SELF-ORGANIZING CULTURED NEURAL NETWORKS

IMAGE ANALYSIS TECHNIQUES FOR LONGITUDINAL TRACKING AND MODELING OF THE UNDERLYING NETWORK STRUCTURE

AÑO 2015

Author:
Daniel de Santos Sierra
MSc in Computer Science

Supervisors:
Irene Sendiña Nadal
PhD in Physics

Stefano Boccaletti
PhD in Physics
The thesis analyzes the morphological evolution of assemblies of living neurons, as they self-organize from collections of separated cells into elaborated, clustered, networks.

In particular, it contributes with the design and implementation of a graph-based unsupervised segmentation algorithm, having an associated very low computational cost. The processing automatically retrieves the whole network structure from large scale phase-contrast images taken at high resolution throughout the entire life of a cultured neuronal network. The network structure is represented by a mathematical object (a matrix) in which nodes are identified neurons or neurons clusters, and links are the reconstructed connections between them. The algorithm is also able to extract any other relevant morphological information characterizing neurons and neurites. More importantly, and at variance with other segmentation methods that require fluorescence imaging from immunocyto-chemistry techniques, our measures are non invasive and entitle us to carry out a fully longitudinal analysis during the maturation of a single culture.

In turn, a systematic statistical analysis of a group of topological observables grants us the possibility of quantifying and tracking the progression of the main networks characteristics during the self-organization process of the culture. Our results point to the existence of a particular state corresponding to a small-world network configuration, in which several relevant graphs micro- and meso-scale properties emerge.

Finally, we identify the main physical processes taking place during the cultures morphological transformations, and embed them into a simplified growth model that quantitatively reproduces the overall set of experimental observations.

Keywords
neuron image segmentation; cultured neuronal network; neurite tracing; complex networks; network topology analysis; automated tracing; high throughput; neuronal morphology; light microscopy; connectome reconstruction; in silico model
Esta tesis estudia la evolución estructural de conjuntos de neuronas como la capacidad de auto-organización desde conjuntos de neuronas separadas hasta que forman una red (clusterizada) compleja.

Esta tesis contribuye con el diseño e implementación de un algoritmo no supervisado de segmentación basado en grafos con un coste computacional muy bajo. Este algoritmo proporciona de forma automática la estructura completa de la red a partir de imágenes de cultivos neuronales tomadas con microscopios de fase con una resolución muy alta. La estructura de la red es representada mediante un objeto matemático (matriz) cuyos nodos representan a las neuronas o grupos de neuronas y los enlaces son las conexiones reconstruidas entre ellos. Este algoritmo extrae también otras medidas morfológicas importantes que caracterizan a las neuronas y a las neuritas. A diferencia de otros algoritmos hasta el momento, que necesitan de fluorescencia y técnicas inmunocitoquímicas, el algoritmo propuesto permite el estudio longitudinal de forma no invasiva posibilitando el estudio durante la formación de un cultivo.

Además, esta tesis, estudia de forma sistemática un grupo de variables topológicas que garantizan la posibilidad de cuantificar e investigar la progresión de las características principales durante el proceso de auto-organización del cultivo. Nuestros resultados muestran la existencia de un estado concreto correspondiente a redes con configuración small-world y la emergencia de propiedades a micro- y meso-escala de la estructura de la red.

Finalmente, identificamos los procesos físicos principales que guían las transformaciones morfológicas de los cultivos y proponemos un modelo de crecimiento de red que reproduce el comportamiento cuantitativamente de las observaciones experimentales.
Contents

Abstract

Resumen

Introduction and Context 1

1 Methods 5

1.1 Neuronal Cultures from Schistocerca Gregaria 6

1.1.1 Lifecycle 6

1.1.2 Nervous System 7

1.1.3 Frontal Ganglion 9

1.1.4 Cell Culture Protocol 10

1.2 Image Analysis 11

1.2.1 Connectome reconstruction by image analysis 12

1.2.2 Image Analysis and Graph Theory 17

1.2.3 Image analysis performance measure 19

1.3 Complex Networks 20

1.3.1 Basic Concepts in Graph Theory 21

1.3.2 Complex Networks Models 25

2 Image analysis 35

2.1 Multi-layer Graph-Based Segmentation Algorithm 36

2.1.1 Graph initialization: The zero layer 39

2.1.2 Nodes Aggregation: From one layer to the next 40

2.1.3 Communities: the next layer 42

2.1.4 Stop condition: closing the loop 42

2.1.5 Parameter analysis 43

2.1.6 Computational cost analysis 45
2.2 Application to invertebrate neuronal cultures ................................................. 45
  2.2.1 Neuronal cultures and imaging .............................................................. 46
  2.2.2 Culture imaging artifacts ...................................................................... 47
  2.2.3 Image processing strategy .................................................................... 51
  2.2.4 Image segmentation pipeline ................................................................. 53
  2.2.5 Network structure representation and measures ..................................... 56
  2.2.6 Algorithm’s accuracy evaluation ............................................................. 57
  2.2.7 Computational cost .............................................................................. 65

3 Results ............................................................................................................... 67
  3.1 Culture developmental stages .................................................................... 68
    3.1.1 Spreading stage ................................................................................. 69
    3.1.2 Migration and aggregation stages ....................................................... 71
  3.2 Network statistics ....................................................................................... 72
    3.2.1 Formation of the giant connected component ...................................... 72
    3.2.2 Emergence of a small-world structure ................................................. 74
    3.2.3 Emergence of assortative and single scale networks ........................... 77

4 Data driven model ............................................................................................. 81
  4.1 First model: Isotropic growth .................................................................... 81
  4.2 Second model: Anisotropic growth ............................................................ 88
  4.3 Model analysis: transition to clustered networks ....................................... 92

5 Conclusions and Perspectives ........................................................................... 97
  5.1 Conclusions ............................................................................................... 97
  5.2 Future work ............................................................................................. 99
  5.3 First steps in the future ............................................................................. 101
    5.3.1 NEural SOrter ............................................................................... 101

Appendices .......................................................................................................... 105

A Multi-layer graph-based algorithm pseudocode ............................................. 107

B Multi-graph based segmentation algorithm applied to hand biometry .......... 111
Along the past decades, cultured neuronal networks (CNNs) have constituted a fundamental tool for scientists, as one of the benchmark models for the study of the central nervous system. They, indeed, allow conducting very well controlled laboratory experiments, thus providing a systematic way to approach fundamental questions, as for example, unveiling the principles and mechanisms underlying memory, connectivity, and even information processing of their in vivo counterparts [1, 2, 3, 4, 5].

CNNs have also important practical applications, when computer-connected to a real or a simulated robot (to create what is called a hybrot [6, 7, 8] or an animat [9, 10], respectively). In that situations, scientists are then endowed with the possibility of studying some basic neuronal processes in realistic contexts, such as learning and plasticity. Possibly, the most relevant advantage of CNNs is the unique option they offer of following the footprints of the self (or induced) organization of the network’s functionality and dynamics (usually by means of a multi-electrode array [MEA] or calcium fluorescence, recording the CNN electrophysiological data, or inducing electrical stimulations in given spatial positions) together with the monitoring and tracking of the structural organization of the neurons’ connectivity along the entire course of the culture’s growth [11, 12, 13].

Although culturing neurons on top of a MEA equipped chamber implies, in general, only mild constraints, following, the development of the culture’s structure is a far more delicate issue. Indeed, image-based biology systems essentially require to gather sequential imaging of the culture and its processing to seek the evolution of the main network’s indicators and measures along the CNN’s maturation [14, 15, 16, 17].

The main drawback of image processing tools in segmenting neurons and neurons’ connection is that they need pictures with a high level of contrast. This traditionally led experimentalists to rely on immunocytochemistry techniques, which however implies cell fixation and therefore death [18, 19, 20, 21, 22, 23, 24, 25, 26, 27].

This Thesis comes up with a novel approach for non-invasively image procedure. We describe here a graph-based segmentation algorithm which operates on large scale
images acquired by phase-contrast microscopy, and therefore by a fully non-invasive technique, that is, without the need of adding chemicals to the culture. The algorithm accurately identifies the relevant network’s units, and reconstructs the wiring of network connectivity with an overall computational cost (in terms of time and memory) which scales linearly with the image size. We show that we are able to fully track the main parameters characterizing the morphology and network’s topology of a single culture during its maturation, and to identify the basic mechanisms that take place at different stages of the culture development.

With the aim of studying the network’s topology and why and how an assembly of isolated (cultured) neurons self-organizes to form a complex neural network, we carry out a study based on complex network theory [4, 5, 1].

Some previous studies highlighted the fact that the structuring of a neuronal cultured network before the attainment of its mature state is not random, being instead governed and characterized by processes eventually leading to configurations which are comparable to many other real complex networks [28]. In particular, networking neurons simultaneously feature a high overall clustering and a relatively short path-length between any pair of them [17]. Such configurations, which in graph theory are termed small-world [29], are ubiquitously found in real-world networking systems. Small-world structures have been shown to enhance the system’s overall efficiency [30, 31], while concurrently warranting a good balance between two apparently antagonistic tendencies for segregation and integration in structuring processes, needed for the network’s parallel, and yet synthetic performance [32].

In this Thesis, we experimentally investigate the self-organization into a network of an in vitro culture of neurons during the course of development, and explore the changes of the main topological features characterizing the anatomical connectivity between neurons during the associated network’s growth. To that purpose, dissociated and randomly seeded neurons are initially prepared, and the spontaneous and self-organized formation of connections is tracked up to their assembling into a two dimensional clustered network.

Most existing studies in neuronal cultures restricted their attention to functional networks (statistical dependence between nodes activities) and not to the physical connections supporting the functionality of the network [33]. The reason behind this drawback is that the majority of investigations focused on excessively dense cultures, hindering the observation of their fine scale structural connectivity. Although there are studies striving to indirectly infer the underlying anatomical connectivity from the functional network, it has been shown that strong functional correlations may exist with no direct physical connection [34]. Only few studies dealt with the physical wiring circuitry. However, on the one hand, only small networks were considered; on the other hand, how the network state evolves during the course of the maturation process has not been investigated [17].

Here, instead, we focus on intermediate neurons’ densities, and provide a full tracking of the most relevant topological features emerging during the culture’s evolution. In particular, we show experimentally that in vitro neuronal networks tend to develop from a random network state toward a particular networking state, corresponding to a small-world configuration, in which several relevant graph’s micro- and meso-scale properties
emerge. Our approach also unveils the main physical processes underlying the culture’s morphological transformation, and allows using such information for devising a proper growth model (in silico networks), qualitatively and quantitatively reproducing the set of our experimental evidence. Together with confirming several results of previous works on functional connectivity [13], or on morphological structuring at a specific stage of the cultures’ evolution [17], we offer a systematic characterization of several topological network’s measures from the very initial until the final state of the culture. Such a longitudinal study of the network structure highlights as yet unknown self-organization properties of cultured neural networks, such as i) a large increase in both local and global network’s efficiency associated to the emergence of the small-world configuration, and ii) the setting of assortative degree-degree correlation features.

**Thesis motivation and aims**

The main motivation of this Thesis is to analyze the structure of in vitro neuronal networks to reveal the principal mechanisms of culture self-organization.

With this objective, the main aims of this thesis are stated as follows:

- Propose and implement an image-segmentation algorithm able to extract the network structure from culture phase-contrast pictures without adding dyes, chemicals or culture fixation.
- Evaluate the image-segmentation algorithm.
- Culture network analysis based on graph theory to unveil the self-organization principles of in vitro cultures.
- Categorization of these in vitro networks into complex networks models in graph theory.
- Propose an in silico network model that reproduces the growth of in vitro networks.

**Thesis contents description**

Finally, this introduction ends with the thesis general layout.

First of all, the thesis begins introducing the framework we have used in the whole research. In Chapter 1, the invertebrate animal used as culture resource is presented as well as the main works carried out with it (Section 1.1). Afterwards, some imaging procedures for connectome reconstruction in the literature are introduced (Section 1.2). Finally, the basic principles to complex network management is brought in (Section 1.3).

After setting up in the initial concepts, in Chapter 2, a general purpose image segmentation algorithm based on multi-layer graphs is presented (Section 2.1). Later, this algorithm is modified in order to deal with the invertebrate neuronal culture pictures (Section 2.2).
The algorithm’s outcome, i.e. the structure of the underlying network, will be analyzed in Chapter 3. In this chapter we will distinguish the main self-organization principles (Section 3.1) and the structural properties acquired during the culture life (Section 3.2).

By analysis of these cultures, and supported by the literature, an in silico model will be presented in Chapter 4. This simple model is based on neurite spreading and tension forces between connected neurons. These tension forces produce the cell migration and clustering. Firstly, we introduce an isotropic neurite growth model that fits the behavioural trend of topological measures of in vitro cultures (Section 4.1). Secondly, this model is improved with an anisotropic neurite growth model achieving a better fitting at topological and morphological levels (Section 4.2). To finish this chapter, we study in Section 4.3 the in silico network self-organization as a function of both, the cell density and cell adhesion to the substrate.

Finally, coming this thesis to an end with the conclusions and future work in Chapter 5.
CHAPTER 1

Methods

Contents

1.1 Neuronal Cultures from *Schistocerca Gregaria* ............... 6
   1.1.1 Lifecycle ................................................. 6
   1.1.2 Nervous System ........................................... 7
   1.1.3 Frontal Ganglion ......................................... 9
   1.1.4 Cell Culture Protocol .................................... 10

1.2 Image Analysis .................................................. 11
   1.2.1 Connectome reconstruction by image analysis ............... 12
   1.2.2 Image Analysis and Graph Theory .......................... 17
   1.2.3 Image analysis performance measure ....................... 19

1.3 Complex Networks ............................................... 20
   1.3.1 Basic Concepts in Graph Theory ........................... 21
   1.3.2 Complex Networks Models ................................ 25

This thesis is focused on the study of cultured neuronal networks (CNNs) self-organization. Therefore, we study here how CNNs change through days *in vitro* from the very initial up to the day when we are not able to extract the network structural information.

To that end, we need three essential frameworks. The former, we have to use a biological model to do the neuronal cultures and we will track the structural changes by microscopy images. The use of these images brings up to the second framework, that is, we need an image processing tool suited to extract the structural network information.
Chapter 1. Methods

automatically and represent it in a tractable alternative, i.e. a mathematical representation of the network structure. Finally, the latter framework is the graph theory. Representing the CNN as a graph allow us to trace changes into the network topology and distinguish between the network development stages along days in vitro.

In this Chapter, we present these three frameworks. In Section 1.1 we present the biological model used, the *Schistocerca Gregaria*’s frontal ganglion. Afterwards, in Section 1.2 we review the state-of-the-art in connectome reconstruction by image analysis. And, finally, in Section 1.3, we state the principles and definitions of the complex network analysis and the usual measures we take into account to track the network structural formation.

1.1 Neuronal Cultures from *Schistocerca Gregaria*

*Schistocerca gregaria* (Forskål) or Desert Locust belongs to the group of orthopteroid insects in the family Acrididae. This family also includes many grasshoppers. Locust and grasshoppers have a similar morphology but they have different behaviour. Locusts show a strong tendency to group together and become gregarious, and then to migrate in large swarms while grasshoppers rarely aggregate. *Schistocerca gregaria* can usually be found in North Africa, Arabia or India.

The analysis of the locusts nervous system started from the promotion by the Anti-Locust Research Centre in London in order to solve the problem with locust in Africa [35]. Thereafter, the interest in studying this nervous system started growing due to some advantages of this system over the mammals’ nervous system. For instance, locusts have a small number of neurons and the access to them is not impeded by a pulsating blood supply. In addition, many neurons are large enough to place electrodes on them allowing the analysis of the signals that occur within them. The large size of some motor neurons, interneurons and sensory neurons allows treating them as identified individuals rather than members of a population. Finally, muscles are innervated by motor neurons in the same way as those of vertebrates. For this reason, the analysis of the locusts nervous system is a good starting point for understanding the mechanisms of more complex systems.

Postembryonic development through five larval instars means that the neuronal organisation of certain motor patterns can be analysed before the locust is able to express these patterns as behaviour; movable wings, for example, are only present in adults, yet the motor pattern for flying can be expressed in early larvae. As adults, locusts show a wide variety of behaviour that is worthy of analysis, both at ethological and neurobiological levels [35].

1.1.1 Lifecycle

The lifecycle of locusts [35, 36] consists of three phases: egg, larva and adult. Eggs are lain in moist ground during the rainy season and the embryonic development takes place below ground. The larval stage, larvae, nymphs or hoppers, is divided into five instars
1.1. Neuronal Cultures from *Schistocerca Gregaria* (Figure 1.1). The first instar is a miniature version of the adult but without movable wings. In the following molts, the locust increases in size. From the third instar two wing buds appear but they can not be used for flight. Fully movable wings are formed at the final molt to the adult (Figure 1.2).

From egg to adult may take from a few weeks to 6 months depending on prevailing conditions. In controlled conditions it usually takes 6 weeks.

![Figure 1.1](image)

**Figure 1.1** – *Chortoicetes terminifera* nymph stages. After first instar larva hatches out, the larva increases in size across different molts. From the third molt, two small wings appear and begin to develop but these cannot be used to fly (they are not fully movable wings) until it becomes an adult.

1.1.2 Nervous System

The classical division of the Nervous System proposed by Snodgrass [37] divides the Nervous System into three parts: central, peripheral and sympathetic nervous systems. An alternative division given by Nesbitt [38] distinguishes brain, ventral ganglionic chain, stomodeal or stomatogastric and peripheral nervous systems. Nevertheless the nervous system distinctions remains blurred [35]. In this thesis, the division proposed by Burrows [35] is followed, and, the Nervous System is considered to be composed of the somatic nervous system, the peripheral nervous system and the visceral nervous system.

The somatic nervous system consists of a ventral chain of bilaterally symmetrical segmented ganglia, including the brain, that are linked by paired bundles of axons called connectives running the length of the locust.

The peripheral nervous system consists of the axons and terminals of the motor and other efferent neurons, and the axons, terminals and cell bodies of sensory neurons, most of which have their cell bodies in the periphery close to their receptor.

The visceral nervous system consists of a series of small ganglia called either the stomodeal or stomatogastric ganglia innervate the anterior region of the gut (stomodeum), whereas the posterior portion of the gut is innervated by nerves from the terminal abdominal ganglion in the somatic chain.

The brain of *Schistocerca Gregaria* lies above the oesophagus in dorsofrontal portion of the head and is covered dorsally and laterally by head musculature and airsacs. It is divisible into three paired lobes: protocerebrum, deutocerebrum and tritocerebrum (Figure 1.3) [35, 39, 40].

---

Figure 1.2 – *Schistocerca Gergaria* adulto. Ref. Burrows. Intentar poner una foto de una muestra con el tamaño que tiene.

Figure 1.3 – Extracted from Burrows [35]
The protocerebrum is the largest lobe of the brain. It is formed by the optic lobes, the mushroom bodies and the central body complex and it is linked with the corpora cardiaca.

The deutocerebrum consist of paired lobes lying just ventral to the protocerebrum. The deutocerebrum is subdivided in the antennal lobes, the antennal mechanosensory and motor centers, and the glomerulas lobes. An antennal lobe is the main processing region for the primary olfactory signals.

The tritocerebrum, the smallest part of the brain, lies posterior and ventral to the deutocerebrum. It connects the brain to the stomatogastric nervous system, where it acts as a higher ordered center [40]. Besides its connection with the retrocerebral complex (corpora cardiaca and corpora allata), it also contains tritocerebral motoneurons innervating the labrum.

The stomatogastic ganglia or stomodeal ganglia are a serie of small aggregations of neurons that are closely associated with the brain, the corpora cardiaca and allata, and the anterior portion of the gut. This system consists of frontal ganglion, oesophageal ganglion or corpora cardiaca and hypocerebral ganglion.

1.1.3 Frontal Ganglion

The frontal ganglion (FR GNG [39] or FG [41, 42, 36]) is an essential part of the stomatogastric nervous system of insects [43, 41, 42, 44, 36]. The frontal ganglion is an unpaired structure that lies in the forehead, in front of the brain on the dorsal side of pharynx [39, 41]. The FG is linked anteriorly by paired frontal connectives to the tritocerebrum. Posteriorly, a small recurrent nerve (nervus recurrens) passes backwards along the pharynx to connect it with the hipocerebral ganglion [35] (located behind the brain) which is associated with the corpora cardiaca [41]. These and other three pairs of efferent nerves (the anterior, median and posterior nerves) branch onto the dilator muscles of the gut in a rostrum to caudal order. The FG is the major source of foregut muscles innervation.

The frontal ganglion functionality is related to the foregut movements. Experiments where FG nerves were cut, or in ganglionectomy experiments, indicate that in locusts, as in other insects [45, 46, 47], the FG plays an important role in feeding, moulting behaviours and, in female insects, ovarian development. To be specific, the frontal ganglionectomy produces a decrease in feeding activity and food intake in *Schistocerca gregaria*, with the body mass remaining constant (similar to not feed animals) [43]. The role of the FG during moulting has been detailed for a number of species. There are two stages of ecdysis in which the insect needs to apply pressure on the body wall. The first stage is during rupture of the old cuticle, and the second stage, when expanding the new cuticle and wings. The main mechanism of this process is the inflation of the gut with air [46, 42]. The FG controls a foregut motor pattern that is used first to remove moulting fluids (haemolymph) from the space between the old and new cuticle prior to ecdysis, and second for air swallowing.

The frontal ganglion is encased by a fine perineural sheath, or neural lamella, consisting largely of connective tissue, under which is a thin specialised layer of glial cells called the perineurium, and characterised by a central neuropil surrounded dorsally and
latterly by one or two layers of neuronal cell bodies [35, 41]. The brain and ganglia are bathed in the haemolymph. The ganglion sheath acts, like a blood-brain barrier, providing an ionic barrier so that the ionic concentration of the fluid surrounding the membranes of the neurons are different from those in the haemolymph.

The locust FG’s shape is like a 'somewhat flattened pear', whose major axis is 200-250 µm [41]. The FG contains around 100 neurons whose bodies range in size from 10 to 50 µm [35, 41]. Between 17 and 19 neurons of the frontal ganglion innervate the visceral muscles of the foregut via the outer frontal visceral nerve [48]. A similar species which can be compared to Schistocerca gregaria is Locusta migratoria. No differences have been observed at organisation level between them [35]. Adult Locusta migratoria’s FG, whose major axis is 140 µm, contains 80-100 cells whose bodies range in size from 7 to 45 µm [36].

1.1.4 Cell Culture Protocol

Animals and dissection

Schistocerca gregaria were reared under crowded conditions as previously described [41, 49]. Adults of both sexes were used indistinctly. Locusts were briefly anaesthetised in CO₂ and fixed dorsal side up on a transparent Sylgard plate (Sylgard 182 silicon Elastomer, Dow Corning Corp., Midland, MI, USA). The FG was easily accessible by opening a window in the head cuticle, cutting out most of the fronts and clearing fat tissue and air sacs as required. The ganglion was removed from the animal after cutting the connectives linking it to the brain and the nerves leading to the muscles and immersed in sterile cold saline.

Cell culture preparation

After removing the cold saline, 500 µl of collagenaze/dispace mixture (2mg/ml) is added. The tube is wrapped with an aluminium foil and kept in an incubator at 37°C for one hour.

After one hour, the tube is centrifugated for 10 minutes at 1500 r.p.m. in a balanced bench-top centrifuge by placing a tube with 500 µl of water. The centrifugation stops the activation of the enzyme.

The ganglia is washed by replacing the 500 µl enzyme with 1000 µl saline and centrifugating (balanced with 1000 µl) for 10 min at 1500 r.p.m.

Finally, the saline is replaced with 1000 µl of culture medium and centrifugated again for 10 min at 1500 r.p.m.

900 µl of culture medium are removed and the cell dissociation was carried out by flushing 20 times 70 µl out of the remaining 100 µl by means of a pre-wetted regular tip, then by flushing 40 times the whole 100 µl through a pre-wetted thin tip, until nothing in suspension is differentiated.

After cell dissociation, 50 µl are resuspended within a circle (r ∼ 5 mm) precoated with culture substrate Concanavalin A (0.5 mg/ml) in a Petri dish (35 mm), Figure 1.4.
1.2. Image Analysis

The Central Nervous System (CNS) has been studied since the XVII century. The improvements into physical devices like microscopes, staining techniques and so on have allowed researchers to go deeper and deeper into the various elements of the CNS.

It was not until the XIX century when the names for the brain elements were given. In chronological order, in 1889, Wilhelm His coined the term *dendrite*. In 1891, Heinrich Wilhelm Gottfried von Waldeyer-Hartz described the basic structural unit of the nervous system calling it as *neuron*. Rudolph Albert von Kölliker, *circa* 1896, coined the term *axon* and finally, in 1897, Charles Scott Sherrington introduced the term *synapse*.

The study of the structure, function and development of the nervous system plays a key role in order to understand higher-order cognitive functions [50, 51, 24]. Accordingly, the development of computational methods and tools for the study of nervous systems at structural and functional level is increasingly necessary [52]. The morphological properties of the axonal and dendritic trees are basis of neuronal phenotype and take
C hapter 1. Methods

Figure 1.5 – Culture medium and haemolymph-conditioned medium. The medium was made from mixing Leibovitz L-15 medium with L-glutamine (Sigma, L-1518) and Penicillin-Streptomycin Amphotericin B solution (Biological Industries, #03-033-1, 100 ml). The final culture medium was enriched with proteins by extracting haemolymph (500 µl) from alive locusts. To extract the haemolymph, the locusts were punctured just after the midleg and the blood coming out was pipeted. The medium was enriched with 5% of hemolymph (2 ml and 100 µl hemolymph) and then was filtered twice, first through a 0.45 µm filter and then through a 0.20 µm filter.

part in the network connectivity [53, 24]. These analyses allow studying the intrinsic and extrinsic factors influencing neuronal development. Therefore, the analysis of image datasets requires representing the network within the image like graphs which capture the essence of the image information (point coordinates, local thickness, connectivity, and so on) [54, 55, 56, 24].

1.2.1 Connectome reconstruction by image analysis

The term connectome is referred to the neural connectivity at different organization levels [57, 58]. In the brain, we can distinguish between two organization levels, macroscopic and microscopic. At macroscopic level, we are able to recover the functional connections between brain regions through magnetic resonance imaging at millimeter resolution [59]. Whilst, at microscopic level, also known as projectome, a description of the real connectivity in the brain is given by means of optical microscopy at micrometer resolution [60].

Connectome reconstruction history started from the XVII century by means of freehand drawings and evolved through camera lucida [61], wax models [62, 63], acetate films, microtome [64], among others, giving more accurate representations with less visual impressions. Nowadays, the evolution of computers and their capabilities allows the connectome study with automated image processing tools.

First attempts to obtain digital reconstructions of neuronal morphology with the help


<table>
<thead>
<tr>
<th>Round</th>
<th>Data set</th>
<th>#trees</th>
<th>#branches</th>
<th>#image stacks</th>
<th>#images</th>
<th>image dimensions (X×Y px)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training</td>
<td>climbing fiber</td>
<td>1</td>
<td>184</td>
<td>1</td>
<td>34</td>
<td>6120 × 4343</td>
</tr>
<tr>
<td></td>
<td>olfactory</td>
<td>3</td>
<td>170</td>
<td>3</td>
<td>212</td>
<td>512 × 512</td>
</tr>
<tr>
<td></td>
<td>neocortical</td>
<td>14</td>
<td>224</td>
<td>6</td>
<td>290</td>
<td>512 × 512</td>
</tr>
<tr>
<td></td>
<td>neuromuscular</td>
<td>6</td>
<td>218</td>
<td>158</td>
<td>16682</td>
<td>1024 × 1024</td>
</tr>
<tr>
<td></td>
<td>CA3 hippocampal</td>
<td>14</td>
<td>212</td>
<td>1</td>
<td>110</td>
<td>5360 × 3420</td>
</tr>
<tr>
<td>Qualifier</td>
<td>climbing fiber</td>
<td>1</td>
<td>200</td>
<td>1</td>
<td>28</td>
<td>5100 × 3302</td>
</tr>
<tr>
<td></td>
<td>olfactory</td>
<td>3</td>
<td>176</td>
<td>3</td>
<td>244</td>
<td>512 × 512</td>
</tr>
<tr>
<td></td>
<td>neocortical</td>
<td>20</td>
<td>200</td>
<td>6</td>
<td>290</td>
<td>512 × 512</td>
</tr>
<tr>
<td></td>
<td>neuromuscular</td>
<td>6</td>
<td>226</td>
<td>158</td>
<td>16682</td>
<td>1024 × 1024</td>
</tr>
<tr>
<td></td>
<td>CA3 hippocampal</td>
<td>35</td>
<td>260</td>
<td>3</td>
<td>444</td>
<td>varies by section</td>
</tr>
<tr>
<td>Final</td>
<td>climbing fiber</td>
<td>1</td>
<td>184</td>
<td>1</td>
<td>26</td>
<td>7000 × 3000</td>
</tr>
<tr>
<td></td>
<td>olfactory</td>
<td>3</td>
<td>170</td>
<td>3</td>
<td>251</td>
<td>512 × 512</td>
</tr>
<tr>
<td></td>
<td>neocortical</td>
<td>21</td>
<td>246</td>
<td>10</td>
<td>644</td>
<td>512 × 512</td>
</tr>
<tr>
<td></td>
<td>neuromuscular</td>
<td>15</td>
<td>416</td>
<td>156</td>
<td>14016</td>
<td>512 × 512</td>
</tr>
<tr>
<td></td>
<td>CA3 hippocampal</td>
<td>21</td>
<td>210</td>
<td>4</td>
<td>423</td>
<td>varies by section</td>
</tr>
<tr>
<td></td>
<td>visual cortical</td>
<td>25</td>
<td>158</td>
<td>53</td>
<td>9314</td>
<td>1024 × 1024</td>
</tr>
</tbody>
</table>

Table 1.1 – DIADEM datasets technical properties. Datasets from a variety of animal species, brain regions, cell and arbor types, histology and microscopy. Manual or semiautomatic labeling of dendrites and axon trees. Extracted from [25]

of computers date from 1965 [65]. These consisted in using the computer to interact with the microscope and to store point coordinates indicated manually by a human operator [24].

From 1975 to 1986, the entire nematode *Caenorhabditis elegans* connectome was obtained [66, 67] by means of serial section with electron microscopy. In 2009, the peripheral neuromuscular circuit (a mammalian subnetwork) was reconstructed [68] as result of confocal fluorescence microscopy. In 2009-2010, the DIADEM challenge [63, 25, 58] (DIgital reconstruction of Axonal and DEndritic Morphology) was born with the aim of producing a breakthrough in automatic reconstruction of connectomes. In this challenge, six data sets were provided from different laboratories and with different kind of cultures (cerebellar cortex, antennal lobe and neocortex among others (Table 1.1)) and acquisition techniques (confocal, transmitted light brightfield, and so on).

In image analysis, the image segmentation is the most important process in order to extract the connectome.

Image segmentation is the process of assigning to each image element a label indicating which segment it belongs to. According to [24], automated image segmentation is hampered by noise (inevitable statistical fluctuations as well as other irrelevant structures), low resolution (ultimately limited by diffraction), inhomogeneous contrast (nonperfect distribution of the dye), and background gradients (nonuniform illumination). Minimization of these artifacts within the boundary conditions imposed by a given experiment is of key importance [69, 70]. In this case, a complete image segmentation
method involves four processing steps: image preprocessing (whose aim is to improve the image), soma segmentation (whose is focused on extracting the cell boundaries), neurite trees (which target is to extract the dendrites and axon structure) and spines detection.

Next sections are focused on reviewing the methods for soma and neurite tree segmentation which are one of the main problems handled in this thesis.

**Soma Segmentation**

Soma identification or segmentation plays an important role into the neurite tree segmentation and spine segmentation due to it is used as the starting point to these segmentation procedures, and it is more important as the amount of neurons to detect within an image grows.

An approach to facilitate the soma segmentation is to stain them differently, for instance, DAPI staining facilitates the segmentation of the soma’s nuclei that it can be used as a seed segmentation to find the cell body boundaries [71]. If only one stain is used, the concentration levels, and therefore, the soma intensity is higher than in the neurites. A similar case can be found if we consider phasecontrast microscopy where the cell bodies often yield more contrast than their processes [72, 73].

Table 1.2 and Table 1.3 list some of the works in connectome reconstruction. It can be noted that most of the works start from images of stained cultures.

Table 1.4 and Table 1.5 briefly summarize the methods used in order to segment the soma. It can be observed that most of the works first carry out the pre-processing of the image in order to improve image quality by means of histogram equalization, filters, edge detectors, background substraction or correction and so forth, and then a threshold is used to recover the soma’s areas and the results are improved through morphological operations.

**Neurite Tree Segmentation**

According to Meijering [24], segmentation methods to extract neurite trees can be classified into global processing or local exploration methods [80, 81].

Global algorithms usually follow the same steps: binarization, skeletonization, rectification, and graph representation. The binarization step aims to produce an initial segmentation of the neurites. Usually, (adaptive) thresholds are used for this purpose but it is known to be one of the most error-prone segmentation methods. The skeletonization step is used to extract the neurite center line as a descriptor of the neurite tree. The result of the skeletonization step often contains errors, like gaps or branches, and ambiguities, such as loops or crossings. The graph representation is usually carried out by considering as graph nodes those points symbolizing terminations, bifurcations and inflections, and graph links the neurite connecting those points [80, 82].

Local algorithms explore an image locally around relevant structures rather than processing the entire image. Unlike global methods, where the critical points are identified in the last stages, local methods often start with the detection of topologically relevant
Table 1.2 – Works in Image Processing for cultured networks. Reference stands for the first author surname, publication year and reference number. Culture stands for the cell used to check the image processing accuracy. Dye is Y for those works using fluorescence techniques or dyes and N in other case. Density/Size depicts the picture size in pixels or the cell density for the culture. 2D/3D denotes the works using 2D images or 3D reconstructions. Long means works studying longitudinally the culture life (Y) or those that studying the culture at a specific time (N). A/M/S shows those programs working automatically (A), manually (M) or semiautomatically (S). Lens indicates the magnification used to take the pictures. *\(^1\) Studies at 24, 48 and 72 hours. More works in Table 1.3.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Culture</th>
<th>Dye</th>
<th>Density/Size</th>
<th>2D/3D</th>
<th>Long</th>
<th>A/M/S</th>
<th>Lens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Masseroly, 1993, [74]</td>
<td>Mice brain slices</td>
<td>Y</td>
<td>512 × 512 px&lt;br&gt;156710 µ m^2&lt;br&gt;380222 µ m^2</td>
<td>2D</td>
<td>N</td>
<td>S</td>
<td>20×</td>
</tr>
<tr>
<td>Al-Kofahi, 2003, [75]</td>
<td>Hippocampal E18 Sprague-Dawley</td>
<td>Y</td>
<td>2.6 × 10^4 cell/cm², 768 × 512 px</td>
<td>2D</td>
<td>N</td>
<td>A</td>
<td>10×</td>
</tr>
<tr>
<td>Weaver, 2003, [72]</td>
<td>Retinal tissue E6 chick White Leghorn</td>
<td>N</td>
<td>640 × 480 px</td>
<td>2D</td>
<td>N</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Shah, 2004, [73]</td>
<td>Dorsal root ganglia E9 chick White Leghorn</td>
<td>N</td>
<td></td>
<td>2D</td>
<td>N</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Al-Kofahi, 2006, [76]</td>
<td>Cerebellar Granule Neurons P8 rat Sprague-Dawley NG108-15 cells (ATCC, Cat. #: HB-12317)</td>
<td>N</td>
<td>1470 × 1180, 1300 × 1080 and others</td>
<td>2D</td>
<td>N(*(^1))</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Mitchell, 2007, [77]</td>
<td>Dorsophila neurons (CNS)</td>
<td>Y</td>
<td>300000 cells/ml</td>
<td>2D</td>
<td>N</td>
<td>S</td>
<td>20×</td>
</tr>
</tbody>
</table>

*\(^1\) Studies at 24, 48 and 72 hours. More works in Table 1.3.
Table 1.3 – Works in Image Processing for cultured networks. Reference stands for the first author surname, publication year and reference number. Culture stands for the cell used to check the image processing accuracy. Dye is Y for those works using fluorescence techniques or dyes and N in other case. Density/Size depicts the picture size in pixels or the cell density for the culture. 2D/3D denotes the works using 2D images or 3D reconstructions. Long means works studying longitudinally the culture life (Y) or those that studying the culture at a specific time (N). A/M/S shows those programs working automatically (A), manually (M) or semiautomatically (S). Lens indicates the magnification used to take the pictures. *13 parameters. More works in Table 1.2.
1.2. Image Analysis

<table>
<thead>
<tr>
<th>Reference</th>
<th>Pre-processing</th>
<th>Soma Segmentation</th>
<th>Tree Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Masseroli</td>
<td>Histogram equalization</td>
<td>THR (manual)</td>
<td>THR</td>
</tr>
<tr>
<td>1993, [74]</td>
<td>filter</td>
<td>MO</td>
<td>MO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Skeleton</td>
</tr>
<tr>
<td>Al-Kofahi</td>
<td></td>
<td>ATHR</td>
<td>TRAC (5 kernels)</td>
</tr>
<tr>
<td>2003, [75]</td>
<td></td>
<td>MO</td>
<td></td>
</tr>
<tr>
<td>Weaver</td>
<td>Laplacian HPF</td>
<td>THR</td>
<td>Local ATHR tests (polar, ridge, . . .)</td>
</tr>
<tr>
<td>2003, [72]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shah</td>
<td></td>
<td>THR (manual)</td>
<td>Sobel filter</td>
</tr>
<tr>
<td>2004, [73]</td>
<td></td>
<td>MO</td>
<td>MO</td>
</tr>
<tr>
<td>Al-Kofahi</td>
<td>Image Normalization</td>
<td>ATHR</td>
<td>TRAC</td>
</tr>
<tr>
<td>2006, [76]</td>
<td>Wiener filter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mitchell</td>
<td>BG correction</td>
<td>THR (blue channel)</td>
<td>HPF (green channel)</td>
</tr>
<tr>
<td>2007, [77]</td>
<td></td>
<td>Watershed filter</td>
<td>THR</td>
</tr>
<tr>
<td>Narro</td>
<td>Sobel filter</td>
<td></td>
<td>Gaussian blur</td>
</tr>
<tr>
<td>2007, [20]</td>
<td>BG substraction</td>
<td></td>
<td>Skeleton</td>
</tr>
<tr>
<td></td>
<td>Manual selection</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.4 – Image reconstruction algorithms divided into three phases: Image pre-processing, soma segmentation and tree segmentation. Glossary: MO = Morphological Operations, THR = Threshold, ATHR = Adaptive Threshold, TRAC = Tracing algorithm, HPF = high pass filter, LPF = low pass filter, BG = Background, LoG = Laplacian of Gaussian. More algorithms in Table 1.5.

points and iteratively predict the next point on the neurite by fitting a model (shape). Local algorithms to neurite tracing are based on mean-shift tracing [83], active contour [84, 85], region-growing [86] and minimizing energy functions [78] among others.

Table 1.4 and Table 1.5 briefly summarize the methods used in neurite tree segmentation.

1.2.2 Image Analysis and Graph Theory

As seen in the previous section and in Tables 1.2 and 1.3, most authors use global methods (like (adaptive) threshold, edges, morphological operations, and so on) or local methods (such as snakes, active contours and region-growing among others).

In others research fields, another approach is used that combines global and local information. Graph-based algorithms usually consider the segmentation problem like a graph-partition problem where each data (a pixel within an image in case of image segmentation) is represented by a node within the graph and the nodes are related by similarity functions.

Traditional graph cut methods such as minimum cut [87], ratio cut [88], average cut [89], normalized cut [90] and min/max cut [91] try to minimize or maximize an objective function to reach the data segmentation. However, obtaining the best segmentation with these methods is often a NP-hard problem [92].

By using spectral methods, the cut problem can be solved in polynomial time by
<table>
<thead>
<tr>
<th>Reference</th>
<th>Pre-processing</th>
<th>Soma Segmentation</th>
<th>Tree Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vallotton 2007, [23]</td>
<td>Gaussian filter</td>
<td>MO</td>
<td>Gaussian filter  Nonmaximum suppresion  MO</td>
</tr>
<tr>
<td>Zhang 2007, [78]</td>
<td>Intensity adjustment</td>
<td>fuzzy c-means</td>
<td>Seed point detection  Local minima  1D Gaussian LPF  TRAC</td>
</tr>
<tr>
<td>Leandro 2009, [82]</td>
<td>THR</td>
<td>MO</td>
<td></td>
</tr>
<tr>
<td>Schmitz 2011, [79]</td>
<td>2D adaptative Wiener LPF</td>
<td>THR</td>
<td>Gaussian filter  TRAC</td>
</tr>
<tr>
<td>Türetken 2011, [26]</td>
<td></td>
<td></td>
<td>Tubularity measure  THR  Graph construction  k-Minimum Spanning Tree  Maximization of a global function</td>
</tr>
<tr>
<td>Zhao 2011, [27]</td>
<td></td>
<td></td>
<td>3D filter  TRAC</td>
</tr>
<tr>
<td>Pani 2014, [22]</td>
<td>LoG filter</td>
<td>Gaussian blur filter  THR</td>
<td>BG substraction  Multiscale segmentation  THR  Gaussian blur (enhance edges)  LoG filter</td>
</tr>
</tbody>
</table>

**Table 1.5** – Image reconstruction algorithms divided into three phases: Image pre-processing, soma segmentation and tree segmentation. Glossary: MO = Morphological Operations, THR = Threshold, ATHR = Adaptive Threshold, TRAC = Tracing algorithm, HPF = high pass filter, LPF = low pass filter, BG = Background, LoG = Laplacian of Gaussian. More algorithms in Table 1.4.
1.2. Image Analysis

relaxing the original discrete optimization problem to the real domain [93]. Most spectral clustering algorithms are based on the Laplacian matrix spectrum considering two Laplacian matrices: un-normalized and normalized. Other approaches use the probability transition matrix [94] or modularity matrix [95, 96, 97, 98] among others.

1.2.3 Image analysis performance measure

Image segmentation algorithm must be evaluated by comparing the segmentation result with a hand-made segmentation or a semiautomatic segmentation, known as ground truth. This comparison is carried out by measuring the proportion of well classified pixels, that is, those pixels classified in the same way as in the ground truth (True Positive pixels or False Negative pixels) and the rate of bad classified pixels, meaning, the image segmentation algorithm classifies some pixels in a different way from the ground truth (False Positive pixels or False Negative pixels).

The accuracy of the system are usually given by the F-score defined by the precision of the segmentation and the recall or sensitivity. In that sense, F-score is a value within the interval [0, 1] where 0 means bad segmentation and 1 means the best segmentation (i.e. the segmentation result is the same as the ground-truth).

We provide definitions to quantify an algorithm accuracy [75, 99, 15]

**Definition 1.2.1** (Element comparison). Let $A$ and $B$ be binary matrices where $B$ represents the proposed system result and $A$ the matrix that has to be compared with. We define:

- **True positive** (TP) when $a_{ij} = b_{ij} = 1$
- **True negative** (TN) when $a_{ij} = b_{ij} = 0$
- **False positive** (FP) when $a_{ij} = 0$ and $b_{ij} = 1$
- **False negative** (FN) when $a_{ij} = 1$ and $b_{ij} = 0$

where $a_{ij}$ and $b_{ij}$ are the elements of the matrices $A$ and $B$ respectively.

**Definition 1.2.2** (Precision or Confidence). Let $P$ (Precision or Confidence) be the number of true positive elements divided by the number of elements in the result (i.e. $b_{ij} = 1$).

\[ P = \frac{TP}{TP + FP} \]

**Definition 1.2.3** (Recall or Sensitivity). Let $R$ (Recall or Sensitivity) be the number of true positive divided by the number of elements in the ground truth (i.e. $a_{ij} = 1$)

\[ R = \frac{TP}{TP + FN} \]
Definition 1.2.4 (F-score). Let \( F \) be the f-score measure defined as:

\[
F = \frac{2RP}{R+P}
\]

\( F \) is a value within the interval \([0, 1]\) where 0 means the worst score, i.e. \( B \) is opposite to \( A \), and 1 means the best score, i.e. \( B \) is equal to \( A \).

1.3 Complex Networks

Complex networks are currently being studied across many fields of science [100, 101, 102, 103]. Numerous systems in nature can be represented by means of complex networks. In essence, a complex network consists of a set of nodes connected by links, where network nodes stand for elements in the system to be described and the network links symbolize the relationship between those nodes. For instance, the world is a network of countries connected by airlines, the countries are themselves networks of cities connected by roads. The social networks are networks of people connected by friendship or acquaintanceship. The brain is a network of neurons.

The study of complex networks has its roots in a branch of discrete mathematics known as graph theory [28, 104]. The first proof in graph theory was held in 1736 when the Swiss mathematician and physicist Leonhard Euler solved the problem of the Königsberg’s bridges [105]. Euler proved, by representing the problem as a network, that it is not possible to cross exactly one time each of the seven bridges connecting the two islands in the Pregel river and its shores.

In the 1950s and 1960s, random graphs discovery was a major breakthrough [106, 107]. In random graphs, connections between the network nodes are present with a likelihood \( p \). On the one hand, this model showed that properties of the graphs often undergo a sudden transition, known as phase transition, as a function of increasing \( p \). On the other hand, this model can not explain some properties of real networks, for instance, the fact the distance between nodes in sparse networks were often much smaller than expected [104].

In 1967, the first empirical proof of this small-world phenomenon was quantified by Stanley Milgram [108]. Milgram asked randomly selected people in Nebraska to send letters to a distant target individual in Boston, identified only by his name, occupation and rough location. However, the subjects were only allowed to send the letter to another person they knew well, and who might possibly be a little closer to the target in Boston. As result, many letters arrived at the addressee, and on average each letter was sent only 5.5 times.

In late 1990s, Barabási et al. [109, 110] noted that many large-scale networks are scale-free, namely, their connectivity distributions are power-law. They concluded that “the development of large networks is governed by robust self-organizing phenomena that go beyond the particulars of the individual systems”.
In section 1.3.1 we introduce the basic concepts in graph theory and, in more detail, three concepts - the average path length, clustering coefficient and degree distribution - that play a key role in the field. These concepts have been used to categorize networks following the classification that we present in section 1.3.2 and implemented in Chapter 1.

### 1.3.1 Basic Concepts in Graph Theory

Graph theory is the natural framework for the mathematical analysis of complex networks. Formally, a complex network can be represented as a graph [28].

A graph is an ordered pair $G = (N, E)$ where $N$ is the set of nodes, also called vertices, in the graph and $E$ stands for the set of edges connecting those nodes ($E$ is an unordered or ordered set of graph’s nodes if the graph is undirected or directed respectively).

A node, $n_i$, is usually referred to by its order $i$ in the set $N$, $N \equiv \{n_1, n_2, \ldots, n_{|N|}\}$ where $|N|$ stands for the number of nodes in the set, its cardinality.

An edge, $e_{ij}$ is usually referred to by the nodes’ indexes that is connecting (nodes $i$ and $j$).

Usually, graphs are represented by a matrix, $M$, of size $|N| \times |N|$ where the row or column $i$ stands for the node $i$ in the graph. The matrix states the existence of a link between two nodes. This matrix is known as adjacency matrix.

$$M(i, j) = \begin{cases} 1 & \text{if } e_{ij} \in E \\ 0 & \text{otherwise} \end{cases}$$

Henceforth, the $e_{ij}$ nomenclature is used to refer to the value in $M(i, j)$.

An undirected graph fulfills that the adjacency matrix is symmetric and the diagonal elements are 0, i.e. $e_{ij} = e_{ji}$ and $e_{ii} = 0$, while in a directed graph (or digraph) the adjacency matrix is asymmetric and if there is a link ($e_{ij}$), its inverse ($e_{ji}$) does not exist, i.e. if $e_{ij} = 1$ then $e_{ji} = 0$. A graph is called mixed graph when there are directed and undirected links.

It may be noted that these three graphs (undirected, directed and mixed graphs) do not allow loops or several links between their nodes. A graph contains a loop if there is a link which starts and ends on the same node, i.e. $e_{ii} = 1$. Graphs with loops or multiple links connecting two nodes are called multigraph.
A weighted graph is an order triplet, $G = (N, E, W)$ where $N$ and $E$ are sets defined as in the directed and undirected graphs and $W$ is a set of values (weights), $w_{ij} \in W$, which are real numbers attached to the links. $W$ is usually represented by a $|N| \times |N|$ matrix (weights matrix $M^W$) where each element, $M^W(i, j)$ henceforth $w_{ij}$, represents the weight of the link connecting nodes $i$ and $j$. If there is no link connecting nodes $i$ and $j$ then $w_{ij} = 0$.

Weighted graphs can also be categorized as undirected, directed, mixed or multigraph weighted graphs.

Given a graph $G$ of size $|N|$, the number of edges $|E|$ is at least 0 and at most $\frac{1}{2}|N|(|N| - 1)$ for undirected graphs and at most $|N|(|N| - 1)$ for directed graphs. We define the link density, $\Delta$, of an undirected graph in Equation 1.1 and for directed graphs in Table 1.6.

The link density of a graph is within the interval $[0, 1]$, where $\Delta \ll 1$ means the graph is sparse, $\Delta \approx 1$ means the graph is dense and $\Delta = 1$ means the graph is complete, namely, the graph has all the nodes linked between them, denoted by $K_N$.

$$\Delta = \frac{2 \cdot |E|}{|N|(|N| - 1)} \quad (1.1)$$

A node $j$ is reachable from another node $i$ if there is a walk connecting them.

A walk is a sequence of nodes and edges that begins in $i$ and ends with $j$. The walk length is defined as the number of edges in the sequence. A walk without edge repetition is named tail. A walk without node repetition is denominated as path. The shortest path or geodesic is the walk with minimal length between two nodes. A walk starting and ending in the same node is a cycle. A walk is considered as simple cycle or circle when the walk is a cycle without edge and node repetitions.

A graph is a connected graph if for each pair of nodes $i$ and $j$ exists a walk connecting them, in other case, it is named unconnected graph. A graph is made up of components. A graph component is a subgraph, $G' = (N', E')$ such that $N' \subseteq N$ and $E' \subseteq E$ and $G'$ is a connected graph.

A connected graph is formed by one component known as giant component, while an unconnected graph is formed by several connected components, i.e. $N = N'_1 \cup \ldots \cup N'_s$ and $E = E'_1 \cup \ldots \cup E'_s$ where $s$ is the number of connected components within the graph.

Node degree, degree distributions and correlations
The degree $k_i$ of a node $i$ is the number of links incident to the node. Undirected graphs’ node degree is defined in Equation 1.2.

$$k_i = \sum_{j \in N} e_{ij} \quad (1.2)$$

In directed graphs, the degree of the node has two components, the number of outgoing links (out-degree, $k_{\text{out}}^i$) and the number of ingoing edges (in-degree, $k_{\text{in}}^i$) (Table 1.6). The total degree is defined as $k_i = k_{\text{out}}^i + k_{\text{in}}^i$.

The degree distribution $P(k)$ (or $P_k$) is referred to the fraction of nodes having degree $k$ and it is used as the basis for graph topological characterization. Directed graphs consider two distribution $P(k_{\text{in}})$ and $P(k_{\text{out}})$. The cumulative degree distribution $P_{\geq}(k)$ is the fraction of nodes within the graph with degree $k$ or bigger, and $P_{\leq}(k)$ is the fraction of nodes with degree $k$ or lower.

Information on how the degree is distributed among the nodes of an undirected graph can be obtained either by plotting $P(k)$, or by the calculation of the moments of the distribution. The $n$-moment of $P(k)$ is defined as

$$\langle k^n \rangle = \sum_k k^n \cdot P(k) \quad (1.3)$$

The first moment, $\langle k^1 \rangle$ or $\langle k \rangle$, is the mean degree of $G$. The second moment, $\langle k^2 \rangle$, measures the fluctuation of the connectivity distribution.

Assortativity is referred to the network preference to connect similar nodes between them. Assortativity is measured in terms of node’s degree distributions. We define the average nearest neighbors degree of node $i$ as

$$k_{\text{nn},i} = \frac{1}{k_i} \sum_{j=1}^{\left|\mathcal{N}\right|} e_{ij} \cdot k_j \quad (1.4)$$

By using the Equation 1.4 one can compute the average degree of the nearest neighbors of nodes with degree $k$, denoted as $k_{\text{nn}}(k)$. Graphs are classified as assortative if $k_{\text{nn}}(k)$ is an increasing function of $k$, whilst if $k_{\text{nn}}(k)$ is a decreasing function of $k$ is named disassortative [111]. Assortative means that nodes are connected to another nodes with similar degrees whilst disassortative means that there are hub nodes (nodes with high connectivity) to which nodes with low degree are connected.

---

**Node degree** number of links at node $i$

**Degree distribution** $P(k)$ fraction of nodes with degree $k$

**Cumulative degree distribution**

- $P_{\geq}(k)$ fraction of nodes with at least degree $k$
- $P_{\leq}(k)$ fraction of nodes with at most degree $k$

**$n$-moment**

- $\langle k \rangle$: $G$ degree mean
- $\langle k^2 \rangle$: $G$ degree variance
- $\langle k^3 \rangle$: $G$ degree skewness
- $\langle k^4 \rangle$: $G$ degree kurtosis

---

**Assortativity & Disassortativity**

**Assortativity:** nodes are connected to nodes with similar degrees.

**Disassortativity:** low degree nodes are connected to hub nodes.
Shortest path lengths, diameter and betweenness

The shortest path or geodesic is the walk with minimal length between two nodes. The shortest paths between pairs of nodes in the graph give a representative information about the graph structure and they can be represented as a matrix $D$ where each element within the matrix, $d_{ij}$, depicts the shortest path between node $i$ and $j$.

In a connected graph, the maximum value of the matrix $D$ is called the diameter of the graph, and it represents the longest geodesic length between nodes within the graph.

The mean value of the geodesic lengths over all couples of nodes is known as the average shortest path length or characteristic path length and it stands for the typical separation between two nodes.

$$L = \frac{1}{|N|(|N| - 1)} \sum_{i,j \in N} d_{ij} \quad (1.5)$$

In a disconnected graph, i.e. when there is at least a pair of nodes $(i,j)$ which are not connected through a path ($d_{ij} = \infty$), the definition in Equation 1.5 diverges. In order to avoid this issue, the characteristic path length is usually calculated over the largest connected component in the graph [29]. An alternative approach is reckoned on the graph efficiency [112] as the harmonic mean of geodesic lengths as an indicator of the network traffic capacity.

$$E = \frac{1}{|N|(|N| - 1)} \sum_{i,j \in N, i \neq j} \frac{1}{d_{ij}} \quad (1.6)$$

A measure of the importance of a node, $i$, is given by the node betweenness (betweenness centrality), $b_i$, [113, 114, 115, 116] which counts for the number of shortest paths connecting two nodes, $j$ and $k$, that pass through that node, $i$.

$$b_i = \sum_{j,k \in N, j \neq k} \frac{n_{jk}(i)}{n_{jk}} \quad (1.7)$$

where $n_{jk}$ is the number of shortest paths connecting $j$ and $k$, while $n_{jk}(i)$ is the number of shortest paths connecting $j$ and $k$ and passing through $i$.

The edge betweenness is defined as the number of shortest paths between pairs of nodes that run through that edge.

Clustering / Transitivity
1.3. Complex Networks

Clustering measures the acquaintance property of the network, where two nodes with a common neighbor are likely to connect each other [113], i.e. they form a triangle. The fraction of connected triples of nodes (triads) which also form triangles is known as transitivity of the graph[117, 118, 119]

\[ T = \frac{3 \times \# \text{ of triangles in } G}{\# \text{ of connected triples of vertexes in } G} \quad (1.8) \]

Watts and Strogatz defined the graph clustering coefficient in [29]. The local clustering coefficient of node \( i \), \( c_i \), expresses how likely is \( e_{jm} = 1 \) for two neighbors \( j \) and \( m \) of node \( i \). The local clustering coefficient is defined as the ratio between \( t_i \) (the number of triangles around node \( i \), Equation 1.10) and \( k_i(k_i-1)/2 \) (the total number of edges if all \( i \) neighbors formed triangles with \( i \)).

\[ c_i = \frac{2t_i}{k_i(k_i-1)} = \frac{\sum_{j,m} e_{ij}e_{jm}e_{mi}}{k_i(k_i-1)} \quad (1.9) \]

\[ t_i = \frac{1}{2} \sum_{j,m \in N, j \neq m} e_{ij}e_{jm}e_{mi} \quad (1.10) \]

The clustering coefficient of the graph is then given by the average of \( c_i \) over all the nodes in \( G \):

\[ C = \langle c \rangle = \frac{1}{|N|} \sum_{i \in N} c_i \quad (1.11) \]

By definition, \( 0 \leq c_i \leq 1 \) and \( 0 \leq C \leq 1 \). The differences between \( C \) and \( T \) are illustrated in [117, 30]. The average of \( c_i \) taken over all nodes with a given degree \( k \), \( c(k) \), is the so-called clustering coefficient of a connectivity class \( k \).

An alternative measure of the clustering properties of \( G \) is the local efficiency [120, 30], defined as

\[ E_{loc} = \frac{1}{|N|} \sum_{i \in N} E(G_i) \quad (1.12) \]

where \( E(G_i) \) is the efficiency of \( G_i \) (Equation 1.6) and \( G_i \) is the subgraph formed by the neighbors of the node \( i \).

1.3.2 Complex Networks Models

Regular Networks
Regular networks are classified into two categories, lattice and star-shaped coupled networks.

A lattice is a regular graph in which every node is joined only by a few of its neighbors. A nearest-neighbor lattice with a periodic boundary condition consists of \( N \) nodes positioned in a ring, where each node \( i \) is adjacent to its \( K \) first-neighboring nodes, where \( K \) is the expected node degree (\( K \) being an even integer).

The clustering coefficient for lattice networks is defined as
\[
C_{\text{reg}} = \frac{3(K-2)}{4(K-1)}
\]
and the average shortest path is defined as
\[
L_{\text{reg}} = \frac{|N|}{2K}.
\]
For a large \( K \), the clustering coefficient of lattice networks is approximately \( 3/4 \). For the aforementioned, the regular network average shortest path is quite large and tends to infinity as \( |N| \to \infty \), [103].

A star-shaped network is a regular network in which there is a center node and each of the other \( |N| - 1 \) nodes only connect to this center but not among themselves. In this case, the average path length tends to 2 and the clustering coefficient tends to 1, as \( |N| \to \infty \).

Random Graphs

Random graphs were first studied by Erdős and Rényi (ER) [107] with the original motivation of analyzing, by means of probabilistic methods, the properties of graphs as a function of the increasing number of random connections. A remarkable discovery was that important properties of random graphs vary as a function of \( p \), showing a dramatic change at a critical probability \( p_c \), corresponding to a critical average degree \( \langle k \rangle_c = 1 \).

ER showed that,

- if \( p < p_c = \frac{1}{|N|} \), the graph has no components of size greater than \( O(\ln|N|) \).
- if \( p = p_c = \frac{1}{|N|} \), the largest component has size \( O(|N|^{2/3}) \).
- if \( p > p_c = \frac{\ln|N|}{|N|} \), the graph has a component of \( O(|N|) \) and no other component has more than \( O(\ln|N|) \) nodes.
- if \( p > p_c = \frac{\ln|N|}{\langle k \rangle} \), then almost every random graph is connected.

The clustering coefficient for an ER random network is defined as
\[
C_{\text{rand}} = p = \frac{\langle k \rangle}{|N|} \ll 1
\]
and the average shortest path is defined as
\[
L_{\text{rand}} \sim \frac{\ln|N|}{\langle k \rangle}.
\]
For large-scale random networks, the ER model generates an homogeneous network where the degree distribution is well approximated by a Poisson distribution [103, 28]

\[
P(k) = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}
\]
1.3. Complex Networks

Small-World Models

The small-world is the property which indicates that two nodes within the graph are connected by a path whose length is much smaller than the number of nodes. For instance, regular lattices are clustered but do not exhibit the small-world effect. However, star-shaped regular networks are highly clustered and exhibit the small-world effect but, actually, most of the real-networks do not present a star-shaped structure. Moreover, random graphs show the small-world effect but do not show clustering. Therefore, regular lattice model and the ER random model both fail to reproduce some important features of many real networks.

A small-world network preserves the local neighborhood (as for regular lattices [121]) and the network diameter increases logarithmically with the number of vertices (as for random graphs [122]).

Watts and Strogatz (WS) [29] proposed a small-world network model based on a rewiring procedure of the edges of a lattice network with a probability $p$.

The WS algorithm starts with a lattice network with $N$ nodes arranged in a ring. Each edge is randomly rewired with probability $p$. Varying $p$ from $p = 0$ to $p = 1$ the transition between the lattice and randomness, respectively, can be monitored (Figure 1.6).

Rewiring within this context means shifting one end of the connection to a new node chosen at random from the whole network, with the constraints that loops and multiple edges are not allowed. This procedure introduces $pNK/2$ long-range edges.

Therefore, the clustering coefficient $C(p)$ and the average path length $L(p)$ in the WS small-world model can be considered as a function of the rewiring probability $p$. A regular ring lattice ($p = 0$) is highly clustered ($C(0) \cong 3/4$) but has a large average
Figure 1.7 – Average path length and clustering coefficient of the WS small-world model as a function of the rewiring probability \( p \). Both are normalized to their values for the original regular lattice \( (p = 0) \). Note that there is a broad range for the rewiring probability \( p \) where networks have clustering that is similar to that of the regular network, and a path length that is similar to that of the random network. Within this range, networks exhibit small-world attributes. The data shown in the figure are averages over 100 random realizations of the rewiring process, and have been normalized by the values \( L(0), C(0) \) for a regular lattice. All the graphs have 500 vertices and an average degree of 4 edges per vertex.

Amaral et al. [123] studied diverse real-world networks and evidenced the occurrence of three classes of small-world networks, namely, scale-free networks, broad-scale networks and single-scale networks.

Scale-free networks are characterized by a degree distribution that decays with a power law tail (e.g. world-wide web, electric power grid, network of world airports). Broad-scale networks are characterized by a degree distribution that has a power law regime followed by a sharp cutoff (e.g. movie-actor network, acquaintance network of Mormons).

Single-scale networks are characterized by a connectivity distribution with a fast decaying tail, such as exponential or Gaussian (e.g. neuronal network of the worm Caenorhabditis elegans).

Scale-free networks emerge in the context of a growing network in which new nodes prefer to connect to the highly connected nodes in the network. When there are constraints limiting the addition of new links, like aging of the vertices or cost of adding links to the vertices or the limited capacity of a vertex, then broad-scale or single-scale small-world networks appears [123].

path length \( (L(0) \approx N^2 K \gg 1) \). For a small rewiring probability, the network clustering coefficient does not differ from its initial value \( (C(p) \sim C(0)) \), but the average path length downfall rapidly and it is in the same order as that of random networks \( (L(p) \sim L(1)) \) (Fig. 1.7) [103].
Table 1.6 – Mathematical definitions of complex network measures. Extracted from [124] and expanded.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Binary and undirected definitions</th>
<th>Weighted and directed definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic concepts and notation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N$ is the set of all nodes in the network, and $n$ is the number of nodes. $L$ is the set of all links in the network, and $l$ is the number of links. $(i,j)$ is a link between nodes $i$ and $j$, $(i,j) \in N$. $e_{ij}$ is the connection status between $i$ and $j$: $e_{ij} = 1$ when the link $(i,j)$ exists (when $i$ and $j$ are neighbors); $e_{ij} = 0$ otherwise ($e_{ii} = 0$ for all $i$). We compute the number of links as $l = \sum_{i,j \in N} e_{ij}$ (to avoid ambiguity with directed links we count each undirected link twice, as $e_{ij}$ and as $e_{ji}$).</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| **Link density** | | |
| The fraction of possible links that exist in a graph. | $\Delta = \frac{2 \cdot l}{n(n-1)}$ | $\Delta^\rightarrow = \frac{l}{n(n-1)}$ |

| **Degree** | | |
| Degree of a node $i$, $k_i = \sum_{j \in N} e_{ij}$, $(\text{Directed})$ total degree of $i$, $k_i = \sum_{j \in N} e_{ij}$, $(\text{Directed})$ out-degree of $i$, $k_i^{\text{out}} = \sum_{j \in N} e_{ij}$, $(\text{Directed})$ in-degree of $i$, $k_i^{\text{in}} = \sum_{j \in N} e_{ji}$, $(\text{Direct})$ total degree of $i$, $k_i = k_i^{\text{out}} + k_i^{\text{in}}$. | Weighted degree of $i$, $k_i = \sum_{j \in N} w_{ij}$, Shortest weighted path length between $i$ and $j$, $d_{ij}^{\text{w}} = \sum_{e_{uv} \in g_{i \rightarrow j}} f(w_{uv})$, where $f$ is a map (e.g., an inverse) from weight to length and $g_{i \rightarrow j}$ is the shortest weighted path between $i$ and $j$. Shortest directed path length from $i$ to $j$, $d_{ij}^{\text{d}} = \sum_{e_{ij} \in g_{i \rightarrow j}} e_{ij}$ where $g_{i \rightarrow j}$ is the directed shortest path from $i$ to $j$. | |

| **Shortest path length** | | |
| Shortest path length (distance), between nodes $i$ and $j$, $d_{ij} = \sum_{e_{uv} \in g_{i \rightarrow j}} e_{uv}$, where $g_{i \rightarrow j}$ is the shortest path (geodesic) between $i$ and $j$. Node that $d_{ij} = \infty$ for all disconnected pairs $i, j$. | (Weighted) geometric mean of triangles around $i$, $t_i^{\text{w}} = \frac{1}{2} \sum_{j,h \in N} (w_{ij} w_{ih} w_{jh})^{\frac{1}{2}}$. Number of directed triangles around $i$, $t_i^{\text{d}} = \frac{1}{2} \sum_{j,h \in N} (e_{ij} + e_{ji}) (e_{ih} + e_{hi}) (e_{jh} + e_{hj})$. | |

| **Number of triangles** | | |
| Number of triangles around a node $i$, $t_i = \frac{1}{2} \sum_{j,h \in N} e_{ij} e_{ih} e_{jh}$. | Weighted clustering coefficient [125], $C_i^{\text{w}} = \frac{1}{n^2} \sum_{i,j \in N} \frac{2t_{ij}^{\text{w}}}{(k_i + k_j) (k_i + k_j - 1)}$. See [126] for other variants. Directed clustering coefficient ([127]), $C_i^{\text{d}} = \frac{1}{n} \sum_{i \in N} \frac{t_i^{\text{d}}}{(k_i^{\text{out}} + k_i^{\text{in}}) (k_i^{\text{out}} + k_i^{\text{in}} - 1) - 2 \sum_{j \in N} e_{ij} e_{ji}}$. | |

| **Clustering coefficient** | | |
| Clustering coefficient of the network [29], $C = \frac{1}{n} \sum_{i \in N} C_i = \frac{1}{n^2} \sum_{i \in N} \frac{2t_i}{k_i (k_i - 1)}$, where $C_i$ is the cluster coefficient of node $i$. (For $k_i < 2$, $C_i = 0$.) | Weighted clustering coefficient [125], $C_i^{\text{w}} = \frac{1}{n^2} \sum_{i,j \in N} \frac{2t_{ij}^{\text{w}}}{(k_i + k_j) (k_i + k_j - 1)}$. See [126] for other variants. Directed clustering coefficient ([127]), $C_i^{\text{d}} = \frac{1}{n} \sum_{i \in N} \frac{t_i^{\text{d}}}{(k_i^{\text{out}} + k_i^{\text{in}}) (k_i^{\text{out}} + k_i^{\text{in}} - 1) - 2 \sum_{j \in N} e_{ij} e_{ji}}$. | |

| **Transitivity** | | |
| Transitivity of the network [117], $T = \sum_{i \in N} \frac{2t_i}{k_i (k_i - 1)}$. Note that transitivity is not defined for individual nodes. | Weighted transitivity [124], $T_i^{\text{w}} = \frac{1}{n} \sum_{i \in N} \frac{2t_{ij}^{\text{w}}}{(k_i^{\text{out}} + k_i^{\text{in}}) (k_i^{\text{out}} + k_i^{\text{in}} - 1) - 2 \sum_{j \in N} e_{ij} e_{ji}}$. Directed transitivity [124], $T_i^{\text{d}} = \frac{1}{n} \sum_{i \in N} \frac{t_i^{\text{d}}}{(k_i^{\text{out}} + k_i^{\text{in}}) (k_i^{\text{out}} + k_i^{\text{in}} - 1) - 2 \sum_{j \in N} e_{ij} e_{ji}}$. | |

Continued on next page
Table 1.6 – continued from previous page

<table>
<thead>
<tr>
<th>Measure</th>
<th>Binary and undirected definitions</th>
<th>Weighted and directed definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Local efficiency</strong></td>
<td>Local efficiency of the network [120],</td>
<td>Weighted local efficiency [124],</td>
</tr>
<tr>
<td></td>
<td>$E_{loc} = \frac{1}{n} \sum_{i \in N} E_{loc,i}$</td>
<td>$E_{loc}^w = \frac{1}{2} \sum_{i \in N} \left( w_{i,j} w_{i,k} [d_{jh}^w (N_i)]^{-1} \right)^{\frac{1}{2}}$</td>
</tr>
<tr>
<td></td>
<td>$E_{loc,i} = \sum_{j,h \in N, j \neq i} e_{ij} e_{ih} [d_{jh} (N_i)]^{-1}$</td>
<td>$k_i (k_i - 1)$</td>
</tr>
<tr>
<td></td>
<td>where $E_{loc,i}$ is the local efficiency of node $i$, and</td>
<td>Directional local efficiency [124],</td>
</tr>
<tr>
<td></td>
<td>$d_{jh} (N_i)$ is the length of the shortest path between</td>
<td>$E_{loc}^+ = \frac{1}{2} \sum_{i \in N} \left( \sum_{j,h \in N, j \neq i} (e_{ij} + e_{ji}) (e_{ih} + e_{hi}) \left( [d_{jh} (N_i)]^{-1} + [d_{ih} (N_i)]^{-1} \right) \right)$</td>
</tr>
<tr>
<td></td>
<td>$j$ and $h$, that contains only neighbors of $i$.</td>
<td></td>
</tr>
<tr>
<td><strong>Modularity</strong></td>
<td>Modularity of the network [128],</td>
<td>Weighted modularity [93],</td>
</tr>
<tr>
<td></td>
<td>$Q = \sum_{u \in M} \left[ e_{uu} - \left( \sum_{v \in M} e_{uv} \right)^2 \right]$</td>
<td>$Q^w = \frac{1}{l^w} \sum_{i,j \in N} \left( w_{ij} - \frac{k_i^w k_j^w}{l^w} \right) \delta_{m_i,m_j}$</td>
</tr>
<tr>
<td></td>
<td>where the network is fully subdivided into a set of</td>
<td>Directed modularity [97],</td>
</tr>
<tr>
<td></td>
<td>non-overlapping modules $M$, and $e_{uv}$ is the proportion of all</td>
<td>$Q^+ = \frac{1}{l} \sum_{i,j \in N} \left( e_{ij} - \frac{k_i^+ k_j^+}{l} \right) \delta_{m_i,m_j}$</td>
</tr>
<tr>
<td></td>
<td>links that connect nodes in module $u$ with nodes in module $v$.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>An equivalent alternative of the modularity [98] is given by $Q = \frac{1}{l} \sum_{i,j \in N} \left( e_{ij} - \frac{k_i^l k_j^l}{l} \right) \delta_{m_i,m_j}$,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>where $m_i$ is the module containing node $i$, and $\delta_{m_i,m_j} = 1$ if $m_i = m_j$, and 0 otherwise.</td>
<td></td>
</tr>
</tbody>
</table>

**Measures of centrality**

<table>
<thead>
<tr>
<th>Measure</th>
<th>Closeness centrality of node $i$ [116],</th>
<th>Weighted closeness centrality,</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_i^{-1} = \frac{n - 1}{\sum_{j \in N, j \neq i} d_{ij}}$</td>
<td>$(L_i^w)^{-1} = \frac{n - 1}{\sum_{j \in N, j \neq i} d_{ij}^w}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Directed closeness centrality,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$(L_i^+)^{-1} = \frac{n - 1}{\sum_{j \in N, j \neq i} d_{ij}^+}$</td>
</tr>
<tr>
<td><strong>Betweenness centrality</strong></td>
<td>Betweenness centrality of node $i$ [116],</td>
<td>Betweenness centrality is computed equivalently on weighted and directed networks, provided that path lengths are computed on respective weighted or directed paths.</td>
</tr>
<tr>
<td></td>
<td>$b_i = \frac{1}{(n - 1)(n - 2)} \sum_{h,j \in N, h \neq i,j \neq i} \rho_{h,j(i)} / \rho_{h,j}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>where $\rho_{h,j}$ is the number of shortest paths between $h$ and $j$, and $\rho_{h,j(i)}$ is the number of shortest paths between $h$ and $j$ that pass through $i$.</td>
<td></td>
</tr>
</tbody>
</table>
### 1.3. Complex Networks

#### Table 1.6 – continued from previous page

<table>
<thead>
<tr>
<th>Measure</th>
<th>Binary and undirected definitions</th>
<th>Weighted and directed definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Within-module degree z-score of node $i$ $[129]$</td>
<td>$z_i = \frac{k_i(m_i) - \bar{k}(m_i)}{\sigma_k(m_i)}$</td>
<td>$z_i^w = \frac{k_i^w(m_i) - \bar{k}^w(m_i)}{\sigma_k^w(m_i)}$</td>
</tr>
<tr>
<td></td>
<td>where $m_i$ is the module containing node $i$, $k_i(m_i)$ is the within-module degree of $i$ (the number of links between $i$ and all other nodes in $m_i$), and $\bar{k}(m_i)$ and $\sigma_k(m_i)$ are the respective mean and standard deviation of the within-module $m_i$ degree distribution.</td>
<td>Within-module out-degree z-score, $z_i^{out} = \frac{k_i^{out}(m_i) - \bar{k}^{out}(m_i)}{\sigma_k^{out}(m_i)}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Within-module in-degree z-score, $z_i^{in} = \frac{k_i^{in}(m_i) - \bar{k}^{in}(m_i)}{\sigma_k^{in}(m_i)}$</td>
</tr>
<tr>
<td>Participation coefficient of node $i$ $[129]$</td>
<td>$y_i = 1 - \sum_{m \in M} \left( \frac{k_i(m)}{k_i} \right)^2$</td>
<td>Weighted participation coefficient, $y_i^w = 1 - \sum_{m \in M} \left( \frac{k_i^w(m)}{k_i^w} \right)^2$</td>
</tr>
<tr>
<td></td>
<td>where $M$ is the set of modules (see modularity), and $k_i(m)$ is the number of links between $i$ and all nodes in module $m$.</td>
<td>Out-degree participation coefficient, $y_i^{out} = 1 - \sum_{m \in M} \left( \frac{k_i^{out}(m)}{k_i^{out}} \right)^2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In-degree participation coefficient, $y_i^{in} = 1 - \sum_{m \in M} \left( \frac{k_i^{in}(m)}{k_i^{in}} \right)^2$</td>
</tr>
<tr>
<td><strong>Network motifs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anatomical and functional motifs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$J_h$ is the number of occurrences of motif $h$ in all subsets of the network (subnetworks). $h$ is an $n_h$ node, $l_h$ link, directed connected pattern. $h$ will occur as an anatomical motif in an $n_h$ node, $l_h$ link subnetwork, if links in the subnetwork match links in $h$ $[130]$. $h$ will occur (possibly more than once) as a functional motif in an $n_h$ node, $l'_h$ ≥ $l_h$ link subnetwork, if at least one combination of $l_h$ links in the subnetwork matches links in $h$ $[59]$.</td>
<td>(Weighted) intensity of $h$ $[125]$, $I_h = \sum_u (\Pi_{(i,j) \in L_{h,u}} w_{ij}) \frac{\hat{h}}{k}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>where the sum is over all occurrences of $h$ in the network, and $L_{h,u}$ is the set of links in the $u$th occurrence of $h$. Note that motifs are directed by definition.</td>
</tr>
<tr>
<td>Motif z-score</td>
<td>$z_h = \frac{J_h - \langle J_{rand,h} \rangle}{\sigma_{J_{rand,h}}}$ where $\langle J_{rand,h} \rangle$ and $\sigma_{J_{rand,h}}$ are the respective mean and standard deviation for the number of occurrences of $h$ in an ensemble of random networks.</td>
<td>Intensity z-score of motif $h$ $[125]$, $z_h^I = \frac{I_h - \langle I_{rand,h} \rangle}{\sigma_{I_{rand,h}}}$ where $\langle I_{rand,h} \rangle$ and $\sigma_{I_{rand,h}}$ are the respective mean and standard deviation for the intensity of $h$ in an ensemble of random networks.</td>
</tr>
</tbody>
</table>

Continued on next page
Table 1.6 – continued from previous page

<table>
<thead>
<tr>
<th>Measure</th>
<th>Binary and undirected definitions</th>
<th>Weighted and directed definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motif fingerprint</td>
<td>$n_h$ node motif fingerprint of the network $[59]$, $F_{nh}(h') = \sum_{i \in N} F_{nh,i}(h') = \sum_{i \in N} J_{h', i}$</td>
<td>$n_h$ node motif intensity fingerprint of the network, $F_{nh}^I(h') = \sum_{i \in N} F_{nh,i}^I(h') = \sum_{i \in N} I_{h', i}$</td>
</tr>
<tr>
<td>Measures of resilience</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Degree distribution</td>
<td>Cumulative degree distribution of the network $[109]$, $P_\geq(k) = \sum_{k' \geq k} p(k')$</td>
<td>Cumulative weighted degree distribution, $P_w(k^w) = \sum_{k' \geq k^w} p(k')$</td>
</tr>
<tr>
<td>Average neighbor degree</td>
<td>Average degree of neighbors of node $i$ $[131]$, $k_{n,n,i} = \sum_{j \in N} e_{ij} k_{ij} / k_i$</td>
<td>Average weighted neighbor degree (modified from $[132]$, $k_{n,w,i}^w = \sum_{j \in N} w_{ij} k_{ij}^w / k_i^w$</td>
</tr>
<tr>
<td>Assortativity coefficient</td>
<td>Assortativity coefficient of the network $[111]$, $r = \frac{2^{-1} \sum_{(i,j) \in E} w_{ij} k_{ij} - 2^{-1} \sum_{(i,j) \in E} \frac{1}{2} (k_i + k_j)}{2^{-1} \sum_{(i,j) \in E} \frac{1}{2} (k_i + k_j)^2}$</td>
<td>Weighted assortativity coefficient (modified from $[133]$, $r_w = \frac{2^{-1} \sum_{(i,j) \in E} w_{ij} k_{ij}^w - 2^{-1} \sum_{(i,j) \in E} \frac{1}{2} (k_i^w + k_j^w)}{2^{-1} \sum_{(i,j) \in E} \frac{1}{2} (k_i^w + k_j^w)^2}$</td>
</tr>
<tr>
<td>Other concepts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Degree distribution preserving network</td>
<td>Degree-distribution preserving randomization is implemented by iteratively choosing four distinct</td>
<td>The algorithm is equivalent for weighted and directed networks.</td>
</tr>
<tr>
<td>randomization</td>
<td>nodes $i_1, j_1, i_2, j_2 \in N$ at random, such that links $(i_1, j_1), (i_2, j_2) \in L$, while</td>
<td>In weighted networks, weights may be switched together with links; in this case, the weighted</td>
</tr>
<tr>
<td></td>
<td>links $(i_1, j_2), (i_2, j_1) \notin L$. The links are then rewired such that $(i_1, j_2), (i_2, j_1) \in L$</td>
<td>degree distribution is not preserved, but may be subsequently approximated on the topologically</td>
</tr>
<tr>
<td></td>
<td>“Latticization” (a lattice-like topology) results if an additional constraint is imposed, $</td>
<td>i_1 + j_1</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Measure of network small-worldness</th>
<th>Binary and undirected definitions</th>
<th>Weighted and directed definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network small-worldness [135],</td>
<td>$S = \frac{C}{C_{\text{rand}}} \frac{L}{L_{\text{rand}}}$</td>
<td>Weighted network small-worldness,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$S^w = \frac{C^w}{C_{\text{rand}}^w} \frac{L^w}{L_{\text{rand}}^w}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$S^{\rightarrow} = \frac{C^{\rightarrow}}{C_{\text{rand}}^{\rightarrow}} \frac{L^{\rightarrow}}{L_{\text{rand}}^{\rightarrow}}$</td>
</tr>
<tr>
<td></td>
<td>where $C$ and $C_{\text{rand}}$ are the clustering coefficients, and $L$ and $L_{\text{rand}}$ are the characteristic path lengths of the respective tested network and a random network. Small-world networks often have $S \gg 1$.</td>
<td>In both cases, small-world networks often have $S \gg 1$.</td>
</tr>
</tbody>
</table>
CHAPTER 2

Image analysis

Contents

2.1 Multi-layer Graph-Based Segmentation Algorithm . . . . . . 36
  2.1.1 Graph initialization: The zero layer . . . . . . . . . . . . . 39
  2.1.2 Nodes Aggregation: From one layer to the next . . . . . . . 40
  2.1.3 Communities: the next layer . . . . . . . . . . . . . . . . . . . 42
  2.1.4 Stop condition: closing the loop . . . . . . . . . . . . . . . . . . 42
  2.1.5 Parameter analysis . . . . . . . . . . . . . . . . . . . . . . . . 43
  2.1.6 Computational cost analysis . . . . . . . . . . . . . . . . . . . 45

2.2 Application to invertebrate neuronal cultures . . . . . . . . 45
  2.2.1 Neuronal cultures and imaging . . . . . . . . . . . . . . . . . . . 46
  2.2.2 Culture imaging artifacts . . . . . . . . . . . . . . . . . . . . . 47
  2.2.3 Image processing strategy . . . . . . . . . . . . . . . . . . . . . 51
  2.2.4 Image segmentation pipeline . . . . . . . . . . . . . . . . . . . 53
  2.2.5 Network structure representation and measures . . . . . . . . 56
  2.2.6 Algorithm’s accuracy evaluation . . . . . . . . . . . . . . . . . . 57
  2.2.7 Computational cost . . . . . . . . . . . . . . . . . . . . . . . . . 65

Possibly, the most relevant advantage of CNNs is the unique option they offer of following the footprints of the self (or induced) organization of the networks functionality and dynamics [usually by means of a multielectrode array (MEA) or calcium fluorescence, recording the CNN electrophysiological data, or inducing electrical stimulations in given spatial positions] together with the monitoring and tracking of the structural
organization of the neurons connectivity along the entire course of the cultures growth [16, 14, 136]. While culturing neurons on top of a MEA equipped chamber implicates, in general, only mild constraints, following the development of the cultures structure is a far more delicate issue. Indeed, image-based systems biology essentially requires to gather sequential imaging of the culture and its processing in order to seek the evolution of the main networks indicators and measures along the CNNs maturation [76, 15, 136]. Now, it turns out that the accuracy of the existing image processing tools (as e.g. NeuronJ [18], NeuriteTracer [19], NeuronMetrics [20], NeurphologyJ [21], MorphoNeuroNet [22], HCA-Vision [23], and as we discussed in Chapter 1 (Section 1.2), in segmenting neurons and neurons connections depends crucially on furnishing, as inputs, pictures with a high level of contrast. This traditionally led experimentalists to rely on immunocytochemistry techniques, which however implies cell fixation and death. As a result, while comparing different CNNs at different stages of their maturation is nowadays a common practice, tracking variations of the networks structure on a single CNN requires a completely novel approach, wherein image processing could analyze non-invasively image acquisitions.

In this thesis, we describe in Section 2.1 a novel graph-based segmentation algorithm which is adapted in Section 2.2 to operate on large scale images of invertebrate neuronal cultures of acquired by phase-contrast microscopy, and therefore by a fully non invasive technique, i.e. without the need of adding chemicals to the culture. The algorithm accurately identifies the relevant networks units, and reconstructs the wiring of network connectivity with an overall computational cost (in terms of time and memory) which scales linearly with the image size.

## 2.1 Multi-layer Graph-Based Segmentation Algorithm

Graph-based segmentation algorithms are powerful tools to segment an image without loosing the spatial information.

The proposed algorithm is a multi-layer graph-based segmentation procedure which groups neighbor pixels with similar properties creating communities of pixels. In each step, communities can be merged with other neighbor communities creating a bigger one. The algorithm merges communities until a goal, known as stop condition, is reached like e.g. a given number of steps or clusters (Figure 2.1).

In that sense, each step can be considered as a layer of the segmentation procedure while the relationship between communities and pixels creates the graph structure.

The algorithm can be broken down into the following actions:

**Initialization** Each image pixel is considered as a graph’s node, and the relationship between nodes is defined. Usually a 4- or 8- *neighborhood* layout.

**Aggregation** A similarity function is used to group those nodes matching a certain criterion. Typically, the adopted rule is such that the similarity value between two nodes is below a given threshold.

**Communities** The new nodes and their relationship (neighborhood) are defined.
2.1. Multi-layer Graph-Based Segmentation Algorithm

Figure 2.1 – Schematic illustration of the layer aggregation process in the graph-based algorithm for image segmentation (left and right columns are equivalent representations). The multi-layer algorithm starts defining the first layer (the zero-layer $L_0$) as an input image (10x10 pixels of blue, yellow, red, green and white color) which is mapped into a graph whose nodes are the pixels of the image and each node’s neighborhood is formed by its 8 nearest neighbors. Solid lines in each layer depict the relationship (connectivity) between nodes which, for $L_0$ is defined only by the spatial location. For the next layers, the algorithm merges those nearby nodes whose relationship (similar color) is below a given threshold. The merged nodes in layer $L_0$ (for example the blue pixels which are neighbors in the grid) are mapped into one node (or region in the representation on the left) in the next layer $L_1$ and, two nodes are said to be neighbors if the regions they are coming from are neighbors in $L_0$. In the representation on the left, nodes sharing similar features are represented as regions while in the one on the right, each region is represented by the first node in that region. Dashed lines between layers account for those nodes in the lower layer that are grouped into the same node in the upper one. Notice that, for clarity, not all the dashed lines are drawn. Finally, the algorithm stops when a condition criterion is met (e.g. a given number of layers or number of regions). In this example, the algorithm transforms a grid representation into a single fade violet node (or region).
Stop condition The aggregation and community creation actions are repeated until a stop condition is reached. The aggregation criterion can be changed at each step in order to group those communities whose relationship does not meet the above criterion.

The following definitions, taken from the general graph theory, are necessary to introduce the segmentation algorithm.

**Definition 2.1.1** (Weighted graph). A weighted graph, $G$, is defined by $G = \{V, E, W\}$ where $V$ is the set of nodes $v_i \in V$, $E$ stands for the set of edges $e_{ij} \in E$ connecting nodes $v_i$ and $v_j$, and the set $W$ provides a value $w_{ij}$ to the edge $e_{ij}$, weight, representing the similarity between the connected nodes $v_i$ and $v_j$.

**Definition 2.1.2** (Multi-layer graph). Let $G^L = \{V^L, E^L, W^L\}$ be the graph in layer $L$. We define a multi-layer graph as $G = \{V, E, W\}$ where:

- $V = V^0 \cup V^1 \cup \ldots \cup V^L$
- $E = \{e_{ij} : \exists v_j \in V^L \land \exists v_i \in V^{L+1}\} \cup E^0 \cup \ldots \cup E^L$
- $W = W^0 \cup W^1 \cup \ldots \cup W^L$

The proposed algorithm calculates several quantities in the merged regions which are stored as nodes’ properties. For example, if the input matrix is a RGB image, each node is characterized by the mean, maximum and minimum color of the neighborhood’s color calculated in each component R, G and B. The following definition 2.1.3 provides a full description of a node and node’s properties.

**Definition 2.1.3** (Node). Let $v^L_i(s)[f]$ be a node in the set $V^L$ of the graph $G^L$ where:

- $i$ is the node’s identifier
- $L$ is the graph’s layer
- $s$ is the property name (mean, min, max and nelements)

\[
\begin{align*}
&v^L_i(\text{mean}), v^L_i(\text{max}), v^L_i(\text{min}) \in \mathbb{R}^F \\
&\text{nelements} \in \mathbb{Z}^+
\end{align*}
\]

- $f \in [1, F]$ being $F \in \mathbb{Z}^+$ the number of components (e.g. $F = 3$ for RGB images, while $F = 1$ in the case of gray-scale images)

**Definition 2.1.4** (Edge). Let $v^L_i$ and $v^L_j$ be a pair of nodes at layer $L$. We define an edge $e_{ij}^L$ as

\[
e_{ij}^L = \begin{cases} 
1 & \text{if } v_j \in N^L(i) \\
0 & \text{otherwise}
\end{cases}
\]

where $N^L(i)$ represents the neighborhood of the node $v^L_i$. 


The set $N^L(i)$ will be formally defined for the zero $L_0$ and non-zero layers $L_i$ in Sections 2.1.1 and 2.1.3, respectively.

The relationship between neighboring nodes in the same layer, known as similarity function, measures how similar the properties of those nodes are. In our case, in order to weigh the edges between nodes (Definition 2.1.6), we consider the Euclidean distance between the mean property of the nodes (Definition 2.1.5). Other more complex definitions of distances beyond the scope of this Thesis.

**Definition 2.1.5 (Similarity function).** Let $\sigma(v^L_i, v^L_j)$ be the similarity function between nodes $v^L_i$ and $v^L_j$ in $V^L$ defined as the Euclidean distance between the mean properties of those nodes

$$\sigma(v^L_i, v^L_j) = \sqrt{\sum_{f=1}^{F} (v^L_i(\text{mean})[f] - v^L_j(\text{mean})[f])^2}$$

**Definition 2.1.6 (Link weight).** Let $w^L_{ij} \in W^L$ be the weight of the link between nodes $v^L_i$ and $v^L_j$ in $V^L$ defined as

$$w^L_{ij} = \sigma(v^L_i, v^L_j)$$

**Property 2.1.1 (Similarity function properties).** The similarity function $\sigma$ fulfills the following properties

1. is symmetric, $\sigma(v^L_i, v^L_j) = \sigma(v^L_j, v^L_i)$
2. is positive defined, $\sigma(v^L_i, v^L_j) \geq 0$
3. $\sigma(v^L_i, v^L_j) = 0 \Leftrightarrow \forall f, v^L_i(\text{mean})[f] = v^L_j(\text{mean})[f]$

We can establish an equivalence relation between nodes in $V^L$ if the edge’s weight between them is below a given threshold:

$$\forall i, j \in D^L \text{ where } |D^L| = |V^L|$$

$$v^L_i \sim v^L_j \Leftrightarrow j \in N^L(i) \land w^L_{ij} \leq \text{thr}^L$$

(2.1)

where the set $D^L$ contains the nodes’ indices of layer $L$.

**2.1.1 Graph initialization: The zero layer**

Given an input matrix $P = (P_{yx}) \in \mathcal{M}_{M_{\text{height}} \times M_{\text{width}}} (\mathbb{R}^F)$ where $x \in [0, M_{\text{width}})$, $y \in [0, M_{\text{height}})$ and $F$ is the number of features. The zero layer considers the matrix $P$ as an undirected weighted graph $G^0 = (V^0, E^0, W^0)$, in which every element $(P_{yx})$ of the
matrix is a node \( v_i^0 \) in \( V^0 \). Elements of \( V^0 \) are ordered by a bijective function that sorts the elements by columns from left to right, as given by the Equation 2.2:

\[
V^0 = \{ v_i^0 : i \in D^0 \} \quad \text{where} \quad D^0 = \{ i : i = x \cdot M_{\text{height}} + y \} \\
|D^0| = M_{\text{height}} \times M_{\text{width}}
\]  

(2.2)

Nodes properties are initialized to the values of the matrix \( P \) as

\[
\forall v_i^0 \in V^0, \\
v_i^0(\text{mean})[f] = v_i^0(\text{max})[f] = v_i^0(\text{min})[f] = P_{yx}[f] \\
v_i^0(\text{nelements}) = 1
\]

Notice that at this step the fields \text{mean}, \text{max} and \text{min} of each node are initialized to the same values.

We consider the neighborhood \( N^0(i) \) of the node \( i \) to be formed by its 8 first neighbors in the grid (Equation 2.3).

\[
\forall i \in D^0, N^0(i) = \{ v_j \in V^0 : |i - j| \in \{1, M_{\text{height}}, M_{\text{height}} + 1, M_{\text{height}} - 1\} \}
\]  

(2.3)

Computation is more efficient if we consider a directed weighted graph \( G^0 = (V^0, E^0, W^0) \) defining a node’s neighborhood as follows,

\[
\forall i \in D^0, N^0(i) = \{ v_j \in V^0 : i - j \in \{1, M_{\text{height}}, M_{\text{height}} + 1, M_{\text{height}} - 1\} \}
\]  

(2.4)

In this case, the edge direction should not be understood as an edge direction in graph theory as a computational dependence, which means that node \( v_i \) can be computed after all its neighbors \( N^L(i) \) pointing at \( v_i \) have been computed. Therefore, the graph is undirected and basic concepts from graph theory as path and connected components are still applicable, and edge directionality is taken into account just for fast node feature calculation.

Figure 2.2 sketches how the graph in the zero layer is created from a given matrix the decisions made to enhance the computational time performance.

### 2.1.2 Nodes Aggregation: From one layer to the next

Given a graph layer \( G^L \) we move from this layer to the next one \( G^{L+1} \) using the equivalence relation between nodes in \( V^L \) introduced in Equation 2.1. Once this equivalence relation has been established, we calculate the graph’s connected components and label each component in order to identify each component as a node in the new layer.

The next definitions, taken from the graph theory, are necessary to provide a full framework of the layer description:
2.1. Multi-layer Graph-Based Segmentation Algorithm

Figure 2.2 – Schematic illustration of the transformations implemented by the segmentation algorithm to map a given matrix (e.g. an image) into a connected graph. (Left) Each matrix element (image pixel) is a nodes graph (circles) connected to its 8 closest neighbors in the grid. (Right) Fast computation of each nodes features is carried out by establishing an order among nodes and a link directionality. Nodes features can be computed only after all nodes features pointing at it have been previously calculated. Features are needed to decide the connectivity and the components of the next layer.

Definition 2.1.7 (Path). A path in the graph \( G^L \) is a sequence of edges that connect two nodes in the graph. Each edge in the sequence connects two equivalents nodes \( i \) and \( j \), i.e., \( j \in N^L(i) \land w_{ij}^L \leq \text{thr}^L \) as introduced in Eq. 2.1:

\[
e_{ij} \sim e_{jk} \sim e_{kl} \sim \ldots
\]

Definition 2.1.8 (Connected Component). Let \( G_1 = \{ V_1, E_1, W_1 \} \) and \( G_2 = \{ V_2, E_2, W_2 \} \) be subgraphs of \( G = \{ V, E, W \} \) such that \( V_1 \cap V_2 = \phi \) and \( V_1 \cup V_2 = V \).

\( G_1 \) is a connected component of \( G \) if

- all nodes in \( V_1 \) are interconnected by a path of edges in \( E_1 \)
- there are no edges in \( G \) connecting one node in \( V_1 \) with any other node in \( V_2 \)

Definition 2.1.9 (Connected Component Labeling). Let \( G^L_k \) be a connected component of \( G^L / \sim \). The label \( k \) is defined as:

\[
k = \min_i \{ v_i^L \in V_k^L \}
\]

The connected component labels defined in this way are used as identifiers of the next layer’s nodes

\[
D^L = \left\{ k : k = \min_i \{ v_i^L \in V_k^L \} \right\}
\]
2.1.3 Communities: the next layer

Once all the connected components of \( G^L/\sim \), \( G^L_k = \{ V^L_k, E^L_k, W^L_k \} \) are determined in layer \( L \), each node in layer \( L + 1 \) will be identified with one component in layer \( L \), such that the number of nodes in layer \( L + 1 \) will be equal to the number of connected components detected in layer \( L \), and such that each node represents all nodes in that component.

We define the graph \( G^{L+1} = \{ V^{L+1}, E^{L+1}, W^{L+1} \} \) where

\[
\forall v^{L+1}_k \in V^{L+1},
\]

\[
v^{L+1}_k (\text{nelements}) = \sum_{v^L_i \in V^L_k} v^L_i (\text{nelements})
\]

\[
v^{L+1}_k (\text{mean}) [f] = \frac{\sum_{v^L_i \in V^L_k} v^L_i (\text{mean}) [f] \cdot v^L_i (\text{nelements})}{\sum_{v^L_i \in V^L_k} v^L_i (\text{nelements})}
\]

\[
v^{L+1}_k (\text{max}) [f] = \max_{v^L_i \in V^L_k} (v^L_i (\text{mean}) [f])
\]

\[
v^{L+1}_k (\text{min}) [f] = \min_{v^L_i \in V^L_k} (v^L_i (\text{mean}) [f])
\]

Graph’s edges, \( E^{L+1} \), connect those component whose nodes in the graph \( G^L \) were neighbors but they were not in \( G^L/\sim \), i.e., they did not fulfill the equivalence relationship of similarity.

Finally, the neighborhood \( N^{L+1}(i) \) of node \( i \) in layer \( L + 1 \) is defined as

\[
N^{L+1}(i) = \left\{ v^{L+1}_j : \exists v^L_i \in V^L_i \land \exists v^L_j \in V^L_j \land i \neq j \land e^L_{ij} \in E^L \land e^L_{ij} = 1 \right\}
\]

2.1.4 Stop condition: closing the loop

The algorithm iterates the nodes aggregation (Section 2.1.2) and community creation steps (Section 2.1.3) until a given condition is matched, usually called stop condition. In our particular case, the algorithm is stopped when a given number of layers is reached.

Moreover, at each iteration, the threshold condition of the similarity function \( \sigma \) is modified by

\[
\text{thr}^{L+1} = \text{thr}^L + 0.1
\]

The initial threshold \( \text{thr}^0 \) and the number of layers (depth level) constitute the algorithm parameters and whose impact on the algorithm’s functioning will be discussed in the next Section 2.1.5.
2.1. Multi-layer Graph-Based Segmentation Algorithm

Figure 2.3 – Image segmentation dependence on the chosen threshold. The segmentation of an original RGB image is performed using different values of the threshold needed by the algorithm. Each segmented cluster is represented by its mean color (in each R, G and B components) and its composed of neighboring pixels with similar features (similar RGB color). As the threshold increases, all pixels are grouped into just one cluster.

2.1.5 Parameter analysis

The presented multi-layer graph-based segmentation algorithm relies on two parameters only: the initial threshold fixing the edge weight below which two nodes are connected and the number of layers or depth level that changes the initial threshold by using an incremental function, as seen in Section 2.1.4.

The threshold parameter allows the algorithm to group neighboring pixels with a certain degree of freedom. For example, if low thresholds values are considered, we are consenting to merge those most similar neighbor groups which may result into small groups whose standard deviation between the features is low. On the other hand, considering high threshold values, we will likely be grouping pixels that are neighbors but whose features may be different resulting into large groups whose standard deviation among features high. Figure 2.3 depicts the outcome difference between choosing a low threshold level and a high one. It can be noted that an extremely low threshold level results into a not segmented image (in the sense that each pixel in the original image corresponds to one graph’s node) while an excessively high threshold level gives rise to a single node image, i.e. all pixels are grouped into one cluster.

Regarding the second parameter of the algorithm, the depth level allows the algorithm to merge those groups of pixels which are quite similar but not enough to be merged in the previous level. The incremental function introduced in Section 2.1.4 can be tuned in order to control how fast groups merge to produce the next layer, but a deeper insight of the effects of this function is beyond the scope of this thesis.

The merging of nodes in one layer depends on the nodes in the previous level, therefore, executions of the algorithm starting with different sets of parameters but ending with the same threshold value produce different outcomes. For example, if we choose a
Figure 2.4 – Image segmentation behavior as a function of the threshold and of the depth. An RGB image (top) is segmented using different values of the initial threshold (vertical axis) and of the depth (horizontal axis). Each cluster is represented by its mean color (mean of R, G, and B). Moving from left to right keeping fixed the value of the initial threshold, it can be seen that nodes from previous levels merge into one node in higher levels. Furthermore, if comparing the algorithm executions ending with the same threshold level, e.g., starting with $\text{thr} = 2$ and $\text{dpth} = 20$, the final threshold value is $\text{thr} = 4$, the same as starting with $\text{thr} = 4$ and $\text{dpth} = 0$, the final outcome is different because the speed at which the pixels are grouped is not the same. When $\text{thr} = 1$ and $\text{dpth} = 20$, only the most similar neighbor nodes are grouped at the beginning so that the difference with the neighbors at the next level is higher and therefore more difficult to group. In the case of $\text{thr} = 4$ the algorithm allows to create larger clusters with less similar nodes.
threshold 2 and a depth 20, the algorithm ends with a threshold 4 \((2 + 0.1 \cdot 20 = 4)\). The algorithm execution with these two set of parameters \((\text{thr} = 2, \text{dpth} = 20)\) and \((\text{thr} = 4, \text{dpth} = 0)\) yields different outcomes. The reason behind this is that in the zero layer (i.e. when nodes are the image pixels) there is a continuity between the colors of neighboring pixels, i.e. the distance between their features is small. By contrast, in the next layer, when nodes are merged and neighboring clusters are compared between them, the distances become larger, and therefore, more difficult to bond in the next step.

In conclusion, the choice of the threshold and depth parameters depends on the image type, the feature space (in our case RGB space) and the goal to be achieved, so that it should be carefully studied in each case.

### 2.1.6 Computational cost analysis

The computational cost of the multi-layer graph-based algorithm is evaluated in terms of the performing time invested in each layer and in the whole process. The computational cost in each graph-layer grows linearly with the number of nodes as it is shown in the code, Appendix A. This is also supported by the linear regression plotted in Figure 2.5(i) where the algorithm was executed with images, different sizes and the depth parameter was fixed to 1. In Figure 2.5(ii) we study how the depth level affects the total amount of time. Depending on the original image, the number of segmented nodes from one layer to the next one can decrease faster or slower, yielding lower or larger processing times, respectively. As shown in 2.5(ii), the same image but with five different backgrounds of growing heterogeneity (not shown), needs larger times of computation to achieve the stop condition.

### 2.2 Application to invertebrate neuronal cultures

Along the past decades, cultured neuronal networks (CNNs) have constituted a fundamental tool for scientists, as one of the benchmark models for the study of the central nervous system. They, indeed, allow performing very well controlled laboratory experiments, thus providing a systematic way to approach fundamental questions, as e.g. unveiling the principles and mechanisms underlying memory, connectivity and even information processing of their in-vivo counterparts.

CNNs have been studied with two main tools, electrodes and images. Usually, electrode techniques give information about how a network works by detecting the information flows and creating functional networks. Functional networks show the relationship between areas into a culture but tell nothing about how the cells in the culture are interconnected. At the other hand, culture images give information about how the cells are interconnected allowing us to reconstruct the whole network but saying nothing about how the information flows.
Chapter 2. Image analysis

Figure 2.5 – Computational cost of the graph-based segmentation algorithm. (i) Segmentation times in the zero-layer (black dots) as a function of the image size (in pixels). The linear regression shows that the computational cost scales linearly with the image size and, by extension, the scaling also applies in each level, i.e. $O(n)$ where $n$ is the number of nodes in each level. (ii) Segmentation times as a function of the depth level (stop condition). Each curve corresponds to the segmentation of the same image of size 310x518 pixels but different backgrounds of growing color heterogeneity from Pic 1 to Pic 5. For a given depth level, the number of segmented nodes and, therefore, the computation time, increases with the non-uniformity of the image. Performance times correspond to a PC computer Intel Core i3 @ 3.30 GHz and 4GB RAM.

2.2.1 Neuronal cultures and imaging

Primary neuronal cultures were prepared from the frontal ganglion of adult locusts *Schistocerca Gregaria*. Frontal ganglia dissection and dissociation procedures are described in Section 1.1.4. In brief, neuronal cells are isolated (and removed from their original neurites) by enzymatic and mechanical dissociation. Cells are then re-suspended in Leibovitz medium (L-15) with L-glutamine (Sigma-Aldrich, L4386), supplemented with 0.01% penicillin-streptomycin (Biological Industries, Israel), and seeded on a Concanavalin A (Sigma-Aldrich, C0412) pre-coated circular area ($r \sim 2.5$ mm) of a 35 mm Petri dish at a density of about 50 to 150 cells per mm$^2$, and left eventually for 2 hours to allow adherence. Neurons are then incubated with 2 ml conditioned medium L-15 enriched with 5% locust hemolymph, and cultured in darkness for 18 days *in-vitro* (DIV) under controlled temperature (29°C) and humidity (70%). No medium changes are done to no affect the network topology.

High-resolution and large scale phase-contrast images are acquired daily with an inverted microscope (Eclipse Ti-S, Nikon Instruments) equipped with a motorized XYZ stage (H117 ProScan, Prior Scientific), and using a charge coupled device camera (DS-Fi1, Nikon Instruments) with a 10x air (Achromat, ADL, NA 0.25) objective. The automated control of the motorized XYZ stage and camera is performed using the NIS-Elements software (Nikon Instruments Software, Nikon). Mosaic images with a pixel size of 1.34 µm are captured with the large image method implemented in NIS-Elements,
2.2. Application to invertebrate neuronal cultures

Figure 2.6 – (Left) Primary culture of dissociated neurons from 12 frontal ganglia of *Schistocerca Gregaria* after 7 days in-vitro (DIV). The red marker is used to easily locate the neurons in the Petri dish. (Right) Enlarged area of the outlined region on the left.

which does automatic blended stitching with an overlap of 25%. Therefore, at each day of measurement, the result is a large high-resolution jp2 image file consisting of an ensemble of images acquired with 10x magnification in mosaic. An example of a typical acquisition is shown in Fig.2.6 for a culture after 7 DIV, where neurons, ranging from 10 to 50 µm in size, grow neuronal processes (neurites) trying to target neighboring cells.

2.2.2 Culture imaging artifacts

The aim of this section is to review and discuss the issues we can find during image acquisition that impair image processing performance. These issues can be split up into three different categories depending on the sources:

1. from the researcher decisions, e.g. cell density, using marking signs to locate the culture, microscope magnification configuration, and so on.

2. from the microscope calibration, e.g. light conditions, and XYZ stage calibration, among others, and

3. from the culture itself, e.g. debris caused by enzymatic and mechanical dissociation, dead cells, air bubbles, and so forth.

We can regroup these problems into two classes attending whether they are culture or not dependent. Problems that are not culture related usually affect the image background (see Fig.2.7) while those problems coming directly from the culturing process (density, garbage, bubbles, . . . ) usually impact the extraction of the underlying network structure (as sketched in Fig.2.8).
(a) Red mark used to highlight the area where the cultured cells are located in the Petri dish. (i) RGB image cut with the red mark. (ii) Trail left by the red marker after filtering the red layer of the RGB image.

(b) Wrong illumination setups produce intensity gradients like the one in the top-left corner. (i) Red layer of a background image. (ii) Smoothed image to highlight the background gradient. (iii) Histogram equalization of the smoothed image to expose the intensity gradient.

(c) Each image acquisition is an overlay of smaller frames each one of them affected by an intensity gradient. The resulting image displays a tessera effect. (i) Red layer of a background image with a tessera effect. (ii) Smoothed image of the background image. (iii) Histogram equalization of the smoothed image to highlight the tessellation.

(d) A motorized XYZ platform is used to automatically capturing a large image from a set of smaller images. A bad alignment of the Petri dish with the robotic platform or a small error calibration of the platform gives rise that some areas be out of focus. (Top-left) focused frame, (Bottom-right) blurred frame. Culture 7 DIV age. Picture code: 15102012-17102012-C2-07.

Figure 2.7 – Culture independent imaging problems affecting the quality of the image background.
Regarding the first class of image problems, culture unrelated, and depicted in Fig. 2.7, the use of a red marker to highlight the boundary area of the culture in order to easily locate it in the Petri dish, it is an undesirable feature we need to remove before segmenting the image. We manage this problem by working with the red layer of the RGB image and disregarding the other two layers. With this action, we filter the red information and, therefore, the red mark is removed from the image although some other problems may appear associated with the trail left by the marker pen (Figure 2.7a).

As explained before, images to be analyzed are mosaics of small images which are arranged together to produce a large picture of the whole culture. The image acquisition process can be set to manual, semi-automatic or automatic. Once the top-bottom and left-right limits of the region of interest are defined, the XYZ microscope stage moves through the pre-defined area. In the manual setting, the user can focus the frame prior the camera shoots, while in the semi-automatic setting, the focus can be set every a given number of frames. And finally, in the automatic setting, the user focused at the beginning of the acquisition and all the rest of frames are captured with the same focus. The automatic process takes about 5 to 10 minutes while the manual configuration takes about 30 to 45 minutes depending on the size of the region of interest. Therefore, this type of focus decisions may result in some frames are better focused than others. By way of illustration, in the manual acquisition setting, the user can change the focus at every picture and, at consequence two consecutive frames may have different focus. In the automatic setting, as the objective’s vertical position is fixed at the beginning, if the XYZ stage is not well leveled or the focus is not well determined then some region in the picture may be better focused than others (Figure 2.7d).

During the acquisition, wrong light conditions may produce illumination gradients at the level of the whole picture (like in 2.7b) or affect each frame giving rise to a tessera-like image (Figure 2.7c).

The process of culturing also adds an undetermined number of objects that could interfere with the network structure extraction. For instance, during the enzymatic and mechanical dissociation of the frontal ganglia to break the neurons apart and remove them from all the dendrites and axons, a lot of debris is produced whose size can be as large as a ganglion sheath (Fig. 2.8a(i)) or smaller than the neurons soma (Fig. 2.8a(ii)). The cell culture medium used to keep neurons alive may also provoke light fluctuations caused by liquid bubbles or particles in suspension as shown in Fig. 2.8a(iii). Finally, a high content of glia cells makes the network structure extraction almost impossible without using dyes (Figure 2.8a(iv)).

Sometimes it may happen that the culture grows outside the expected area because the cell suspension was mistakenly dropped (Figure 2.8b). In those cases, we can still recover the network structure grown on top of the red mark by taking the red layer from the RGB image as previously explained.

Culture network development involves several phases as we will see in Chapter 3. After cell dissociation and few hours in culture, cells start to throw neurites trying to connect to other cells. Usually these neurites are not point to point lines but more like bifurcation trees spatially extending to reach neighboring neurons. This type of...
Chapter 2. Image analysis

(a) Debris and objects typically present in a neuronal culture: (i) Sheath of a frontal ganglion, (ii) liquid bubbles and suspended particles (dark tones in the equalized histogram image), (iii) debris from the dissociation process, and (iv) glia cells. Pictures taken from a culture 21 DIV age. Picture code: 07092012-19092012-C1-21.

(b) Development of the culture outside the red marked circle. Culture at DIV 9. Picture code: 07092012-17092012-C2-09.

(c) High density of projections, thin neurites, crossing neurites, etc., make difficult the recovering of the network structure. (Culture at DIV 12. Picture code: 07092012-21092012-C2-12)

(d) Cell migration produces cell clustering which is a 3D structure focusing at different heights. This turns impossible knowing the number of cells belonging to one cluster or to identify which neuron is making synapsis. Culture at DIV 14. Picture code: 15102012-17102012-C2-14.

Figure 2.8 – Image issues due to the process of culturing that difficult the extraction of the neuronal network structure.
growth may give rise to high neurite density regions hindering the network structure extraction. In particular, if two projections are intersecting each other we can not distinguish whether there is a synapse between them or they are forming independent paths. Moreover, as said before, we choose a 10x working objective, whose magnification allows us to study the network with a good resolution but it makes difficult to identify those neurites that are quite tiny or distinguish between dendrite and axon connections (Figure 2.8c).

Immediately after a neuron establishes a connection with another one [137, 138, 139], neurons pull each other and start to migrate giving rise to a cluster of neurons where synapses are mediated through gap junctions. We will study this phenomenon and others acting in the network structure development in Chapter 3. Single neuron detection inside the cluster is an unattainable work, even by eye inspection, and, in addition, when several neurons conjoin, some of them are pushed up producing a 3D structure (Figure 2.8d) making impossible to infer the total number of cells composing the cluster just from the cluster’s 2D volume.

Finally, another important factor to take into account is regarding the acquisition of large regions of interest producing huge image sizes ranging from 50 to 300 MB (*.jp2 files) which makes the image analysis computationally demanding and hard to manage.

2.2.3 Image processing strategy

This section is focused on justifying why graph-based segmentation algorithms, like the one proposed in this thesis, are more powerful compared to other algorithms like clustering, edge detection, active contours, etc. or other object detection algorithms (See Section 1.2 for references).

When analyzing a representative Schistocerca Gregaria culture image and extracting the background and foreground gray scale histograms, we can discern two Gaussian distribution overlapping each other with the background histogram hiding the foreground color pixels, as shown in Figure 2.9 by the green shaded area.

Clustering algorithms, like k-means [140], do not take into account the spatial information of each pixel and classify those pixels only by their color information. As consequence they split the histogram into sections demarcated by thresholds. This kind of algorithms are quite sensitive to the selected thresholds and even a good choice of them involves the loss of many non-background pixels, as it can be seen in Figure 2.9 if those pixels into the overlapping green area are removed.

In the case of edge-based algorithms, they would yield a large amount of noise due to image issues like culture medium bubbles or particles in suspension, debris, and light gradients (Section 2.2.2).

Active contours or snakes algorithms [141, 142] involve the active cooperation of the user to create the seeds needed by those algorithms, although they can also be initialized using the edge-based algorithm’s output. However, they are very computationally demanding making them very unsuitable to use within large images.

Object search-based algorithms, like Haar Cascades, have the drawback that they need an initial training in order to function. In our particular case, we would need to
provide for the learning process, a set of images with neurons and neurites for later recognition. The major issue in this task is that due to the great variety of objects (clusters with different number of neurons, shapes and orientations) it is difficult to provide enough representative images to recover the network structure.

Graph-based algorithms, on their turn, have the advantage over other clustering algorithms that they take into account the color (or feature) information and the spatial location of each pixel, allowing to merge two pixels if they have similar features and they are neighbors. Usually, these algorithms are highly demanding processes of memory (RAM) and time. By way of illustration, those graph algorithms minimizing the number of removed edges (cut methods) are NP-Hard problems [90], i.e. there exists no deterministic polynomial-time solution to compute those problems. Furthermore, those using eigenvector decomposition to determine graph partitions are also computationally expensive (\(O(n^3)\) where \(n\) is the number of nodes). In other words, in our case that we have around 100 million nodes, eigenvector decomposition is unaffordable.

The proposed algorithm is a graph-based algorithm where the connected components are calculated. These connected components are considered to be the nodes of a new graph (layer). Full description of the main algorithm has already been given in Section 2.1 while changes made to the algorithm to segment images from invertebrate neuronal cultures are introduced in Section 2.2.4.

![Figure 2.9 - Typical grayscale intensity histograms of neuronal culture images. Background (blue curve) and foreground (red curve) histograms are plotted separately for comparison. Green shaded area represents the grayscale level range where both histograms share the same colors. (Inset) Same histograms in log-linear scale to amplify the foreground.](image-url)
2.2. Application to invertebrate neuronal cultures

2.2.4 Image segmentation pipeline

The proposed image-processing algorithm for the network structure extraction of *Schistocerca Gregaria* neuronal culture consist of three main tasks. In the first task, it takes the red layer from a phase-contrast RGB-image to segment, separating the background pixels from those which are not, i.e. the foreground. The second task applies morphological operations to the foreground pixels to differentiate between pixels belonging to neurons and pixels from neurites. After that, a skeletonization operation to the neurites pixels is carried out to extract the neurite bifurcation points and, finally, in the third task the adjacency matrix representing the network structure is created. Figure 2.10 depicts the algorithm’s pipeline for the network structure extraction.

**Graph-based image segmentation and thresholding**

A first action to provide an accurate and fast segmentation of a CNN (culture neural network) from phase-contrast illumination images is to aggregate areas with similar features according to a specific function along different scales. The first issue the image segmentation process has to deal with is to merge into a single cluster all the background pixels to easily remove them afterward. Therefore, the main problem is that background regions are not connected, i.e. background pixels are surrounded by foreground ones.
which form a barrier.

We have adapted the proposed algorithm in Section 2.1 in order to avoid the aforementioned problems. In particular, the neighborhood function defined in Equation 2.4, where we introduced a neighborhood formed by the 8 nearest pixels, is modified by adding a non-local neighborhood formed by 4 extra pixels as follows:

\[
\forall i \in D^0, N^0(i) = \{ v_j \in V^0 : i - j \in \{1, M_{\text{height}}, M_{\text{height}} + 1, M_{\text{height}} - 1, 100, 100 \cdot M_{\text{height}} \} \} \tag{2.5}
\]

Local neighborhood attempts to link those pixels belonging to the same region, while non-local neighborhood aims for linking those background regions separated by foreground barrier.

In order to deal with large size images, the graph-based algorithm decomposes the

Figure 2.11 – Foreground segmentation process. (i-top) Red layer from a phase-contrast large RGB image, (i-bottom) and corresponding intensity histogram. (ii-top) Clusterized image after applying the graph algorithm to the image on the left. Each cluster is represented by the mean color intensity of the pixels belonging to the cluster. (ii-bottom) Intensity histograms of the original image (green curve), and of the segmented image (blue curve) where most of the background pixels are merged into clusters with the same average intensity, \(\sim 128\). The red line represents the threshold used to binarize the returned image and to remove the background pixels. (iii) Foreground segmentation process outcome. Detected neurons and neurites are highlighted in yellow. Culture 8 DIV age. Picture code: 07092012-19092012-C2-08.
images in 3x3 smaller images and works separately on each image cut to further group all the outcomes into a single solution.

Each cluster identifier is replaced by the average color in the region obtaining a new image as the one shown in Figure 2.11(ii-top). If we compare the histograms of the original and of the clusterized image (green and blue lines in Figure 2.11(ii-bottom)), we observe that all the pixels belonging to the background are grouped in a delta function, while the rest of pixels are uniformly distributed. Subsequently, a straightforward segmentation is carried out with threshold-based segmentation (red line of Figure 2.11) obtaining two binary masks: one corresponding to those pixels belonging to background and the other with those pixels forming the neuronal network mask (Figure 2.11(iii)).

**Neuronal cluster detection**

Given the neural network mask, in order to segment both isolated and clustered neurons we first distinguish between ROIs (regions of interest) larger and smaller than $10^5$ pixels, to subsequently perform selective morphological operations on each class of ROI, being all of them standard techniques in image processing. On the large ROIs, we performed two consecutive erosions using rectangles as structural elements. After first eroding with a 10x30 rectangle, regions smaller than 500 pixels are eliminated, and a second erosion is performed using a rectangle of 30x10 pixels. Finally, a dilation is applied using a disk with radius 10 pixels to smoothen the contours. The erosion in the smallest ROIs is carried out by performing 8 successive linear erosions at 45° from each other. After that, regions smaller than 100 pixels are eliminated and contours are eventually smoothed by means of a dilation with a 5 pixels disk. The combination of the two resulting ROIs yields the final neuronal cluster mask.

**Neurite detection**

After segmenting the neuronal clusters, the corresponding mask is subtracted from the neuronal network mask to proceed with the neurite detection. To that end, we apply a further morphological operation consisting in linearly dilating at 45°, -45°, 30° and -30° to connect those areas which were separated due to small errors in the initial background segmentation, especially relevant in the case of thinner neurites. The segmentation is then finalized by performing a dilation with a 5 pixels circular mask, and those holes of size smaller than 500 pixels are filled in afterwards. The neurite mask is then generated from the skeletonized image, after which the neuronal cluster and neurite masks are combined to produce the CNN mask as shown in Fig. 2.10(iii), were branching points and ends of neurites are marked as yellow pixels, neurites as blue lines, and neuronal clusters as green areas.

**Adjacency matrix reconstruction and analysis**

The obtained CNN mask is used in a twofold way: to extract morphological parameters characterizing both neuronal clusters (number, size, centroid, roundness, etc.) and neu-
Figure 2.12 – Extracted adjacency matrices. (i) Unweighted bipartite graph. Circles and diamonds stand for cluster and branching nodes respectively and blue lines represent a physical connection between two nodes. (ii) Unweighted cluster graph formed by nodes representing clusters and links representing the existence or not of a path between cluster nodes through bifurcation nodes. (iii) Weighted bipartite graph. The link’s thickness stands for the link’s weight. This weight is computed as the neurite length connecting two nodes. (iv) Weighted cluster graph. Links are weighted by the minimum-neurite length between nodes (clusters).

rites (length, orientation, etc.), and to extract the actual adjacency matrix encoding the topology of the neuronal network, as shown in Fig. 2.10(iv). The adjacency matrix is constructed as a binary and undirected graph, whose nodes are either branching points or cluster centroids, and two nodes are linked if there is a neurite process connecting them. Treating all links as identical, i.e. ignoring edge length and edge directionality, such a graph can be described in terms of a symmetric matrix \( A \) whose elements \( a_{ij} \) are equal to 1 if nodes \( i \) and \( j \) are linked, and 0 otherwise.

In Section 2.2.5 we describe in detail the adjacency matrix representing the culture structure and the measures taken to analyze it.

2.2.5 Network structure representation and measures

The graph representing the culture structure is bipartite since it is made of two types of nodes. However, in this thesis, we focus on the network statistical properties at the level of the neuronal clusters, ignoring the dynamics of both neurite connections and branching points. Therefore, we ended up with a subgraph defining the connectivity among the neuronal clusters in such a way that two of them are linked either directly or through a connected path of branching points. Such an object allows us to calculate all classical parameters characterizing the topology of a complex network (degree distribution, shortest path, node clustering, etc. See Section 1.3) and the network morphology (cluster areas and cluster centroids, culture area, neurite lengths, and so forth). We analyze the evolution along days in vitro of these measures in Chapter 3.

Other matrix representations are also provided by the algorithm and they can be analyzed in a similar way (Figure 2.12). Among others, the algorithm provides the
full structure representation, i.e. the bipartite graph where nodes represent cluster and bifurcation, and links representing connections between nodes.

The algorithm provides network structure representation by unweighted and weighted adjacency matrices.

Unweighted matrices were mentioned above and enumerated here:

- **Bipartite graph** (Figure 2.12(i)) where nodes represent clusters and branching points, and links represent connections between nodes.

- **Cluster graph** (Figure 2.12(ii)) where nodes represent clusters and links represent the existence or not of a path between cluster nodes through bifurcation nodes.

Weighted matrices are the same kind as unweighted matrices but here the matrices’ links are weighted by the neurite length connecting two nodes (cluster/branching point) in the case of the bipartite graph (Figure 2.12(iii)) and the minimum-neurite length between nodes (clusters) for the cluster graph (Figure 2.12(iv)).

The analysis of the bipartite graphs and weighted matrices is beyond the scope of this thesis.

### 2.2.6 Algorithm’s accuracy evaluation

The evaluation of the algorithm’s performance to extract the neuronal network structure is a matter of contention. To the best of our knowledge, there is no automated software dealing with segmentation of large size phase-contrast images of CNNs to compare with. Therefore, to quantitatively assess the capability of our proposed algorithm to reconstruct the CNN, we design a manual and an automated evaluation procedure. In the former case, we compare the topological similarity between the manual and algorithm’s graph solutions while in the latter, segmentation results will be compared to ground-truth synthetic images whose network structure is known. To test the robustness of our segmentation algorithm, different levels of blurring were added to the synthetic images by applying a Gaussian filter of the same size of the images (blur kernel from MATLAB) and using the blur metric introduced in Ref. [143].

#### Manual evaluation

Manual evaluation is carried out, by eye inspection, labeling clusters of neurons and neurite’s bifurcation points and connecting those labels to create the graph representing the network structure.

Gainc (Graph Algorithms In Matlab Code\textsuperscript{1}) is used to create graph representations by clicking and dragging. This tool was modified to show a background image which is used as a template by the user to mark on it the corresponding region of interest as nodes (the centroid of clusters of neurons and neurite’s bifurcation points) and connect them with links if it exists a neurite connecting those regions.

\textsuperscript{1}gainc website: http://www.mathworks.es/matlabcentral/fileexchange/24134-gaimc---graph-algorithms-in-matlab-code
Due to the tool limitations, the image used as a background for gaimc tool cannot be zoomed in or out due to tool limitations. Hence, the image size must be fixed before starting the labeling process.

**Manual annotation limitations**

The selection of a fixed image size entails the following problems. First, if the image size (in pixels) is bigger than the screen resolution then the image is automatically scaled to fit into the screen. As a consequence, image details like thin neurites could be lost if the scaling is significant. In such a situation, the user could not distinguish thin neurites that the proposed algorithm could. Second, if the image size (in pixels) is lower than the screen resolution, the number of neurons in the accessible image area is quite small and, therefore the connectivity is extracted in a very local manner not allowing the user to mark those neurites leaving that image area.

Manual labeling also entails subjective judgments [144]. By way of illustration, Figure 2.13 depicts some of the possible cases in which the user can make subjective decisions. For example, if a neurite is blurred to get lost and then reappears, the user could label it as a link or think that maybe is not the same neurite (Fig.2.13.i); or if two neurons are quite close, the user could label them as a cluster of neurons or as separated neurons linked by a neurite (Fig.2.13.ii); or it may happen the user must approximately mark the cluster centroid. This could be a problem for those clusters with C-like form where the cluster centroid has to be placed outside the cluster and the user decides to place it inside (Fig.2.13.iii).

**Manual evaluation decisions**

In order to overcome those manual annotation limitations and to compare the connectivity matrices resulting from the manual annotation and from applying our segmentation algorithm to the same area, we consider images of size 1000x1000 pixels. To proceed with the comparison of the two adjacency matrices, the following rules are taken into account:

1. a node in the manual solution graph is identified as the same node in the algorithm’s solution if the Euclidean distance between their positions is below a given threshold and vice versa.

2. those nodes not having a reciprocal mapping are removed from both graphs, and

3. those nodes that are connected through a deleted node become connected through a direct link.

This reduction process avoids the subjective judgments issue since each node has to be detected by the two labeling methods and the evaluation will reflect the ability of the algorithm to find and existing path between two detected clusters.
Subjective decisions during manual image annotation. (i) Thin or blurred neurites can make the user to decide either labeling the link or not. (ii) Close neurons can be labeled as single neurons or as a cluster of neurons. (iii) C-like shape clusters have the centroid outside the cluster and the user may decide to mark it either inside it or not. Scale bar 200 µm. Culture 6 DIV old. Picture code: 07092012-21092012-C1-6.
Chapter 2. Image analysis

Figure 2.14 – Background images generated by taking a picture of a petri dish with *Schistocerca Gregaria* culture medium and used to create the synthetic database. The whole image is composed by 12x10 frames resulting into a image with dimension 9000x9000 pixels. (i) Background with tesseration effect. (ii) Background with tesseration effect and red circle. (iii) Background with illumination gradient and red circle.

Figure 2.15 – Objects, manually segmented, used to create the synthetic database. From (i) to (xi) neurons with different sizes and shapes. (xii) neurite.

**Automatic evaluation**

In the automated evaluation instead, we create a synthetic database of images emulating the observed CNNs of *Schistocerca Gregaria*. The image database is built using somas and neurites extracted from real images, and the background under the same optical conditions as in the experiments when only the culture medium is present. Regarding this element, three different backgrounds were used for the construction of the image database portraying the image problems discussed in Section 2.2.2, about tesseration, illumination gradients and red location marks.

Figure 2.15 shows the eleven different neurons sizes and shapes and one sample of neurites used to create an artificial neuronal network as it is explained below.
2.2. Application to invertebrate neuronal cultures

Figure 2.16 – Synthetic images database creation. (i) Neurons arbitrarily taken from the set shown in Fig. 2.15, are randomly distributed into an empty board of size 9000x9000 pixels. Here only a segment of 4000x4000 pixels is shown. (ii) Each connection between neurons is established by adding a succession of the same neurite cut whose thickness is randomly set. Connectivity is determined by randomly selecting links from the Delaunay triangulation of the cluster centroids. (iii) Background is added and borders between background and network are blurred.

Synthetic images database

Neurons are arbitrarily chosen from the sample set and randomly placed in a 9000x9000 pixels square grid, allowing for the formation of larger clusters if neurons intersect. In this later case, pixels are set to the maximum value of intensity of the overlapping pixels. Next, cluster centroids are computed and a Delaunay triangulation is performed to obtain a connectivity matrix where some of those connections are randomly removed. Every pair of linked cluster centroids are physically connected in the image by placing a neurite whose thickness is randomly scaled and whose length and orientation is given by the Euclidean distance and relative angle of the straight line between them. After rescaling, neurites are concatenated until the desired length is reached. Finally, the background is added into the image by blurring the borders between the artificial network and background. Figure 2.16 depicts the process to create a single synthetic neuronal network.

Automatic evaluation limitations and decisions

Synthetic images creation has the two following problems. The first one is related to those neurons with high connectivity and particularly small size which could end up being hidden by the neurites branches linking to them. The second one happens when two neurites touch each other before reaching the target neuron. We treat this contact point as a bifurcation point between neurites.
Evaluation outcome

As for the algorithm accuracy in reconstructing the CNN, i.e. understanding accuracy as the capability to properly detect a link, we compare the adjacency matrices by checking fails and hits (see Section 1.2.3) coming from manual annotation or from synthetic images whose structure is known with the adjacency matrix resulting from the algorithm’s segmentation.

To this purpose, we applied the algorithm to small size images (1000x1000 pixels) of a 6 DIV neuronal culture, and manually annotated each image as previously described. An example of such a comparison is shown in Figure 2.17 after an original image (left column) is being manually (middle column) and automatically (right column) annotated, where neurons (green circles), bifurcations (yellow diamonds) and neurites (blue straight lines) are labeled. The resulting F-score obtained for 32 images is 72.94 ± 2.99% (R = 65.14 ± 4.15% and P = 91.90 ± 2.52%) (± refer to the standard error of the mean) and the level of coincidence $E = 79.45 ± 2.18\%$, indicating that the segmentation algorithm is able to statistically detect, with high precision, most of the existing links in the manually annotated graph.

In addition to the problems associated with the user bias, discussed in the “Manual Evaluation” Section, the manual evaluation of the performance restricted to small areas of the culture, hampers the possibility of evaluating the ability of the algorithm to detect long-range connections and, therefore, of working with a real ground-truth topology to compare with. In order to overcome this issue, we devised an automated method of evaluation based on synthetic images of neuronal networks whose topology is known. Left panel of Fig. 2.18 shows a detail of one artificial network designed as explained before. The underlying network structure used to create the artificial network image is shown in the middle panel while the right one depicts the graph solution rendered by the algorithm. We used 16 images (9000x9000 pixels) to evaluate the performance with different network configurations and backgrounds, obtaining and accuracy of about 80.35±0.70% (± means standard error). In particular, the accuracy to recover the nodes (neurons) of the synthetic networks lowers to 67.42% due to small size neurons masked by a large number of neurites ending at them.

Automatic evaluation gives us the possibility of comparing the neuronal network mask obtained just after the image segmentation (Section 2.2.3) with the exact location of neurons and neurites used in the synthetic image creation. The synthetic image can be mapped into a 9000x9000 matrix, encoding the existence of a neuron or a neurite pixel in the image if the matrix element equals 1 and 0 otherwise. Taking this matrix as ground-truth, we evaluated the algorithm’s capacity for recovering the neuronal network mask (or, equivalently, for distinguishing the background from the foreground) yielding a 96.56 ± 0.30% of F-score.

Taking advantage of this framework, we measure the robustness of the algorithm’s segmentation by blurring the synthetic images with a Gaussian filter. The blur level was measured as in Ref. [143] and it is within the interval [0, 1] where 0 means sharp and 1 means blurred. Figure 2.19 shows the effect on the accuracy of the segmentation process as a function of the blur level. From the left to right, top panels show the synthetic
Figure 2.17 – Manual versus algorithm labeling. From left to right, culture image cuts (1000x1000 pixels) to be analyzed (left column), corresponding manual labellings (middle column), and algorithms outcome (right column). Green circles mark neurons or clusters of neurons, yellow diamonds stand for bifurcation points and blue lines symbolize neurites connecting both neurons and bifurcations. On the right, a diagram of the original image showing where the images cuts have been taken. Scale bar 1mm. Picture code: 07092012-21092012-C1-6.
Chapter 2. Image analysis

Figure 2.18 – Evaluation of the algorithms accuracy using synthetic images. Image segment (2000x3000 pixels) of an artificial neuronal network (left panel), network structure used to create the synthetic image (middle panel), and the algorithms solution when applied to the image on the left (right panel). Green circles mean neuron node. Yellow diamonds stands for bifurcation points and blue lines symbolize the neurites connecting nodes.

Figure 2.19 – Behavior of the image segmentation algorithm upon image blurriness. (Top row) Synthetic images with different blur levels: (i) 0.22, (ii) 0.33, (iii) 0.38, and (iv) 0.41. (Bottom row) Comparison between the image segmentation outcome (in yellow) and the ground truth mask (in red). The resulting F-scores about the accuracy in detecting non-background pixels are, from (i) to (iv), 98%, 94%, 88% and 73%, respectively.
2.2. Application to invertebrate neuronal cultures

Figure 2.20 – Robustness of the algorithms segmentation to blurriness. Algorithms accuracy processing synthetic images as a function of the blur level. Green dots are the F-scores of all the analyzed synthetic images for a given blur level. Purple solid (dashed) line represents the average (standard deviations) value. Blue area stands for the usual blurriness values observed in real culture images (0.23 ± 0.02).

blurred images for values of blurring ranging from 0.22 to 0.41, while the bottom panels show the differences between the ground-truth (in red) and the segmented mask (in yellow). The more yellow is the picture the better is the accuracy (98% in the leftmost panel and 73% for the rightmost one).

In a more systematic way, Figure 2.20 quantifies the F-score in the location of neurons and neurites’ pixels as a function of the blur level, showing a relatively slow decay up to blur levels of about 0.35, after which the accuracy no longer sustains. As a matter of comparison, the graph also shows the usual blurriness value of a real culture image (0.23 ± 0.02), which falls within the region where the algorithm exhibits a more stable behavior.

2.2.7 Computational cost

Figure 2.21(i) shows the overall processing time for each one of the images of the data set. The height of the bars accounts for the total time consumed in both the segmentation and analysis tasks (time consumed by specific tasks are color coded), and is computed assuming that the graph-based algorithm is implemented in C++ and the rest of the pipeline uses MATLAB running in a PC computer (Core i5, 6GB RAM, 3.00 GHz). As it can be observed from the inset of Fig. 2.21(i), background segmentation and adjacency matrix extraction are the most demanding tasks, consuming respectively 30% and 45% of the total computational time. To further quantify the computational efficiency of the algorithm, we plot in Fig. 2.21(ii) the image processing time as a function of the image size (in pixels). The algorithm is able to manage images as large as 125 Mpixels in less than 6 minutes (while the segmentation task only takes 1.6 min, see inset of Fig.2.21(ii)) without RAM saturation. For larger image sizes, the RAM becomes saturated and the
Figure 2.21 – Computational cost. (i) Image processing time for each one of the whole set of images ordered according to their sizes (from left to right). Each bar is proportionally divided in colored segments (see legend) according to the time each specific task consumes. (Inset) Task time percentage in relation to the whole process. (ii) Image processing time as a function of the image size (in pixels). Blue dots account for the processing time when the RAM is not saturated while red diamonds account for the processing time when hard drive is used to caching data (poor performance). Linear regressions show that the algorithm computational cost scales linearly with the image size, i.e. $O(n)$ where $n$ is the image size. (Inset) The same as is in the main plot but time corresponds only to the time consumed during segmentation.

performance is worsened, as the pipeline needs the hard drive to caching data. Moreover, a detailed examination of the code shows that the algorithm performs linearly with the number of pixels, for both the global execution and that limited to the graph-based segmentation algorithm, as shown in the inset. This is also supported by the linear regressions in both RAM and hard drive memory usage region.
In this chapter, we experimentally investigate the self-organization into a network of an in vitro culture of neurons during the course of development, and explore the changes of the main morphological and topological features characterizing the anatomical connectivity between neurons during the associated network’s growth.

The image segmentation algorithm introduced in Chapter 2 takes as an input a gray scale image (red layer from the RGB image) of the culture at a particular day and it extracts the information related to neuron cluster boundaries, neurite traces and connection points between neurites, as well as those between neurites and clusters.

This information is then used to map the neuronal network into a graph $G$ whose nodes are either cluster centroids or connection points, and the edges represent the links as straight lines connecting them. Therefore, our graph is made of two types of nodes: neurons or clusters of neurons ($v_i$) and neurite connection points ($u_i$). Treating all links as identical, i.e. ignoring edge length and edge directionality, this graph can be described
in terms of a symmetric adjacency matrix $A$ whose elements $a_{ij}$ are equal to 1 if nodes $i$ and $j$ are linked, and 0 otherwise.

In this thesis, we focus on the network statistical properties at the level of the $v_i$ nodes, ignoring the dynamics of both neurite connections and branching points. Hence, we extract a graph $G^v$ from $G$ by defining the connectivity among nodes of class $v_i$ in such a way that $v_i$ and $v_j$ are linked either directly or through a connected path of $u_i$ nodes.

In particular, we report the results on 12 cultured networks, which were grown from independent initial sets of dissociated neurons extracted from the frontal ganglion of adult locusts of the *Schistocerca gregaria* species. In all cases, the same protocol, as described in Chapter 1, was used. The density at which cultures are seeded determines the maturation rate and the spatial organization at the mature state [137, 138, 14]. For the purpose of this thesis, 5 cultures of high density (more than 500 neurons after seeding) and 7 of low density (less than 500 neurons after seeding) are considered and monitored during 20 days *in-vitro* (DIV). Neuronal suspensions are seeded in Petri dishes with a pre-coated circular area (diameter $d \sim 5$ mm). During the whole lifetime of an experiment, the culture medium was not changed.

In Section 3.1 we describe the culture development from the very initial up to the day when it is not longer possible to follow the culture evolution using image analysis due to cell detached from the substrate or to massive cell clustering. In Section 3.2 we show experimentally that *in vitro* neuronal networks tend to develop from a random network state toward a particular networking state, corresponding to a small-world configuration, in which several relevant graphs micro- and meso-scale properties emerge.

### 3.1 Culture developmental stages

From the sowing of the somas to the culture’s death, a neuronal culture grows with the aim at reaching a stage where neurons are interconnected among them. A typically observed growth evolution is shown in Fig. 3.1 through snapshots between 0 and 12 DIV. Isolated neurons, that have lost their original neurites, whose size ranges from 10 to 50 $\mu$m, are initially randomly scattered in a two dimensional substrate and end up organizing into a clustered network. To characterize and quantify the morphological changes that a culture experiences during the evolution we need to distinguish at this point between the terms “cluster” and “node”. We define a cluster as a group consisting of one or more neurons physically connected to one another through gap junctions while we will refer to a node in the network as a cluster centroid that is connected with other nodes through neurites establishing chemical synapses. While the former is the result of the image segmentation process and may contain errors due to debris in the culture, the latter ensures that such clusters have developed processes and they are connected to other nodes.

In the next subsections, we will identify two completely different and almost consecutive developmental stages in the growth process: a first spreading stage where all the connectivity forms and a second stage of optimization of the circuitry formed in the
3.1. Culture developmental stages

Figure 3.1 – Evolution of a neuronal culture into a clustered network. The left upper and lower frames represent the initial (DIV 0) and mature (DIV 12) configurations of the whole cultured area (size $7.7 \times 6 \text{ mm}^2$). Rectangles identify a specific area, whose intermediate evolution stages are reported in the other frames, ordered clockwise (see arrows). These latter snapshot correspond to DIV 0, 3, 5, 7, 10 and 12, respectively. Black (white) bars correspond to 500 (100) µm. (Cultures at DIV 0, 3, 5, 7, 10 and 12. Picture codes: 07092012-21092012-C2-(0,3,5,7,10,12))

previous stage. We will characterize these two phases by monitoring the evolution of morphometric parameters of single clusters in high and low density cultures (HD and LD respectively): number and average cluster area (Fig. 3.2), average area of the culture (Fig. 3.4), and number of connecting processes originating from a cluster and average cable length (Fig. 3.3).

3.1.1 Spreading stage

During the first days of culture and following the enzymatic and mechanical dissociation, neuronal bodies try to adapt to the new environment recovering their spherical shape and increasing their sectional areas as shown in Fig. 3.1 when comparing the enlarged panels from days 0 and 3. Quantitatively, this subtle increase at DIV 3, is about 17.8% and 25.8% for low and high density cultures respectively as shown in Figure 3.2 where we monitor the average area of the clusters (panel (i)) and the number of clusters (panel (ii)) as a function of age. We observe that from DIV 0 to 6, both, the average cluster area and the number of clusters stay approximately constant despite errors due to culture debris.

From DIV 3 to DIV 6 (top row of Fig. 3.1), cultures display a very intense phase of network development, in which neurons grow neurites in order to reach not only neighboring neurons or clusters but also neurites. During this growth process, neurites also split and reach other processes forming loops of neurites with no evidence for self-avoidance, giving rise to complex connective patterns with neurite-neurite and neurite-
Chapter 3. Results

Figure 3.2 – Morphological measurements quantifying the time evolution of the average cluster area (i) and number of clusters (ii) for high (more than 500 clusters at DIV 0) and low (less than 500 clusters) density. Error bars stand for the standard errors of the mean (s.e.m).

Figure 3.3 – Time evolution of the average connectivity (i) and of the mean euclidean distance between connected cluster centroids (ii) for high (red up triangles) and low (blue down triangles) densities. In both plots, error bars stand for the standard errors of the mean (s.e.m).
3.1 Culture developmental stages

DIV 0  DIV 2  DIV 5
1mm 1mm 1mm

DIV 8  DIV 12

Figure 3.4 – Time evolution of the culture area represented as the polygons covering all the clusters within the culture. The culture area is computed by adding triangles (blue areas) resulting from a Delaunay triangulation (black lines) of the cluster centroids (yellow points). Last panel shows the time evolution of the average culture area for high (red up triangles) and low (blue down triangles) density cultures. (Cultures at DIV 0, 2, 5, 8 and 12. Picture codes: 07092012-19092012-C1-(0,2,5,8,12))

neuron synapses. Figure 3.3 characterizes this maximum stage of network development by the exponential growth of the average cluster connectivity (i) and the euclidean distances between cluster centroids (ii) We observe from this figure that the maximum connectivity is reached earlier in the more dense cultures than in the less dense ones and that the most distant neurons are reached by day 6 in HD cultures while LD cultures need more time to reach the farther ones (they are more distant).

3.1.2 Migration and aggregation stages

After the initial stage of neurite formation, the growth rate decreases and a different mechanism starts shaping the network: tension is generated along the neurites as they stretch between neurons or bifurcation points to form straight segments [137, 138, ?, 14, 15] and the resulting network is characterized by a random distribution of few clusters of aggregated neurons linked by thick nerve-like bundles. When this force (tension) is stronger than the attachment force (adhesion) to the substrate, the cluster migration is favored giving rise to the formation of clusters of neurons.
This response takes place throughout the culture’s life since the first connection is formed. However it becomes more important when the neurite outgrowth stops and the culture reaches a stage where most of the neurons are connected as it evidenced from the panels at 7, 10 and 12 DIV of the bottom row in Figure 3.1.

This aggregation process is clearly outlined in Fig. 3.2 by the exponential increase of the average cluster area (left panel) and the corresponding decrease in the number of detected clusters (right panel). Due to tension forces, some parallel or close neurites merged into a single and thicker connection together with the retraction of those branches which did not target any neuron, slightly lowering the average connectivity as shown in Fig. 3.3(i). We consider that two neurites are parallel when the links ends are the same, i.e. two neurons connected by two neurites.

The neuronal migration accompanying this aggregation process can be further monitored by looking at how the average distance between connected clusters (Figure 3.3(ii)) starts decreasing slightly after 6 DIV in case of HD cultures. For LD cultures, the average distance decreases sharply after 10 DIV.

Finally, despite the aggregation of clusters, the whole area delimiting the culture remains almost constant. The culture area is measured by adding the triangles resulting from a Delaunay triangulation of the cluster centroids, as depicted in the panels of Fig. 3.4, each one representing a typical day of a culture. Last panel plots the average culture area as a function of the age. This measure is quite sensitive to segmentation errors or culture garbage.

### 3.2 Network statistics

#### 3.2.1 Formation of the giant connected component

During the growth phase, spanning from DIV 0 to DIV 6 in HD cultures, the number of nodes with at least one connection slowly increases with age, while the number of links rises exponentially, reaching a maximum at DIV 6 in HD cultures and DIV 8-10 in LD cultures, Figure 3.5(top). After this time point, the convergence of parallel neurites and neuronal clusterization induces a more gentle decrease in the number of links, accompanied by a slight reduction in the number of nodes. In order to properly compare networks of different size, we need to refer to a measure which is independent of the network size: the link density, defined as the ratio between the total number of measured links and the number of links characterizing the arrangement of the same number of identified nodes in a complete clique configuration. As illustrated in the inset of Figure 3.5(top), at any stage of development, the cultured networks are far from being fully connected (only about 2% of all possible connections exist between nodes), and thus operate in a low-cost regime of sparse anatomical connections.

In such a sparse connectivity regime, we quantify how our networks constrained in 2D space percolate. To do so, we measure the size $S_1$ of the giant connected component (GCC) and the size $S_2$ of the second largest component (GCC2) as a function of the age [145, 146]. Figure 3.5(bottom) shows that the number of nodes forming such connected
Figure 3.5 – Density of the network as a function of culture age (top-row) and first and second adjacency matrix giant component (bottom-row) at high (left-column) and low (right-column) density. (Top-Row) Mean number of nodes (yellow squares), including neurons and clusters of neurons, and links connecting them (green circles). Inset: the link density quantifies the actual number of links divided by that of an all-to-all configuration, $|N| \cdot (|N| - 1)/2$. (Bottom-Row) Log-linear plot of the mean number of nodes having at least one connection (yellow squares), of the mean size of the giant connected component (green circles) and of the second largest connected component (purple circle). In all plots, error bars stand for the standard errors of the mean (s.e.m).
components smoothly increases at the same rate during the first days of the network development, up to the DIV 6 (DIV 8) for HD (LD) cultures when the difference in size between them suddenly and consistently starts to grow. From that point on, the GCC2 starts collapsing and progressively merging into the GCC, and the establishment of an almost fully connected network of clusters characterizes the rest of the culture’s life.

The first days of the cultures’ development (from DIV 0 to DIV 3 in HD and LD cultures) were characterized by networks with very few nodes and links (see Fig. 3.5.Top). After DIV 3, the networks showed a very pronounced increase in the number of links and nodes (from DIV 3 to DIV 6 in HD cultures and from DIV 3 to DIV 8 in LD cultures) preceding a spatial network reorganization eventually driving the graph into its clustered, mature state.

3.2.2 Emergence of a small-world structure

The associated network statistics sheds light on the transition from random to non-random properties with a progression of both the clustering coefficient and the average path length (normalized by the GCC size) as a function of age (see Fig. 3.6.Top). The first significant result is the simultaneous increase in the clustering coefficient ($C$) and decrease in the mean path length ($L$), a clear fingerprint of the emergence of a small-world network configuration. In HD cultures, this configuration becomes prominent at DIV 6 and stays relatively stable through the last two weeks in vitro, while in LD cultures, although this configuration is more fuzzy due to the variability of the data, the same behaviour is observed. To properly asses the significance of this finding and isolate the influence of the variable network size and density, we calculated the values of $C$ and $L$ normalized by the corresponding expected values for equivalent random (and lattice) null model networks (see Fig. 3.6.Bottom). In doing so, we follow the approach used in similar circumstances for the obtainment of null models [16]. According to Watts and Strogatz’s model [29], a small-world network simultaneously exhibits short characteristic path length, like random graphs, and high clustering, like regular lattices. Here, in HD cultures, we found a clear change in the trend at DIV 6 where $L_{\text{rand}}/L \leq 1$, and the average path length of the cultured network starts to be close to that of a random graph and much smaller than that of a regular graph ($L_{\text{reg}}$ is calculated as $L_{\text{reg}} = S_1/(2\langle k \rangle)$). At the same time, the clustering coefficient was much higher (between 30 – 50 times) than that of the corresponding random graphs. Furthermore, in LD cultures, it is not until DIV 8 when this change occurs and it can be observed that the cultures’ mean shortest path is not stable but still is around the mean shortest path of a random network.

These results are in agreement with previous morphological characterizations of in vitro neuronal networks at a single developmental stage [17], where a similar small-world arrangement of connections was evidenced at DIV 6. However, to reinforce the evidence of the emergence of a small-world configuration during the graph development (as well as the fact that here the small-world metrics are not influenced by network disconnect- edness), we also measured the global and local efficiency, as introduced by Latora and Marchiori in [30]. These latter quantities, indeed, are seen as alternative markers of the small-world phenomenon, in that small-world networks are those propagating informa-
3.2. Network statistics

![Network clustering and shortest path properties as a function of culture age at high (left-column) and low (right-column) density.](image)

**Figure 3.6** – Network clustering and shortest path properties as a function of culture age at high (left-column) and low (right-column) density. (Top-Row) Clustering coefficient $C$ (yellow circles) and mean path length $L$ (green squares) normalized by the size of the largest cluster ($S_1$). (Bottom-Row) Log-linear plot of normalized values of $C$ and $L$ with respect to the expected values for an equivalent random network having the same number of nodes and links and preserving the degree distribution: $C_{\text{rand}}/C$ (green circles) and $L_{\text{rand}}/L$ (yellow squares). The average path length is also compared to the value for a regular lattice as $L/L_{\text{reg}}$ (purple triangles) with $L_{\text{reg}} = S_1/2\langle k \rangle$, being $\langle k \rangle$ the average connectivity and $S_1$ the size of the largest connected component. A small-world network requires that $C_{\text{rand}}/C \ll 1$, $L_{\text{rand}}/L \leq 1$ and $L/L_{\text{reg}} \ll 1$. 

$C$ and $L$ are normalized by the size of the largest cluster ($S_1$).
**Figure 3.7** – Local (Top-Row) and Global (Bottom-Row) efficiency as a function of culture age and compared to their respective values for the random graphs of the null model. Both quantities are averaged for the set of cultures at each day of measure (DIV) and error bars represent the standard error of the mean.
3.2. Network statistics

The efficiency curves of the cultured networks are reported in Fig. 3.7 as a function of age, and compared to the efficiency of the equivalent random graphs. Our results indicate that the connectivity structure of the neuronal networks evolve towards maximizing global efficiency (making it similar to the value of random graphs), while promoting fault tolerance by maximization of local efficiency (which is, instead, larger than the local efficiency of a random graph), and both properties are carried out at a relatively low cost in terms of number of links (see again Fig. 3.5.Top).

3.2.3 Emergence of assortative and single scale networks

Turning now our attention to network statistics at the micro-scale, we investigated how the node degree distribution evolved during maturation process. At all ages, cultures appeared to belong to the class of single-scale networks, displaying a well defined characteristic mean node degree. Figure 3.8.Top shows that the cumulative degree distributions $P_{\leq}(k)$ for DIVs 3, 6, 7, and 12 in HD cultures and DIVs 4, 7, 8, 11 in LD cultures had a fast declination with a non monotonous increase in the average connectivity, with most of the nodes having a similar number of connections and only a few ones with degrees deviating significantly from such a number.

The data were fitted to an exponential scaling law $y(k) \sim \exp(-k/b)$ with a level of confidence larger than 95%. It has to be remarked that the distribution of node connections, although always homogeneous, shifted during culture maturation toward much broader distributions, with few highly connected nodes appearing at DIVs 6 and 7. These “hubs” at the peak days of the culture evolution result from a branching process, allowing each single neuron to reach a larger neighborhood. Thus, at variance with scale-free networks [101, 109], our cultured and clustered networks are identified as a single-scale homogeneous population of nodes. This is in agreement with reports on many other biological systems like the neuronal network of the worm *Caenorhabditis elegans* [67, 121], and suggests the existence of physical costs for the creation of new connections and/or nodes limited capability [123].

While the number of neighbors (the degree) is a quantity retaining information at the level of a single node, one can go further to inspect degree-degree correlations, i.e. to quantify whether the degrees of two connected nodes are correlated. It is known, indeed, that biological networks generally feature disassortative network structures [111], that is connections are more likely to be established between high-degree and low-degree nodes. The assortativity coefficient $r$ was calculated as the Pearson correlation coefficient corresponding to the best fit of $\log(\langle k \rangle) \sim p \log(k^p)$. If $r > 0$ ($r < 0$), the network is set to be assortative (disassortative), while depending upon the obtained value of $p$, the degree correlation properties are said to be of a linear ($p = 1$), sub-linear ($p < 1$), or super-linear ($p > 1$) nature. Figure 3.8.Bottom shows the age evolution of the Pearson coefficient $r$ and of the exponent $p$ of the degree correlation ($\langle k_{nn} \rangle = ak^p$). At one hand, as $r$ stays positive during the whole development we can generally conclude that our networks are assortative [147] and, on the other hand, as the degree correlation $p$ is all days under 1 this indicates that it keeps a sub-linear ($p \sim 0.6$) degree-degree correlation.
Figure 3.8 – Degree distribution and degree-degree correlation at high (left-column) and low (right-column) density. (Top-row) Cumulative node degree distributions on a log scale at DIVs 3, 6, 7 and 12 for a high density culture and DIVs 4, 7, 8 and 11 for a low density culture. (Bottom-row) Degree correlation exponent (green circle) measuring the network assortativity and the corresponding Pearson coefficient (purple circle). Both quantities are averaged for the set of cultures at each day of measure (DIV) and error bars represent the standard error of the mean.
regime during the small-world stage.
A series of previous studies [137, 138] singled out tension along neurites and adhesion to the substrate as the two main factors conditioning the neuronal self-organization into a clustered network. Here we go a step ahead, and propose a simple spatial model which not only incorporates migration of neurons but also explicitly considers neurite growth, and the establishment of synaptic connections.

### 4.1 First model: Isotropic growth

The isotropic model is schematically illustrated in Figure 4.1. We start by considering a set of $N$ cells. Cells are small disk of radius $a$ randomly distributed in a 2-dimensional circular substrate of area $S$. The algorithm then makes the connections and positions of such disks evolve at discrete times, each time step $t$ corresponding to a DIV of the culture. The complex process of neurite growth and the establishment of synaptic connections is modeled by associating to each cell a time growing disk in such a way that, two cells are linked at a given time step if their outer rings intersect as shown for DIVs 2 and 3 in Figure 4.1. At each time step $t$, the radius $r_i \geq a$ increases by a quantity $\delta r_i$ which
Figure 4.1 – Schematic representation of how cells get connected. At DIV 0, 4 cells of radius \( a \) are located at random positions. The first iteration of the algorithm, DIV 1, assigns to each cell \( i \) a disk of radius \( r_i \) (blue shade). At the next iteration, DIV 2, the disks’ growth rate decreases, \( r_i' \), and a link between two cells is established when their disks intersect (DIV 3). This process continues until \( T_s \) steps.

Figure 4.2 – Force diagram explaining cell migration and clustering. Tension forces \( T_1, T_2 \) and \( T_3 \) are acting on the central cluster composed of two cells, whose vector sum (red arrow) exceeds the adhesion to the substrate (green arrow). As a result, a new equilibrium state is produced with new tension forces \( T'_1, T'_2 \) and \( T'_3 \), being the central cluster pulled in the direction of the net force approaching the largest cluster.

Decays as

\[
\delta r_i' = \frac{V}{2t} \left( 1 - \frac{1}{K_i} k_{it}^{i-1} \right)
\]  

(4.1)

where \( V \) is the neurite growth velocity (the same for all cells), \( K_i \) a random function of the cell amount in the interval \([1, N]\) and \( k_{it} \) the degree of the node (cell) at the time step \( t \). The term \( k_{it}^{i-1}/K_i \) introduces heterogeneity in the cell population, and represents the fraction of links acquired by the cell in the previous steps from the initial randomly assigned endowment \( K_i \). A very large \( K_i \) indicates that, potentially, a cell is very active and could connect many other cells. The wiring process is iterated up to a given time step \( T_s \), at which the formation of new connections is stopped.

As for the process of cell migration and clusterization, cells or clusters whose distance is less than \( 2a \) are then merged into the same new cluster. Furthermore, whenever two cells are connected, an initial tension \( \vec{T}_{ij} = 0.1\vec{u}_{ij} \) is created between them [14], and it is incremented in 0.1 force units at each time step, being \( \vec{u}_{ij} \) the unit vector along the direction connecting the two cells. The total force acting on a cell or cluster \( i \) is given...
4.1. First model: Isotropic growth

By \( \vec{F}_i = \sum_j \vec{T}_{ij} \) with \( j \) running over the cell indexes connected to \( i \), and not belonging to the same cluster. Furthermore, each cell is “anchored” to the substrate by a force \( F_a = 0.9 \) force units, and the \( i^{th} \) cell can only be detached if there is a net force \( F_i \) acting on it larger than \( F_a \). In the case of a cluster of cells, the adhesion force to the substrate is considered to be the sum of the individual adhesion of the cells composing the cluster. Therefore, cells and clusters move in a certain direction in all circumstances in which the net force acting on them overcomes the adhesion force, and an equilibrium point is reached at a new position in which the new net force balances (or is smaller than) the adhesion to the substrate (see Figure 4.2).

In order to validate our model, we ran a large number of simulations for different values of the model parameters \( N \) (number of neurons), \( V \) (initial neurite spreading velocity) and \( T_s \) (stop time) where \( N = \{100, 300, 500, 700\} \), \( V = \{50, 55, 60, \ldots, 100\} \) and \( T_s = \{4, 6, 8, 10, 12\} \). Each network is simulated from day of culture (DIV 0) to day of death (DIV 19). Every single combination of \( N, V \) and \( T_s \) is reproduced three times. From each simulation, the same measures as in Chapter 3 were computed (Number of neurons, number of links, number of nodes, connected component sizes, average shortest path, clustering and so on).

We compare the statistical topological features of the simulated networks to those measured from the set of high density cultures and the set of low density cultures. Figure

---

**Figure 4.3** - Comparisons between simulated networks (isotropic model) and experimental networks. Each cell within graphs represents the mean correlation coefficient when fitting modeled features to experimental features at a given number of neurons \( N \), initial neurite velocity spreading \( V \) and DIV when neurite stop growing \( T_s \). The features used to these comparisons are the number of nodes, number of edges, average shortest path and clustering. (Top-Row) Mean correlation coefficients between simulations and high density cultures evolution. (Bottom-Row) Mean correlation coefficients between simulations and low density cultures evolution.
4.3 depicts the mean correlation coefficient when fitting the simulated features to the real features. Features used to this purpose are the number of nodes, number of edges, average shortest path and clustering. The top row of this figure shows comparisons of simulations with high density cultures while the top down shows comparisons between simulations and low density cultures. The highest mean correlation coefficient in the top (down) row is in the region where \( N = 700, V = 85 \pm 5 \) and \( T_s = 12 \) (\( N = 300, V = 85 \pm 5 \) and \( T_s = 12 \)).

Figure 4.4 shows a typical output of the evolution of two simulated networks at high (700 neurons) and low (300 neurons) density. The initial number of cells is taken to be of the same order as in the experiments, and we chose as values for the parameters \( V \) and \( T_s \) those with the highest correlation with experiments. Boundary conditions mimic the real experimental setup by canceling the adhesion force to the substrate outside the culture area. Figure 4.4 depicts the spatial organization of clusters after 0, 3, 7, 12 and 19 DIV with \( N = 700, V = 85 \) and \( T_s = 12 \) (Top-row) and \( N = 300, V = 85 \) and \( T_s = 12 \) (Bottom-row).

Despite its relative simplicity, it is remarkable that the model offers a rather good qualitative verification of the trends of all the morphological and structural network characteristics measured in the experiments.

In particular, Figure 4.5 reports a comparison of the mean number of clusters, of mean cluster area and of the mean culture area for high and low density cultures and simulations. The mean cluster area curves have been normalized for comparison due to in the real data we measure the area in terms of number of pixels or \( \mu m^2 \) but in the simulated data the area is measured in terms of number of cells in a cluster. The mean culture area for real and simulated data is subtracted to the series average for analyzing the curves’ trend. It can be observed that simulated networks evolve in the same way
4.1. First model: Isotropic growth

Figure 4.5 – Morphological measures comparison between model and experiments. (i) Number of clusters, (ii) Normalized mean cluster area and (iii) mean culture area trend. Simulation parameters are the same as in the caption of Fig. 4.4, and each point is the ensemble average over 3 independent runs of the growth algorithm.
that real networks do, namely, merging neurons into clusters but preserving the cultured area.

Figure 4.6 shows likeness between real and simulated networks at high and low density by comparing the node and link density and the giant component sizes GCC and GCC2. As in the real case, simulated networks are far from being fully connected up to DIV 6. Then the connected components smoothly merge into the GCC and establish an almost fully connected network of clusters.

Simulated networks, like in real networks, transit from random to non-random network as seen by an increase in the clustering coefficient and a decrease in the average path length (normalized by the GCC size). Figure 4.7 in its top row depicts these changes and compares them with the real data. We can also observe that simulated networks fulfill the small world condition as in real network, that is, $C_{\text{rand}}/C \ll 1$, $L_{\text{rand}}/L \leq 1$ and $L/L_{\text{reg}} \ll 1$.

As in experimental data, the assortativity coefficient $r$ was calculated as the Pearson correlation coefficient corresponding to the best fit of $\log \langle (k) \rangle \sim p \log (k^p)$. It can be seen in Figure 4.8 that $r$ stays positive during the whole development meaning that our simulated networks are assortative like the real ones and, the degree correlation $p$ is all days under 1 symbolizing that they keep a sub-linear ($p \sim 0.6$) degree-degree correlation
4.1. First model: Isotropic growth

Figure 4.7 – Comparison between model and experiment. Legends in each panel clarifies on the topological quantities measured in experiments, and the corresponding trends of the simulated networks. Simulation parameters are the same as in the caption of Fig. 4.4, and each point is the ensemble average over 3 independent runs of the growth algorithm.
regime during the small-world stage.

Despite the simplicity of the model, the isotropic model fits qualitatively and quantitatively the trends of the features at morphological and structural levels of the networks at high and low density. However, an isotropic model for the neurite growth isn’t quite realistic due to in this model neurons cannot connect to distant neurons without linking to the nearest first. If we compare the distance distribution between connected neurons in experimental data and simulated networks (Figure 4.9), we observe this issue, the isotropic model has local connectedness while in the experimental model neurons are able to connect to further neurons.

### 4.2 Second model: Anisotropic growth

We start from the zero-model, previously reported in Section 4.1 that considers a set of $N$ cells randomly distributed with a uniform cell density $\rho$. Each cell is associated with an interaction disk of radius $r_i(t)$ with an initial radius $a$. Whenever two cells disk overlap, a link between them is established, whose length equals the distance between the centers of the two cells disks. If this distance is less than $2a$, then the two cells are merged into a single cluster. The time evolution of the growth of each disk depends on the actual connectivity $k_i(t)$ of the cell. At each discrete time step $t$ (here denoting the sequence of the DIV), the radius $r_i \geq a$ increases by a quantity $\delta r_i(t)$ which decays as seen in Equation 4.1 where $V$ is the neurite growth velocity (the same for all cells), $K_i$ a random number in the interval $[1, N]$, and $k_i$ the degree of the node (cell) at the time step $t$. The term $k_i/K_i$ introduces heterogeneity in the cell population, and represents the fraction of links acquired by the cell in the previous steps from the initial randomly assigned endowment $K_i$. 
4.2. Second model: Anisotropic growth

Figure 4.9 – Comparison between isotropic model and experiment. Normalized histograms of the distance between connected clusters in DIV 5 obtained in the experiments, and in simulations using the isotropic model. All experimental quantities are averaged for the set of six cultures at each day of measure (DIV). The simulation parameters are $N = 700$, $T_s = 12$, and $V = 85$.

Figure 4.10 – Schematic representation of how cells get connected in the anisotropic model. At each step, neurites from each cell growth looking for other neurons or neurites of other cells. This neurite growth is different from each simulated neurite given an anisotropic growth. This process continues until $T_s$ steps.
The only variation that we are considering here with respect to the original model of Section 4.1 is that now the neurite growth velocity \( V \) is no longer isotropic, by modifying the Equation 4.1 so that each cell disk \( i \) has a different radius with respect to cell \( j \), meaning that the radius \( r_{ij} \geq a \) increases by a quantity \( \delta r_{ij} \) which decays as

\[
\delta r_{ij}^t = \frac{\xi_{ij}V}{t} \left( 1 - \frac{1}{K_i^{t-1}} \right)
\]  

(4.2)

where \( \xi_{ij} \) is a random number drawn from a uniform distribution between 0 and 1. This leads to the fundamental consequence that two disks initially located very close to each other do not necessarily establish a connection (Figure 4.10), but they have instead an associated probability of linking, as observed in the experiment. Furthermore, and following Section 4.1, whenever a new link is formed between cells \( i \) and \( j \), a tension force \( T_{ij} = 0.1 \) is created along the line connecting them, and it is incremented in 0.1 force units at each time step. The total force acting on a cell or cluster \( i \) is given by the vectorial sum of all the tension forces acting on it. Finally, each cell is anchored to the substrate by a force \( F_{a} = 0.9 \) force units, and the \( i^{th} \) cell can only be detached if it experiments a net force \( F_{i} \) larger than the adhesion to the substrate. In the case of a cluster of cells, the adhesion force to the substrate is considered to be the sum of the individual adhesions of the cells composing the cluster.

To validate our model, as in Section 4.1, we ran a large number of simulations for
different values of the model parameters $N$, $V$, and $T_s$. Remarkably, when comparing by the mean correlation coefficient (Figure 4.11) when fitting the simulated features to those measured from the experiments (the number of nodes, number of edges, average shortest path and clustering), we found that high correlation values exist only in a very narrow window of $V$ and $T_s$. For instance, the parameter values which better fit the experimental observations for high density cultures are $N = 700$ neurons, $V = 65 \pm 5$ and $T_s = 10 \pm 5$ and for low density cultures are $N = 100$ neurons, $V = 65 \pm 5$ and $T_s = 10 \pm 5$.

Similar results to the isotropic model were obtained when comparing the morphological features (Figure 4.5) and some topological features (Figure 4.6 and Figure 4.8), therefore these graphics are not displayed here. However a more accurate fitting is obtained if we compare the shortest path normalized by the giant component size and the clustering at high and low density (Figure 4.12.Top). Consequently, the anisotropic model suits better the experimental data than the isotropic model when comparing the small-world conditions (Figure 4.12.Bottom).

Finally, we focus on the distance distribution. As depicted by Figure 4.13, in this case we have no abrupt tail distribution as with the isotropic model. On the opposite, as in the
Figure 4.13 – Comparison between anisotropic model and experiment. Normalized histograms of the distance between connected clusters in DIV 5 obtained in the experiments, and in simulations using the anisotropic model. All experimental quantities are averaged for the set of six cultures at each day of measure (DIV). The simulation parameters are $N = 700$, $T_s = 12$, and $V = 85$ for the isotropic model and $N = 700$, $T_s = 12$, and $V = 65$ in the case of the anisotropic model.

anisotropic model long-range connections can be established without making all close-range connections, the distance distribution has a smooth tail as in the experimental data.

4.3 Model analysis: transition to clustered networks

The neuron migration and clustering play a key role in the development of the network’s structure. According to [138], “clustering instability is a phenomenon exhibited by many living and nonliving systems”.

In this section, we examine the relation between cell density and network morphology and topology. For that purpose, we use the in silico model presented in Section 4.2 fixing model’s parameters $V$ and $T_s$ to 65 and 12 respectively and varying both, the cell density (or the number of seeded cells) and cell adhesion to the substrate (previously fixed to $F_a = 0.9$). The tension force between clusters and the cell adhesion to the substrate were considered the two key factors during neuronal self-organization [138]. We ran a larger number of simulations varying the initial number of cells ($N$) and the adhesion force ($F_a$). Specifically, $N = \{50, 100, 150, \ldots, 850\}$ and $F_a = \{0.1, 0.2, 0.3, \ldots, 1.9\}$. Each parameter combination was simulated three times.

We study both the morphology diagram and the topology diagram as a function of cell density and adhesion force. Figure 4.14 depicts some morphology measures revealing different behaviours in different regions, for instance, the average percentage of the grouped neurons (Figure 4.14(i)) defines a region (red-yellow), at the final stage, where neurons have created big clusters and a region (blue) less clusterized. The reader may notice that a poor cell adhesion to the substrate causes a massive migration (Figure 4.14(ii)) and clustering (Figure 4.14(ii)) while, in the other hand, a high adhesion to the substrate cause a moderate migration and, therefore, clustering. Figure 4.14(iii) depicts
4.3. Model analysis: transition to clustered networks

Figure 4.14 – Morphology diagram as a function of cell density and adhesion force. (i) Percentage of clustered neurons at DIV 19. (ii) Average traveling distance by each neuron to its final position at DIV 19. This measure was normalized by the substrate diameter (D). (iii) Average percentage of number of clusters to which each cluster is connected.

The average percentage of connections established from each cluster to the others.

We analyze the self-organization process in terms of the emergence of network structure. To that end, we compare the average shortest path and the clustering coefficient of the in silico model to the theoretic ones in random and regular networks. Figure 4.15 shows these comparison at DIV 19 as a function of neuron density and adhesion to the substrate. Just to call to mind, a small world topology have to meet $C_{\text{rand}}/C \ll 1$ (Fig. 4.15(i)), $L_{\text{rand}}/L \leq 1$ (Fig. 4.15(ii)) and $L/L_{\text{reg}} \ll 1$ (Fig. 4.15(iii)). We differentiate among four regions (Fig. 4.15(iv)): (1) high density cultures bring forth a full connected net at first days since neurons are quite close to each other. However a poor adhesion to the substrate produces the cell migration and clustering. Hence, at the final stage, most neurons are merged into giant clusters highly interconnected. (2) these cultures yield a small world structure at the final stage. This is produced by a moderate migration due to higher adhesion force. (3) low density cultures produce unconnected networks. In this region some small nets appear and they are not able to migrate and, therefore, to create clusters due to the tension force in cells is lower than the adhesion to the substrate. Finally, (4) low density cultures with a poor adhesion force produce a final stage where may or not appear clusters but they are fully unconnected with others.

As way of illustration, Figure 4.16 depicts some networks within these region at first day (DIV 0) and last day (DIV 19).
Figure 4.15 — Small world diagram at DIV 19 as a function of cell density and adhesion force. A small world network requires (i) $C_{\text{rand}}/C \ll 1$, (ii) $L_{\text{rand}}/L \leq 1$ and (iii) $L/L_{\text{reg}} \ll 1$. (iv) regions of the phase diagram with different structure properties: 1) region where neurons are highly grouped presenting a shortest path near to a regular network. 2) small world region where all the conditions are met, i.e. low shortest path $L_{\text{rand}} \sim L$ and high clustering coefficient. 3) Unconnected networks. Some small graphs appear, however most clusters are disconnected 4) Unconnected networks. All clusters are unconnected. Black and white lines are depicted as a help to the eye.
4.3. Model analysis: transition to clustered networks

Figure 4.16 – Initial (DIV 0) and final (DIV 19) network structure examples in different regions of the phase space. (orange) Representative network for those belonging to the region where cells group together the most. (blue) Network for the small world region. (green) Typical network for the region where some small graphs are created but the network remains unconnected. (purple) Fully unconnected network. Every dot within networks represent a cluster and links between cluster are depicted in blue lines.
5.1 Conclusions

Despite the existence of some functional studies of CNNs [136, 13] and also of a few theoretical-numerical attempts to model the relationships between CNNs structure and function [148], no experimental verifications are available so far of such a connection, basically due to the lack of tools allowing a simultaneous tracking of the cultures dynamical activity and morphological/topological changes.

To cater for this need, in this thesis, we developed an algorithm and a set of tools to enable automatic location of neurons and tracing of neurites in non-invasive phase-contrast images, which can be acquired simultaneously with electrophysiological measures (with e.g., multi-electrode arrays), hence potentially allowing for a combined study of network structure and dynamics.

The relevance and value of our work is then to make a first step in the direction of unveiling and uncovering structure/function relationships during the evolution of a CNN. In particular, in this thesis we operated a fully longitudinal inspection of a single
cultures life and maturation process, in which some aspects of the topological and functional organization can be compared [33]. The accuracy of the proposed algorithm (in comparison to the manual and automated evaluations) is very good, and the computational demand is rather low. It is worth saying that one way to improve the validation of our algorithm would have been the use of green fluorescent protein (GFP) transfection or simply anti-horseradish peroxidase (HRP) staining. These tools label the connections and would therefore provide precise information on the physiological, ground-truth topology of the network. Although these labelling techniques are difficult to combine (the former) or incompatible (the latter) with a continuous monitoring of the culture maturation, their availability at a particular age of the culture would have allow the access to other neuronal tracing automated software to compare with and to produce ground truths with most of the inter-neuronal connectivity. Apart from quality image issues, a potential bottleneck of our algorithm is working with very high dense neuronal networks as the extraction of the network structure could be compromised mainly due to neurite crossing or high interconnectedness.

We provided a large scale experimental investigation of the morphological evolution of in vitro primary cultures of dissociated invertebrate neurons from locust ganglia. At all stages of the cultures development, we were able to identify neurons and neurites location in automated way, and extract the adjacency matrix that fully characterizes the connectivity structure of the networking neurons. A systematic statistical analysis of a group of topological observables has later allowed tracking of the main network characteristics during the self-organization process of the culture, and drawing important conclusions on the nature of the processes involved in the culture structuring. At early stages of development (< DIV 3) characterized by a high neurite growth rate, homogeneous node degree distribution and low clustering resulted in a random topology as expected given the fact that neurons were randomly seeded. Following this immature period, neurite growth rate diminished and tension along neurites started to shift the network to a small-world one with path lengths similar to random configurations but presenting high clustering of connections. This transition from random to small-world concurred with the percolation of the culture and the onset of the giant connected network component. Furthermore, the identification of the main physical processes taking place during the cultures morphological transformations, allowed us to embed them into a simple growth model, qualitatively and quantitatively reproducing the overall scenario observed in the experiments.

Our results extend previous studies where network properties of cultures were investigated at a particular developmental stage and for a lesser number of nodes [17]. These results also systematically characterize several topological network measures along the entire cultures evolution, and unveil many yet unknown self-organization properties, such as i) the fact that a small-world configuration spontaneously emerges in connection to a large increase in both local and global networks efficiency, and ii) the evidence that cultures tend to organize in a regime of non trivial degree mixing which, in turn, is characterized by assortative degree-degree correlation features. The evolution from an initial random to a small-world topology has also been reported recently in the context
of a functional network of a cortical culture [13]. However, although functional connectivity correlates well with anatomical connectivity, there are studies showing that strong functional connections may exist between nodes with no direct physical connection [34]. This suggests that future studies are needed in which both anatomical and functional networks are accessible in order to understand their complex entanglement. Given the absence of external chemical or electrical stimulations, we conclude that such complex network evolution and morphological structuring is indeed an intrinsic property of neuronal maturation. Our study therefore contributes to the understanding of the complex processes ruling the morphological structuring of cultured neuronal networks as they self-organize from collections of separated cells into clustered graphs, and may help identifying culture development stages in new, specific and targeted, experiments.

In summary, this thesis contributes with an image segmentation method based on multi-layer graphs, that allows to extract the network structure and, therefore, to study the network development at morphological and topological levels [15]. Additionally, the proposed image segmentation method was used successfully in others research fields. Specifically, it was proposed as an image segmentation method for hand segmentation in biometric recognition reaching an accurately result against different backgrounds [149] (Appendix B).

We researched the network self-organization for in vitro Schistocerca Gregaria’s frontal ganglion’s cultures at high and low densities. Finally, we proposed an in silico network model able to reproduce the development of the in vitro neural network at high and low densities.

5.2 Future work

Several aspects have been left as future work during the research of this thesis. The intention of this section is to collect every consideration appeared within this thesis which is worthy of a deeper insight and research.

Regarding the image segmentation procedure:

- **Enhancement of the image segmentation** considering three possibilities: include an (1) image pre-processing step previous to segmentation; (2) a neurite analysis to distinguish between branching points and crossing neurites; and (3) a distinction between dendrite and axon.

- **Neurite recognition enhancement**. Robust neurite recognition despite of small errors during segmentation. Small segmentation errors lead to a notoriously incorrect net structure.

- **Automatic image acquisition** by means of combining both the algorithm speed and the possible communication with the microscope’s robotic stage. Thus, image acquisition and net structure extraction could be performed simultaneously, reducing substantially the acquisition time.
• Other cultures. The presented algorithm along this thesis is optimized and oriented to *Schistocerca Gregaria* cultures. The algorithm can be easily customized (and its parameters conveniently tuned) to study the spontaneously emerging morphology and organization in cultures of dissociated neurons of other animals, and in particular of vertebrates (rats and mice) given the current level of attention and their vast use in a series of other experiments with CNNs.

• Evaluation. the use of green fluorescent protein (GFP) transfection or simply anti-horseradish peroxidase (HRP) staining. These tools label the connections and would therefore provide precise information on the physiological, ground-truth topology of the network.

• Proper tracking of individual neurons along the culture.

Regarding structure analysis:

• Rest of the extracted adjacency matrices analysis.

• Comprehension about both neurite’s output angle and dendrite’s trees.

Regarding the model:

• State model-based hypothesis about net structure’s behaviour and contrast them based on real in vitro experiments.

• Update former model with neuron dynamics models in an attempt to analyze the relation between functional and topological networks (under a controlled environment).

• A more precise and realistic model considering branches, pruning, neurites’ crossing and optimization.

Regarding global aspects:

• Perform experiments where neurons activity is recorded while net structure is captured by means of image processing. Understand the relationship between functional and topological nets in vitro within these latter experiments.

• Study the fractal dimension on networks.

Some former points are already under development. In fact, next section intends to provide some spotlights on these points, so that the reader may appreciate that the work carried out under this thesis was not only limited to the net structure’s study, but also to the development of the required tools to perform most of the former presented future work.
5.3 First steps in the future

In order to be able to link the structural with the functional network properties, first records were done. In this section, we briefly talk about this first step.

Cultures from Rat Hippocampal E18 (embryo at day 18) and Cortex E18 were done over a multi-electrode array (MEA). This MEA is used to record the electrical activity of the nearest neurons to each electrode. Different MEAs with different spatial configuration of the electrodes were considered. In our case, we use two 60 electrodes solutions. In one of them, the electrodes are positioned in a 6x10 grid while the other has a 4 quadrants configuration with 13 electrodes at each quadrant and 7 electrodes forming a line in the middle of the electrode. We name those electrodes as M-type or M for the 6x10 grid configuration and Q-type or Q for the 4 quadrants configuration (Figure 5.1).

Every culture day, three M-type electrodes and three Q-type electrodes were seeded at the same time with 200,000 cells/MEA. Every recording day (three days per week since the beginning until the 30th day *in-vitro*) the electrical activity were recorded during 15 minutes using an acquisition card provided by Multichannel Systems able to sampling the 60-channels data at 50kHz.

Each single channel records up to six near neurons if they are closed enough. In our case, we record up to three neurons.

First steps to analyze these signals entail both the spike (or action potential) detection over the noise level in each electrode and the sorting (or classification) of the detected spikes. In Section 5.3.1 we give an introduction to a developed tool built with the aim to deal with these signals.

5.3.1 NEural SOrter

To understand how the neurons code their activity it is essential to identify the activity generated by each corresponding neuron (Figure 5.2). Even though many spike-sorting algorithms approaches have been developed to improve the reliability of the sorting...
process, the efficiency and accuracy of this task remains as an unresolved issue in the field of neuroscience. Furthermore, commercial and free software spike sorting applications are not open formats or multiplatform, making the different systems incompatible with each other and a very tedious and expensive task for the users.

NEural SOrter or NESO was originally developed because of the need to have a tool to analyze data files with different file formats from different commercial acquisition systems. Taken advantage of the Neuroshare\(^1\) project and of the companies that develop their own libraries according to the Neuroshare standard, we developed a program able to analyze files that share this policy like *.nev (Blackrock Microsystems, Salt Lake City, UT, US) and *.mcd (Multi Channel Systems, Reutlingen, Germany) file formats (up to date).

NEural SOrter\(^2\) is a C++ application running in Windows, Mac OS X and Linux without special requirements. NESO allows the user to sort spikes from different commercial acquisition systems by using automatic, semi-automatic and manual procedures.

- The manual sorting allows interacting with the sorting classification either to improve the output of the automatic and semiautomatic spike sorting or to group the spikes manually.

\(^1\)www.neuroshare.org
\(^2\)neuralsorter.sourceforge.net
5.3. First steps in the future

Figure 5.3 – Spike waveform Height-Width features. NESO extracts eleven features from each spike waveform, nine of them correspond to time features and two related to voltage. Voltage features are: maximum spike voltage (MSV, colored in red) and minimum spike voltage (mSV, colored in green). Time features are: time of MSV (red), time of mSV (green), time interval between mSV and MSV (blue) and widths at 25%, 50% and 75% of peak heights (pink for mSV and orange for MSV).

- The semi-automatic sorting is split up into two phases. The first one extracts representative features from each spike and the second one classifies the spikes by means of these features. The feature extraction methods implemented are: Principal Component Analysis (PCA) and Height-Width (HW) method. PCA is a mathematical method that decomposes the data covariance matrix (PCA-Cov), correlation matrix (PCA-Cor) or sum of square (PCA-SoS) matrix of waveforms into the eigenvalues after mean extraction and then orders the features that contribute the most to the differences among the waveforms [150]. The HW method takes height and width measurements of the spike waveform like the maximum and minimum spike voltage (MSV and mSV respectively) or times between MSV and mSV among others (Fig 5.3). The features sorting methods classify the spikes with similar features into the same group (or unit). The methods implemented were: Vector-Quantization and K-means.

- The automatic sorting uses spike templates to classify the spikes to the most similar template.

To compare the implemented algorithms with the state-of-the-art we use the work by Wild et al. in [151] where the authors compared three different methods, WaveClus [152], KlustaKwik [153] and Osort [154]. The authors used JW [151] and QQ [152] synthetic signals dataset with noise level between 0.0 and 0.6 as ground-truth for the results given by each algorithm. The noise level “was defined as the reciprocal value to the signal-to-noise ratio SNR”. The spike sorting accuracy was measured as the Adjusted Mutual Information (AMI) [155, 156]. According to the AMI measure, “the value is 0 if the clustering provides information about the true clustering just by chance, and it is 1 if all information is revealed, meaning that the two clustering are the same” [151]. Figure 5.4
Figure 5.4 – Spike sorting evaluation. M1 and M2 extracted features from spikes applying PCA-Corr; M3 and M4 used PCA-Cov; M5 and M6 used PCA-SoS and M7 and M8 used H-W. To classify extracted features, M1, M3, M5 and M7 use K-Means with 5 clusters and 100 iterations and M2, M4, M6 and M8 use V-Q with a threshold of 5. M9 combines different methods provided by NESO in order to enhance spike sorting results.

depicts the spike sorting accuracy of the combined methods (M) of features extraction and clustering for samples of spikes from the JW (20 and 60 seconds recordings) and QQ (30 seconds recordings) data sets. Best results, close to 1, were obtained by mixing methods (M9) (where mean AMI score is 92% for noise level 0.00-0.15 and 79% for noise level 0.45-0.60 among others) followed by PCA-Cov (M3,M4) and PCA-SoS (M5,M6).
Appendices
Multi-layer graph-based algorithm pseudocode

Algorithm 1  Multi-layer graph-based segmentation
1: function doSegmentation(M,threshold,depth)
2:   nodes[0] ← INITIALIZE_NODE(M,0,null)
3:   for i ← 1 to M.height × M.width do
4:     neighbor ← NEIGHBORHOOD_LEVEL0(i)
5:     nodes[i] ← INITIALIZE_NODE(M,i,neighbor)
6:     MERGE_NODES(nodes,i,threshold)
7:   end for
8:   clusternode ← nodes
9:   for l ← 1 to depth do
10:      clusternode ← PREPARE_NEXT_LEVEL(clusternode)
11:      OPTIMIZE_NEIGHBORHOOD(clusternode)
12:      threshold ← threshold + 0.1
13:      for i = 1 to clusternode.length do
14:         MERGE_NODES(clusternode,i,threshold)
15:      end for
16:   end for
17:   R ← CONSTRUCT_RESULT(nodes)
18:   return R
19: end function
Algorithm 2 Initialization
1: function INITIALIZENODE(M,i,neighbor)
2: \[ \text{node.mean} \leftarrow \text{node.max} \leftarrow \text{node.min} \leftarrow M(i) \]
3: \[ \text{node.nelems} \leftarrow 1 \]
4: \[ \text{node.neighbor} \leftarrow \text{neighbor} \]
5: return node
6: end function

Algorithm 3 Initialization
1: function NEIGHBORHOODLEVEL0(i)
2: if \( i == 0 \) then
3: else if \( i \% M_{\text{height}} == 0 \) then
4: \[ \text{neighbor} \leftarrow \{i - M_{\text{height}}, i - M_{\text{height}} + 1\} \]
5: else if \( i \% M_{\text{height}} == M_{\text{height}} - 1 \) then
6: \[ \text{neighbor} \leftarrow \{i - 1, i - M_{\text{height}} - 1, i - M_{\text{height}}\} \]
7: else
8: \[ \text{neighbor} \leftarrow \{i - 1, i - M_{\text{height}} - 1, i - M_{\text{height}}, i - M_{\text{height}} + 1\} \]
9: end if
10: return neighbor
11: end function

Algorithm 4 merging
1: function MERGENODES(nodes,i,threshold)
2: \[ \text{id} \leftarrow \text{LEADERNODE}(nodes,i) \]
3: for each \( \text{neig} \) in \( \text{nodes}[i].\text{neighbor} \) do
4: \[ \text{distance} \leftarrow \text{EUCLEDIANDISTANCE}(\text{nodes}[i],\text{neig}) \]
5: if \( \text{distance} \leq \text{threshold} \) then
6: \[ \text{id}_{\text{neig}} \leftarrow \text{LEADERNODE}(\text{nodes},\text{neig}.\text{id}) \]
7: if \( \text{id}_{\text{neig}} < \text{id} \) then
8: \[ \text{nodes}[\text{id}].\text{id} \leftarrow \text{id} \leftarrow \text{id}_{\text{neig}} \]
9: else
10: \[ \text{id}_{\text{neig}} \leftarrow \text{nodes}[\text{id}].\text{id} \]
11: end if
12: else
13: \( \text{neighbor}'\).append\( (\text{neig}) \)
14: end if
15: end for
16: end function
Algorithm 5 merging
1: function LEADERNODE(nodes, i)
2:    id ← nodes[i].id
3:    while nodes[id].id == id do
4:        id ← nodes[id].id
5:    end while
6:    return id
7: end function

Algorithm 6 merging
1: function PREPARENEXTLEVEL(clusters)
2:    next ← {}
3:    for each node in clusters do
4:        id ← node.id
5:        id_cluster ← LEADERNODE(clusters, id)
6:        if id ≠ id_cluster then
7:            clusters[id].id ← id_cluster
8:            clusters[id_cluster].max ← max(clusters[id_cluster].max, node.max)
9:            clusters[id_cluster].min ← min(clusters[id_cluster].min, node.min)
10:           aux1 ← clusters[id_cluster].mean · clusters[id_cluster].nelems
11:           aux2 ← node.mean · node.nelems
12:           clusters[id_cluster].nelems ← clusters[id_cluster].nelems + node.nelems
13:           clusters[id_cluster].mean ← \frac{\text{aux1} + \text{aux2}}{\text{clusters[id_cluster].nelems}}
14:           clusters[id_cluster].neighbor.insert(node.neighbor)
15:        else
16:            next.append(id_cluster)
17:        end if
18:    end for
19:    return clusters[next]
20: end function
Algorithm 7  merging
1: function OPTIMIZE NEIGHBORHOOD(clusters)
2:     for each node in clusters do
3:         neighbor ← node.neighbor
4:         neighbor′ ← {}  
5:             for each neig in neighbor do
6:                 id ← LEADER NODE(clusters, neig.id)
7:                 neighbor’.insert(id)
8:             end for
9:         node.neighbor ← neighbor'
10:     end for
11: end function

Algorithm 8 merging
1: function CONSTRUCT RESULT(nodes)
2:     for \(i \leftarrow 0\) to \(M_{\text{height}} \times M_{\text{width}}\) do
3:         id ← LEADER NODE(nodes, i)
4:         R[i] ← id
5:     end for
6:     return R
7: end function
Multi-graph based segmentation algorithm applied to hand biometry

Next article was presented on the 49th Annual International Carnahan Conference on Security Technology [149].
Low Computational Cost Multilayer Graph-based Segmentation Algorithms for Hand Recognition on Mobile Phones

Daniel de Santos-Sierra*, Miguel F. Arriaga-Gómez*, Gonzalo Bailador*, Carmen Sánchez-Ávila*
* Group of Biometrics, Biosignals and Security, GB2S
Universidad Politécnica de Madrid
Campus de Montegancedo
Pozuelo de Alarcón, 28223, Madrid, Spain
Email: {dsantos,marriaga,gbailador,csa}@cedint.upm.es

Abstract—Unconstrained and contact-free hand recognition problem with mobile devices is not solved yet because these systems have to deal with hard problems like different backgrounds and illumination. Algorithms to perform an image segmentation in order to create regions in the image with the same semantic meaning are a work in progress. Graph theory has been used successfully in order to reach a good image segmentation in many fields but these algorithms are computational demanding (time and memory) making it very difficult to use on mobile platforms. New algorithms to perform image segmentation are needed in order to adapt biometric technologies to mobile devices. This paper presents a segmentation algorithm based on multilayer graphs. We compared our results with other known segmentation algorithms (NCuts and KMeans) by using a synthetic database with over 400000 images. Our results show that the optimized implementation of the proposed algorithm makes this a powerful tool with high accuracy and low computational cost, improving the accuracy and the execution time of the two other algorithms.

Index Terms—Hand geometry, image segmentation, graph theory, synthetic database

I. INTRODUCTION

Recognition systems based on hand features have been widely used the last decade as one of the systems with higher accuracy and higher acceptability by the user [1], [2], [3], [4].

Biometric recognition based on images, like hand or iris recognition, usually consists of five stages: 1) acquisition, 2) preprocessing, 3) feature extraction, 4) matching and 5) database operations.

In order to achieve the best performance, the preprocessing phase, which aims at locating the hand within a picture, is the most important task and the most sensitive to environmental conditions.

Hand geometry has been widely studied, and it can be classified into three categories according to the image acquisition criteria: 1) constrained and contact-based (it needs a flat platform with pegs to place the hand over during the acquisition) [5], 2) unconstrained and contact-based (peg-free scenario but a flat platform is required) [6] and 3) unconstrained and contact-free (neither pegs nor platforms are needed)[7].

Hand geometry biometrics with mobile devices belongs to this latter category, unconstrained and contact-free. We have to deal with non-controlled requirements, like light variations, different backgrounds, and so forth. These requirements do the preprocessing task to locate the hand into a picture (hand segmentation) more difficult.

One of the most powerful tools to segment an image into regions with semantic meaning uses graph theory. In graph theory a node represents a pixel within an image and a link between two nodes stands for the color-based relationship between the two pixels.

Graph theory has been used successfully in order to reach a good image segmentation in many fields [8], [9]. Some of them are, NCuts (Normalized Cuts)[10], LDC (Lossy Data Compression)[11] and GMA (Gaussian Multi-scale Aggregation)[12].

However, these graphs have to deal with the known problem of dimensionality resulting from converting an image into a weighted graph, that represents the relationship between every pair of nodes. That means that graph theory algorithms are costly in time and memory, making it hard to use with mobile devices. Hence, most of the segmentation algorithms in mobile devices are based on finding pixels similar to skin-colour or using thresholds that don’t work fine with different illuminations and backgrounds.

The main goal of this work is to present an algorithm for the segmentation task based on multilayer graphs. The optimization of the proposed algorithm makes this a powerful tool with a low computational cost.

Finally, the layout of the paper remains as follows: Section II provides a complete description of the segmentation algorithm and the method to generate the synthetic database with more than 400000 hand images with different backgrounds used to evaluate the system. Section III presents a comparative performance study of the algorithm with two known segmentation algorithms, NCuts and KMeans. The performance is measured in terms of computational cost and accuracy. We present in this section a methodology to choose optimal parameters to compute the algorithm. And finally, we conclude in Section IV.
II. METHODS

This section contains the complete description of the methods used to perform the image segmentation (Section II-A) and the method used to create a synthetic hand database from a controlled hand database (Section II-B).

A. Multilayer Graph-based Segmentation Algorithm

The proposed graph-based algorithm aims at providing an accurate segmentation for colour images. The segmentation process consists in grouping pixels with the same semantic meaning, i.e. grouping those pixels that belong to the same object.

The proposed multilayer graph-based algorithm, henceforth MGBA, considers the image as a graph where each pixel at position \((x, y)\) is represented as a node in the graph, and graph edges' weights represent the relationship among the nodes. The MGBA needs two parameters to work properly: 1) threshold to merge regions whose weight is under it and 2) depth or number of layers to be considered.

First MGBA layer considers an input image \(I\) as an undirected weighted graph \(G = (V, E, W)\), in which every pixel of the image is a node in \(V\), and the existence of an edge \(E_{i,j}\) connecting two nodes \(i\) and \(j\) depends on their spatial location and its weight \(W_{i,j}\) depends on a similarity function of those pixels. Specifically, for each node \(i\), we consider its pixel location \((x_i, y_i)\) in the spatial grid. We then consider the neighborhood \(N(i)\) of node \(i\) to be formed by its 8 first neighbors in the grid. The edges are weighted according to the euclidean distance between pixels features (RGB, L*a*b, or other).

Once the relationship between nodes (i.e edges) is defined, we consider those links whose weight is under the threshold value and we compute the connected components in order to get independent regions. Two nodes belong to the same components if there is a path between them (several links under the threshold that connect both), otherwise they belong to different components (i.e different regions).

The independent regions are considered as nodes for the next layer. The new node value is the mean value of the pixels in that component. Edges are defined as neighborhood, that means that if two independent regions are in touch (in the 2D space) then a link between them exists.

The last two processes are repeated as many times as indicated by the depth level. In each layer the threshold value increments its value by 0.1.

In this article, we use the RGB space color to build the MGBA being each node a RGB pixel. The distance between pixels is the euclidean distance between their RGB values. The mean value for each component is computed as the mean of each RGB layer.

B. Synthetic Database

A synthetic database was created to measure the reliability of the segmentation algorithm and as a benchmark to compare with other methods.

We started from a controlled hand-database with controlled background (a blue poster board) that we can remove easily in order to obtain the hand in the picture. The result is considered as the segmentation ground-truth.

Our hand-database has 25 pictures of both hands from 106 users.

We used 91 different backgrounds that we classified into 18 categories (see Table I Background column).

We combined the extracted hand with each background and then we applied a gaussian filter to blur the boundaries between hand and background (Fig. 1). We obtained 482300 syntethic images.

III. RESULTS

This section compares the proposed algorithm with two other algorithms, KMeans and NCuts. These two algorithms start from de a-layer of the L*a*b color space due to the best performance in terms of accuracy is obtained with this layer. KMeans[13] and NCuts[10] have one parameter, the number of segments in which they have to divide the image. As NCuts is a graph-based algorithm, it takes into account the spatial position of each pixel, then it merges those similar pixels that are in touch. On the other hand, KMeans is not a graph-based algorithm that takes into account the pixel's space position, i.e. in its solution exists regions with the same label (they are considered as the same object) but they are separated (they are in different locations). In this case, we improve the KMeans results by considering the segment within the region that most closely matches the ground truth.

On the other hand, MGBA uses all the layers in RGB space color, i.e. it is not required to change the color space but it needs two parameters: threshold and depth (Section II).
In Section III-A we provide the parameters used by the three algorithms. In order to evaluate the performance of MGBA algorithm in the analysis of images, we consider aspects concerning the computational cost (Section III-B) and accuracy (Section III-C).

A. Parameter Selection

In this article we compare the best performance in terms of accuracy of the NCuts and KMeans with the MGBA’s performance.

With NCuts and KMeans we used from 2 to 10 segments for each synthetic image and then we chose the segment that most closely fits with the image’s ground-truth. That means, the parameter has been optimized for each image.

For the MGBA algorithm, we randomly took 20 images with different backgrounds of 10 different hands and we used the combination of different thresholds and depths. Specifically, we used from 1 to 4.9 in 1.3 steps for threshold and from 1 to 1000 in 8 steps for depth. Finally, we selected those parameters that performed the best segmentation within all the backgrounds (in terms of F-score). That means the parameter selection give us the best set of parameters that best works in the 20 randomly selected backgrounds.

B. Computational cost

The computational cost of the MGBA algorithm is studied in terms of processing time in each layer and the processing time in the whole process.

The computational cost in each graph-layer is linear with the number of nodes as shown in the code. This is also supported by the linear regression in Figure 2A where the algorithm was executed with different images with different backgrounds, different sizes and the depth parameter fixed to 1.

As the number of nodes in each layer decreases, the amount of time to process each layer also decreases. In Figure 2B we study how the depth level affects to the total amount of time. We show that this time is dependent of the original image, i.e. depending on the image, the number of nodes from one layer to the next layer could decrease faster or slower than other image.

In order to compare the MGBA execution time with NCuts and KMeans, we took 91 images, one per background, and we analyzed the time needed to perform the best hand segmentation. We used an Intel Core i5 @ 3.00GHz and 6GB RAM and synthetic images of size 776x1296. The execution time with MGBA is $4.77 \pm 2.77$ seconds, while NCuts consumes $651.42 \pm 166.56$ seconds and KMeans consumes $0.37 \pm 0.12$ seconds. The time to convert RGB to $L^*a^*b^*$ has not been taken into account for NCuts and KMeans.

C. Accuracy

In this section we expect to give a comparative description with the NCuts and KMeans algorithms in terms of accuracy.

Figure 3 shows a comparative frame for segmentation evaluation. It can be appreciated visually how well the algorithms work against the different backgrounds. In green, we sketch the true positive pixels (TP), i.e. those pixels well classified. In red, we represent the false positive pixels (FP), i.e. those pixels that the algorithm classifies as hand but they are not. In blue, we depict the false negative pixels (FN), i.e. those pixels that the algorithm classifies as background but they are hand pixels.

Accuracy is measured in terms F-score [14], [15], defined as follows:

$$F = \frac{2RP}{R + P}$$

where $P$ (Precision or Confidence) is the number of true positive pixels divided by the number of pixels in the result ($P = \frac{TP}{TP + FP}$) and $R$ (Recall or Sensitivity) is the number of true positive divided by the number of pixels in the ground truth ($R = \frac{TP}{TP + FN}$). In that sense, F-score is a value within the interval $[0, 1]$ where 0 means bad segmentation and 1 means the best segmentation (i.e. the segmentation result is the same as the ground-truth).

Table I shows the results in terms of F-score of the MGBA algorithm in comparison to NCuts and KMeans. It can be observed that the results for the MGBA algorithm overcome the other two. The reader must notice that the parameters for methods NCuts and KMeans were optimized for each image while the parameters for the MGBA were not (Section III-A).

IV. CONCLUSION

Graph theory has been used in image segmentation providing good performances but with the limitation of being computationally demanding.

We propose a segmentation method based on multilayer graph theory that performs an accurate hand segmentation without being computationally expensive in terms of time and memory.

In order to test the segmentation accuracy, we use a synthetic database with multiple backgrounds.

The results obtained reveal that the performance of the proposed algorithm outperforms existing segmentation algorithms in the state of the art. We compare our proposed method with other graph-based algorithms, NCuts, and with a well known clustering method, KMeans. The results prove that the
A proposed method accuracy is better than NCuts and KMeans in all cases. Also the computational time study shows that the proposed method is more than 100 times faster than NCuts but just 10 times slower than KMeans.

As future work, we consider using different color spaces in order to improve the algorithm accuracy with a more complete evaluation on real databases.

V. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n°610713.

<table>
<thead>
<tr>
<th>Background</th>
<th>Proposed (F%)</th>
<th>NCuts (F%)</th>
<th>KMeans (F%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carpet</td>
<td>98.23 ± 0.01</td>
<td>81.92 ± 0.34</td>
<td>77.40 ± 0.90</td>
</tr>
<tr>
<td>FabricTextures</td>
<td>98.11 ± 0.01</td>
<td>89.68 ± 0.21</td>
<td>81.60 ± 1.30</td>
</tr>
<tr>
<td>Glass</td>
<td>97.85 ± 0.02</td>
<td>95.24 ± 0.14</td>
<td>95.14 ± 0.48</td>
</tr>
<tr>
<td>Grass</td>
<td>97.55 ± 0.03</td>
<td>97.72 ± 0.09</td>
<td>95.10 ± 0.58</td>
</tr>
<tr>
<td>Mud</td>
<td>97.87 ± 0.02</td>
<td>79.08 ± 0.35</td>
<td>84.46 ± 0.67</td>
</tr>
<tr>
<td>Objects</td>
<td>97.99 ± 0.01</td>
<td>78.40 ± 0.29</td>
<td>78.35 ± 0.95</td>
</tr>
<tr>
<td>Paper</td>
<td>98.39 ± 0.01</td>
<td>92.96 ± 0.28</td>
<td>91.08 ± 1.10</td>
</tr>
<tr>
<td>Parquet</td>
<td>98.27 ± 0.01</td>
<td>63.58 ± 0.30</td>
<td>84.20 ± 0.87</td>
</tr>
<tr>
<td>Pavement</td>
<td>98.12 ± 0.01</td>
<td>94.19 ± 0.15</td>
<td>89.35 ± 0.63</td>
</tr>
<tr>
<td>Plastic</td>
<td>98.19 ± 0.01</td>
<td>96.11 ± 0.12</td>
<td>94.31 ± 0.56</td>
</tr>
<tr>
<td>Skin &amp; Fur</td>
<td>98.29 ± 0.01</td>
<td>92.16 ± 0.19</td>
<td>85.62 ± 0.64</td>
</tr>
<tr>
<td>Sky</td>
<td>98.22 ± 0.01</td>
<td>85.44 ± 0.32</td>
<td>92.05 ± 0.67</td>
</tr>
<tr>
<td>Soil</td>
<td>98.26 ± 0.01</td>
<td>85.08 ± 0.20</td>
<td>85.96 ± 0.61</td>
</tr>
<tr>
<td>Stones</td>
<td>98.12 ± 0.01</td>
<td>87.67 ± 0.29</td>
<td>82.65 ± 1.51</td>
</tr>
<tr>
<td>Tiled</td>
<td>98.28 ± 0.01</td>
<td>72.47 ± 0.30</td>
<td>81.34 ± 1.16</td>
</tr>
<tr>
<td>Tree</td>
<td>98.05 ± 0.01</td>
<td>93.50 ± 0.20</td>
<td>92.97 ± 0.49</td>
</tr>
<tr>
<td>Wall</td>
<td>98.22 ± 0.01</td>
<td>81.15 ± 0.27</td>
<td>80.00 ± 0.67</td>
</tr>
<tr>
<td>Wood</td>
<td>98.18 ± 0.01</td>
<td>73.50 ± 0.28</td>
<td>88.68 ± 0.76</td>
</tr>
</tbody>
</table>

TABLE I

SEGMENTATION EVALUATION FOR DIFFERENT METHODS BY MEANS OF F-SCORE WITH 91 DIFFERENT BACKGROUND TEXTURES GROUPED INTO 18 CATEGORIES, TOGETHER WITH THE STANDARD ERROR OF THE MEAN.

REFERENCES

Appendix B. Multi-graph based segmentation algorithm applied to hand biometry


