manner through which biological neural systems implement pose estimation. In particular, it has structurally optimized the relevant neural architectures (i.e. SONATE) in order to derive design principles, which can aid in the efficient design of artificial systems and the investigation of biological systems. This section concludes the paper by discussing the biological relevance of the mirror-symmetry principle and the generalization of the basic CoilNet algorithm.

The results section evinced the fact that certain design principles keep recurring regardless of the particular connectivity that is implemented between second order and vote nodes, i.e.: topography, mirror-symmetry, overlap, centrality and clustering (refer to Table 1). These design principles are found throughout biological neural systems (e.g. [18] and [23]). The principle of mirror-symmetry however, deserves special mention since it is relatively less common in biological systems and, so far, explanations regarding its developmental and evolutionary origin are still in their infancy. It is a currently well known fact that adjacent visual maps (e.g. V2 and V3) are often mirror symmetric (e.g. [24], [25] and [26]). The current paper provides one plausible and novel explanation for the mirror-symmetry of adjacent maps: if two maps have topographic second order connections on a third map then the configuration which favours the minimization of TWL tends to be one where the mentioned maps are mirror symmetric. Apart from calling for further verification, this proposal leads to other interesting questions such as: 1) what/where are the third maps that are being projected to and 2) what types of second order connectivity are being implemented at the third maps? These questions might shed light on several issues regarding the phenomenon of gain modulation.

This paper has demonstrated how CoilNet can be used for TWL minimization. How can the method be generalized,
or less ambitiously, how can it be modified for the ad hoc incorporation of novel optimization goals? For example, apart from minimizing TWL one might wish to minimize the overlap between source and target maps. In this case, it should be possible to add a new set of forces (apart from the usual attraction and repulsion forces) whereby each source node repels every target node and vice versa. In so far as our optimization goals are geometric it should not be difficult to envision a corresponding pattern of forces which implements these goals. Furthermore, to verify that a pattern of forces truly satisfies a set of optimization goals, it is always necessary to compare the resulting stable configurations with those derived from explicit optimization methods.

References


