Characterisation of Disordered Structures

A thesis submitted for the degree of
Doctor of Philosophy

by

Paul Butler BSc(Hons) AMInstP

School of Physical Sciences

University of Kent
Canterbury
U.K.

May 29, 2017

I dedicate this thesis to my parents

Declaration

This thesis has not been submitted as an exercise for a degree at any other university.

Except where stated, the work described therein was carried out by me alone.

I give permission for the Library to lend or copy this thesis upon request.

SIGNED: Paul Butler

Acknowledgements

I would like to start by thanking my supervisors Tomaso Aste and Paul Strange for their support, hard work, patience and encouragement over the years; they have somehow endured me with a smile! Their knowledge and experience have made this thesis possible.

I would also like to thank all the staff in the school of Physical sciences at the University of Kent. With a broad range of knowledge and friendly attitudes I always knew I had somewhere to get advice on some obscure reference I had never heard of!

It would be remissed if I did not also say a huge thank you to all of my friends in SPS for keeping me sane (or as sane as one can be) over my years in Canterbury.

Last and by no means least I would like to thank my parents for their encouragement, patience and support, not only over the time of this thesis but for all my years before.

Abstract

In this thesis I will look at how large, complex structures can be interpreted and evaluated using an information theoretic approach. The work specifically investigates techniques to understand disordered materials. It explains a novel framework using statistical methods to investigate structural information of very large data sets. This framework facilitates understanding of complex structures through the quantification of information and disorder. Large scale structures including granular media and amorphous atomic systems can also be processed. The need to deal with larger complex structures has been driven by new methods used to characterise amorphous materials, such as atomic scale tomography. In addition, computers are allowing for the creation of larger and larger data sets for researchers to analyse, requiring new techniques for storing and understanding information.

As it has become possible to analyse large complex systems there has been a corresponding increase in attempts to scientifically understand these systems. New, man-made, complex systems have emerged such as the stock market and on-line networks. This has boosted interest in their interpretation, with the hopes they can be more easily manipulated or controlled. Crystallography has been applied to great effect in biology, having been used to discover the structure of DNA and develop new drugs (UNESCO, 2013). However it only describes crystal structure, which can be a drawback as a large majority of matter is amorphous. As such it is hoped that interpreting and understanding disorder may lead to similar breakthroughs in disordered materials.

Entropic measures such as the mutual information and Kullback Leibler Divergence are used to investigate the nature of structural information and its impact on the system. I examine how this information propagates in a system, and how it could quantify the amount of organisation in a system that is structurally disordered. The methodology introduced in this thesis extracts useful information from large data sets to allow for a quantification of disorder. The calculated entropy for amorphous packings is generally less than 1 bit with Mutual information between 0 and 0.1 bits. The results verify direct correlation between Mutual Information and the correlation coefficient using various techniques. The Mutual information shows most information is obtained where sphere density is highest, following a similar trend to that of the Radial distribution function, and generally increasing for higher packing fractions. Evidence of the Random Close Packed (RCP) and Random Loose Packed (RLP) limits in two dimensions is shown, as well as evidence of both phases in time-lapsed 3D packings.

The Kullback Leibler Divergence is also explored as a relative measure of disorder. This is achieved by calculating redundant information in packings so that areas of low and high order can be shown. Results present colour maps displaying relative information in random disk packings from which motifs can be identified. For higher packing fractions distinct borders form for areas of low and high information, particularly where crystallisation has occurred. Again, these results show an increase in information for more densely packed structures, as expected, with a Kullback Leibler divergence of between 0 and 1 bits.

Finally I introduce the concept of self-referential order which provides a way to quantify structural organisation in non-crystalline materials, by referencing part of the system in a similar way to a unit cell. This allows a step forward in understanding and characterising disorder, helping to develop a framework to encode amorphous structures in an efficient way. These results show increasing information for higher packing fractions as well as further evidence of RLP and RCP limits around packing fractions of 0.54 and 0.64 respectively.

Contents

List of Tables

List of Figures

List of Acronyms

CPU	Central Processing Unit
FCC	Face-centred Cubic
GPU	Graphics Processing Unit
HCP	Hexagonal Close Packed
IID	Independent Identically Distributed
KC	Kolmogorov Complexity
KDE	Kernel Density Estimator
KLD	Kullback-Leibler Divergence
LSA	Lubachevsky-Stillinger Algorithm
MD	Molecular Dynamics
MI	Mutual Information
MT	Minkowski Tensor
MVG	Multivariate Gaussian
PBC	Periodic Boundary Condition
RCP	Random Close Packed
RDF	Radial Distribution Function
RLP	Random Loose Packed
RSA	Random Sequential Addition
SRO	Self-Referential Order

Chapter 1

Introduction

The aim of this thesis is to investigate methods to quantify structural disorder using a statistical approach. Understanding disordered structures falls into the area of complexity theory and relies upon knowledge in information theory and statistics. While research into complex systems is relatively new, there is a great amount of interest in their study (Wang & Chen, 2003; Newman, 2003).

Complex systems occur throughout nature and man-made systems, from worldwide networks on-line to the microscopic cells that make up all life on earth. Understanding these systems is key to describing the world around us. The relatively well understood crystalline structures that make up around $\sim 30\%$ of all matter only provide part of the picture. The majority (around $\sim 70\%$) are amorphous materials that are constructed by atoms in a disordered arrangement. This disordered state is not well understood with some debate around its very definition. Even the term disorder can be thought of as negative, being a disturbance of order. This definition of order comes from the perspective of crystalline structures. This is not surprising considering the progress that has been made in the field of crystallography (Knight, 2008). X-ray diffraction has been the main tool for studying atomic structures since its creation at the start of the 1900's. Unfortunately this only allows for study of the average relative positions of atoms (Butler et al., 2013). This drawback puts a natural emphasis on crystal structures, leaving amorphous structures ill-understood. Structure became described by its relation to this 'perfect' form of matter.

Despite a focus on crystalline structures, amorphous materials have been shown to have their own forms of organisation. These forms should be better understood to take full advantage of the fabrication of new functional materials. It has proven difficult to encode this information because, in absence of a compact way to encode structural complexity, the processing of this amount of information is still beyond the capability of the world's largest supercomputers. Remarkably, it should be noted that there is not the digital storage capacity to hold all the information from even one gram of matter. This is because all the hard disks and other digital media storage would 'only' store a currently estimated 10^{20} bits (Baez, 2015).

It has been shown that 'order' in amorphous structures can be identified by looking at motifs that are more common or descriptive than others, and thus encodes more information about the system (Kurchan & Levine, 2011). This approach reveals diverging correlation lengths at glass transition (Sausset & Levine, 2011) providing insight on the relations between thermal glass transition and athermal jamming of discrete matter (Biroli & Garrahan, 2013). The method itself can be compared to simplifying data in a crystalline structure to the unit cell. This leads to the concept of finding motifs in a structure that best describes the whole system. With a set of motifs and assembly rules, a shape-filling tessellation can be formed. In this process, a compression in the amount of information required to describe the system can be achieved.

Given the development of new technology such as atomic scale tomography, data sets from amorphous materials could soon be studied on a large scale. Such research would require new techniques to identify structure and statistics in these materials. My thesis focuses on developing a broad framework for dealing with disordered structures, and other complex systems, by incorporating an information theoretic approach to the problem. The general case of hard sphere packings is taken as an example of how this could be achieved allowing for analogy with atomic structures (Frank & Kasper, 1958).

Additionally, this encourages overlapping techniques in the field of packing problems (including the areas of granular and foam materials). These benefits create a dynamic framework which can be applied to such problems.

Amorphous and granular structures can be considered analogous, particularly when looking at their statistics. This is an important connection as such research is well established and gives a basis for developing a broad framework, and dealing with granular materials experimentally is much easier than trying to determine the structure of an amorphous material. In recent years X-ray CT has been used to characterise the structure of 3D random packings of beads (Aste et al., 2007). Such experimentally generated data will be looked at in the final chapters of this thesis. Granular systems have also been likened to foam materials and importantly for this research, atomic packings (Frank & Kasper, 1958) (Biroli & Garrahan, 2013).

The thesis has taken techniques used and developed in a number of scientific fields such as Biology, Physics, Mathematics and Computer Science and hopes to develop a statistical framework for dealing with a large number of problems found across many scientific fields.

Chapter 2

Background

2.1 Packing Problems

Packing problems have been pondered for hundreds of years One can imagine why someone transporting goods would probably want to fit them into the smallest space possible, maximising the amount they could carry, while ancient merchants might try to fill a bag with the minimum amount of produce to maximise profits.

Scientific exploration of packing problems didn't really kick off until much later in our civilisation, towards the end of the renaissance in the 17th century. While many people were involved, it is perhaps the work of Johannes Kepler (figure 2.1) that is most celebrated today. Best known for his laws of planetary motion, Kepler did in fact devote time to studying packing problems. This led to the well known Kepler conjecture published in "On the six-cornered snowflake" in 1611, where he theorises on why snowflakes have hexagonal symmetry (Ball, 2011).

2.1.1 The Kepler Conjecture

The Kepler conjecture states (in his words) the packing obtained by a hexagonal close packed structure "will be the tightest possible, so that in no other arrangement could more pellets be stuffed into the same container".

In short, it describes the most efficient packing possible, whereby, the packing fraction is a maximum (Aste & Weaire, 2000). The packing he proposed was the hexagonal close packed structure (figure 2.3). This gives a packing fraction, the ratio of filled space to all the space in the system, of 0.74, or in other words, leaving 26% of the space empty. The packing fraction is denoted with Φ, and is the sum of the volume of the particles in the system divided by the complete volume of that system.

Figure 2.1: Johannes Kepler (1571-1630)

It would be unfair not to mention the mathematician and Oxford graduate, Thomas Harriot, who, some years previous, in 1606, had been asked by explorer Walter Raleigh the most efficient way to stack cannonballs upon his ship's deck. Naturally as a mathematician he reduced this to a more generic problem, the close packing of equally sized spheres. This had the added bonus of relating it to the discussion of atomic theory, more than two hundred years before it was to be proved. Through the work of Kepler and Rene-Just Hauy it formed some of the framework for crystallography (Ball, 2011; Kunz, 1918). Harriot was most certainly a big influence on Kepler's work on packing problems having discussed his ideas at length in correspondence, which ultimately lead to the formation of the Kepler conjecture.

2.1.1.1 Solving the Kepler Conjecture

While it seemed at first like quite a simple conjecture, proving it was a lot harder, even appearing in the famous 1900 Hilbert list of unsolved mathematical problems. Many scientists devoted not inconsiderable amounts of time in an attempt to find a rigorous mathematical proof. Real progress was made when Carl Friedrich Gauss, 200 years after Kepler's work, managed to prove the conjecture true (well not quite). By placing kissing spheres on a lattice, the conjecture simplifies to an optimisation problem for which the solution can easily be seen as the face centre cubic (FCC) lattice. However, this is obviously not a proof for all cases, and only holds up for any regular lattice arrangement, as it discounts the possibility of the most efficient packing being disordered.

Axel Thue (Born 1863) reduced the problem to two dimensions, much as Joseph Louis Lagrange had done (but like Gauss only proving it for a lattice) in 1773. Thue used triangulation and optimisation showing a hexagon inscribed by a circle, in which no other circles could appear without displacing another. The largest density can then be shown to be the area of the circle divided by the area of the hexagon or

$$\frac{\pi r^2}{2\sqrt{3}r^2} = \frac{\pi}{2\sqrt{3}} = 0.9069 \tag{2.1}$$

where r is the radius of the circle. It should be stated that other more modern proofs exist using Delaunay triangulation and density analysis for dense hard sphere packings (Chang & Wang, 2010).

While it is argued whether or not Thue's argument constitutes a proof, it leads on to later work in three dimensions using the same principles, where the more formal Voronoï analysis is used (see section 4.1.2), with truncation of the Voronoï cells the optimisation parameter (University of Pittsburgh, 2001). This gives the solution as a dodecahedron, for which the packing fraction is

$$\Phi = \frac{\frac{4}{3}\pi r^3}{\frac{1}{4}(15+7\sqrt{5})a^3} \approx 0.755 \tag{2.2}$$

where r is the radius of the circle, a is the edge length and by trigonometry $r = 1.1135a$ This is interesting as, for comparison, the packing fraction for FCC is $\Phi = 0.740$, the same as for hexagonal close packed lattice (HCP). These are the highest packing fractions that have been observed in nature. So was this it, a theoretical maximum that

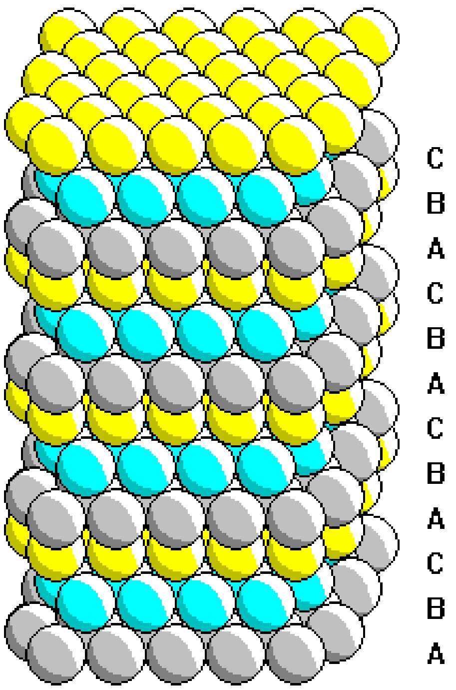


Figure 2.2: Example of an FCC Packing

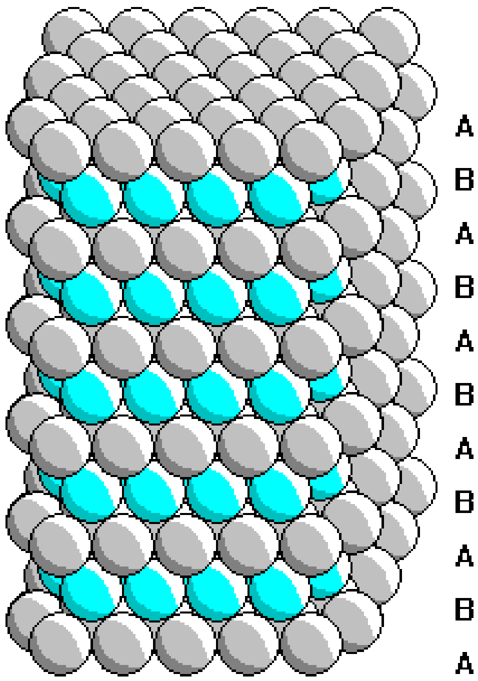


Figure 2.3: Example of an HCP Packing

had just not been observed? Well probably not, as the dodecahedron does not tile in 3D, and so the local maximum Φ is not equatable to the global maximum Φ, therefore some correction term must be used to take this into account. Fejes Toth made an important step to a solution in 1953 by showing the problem could be solved as a minimisation of a function with a finite number of variables, by taking into account the relative position and volume of the various cells (Fejes, 1964). While this proved promising for a solution to the Kepler conjecture, the calculation was far too computationally demanding to be completed. While simplifications of this problem allowed bounds to be set, then tightened as computers became more and more powerful, a complete solution was not made until quite recently.

In 1998, Thomas Hales claimed to have finally solved the problem by using a hybrid Delaunay and Voronoï decomposition approach and building greatly on the work of Toth, combining work in optimisation, linear programming and interval arithmetic (Hales, 2005). While it is generally accepted to be correct, it is an extremely complex proof and in 2003 Hale began work on the "Flyspeck project" or "Formal proof of Kepler", which involves the use of computers to automatically verify the proof but is still an ongoing work (Weisstein, 2014).

2.1.2 Foams

While sphere packing was obviously important for Greengrocers and Admirals alike, a study of spheres alone is not a complete picture, especially if one is a librarian stacking books or instead of oranges the Greengrocer has cake. Research on how poly disperse shapes fit together was perhaps best done experimentally by using foams and, has such, been an area of great interest. A foam is composed mostly of air with liquid interfaces forming a shape filling arrangement of polyhedra (see figure 2.5), and are very common, ranging from our evening bubble bath to the fabric of the universe (like the hypothetical "quantum foam" (Wheeler, 1998)).

The earliest scientific research again goes back to the renaissance, with work looking at simple foams done by Leonardo da Vinci and Robert Boyle (Weaire et al., 2007). Later on, Gottfried Leibniz, laid the foundation of topology, allowing a mathematical basis for the study of structures composed of individual cells (von Leibniz, 1976). Another well known name who worked on packing problems was Lord Kelvin (figure 2.4), who in 1887 asked, what structure, with similar cells of equal volume gives the smallest surface area. This can more easily be thought as cubes. The more cubes are stacked, the larger the surface area becomes disproportionately to the increase in volume. Kelvin proposed a structure to minimise this area, a truncated tetrakaidecahedron (14-sized polyhedron)(Lord Kelvin, 1887). To ensure it conformed to a space filling foam, the hexagonal faces needed to be rounded, as a consequence of Plateau's law for soap films, which define all stable foams (Morgan, 1994).

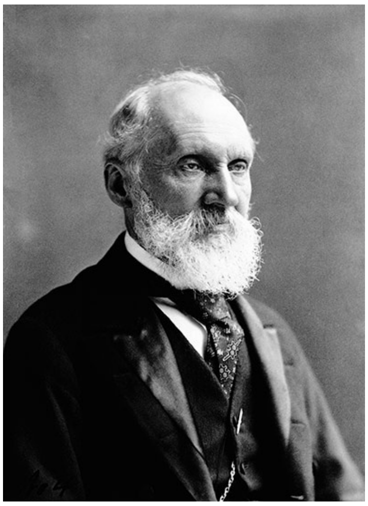


Figure 2.4: Lord Kelvin (1824-1907)

It was only recently this structure was succeeded by the Weaire-Phelan structure (figure 2.6), which is composed of two different cells. The first is an irregular dodecahedron with pentagonal faces, possessing tetrahedral symmetry. The other being a tetrakaidecahedron, with two hexagonal and twelve pentagonal faces, possessing antiprismatic symmetry (Weaire & Phelan, 1994). This structure was used in the design process of the Beijing National Aquatics Centre, built for the 2008 Olympics. The engineering of such shaped structures, due to the minimised surface area, would require less material compared to a similarly sized building, and yet still be a robust and sturdy structure. Other recent examples of foam research are found in Biology, including the study of structures in nature such as beehives (section A.1) in which topology plays an important role in function. These studies allow insight for the development of optimal structures in human engineering (Park & Han, 2013). Intriguingly, cell topology has also been shown to be important, and the use of foam research have allowed insights into how

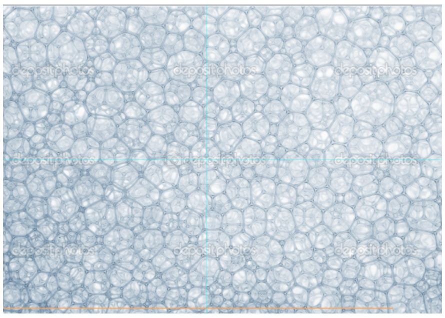


Figure 2.5: A typical foam, comprising air sacs with liquid interfaces

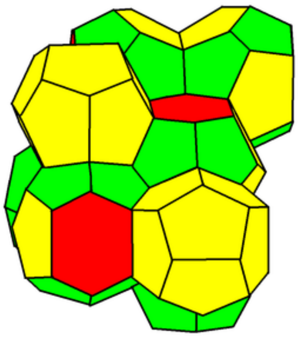


Figure 2.6: The Weaire-Phelan structure

larger cellular structures evolve (Gibson et al., 2011), how tumours may develop and more importantly, why.

2.1.3 Granular Materials

Granular materials became of greater interest due to their similarity to the hard sphere packings described earlier. They are examples of packings of hard non-overlapping entities. However, they are subject to physical forces such as gravity and friction. The easiest examples to think of would be sand on a beach, or sugar in a bowl (I prefer thinking of the latter). Some examples are shown in figures (2.7) and (2.8). Granular materials are of interest in a wide range of fields, from shock waves and explosives to tectonics and other geophysical phenomena. They are also dealt with in large quantities in industry, from salt mines to sugar factories, so understanding their behaviour is of particular importance.

Again a number of well known scientists played at least some part in the story. Charles-Augustin de Coulomb's work on friction was partly based on granular systems (Coulomb, 1821; Rodhes, 1997). Osborne Reynolds did extensive work on the subject, even going so far as saying "it is shown that there is one, and only one conceivable purely mechanical system capable of accounting for all the physical evidence, as we know it, in the Universe" (Reynolds, 1903) in reference to the 'granular' structure of the fabric of the

Figure 2.7: Several examples of everyday granular media

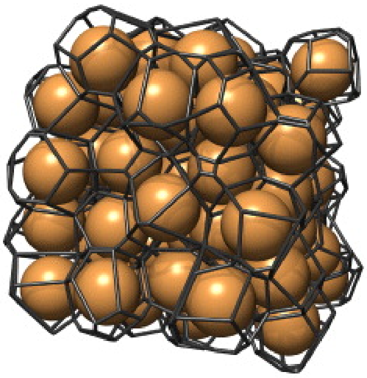


Figure 2.8: A jammed and disordered packing of mono-disperse hard spheres

universe (i.e. the aether). He was known for eye-catching experiments using granular systems, such as filling a container with sand and water and observing it go rigid. This was to show the principle of 'dilatancy' where a rigid granular material must dilate in order to deform (Aste & Weaire, 2000). This also allows us to start forming a concept of jamming, where the granular material becomes rigid as the sand particles can no longer move freely.

2.1.3.1 Jammed Packings

Packings of particles can be considered in a 'jammed' state when the particles no longer have enough freedom to displace the others around it, and so the relative position of the particles becomes fixed, leading to the whole structure becoming rigid. This typically occurs as a system is driven towards increasing density or higher packing fractions. Local jamming may also occur when one or more particles become trapped, but others around it can still be displaced. The density and configuration a packing jams in is highly dependent on the environment, the main ones being:

- The energy in the system, and how equilibrium is achieved, which can have a drastic impact on the arrangement
- Forces applied externally (i.e shearing, straining) can cause or prevent jamming

(Kumar & Luding, 2014)

- The shape of the object (wonderfully portrayed in the M&M's experiment by Paul Chaikin (Donev et al., 2004)) determines its properties
- The poly-dispersity of the objects, where each particle can have a different size. Both the ratio of the sizes, and the relative numbers of particles for each size, are important
- The shape and size of the container; boundaries in particular can encourage certain specific arrangements (Aste et al., 2004)

Current research is ongoing on how these sort of criteria affect packing, and is quite wide ranging. Examples of how spheres pack in cylinders may give insights into arrangements of particles in channels (Chan, 2011). Some research has been done on adjusting aspect ratios of packed spheres (Donev et al., 2004), creating packings of ellipsoids on lattices, while others have created spiky particles, by adding ellipsoids to spheres to explore packings of non-convex particles (Malinouskaya et al., 2009). It is clear from these studies that altering the shape of the particles drastically changes their packing characteristics.

What is still not clear is whether the jamming of amorphous structures, whereby they effectively become a solid, actually corresponds to a phase transition. For example jamming occurs in glassy systems, however, the glass transition is not considered a phase change as it is a gradual and ill-defined change over a range of temperatures. By comparison first-order phase transitions are marked by rapid changes in the structure and properties of a material, which are far easier to define in the well understood crystalline phase. It is important to note that crystalline structures are only exhibited in a small number of the materials around us.

While there are certainly differences, it can be argued the lack of consensus on the issue is simply a lack of understanding of the amorphous phase, and much effort has been placed on better understanding the jammed state. While much modern work is done using computer simulations, many experiments have been carried out to verify their accuracy, for example diffusing wave spectroscopy of granular systems (Kim & Pak, 2010) and the computerised x-ray tomography (CT) of foam balls (Aste et al., 2004).

Further research has provided insight into granular systems, for example spacial heterogeneity, where predictable structures arise in a granular media such as bridges (chains of particles supporting each other), as well as various characteristic structures (Edwards & Oakeshott, 1989a). Other researchers have used a myriad of variables, such as the compactivity (Edwards & Oakeshott, 1989b) and anisotropy (Schrder-Turk et al., 2013). While it is clear from this research that there are changes happening at the jamming limit, it has been difficult to locate an "abrupt change" that one might wish to see for a clear phase transition.

2.1.3.2 The Jamming Limit

The jamming limit is the point at which the packing fraction cannot be driven higher before crystallisation must occur. There is also a lower limit, below which the packing is never dense enough to jam. By extension the jamming limits show the bounds of Φ for which jammed amorphous packings can occur. Investigating these limits has been of particular interest in the field of packing problems.

Auguste Bravais (Bravais, 1949), showed crystal phases can be characterised as a lattice. This restricted the points at which particles could be placed and reduced the number of configurations considerably. Unfortunately for the densest random packings we still have many possible configurations, meaning the problem cannot be simplified as was done for the Kepler conjecture.

Perhaps the most extensive work was initially undertaken by Desmond Bernal (figure 2.9) who created large models, painstakingly assembled by hand (shown in figure 2.10). Despite the perhaps, 'crude nature', by today's standards at least, of the experiments he did show for mono-disperse spheres the random packing fraction did not exceed ≈ 0.64, now known as the random close packed limit (RCP). Beyond this limit crystallisation must occur, and is an intermediary phase between the RCP limit and an HCP structure. Below this limit local crystallisation may occur, but is never certain, as configurations will always exist without it.

Bernal also found a lower bound for jammed states, known as the random loose packing limit (RLP) showing $\Phi \approx 0.55$. However, it is difficult to achieve jamming at the RLP limit, requiring constant pressure. Therefore external forces such as shearing and gravity must be eliminated (Aste & Weaire, 2000). These forces are large compared to the effects of temperature, which has little impact due to the macroscopic nature of the particles. This minimal change in energy means the particles are essentially athermal (Mehta, 2010). These limits also change depending on the shape of the particles as previously seen (Donev et al., 2004).

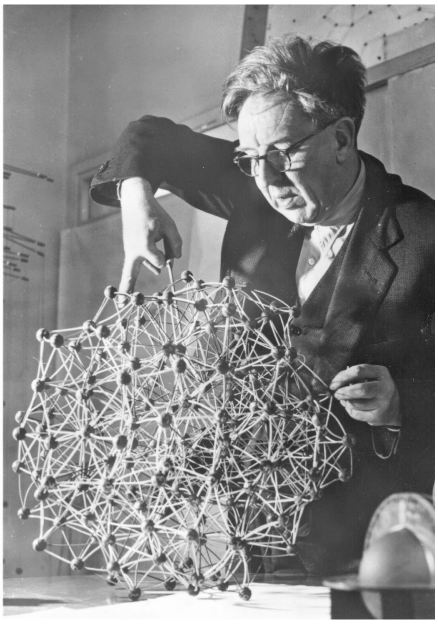


Figure 2.9: Desmond Bernal working on one of his large scale models of amorphous structures

Figure 2.10: Model showing random close-packing by built by Bernal by hand to further understand packing problems

It is clear these materials are complex, sometimes exhibiting liquid and solid characteristics, and certainly their configuration is highly dependent on initial conditions, along with what forces are used to drive a system to equilibrium, and how fast. But it is important to note that strong links have been suggested between hard sphere packing, foams, granular materials, (Weaire et al., 2007) and atomic systems (Frank & Kasper, 1958), and so developing a statistical framework for dealing with one can afford us tools to understand the others.

In 2D the packing limits are less well defined. The maximum packing fraction for any 2D structure is the hexagonal packed structure with $\Phi \approx 0.92$, however the RLP limit is unconfirmed, with little evidence it even exists (Meyer et al., 2005). The RCP, while not as well understood as for the 3D case, is known to be $\Phi \approx 0.82$ (Meyer et al., 2005).

2.2 Self Assembly

Self assembly is where a group of building blocks assemble themselves into some pattern based on matching rules, for example, in molecular scale physics the blocks are molecules and the rules are electromagnetic forces. There are many other examples such as DNA and other protein complexes in biology. Depending on the scale these rules can simply be topological (like a jig saw puzzle), but others can be driven by more complex quantities (such as energy). The number of blocks and rules can be related to complexity, and searching for these 'motifs' can help us describe the system, potentially reducing the amount of information needed to describe the system. The perfect example is that of a unit cell describing an entire crystalline structure. Self assembly takes place on many scales from nanoparticles to entire galaxies (Krasnogor et al., 2011). It is also important that self assembly is only considered to have occurred when order is gained in the assembled structure (Ahnert et al., 2010). Its main areas of application have been within chemistry and biology.

2.2.1 Self Assembly in Disordered Structures

The thesis started by looking at the idea of self assembly which based a measure of structural disorder on the amount of information required to build the system, built by a set of self-assembly rules. This approach started by dividing a system into building blocks and 'colours'. Colours are the different types of interfaces to which the assembly rules would apply for example colour 1 attaches to colour 3. A set of colours and rules are called an assembly kit. From these colours an interaction matrix could be created which contained the rules (Ahnert et al., 2010). Such a technique was presented in the paper *"Self-assembly, modularity and physical complexity"* (Ahnert et al., 2010) (figure 2.11) and was based upon ideas of self-assembly (Rothemund & Winfree, 2000) and molecular biology (Adleman, 1994). This research allowed the authors to assign a value to the disorder of their own structures (i.e. cell proteins). I realised that such a method could be used to quantify disorder using a similar methodology outlined below.

The method applied to disordered structures could follow a plan like this:

- Divide structure into building blocks, which would require finding a motif or motifs that describe the system, and then defining the building blocks;
- Create a contact graph, showing the building blocks and what structures they are in contact with;
- Characterise these contacts to create building rules, describing how these blocks link together;
- Attempt to reduce the complexity by looking at subgroups of the main graphs, essentially scanning at different resolutions;
- Quantify disorder by reducing the information required to a minimum.

In this case the idea can be formalised in relation to the Kolmogorov complexity (discussed in section 2.4.1) of structure A as:

$$K(A) = I(\tilde{S}_A) = \min_{S_A} I(S_A) \tag{2.3}$$

where $I(S_A)$ is the information needed to encode the assembly kit S of structure A and $\tilde{S}_A$ is the minimum assembly kit (Ahnert et al., 2010).

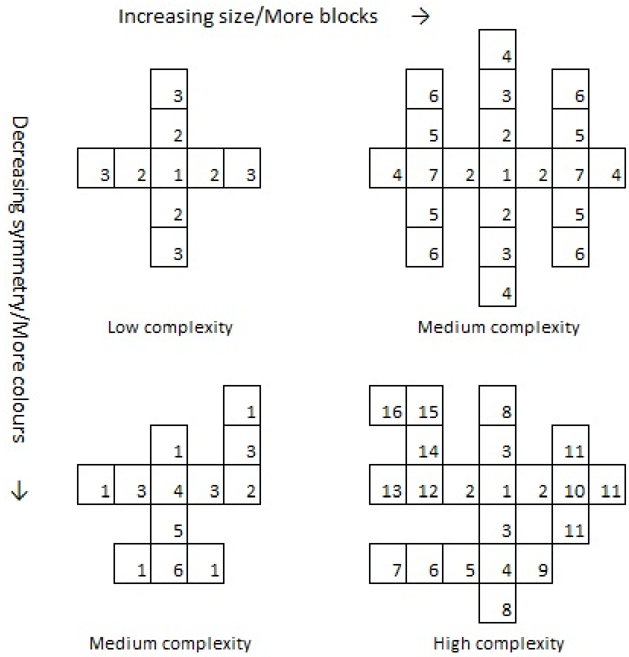


Figure 2.11: An image created to show the concept of modularity and complexity. As the number of colours and rules increases, so does the complexity.

However, for disordered structures it is difficult to divide the structure, as one would either have the number of blocks or the number of colours tending to the number of objects and no information would be saved. Despite this I decided it would be possible to use the amount of information required to store a structure as a basis to measure disorder. Modifying the use of an interaction matrix to one of probabilities could provide insight, based on statistical information gathered from topological techniques. This technique became the foundation of my thesis. While no piece could offer enough information to encode an entire system, it may be possible that some could describe the system better than others.

2.3 Probability Theory

Before we go on to talk about Complexity and Information Theory it is useful to define a few quantities and rules from probability theory.

2.3.1 Probabilities

Probability measures the chance of an event occurring and can be mathematically described as

$$p(X) = \frac{\text{number of occurrences of specific event}}{\text{Total number of events}}$$

let's take the example of a coin flip

$$p(X = heads) = \frac{\text{number of times heads occurs}}{\text{Total number of coin flips}}$$

for one flip $p(X) = 0$ or 1

but, as most people know, the probability should be 0.5, so as the number of events N becomes large, $\lim_{N\to\infty} p(X) \to 0.5$

This leads to the concept of convergence, that a large enough data set is needed to find a close approximation to the real probability distribution function.

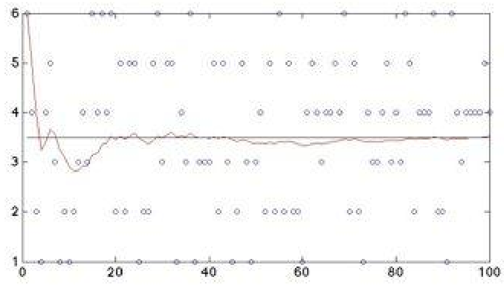


Figure 2.12: Values of a fair six sided die, up to 100 rolls, overlapped with the expectation value given N rolls

As an example take a six-sided dice, you must roll either a 1,2,3,4,5 or 6. The probability of all the possible events summed together must be one, in other words it is certain at least one of the possibilities will occur. In the same way any probability must be between zero and one, as you can not be more than certain something will occur than certain, and analogously, you can be no more certain something will not occur than knowing it will not occur. Formally we can write this as

1. $0 \leq p(x) \leq 1$ and
2. $\sum_{n=1}^{N} p(x) = 1$ or in the continuous case $\int_{-\infty}^{\infty} f(x)\mathrm{d}x = 1$

In the case of a continuous function we must bin our data to compute the probability

$$p(X \in A) = P(a < X < b) = \int_a^b f(x)\mathrm{d}x \tag{2.4}$$

where A is an interval defined between some two values a and b, and $f(x)$ is a continuous function. Here $f(x)$ must also be a probability density function, meaning it follows the same rules as any probability and the integral over the whole function must equal one. In this thesis most data sets have been normalised to achieve this by adding a factor of $\frac{1}{\int_{-\infty}^{\infty} f(x)}$

2.3.2 Conditional Probabilities

Data was examined using several different statistical techniques, and for this reason they will be presented in the results sections. However conditional entropy is an important quantity for many techniques in information theory. Therefore some general points will be made here, beginning by defining some additional terms.

The joint probability is the chance of finding two variables with specific values. Let us take the example of two Bernoulli variables where $x, y \in \{0, 1\}$ if the probability of being either zero or one is equal for both x and y, then there are four equally likely values, meaning $p(x, y) = 0.25$. So

$$p(x, y) = p(x = X \ \& \ y = Y) \tag{2.5}$$

Let us assume x is always 1, then $p(x = 0, y) = 0$, regardless of the value of y. From the above example it can be obtained for $x = 1$ the $p(x = 1, y = 0) = p(y = 0) = 0.5$ as for $y = 1$. These values are typically obtained through a two fold matrix of frequency values, such as the co-occurrence matrix.

The second quantity is the conditional probability $p(x|y)$, this measures the probability of an event x given some other event y. Conditional probability is quite useful when discussing dependency. For example, let us take two independent, identically distributed (iid) variables. As the two events are independent, no information about the second event is gained from the first and $p(x|y) = p(x)$, proven by simple application of Bayes theorem:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{2.6}$$

where $p(y) \neq 0$. Using Bayes theorem it is also possible to show that $p(x|y) \neq p(y|x)$, which is important when thinking about false positives.

2.3.3 Expected Values

Let us take a discrete random variable x. We now define its expectation value as $E(x)$, or the mean value. Generically this is described as

$$E(x) = \sum_{i=1}^{I} x_i p_i \tag{2.7}$$

where p_i is the probability of outcome i. Obviously this is for a discrete case, in the continuous case one would integrate over the values by using the probability density function $f(x)$ as shown in equation (2.8).

$$E(x) = \int_{-\infty}^{\infty} x f(x) dx \tag{2.8}$$

Notice in figure (2.12) that it takes around twenty rolls for the expectation value to converge to its true value of 3.5, this is also interesting as you can not roll a 3.5! This shows the principle of the law of large numbers, where the expected value tends to the average, as N, the number of events, becomes very large. In terms of the probability this can be written as:

$$p(x) = \lim_{N\to\infty} \frac{N(x)}{N} \tag{2.9}$$

Here $N(x)$ is the frequency for a given event x. This directly relates the probability as the relative frequency of an event.

The expected value allows us to calculate deviations in the data, helps us to form a model of the data, and provides tools to quantify uncertainty. Let us again take the example of a discrete variable x which has been modelled by a function $f(x)$. If the data is well defined by $f(x)$, less information is required to express the data. In the perfect case one equation could be used to replace a large number of values, reducing information to a minimum. The first four central moments are most useful to define in this case, the first is the centralised expected value, which must be zero. For clarity μ_k will be used for the kth centralised moment and μ_x will be used for the mean. The standard deviation and 2nd, 3rd and 4th central moments are listed below respectively;

- The Standard Deviation $\sigma = \sqrt{E[(x-\mu_x)^2]}$ (so the average difference a value has from the mean μ), expanding the bracket then gives rise to $\sigma = \sqrt{E(x^2) - \mu_x^2}$
- The Variance is then simply $\mu_2 = \sigma^2 = \langle x^2 \rangle - \mu_x^2$. These give us a quantifiable way to measure the uncertainty and have been used to understand complex systems with popular applications including Chebyshev's Inequality
- The Skewness $\mu^3 = \langle x^3 \rangle - \mu_x^3$ is a measure of how weighted a function is to one side of the mean, in other words is zero for a completely symmetric function, and can be positive or negative depending on the direction of the skew
- The Kurtosis $\mu^4 = \langle x^4 \rangle - \mu_x^4$ measures how unevenly spread the data is, this makes it sensitive to the width and height of the peak/s and tails and high Kurtosis often

suggest complex behaviour[1]

- From these first four important moments it can be seen that for the k^{th} central moment is $\sigma^k = \langle x^k \rangle - \mu_x^k$

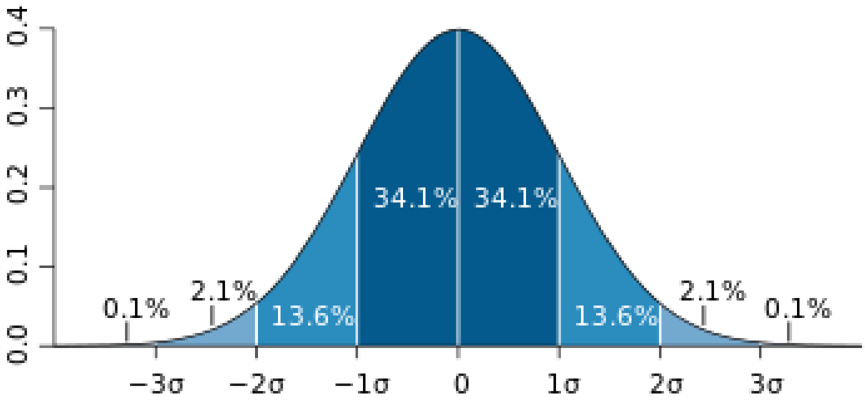


Figure 2.13: Gaussian distribution with % number of values for a given $n\sigma$ away from the mean

2.3.4 The Normal Distribution

If states of the systems are taken as variables, we can begin to build a statistical picture to describe them. This is useful for when we want to define complex systems, at least from the macroscopic perspective. One of several distributions can be used to describe most data, but the most common is the aptly named Normal distribution (figure 2.13) or Gaussian distribution:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu_x)^2}{2\sigma^2}) \tag{2.10}$$

where σ^2 is the variance $= \langle [(x-\mu_x)^2] \rangle$ and μ_x is the mean value of x. The bi-variate form is plotted in figure (2.14) and stated below:

$$p(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp(-\frac{1}{2(1-\rho^2)}[\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}]) \tag{2.11}$$

[1]It is important to note that the normalised $4th$ moment $\tilde{\mu}_4 = \frac{\mu_4}{\sigma_4}$ is more commonly used as the Kurtosis, and was the formula used in this work.

where ρ is the Pearson correlation coefficient

$$\rho = \frac{\langle[(x-\mu_x)(y-\mu_y)]\rangle}{\sigma_x\sigma_y} \tag{2.12}$$

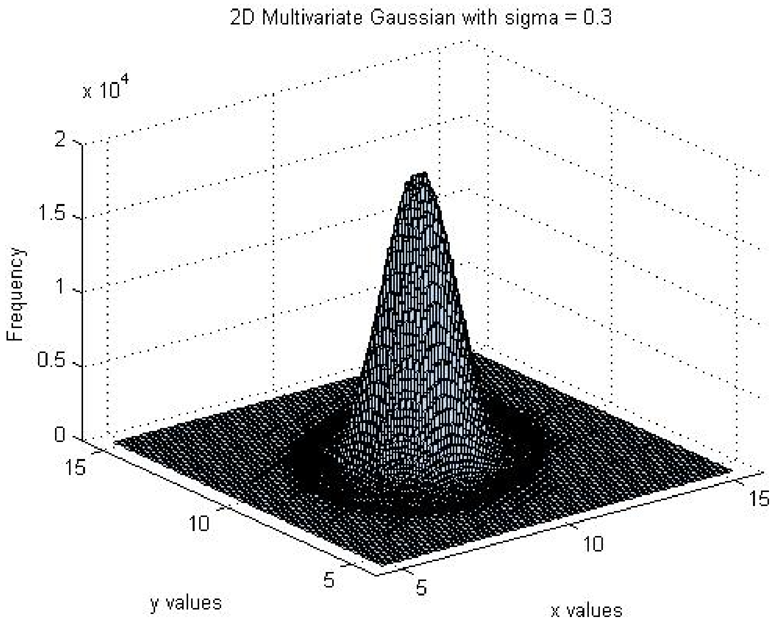


Figure 2.14: An example of a bi-variate Gaussian with a relatively low σ of 0.3

I have stated these quantities implicitly as they are all used later in the text, but, for completeness here I will also include the generalised form of the multi-variate Gaussian which is expressed as:

$$p(x_1, ..., x_n) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\bar{x}-\bar{\mu})^T\Sigma^{-1}(\bar{x}-\bar{\mu})) \tag{2.13}$$

where $\bar{x}$ is a vector containing $(x_1, ..., x_n)$, $\bar{\mu}$ is the vector containing $(\mu_{x_1}, ..., \mu_{x_n})$ and Σ is the covariance matrix and $|\Sigma|$ is its determinate. Σ takes on the size n by n, being the covariance between all elements of $\bar{x}$, so that $\Sigma_{ij} = E[(x_i-\mu_i)(x_j-\mu_j)]$. Some complex systems show statistics that follow a normal distribution except in the tails, making extreme events far more likely. These functions are called fat-tailed distributions and will be talked about a little later, as it is important to understand why these functions tend to behave ’normally’ first.

2.3.4.1 The Central Limit Theorem

The Central limit theorem states the sum of independent, identically distributed (i.i.d.) variables will tend to be normally distributed (Feller, 1945) (figure 2.15), when the variance is finite and each variable is not dependent on any other. Let us again take the example of dice, which individually have a uniform distribution. By taking additional dice as new variables, it can be shown the sum quickly approaches a normal distribution.

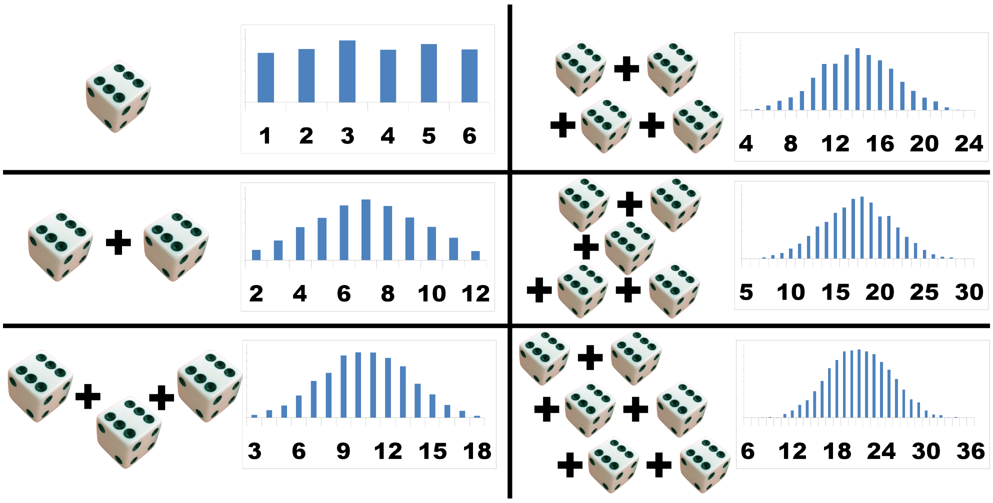


Figure 2.15: An image showing the central limit theorem, using dice as variables, the graphs show sum total score vs frequency

A number of formal proofs have been made of the Theorem including the use of Taylor expansion, cumulants and moments (Filmus, 2010). Since the proof is quite long, I will simply outline the proof made by (Weisstein, 2015). This involves taking the inverse Fourier transform of some arbitrary probability function for i.i.d. points. Allowing the series of points to be substituted in, after much rearrangement and expansion, the Fourier transform can be taken restoring a probability function for the summed variables. After further substitution an answer is obtained in the form of the normal distribution.

The Berry Esseen theorem (Berry, 1941) then states how quickly these i.i.d. variables converge: assuming a zero mean and positive variance, the convergence is $\frac{1}{\sqrt{N}}$

2.3.5 Correlation Techniques

It is also important to introduce some methods of correlation, which shows how related data is to itself, or other sets of data. For example, take 500 points x_i, which would be completely correlated if all produced from the equation $y = 2x + 5$. Equally two completely random numbers should have no correlation. Many sets of data, in reality, lie between these two extremes.

2.3.5.1 Pearson Correlation Coefficient

The Pearson Correlation Coefficient has already been introduced in equation (2.12). It is an important quantity which, as the name suggests, measures correlation. While it is only sensitive to linear correlation, and is not as general as Mutual Information, it is an absolute quantity, with zero being no correlation, and $\rho = 1$ and -1 being completely correlated or anti-correlated respectively. It is in-variate, only being sensitive to how the data is related. This makes it a useful tool as it assumes no apriori information.

2.3.5.2 Co-variance

Many fitting measures use the co-variance, *cov*, to measure the correlation between two sets of data. It is defined by the equation

$$cov(x, y) = \sum_{i=1}^{N} \frac{(x_i - \mu_x)(y_i - \mu_y)}{N} \tag{2.14}$$

This gives a value of zero for uncorrelated variables.

A number of fitting algorithms have built in correlation methods to either improve or confirm a fit. The most common examples of these include linear regression and least squares methods. They do require some assumption that the data are correlated, measuring the difference away from the mean. This is usually done by calculating the vertical offsets leading to a regression coefficient β based on x:

$$\beta = \frac{cov(x, y)}{\sigma_x^2} \tag{2.15}$$

which in terms of fitting is used in the optimisation solution. Mutual Information is given special attention in the methodology, and will only be mentioned here as an entropy based measure of correlation, that calculates shared information between two sets of variables.

2.4 Information Theory

Information theory is, as the name might suggest, the study of information. It aims to understand and quantify information, including its storage, how it's transmitted and received, as well as how we measure uncertainty and disorder. This is particularly important as it shows that the more complex and disordered the data is, the greater the amount of information needed to store it. While information theory is arguably most widely used in the computer sciences today, study is rooted in mathematics, with important applications in statistical physics, biology and even linguistics, as well as an assortment of others (Mezard & Montanari, 2009).

2.4.1 Data Compression and Kolmogorov Complexity

2.4.1.1 Data Compression

Data compression is important to understanding disorder, as the more compressible the information describing an object, the less disorder is associated to it; again an example can be made from a crystalline structure, where the data can be compressed down from describing every atom in a large structure, to a simple unit cell.

Many data compression techniques assign smaller quantities to the most common features, such as Huffman coding (Huffman, 1952). This is a form of lossless coding, meaning no information is lost in compression, and a perfect copy of the original can be recovered from the compressed form. These forms of compression involve a 'key', in which common elements are described by the shortest possible string. Let us take the example of this thesis, if I made the word $disorder = d$. I now save myself seven letters everytime I need to write $disorder$. Over the dozens of times it is mentioned in my thesis, I will have saved myself a few bits of information. This, however, requires the information to encode the 'key' to be taken into account. If we took a more outrageous example and said 'all the information in the universe' $= U$, the information to store U is now tiny, but of course to map the output, one needs to store 'all the information in the universe'.

A number of factors effect compression, including computational time and if there is any loss of information. It effectiveness is measured by a compression ratio, the ratio of information needed to store the compressed data, to the uncompressed data.

2.4.1.2 Kolmogorov Complexity

Let us take a perfect compression, which will always give the minimum amount of information needed to describe some alphanumeric string, s. This compression can be referred to as the Kolmogorov Complexity K (Kolmogorov, 1965), a measure of the least amount of information required to store an object.

Let us take the example of two alphanumeric strings;

$$s_1 = [apv8jhsa646135a9] \text{ and } s_2 = [a1a1a1a1a1a1a1a1]$$

While they are both 16 bits long, it can be seen that s_2 is easily described as "a1 8 times". This is a shorter description and as such s_2 now requires less information to encode than the more complex s_1. However, s_2, could be described in another language with fewer letters, as could s_1. We need to define a universal language that also encodes all objects in the shortest possible length. Using such a language would mean the information required to describe the object is intrinsically an expression of the objects complexity. Therefore we compute the Kolmogorov Complexity as the information associated to an object, using the shortest possible program on a universal Turing machine. A Turing machine (Turing, 1937) has a finite program which manipulates a linear list of cells one at a time, which can take values of zero or one (or be blank). This allows a Turing machine to perform basic operations (Li & Vitanyi, 1993). For true Kolmogorov Complexity the language should be the shortest possible, while maintaining its universality. This is generally accepted to be binary/machine code.

Let us formalise this definition, by using a universal language D, where $D : \{0, 1\}$. Given some object x there exists a descriptor y in language D, so that $D(y) = x$, for all possible x. This follows the idea of data compression where D would be the 'key'. Therefore we can define $y : D(y) = x$ as the set of all possible descriptors of x in language D.

Finally, we wish to find the descriptor y, with the shortest length, to find the Kolmogorov complexity K. This can be expressed as:

$$K_D(x) = \min_y \{|y| : D(y) = x\} \tag{2.16}$$

where $K(x)$ is the Kolmogorov complexity for x for the shortest length of $|y|$, such that $D(y) = x$.

We can relate our universal language D, to an arbitrary description language by means of the invariance theorem. Let us take two description methods D_1 and D_2. The invariance theorem states there is a universal description using a universal Turing machine

U such that

$$K_{U_{D_1}}(x) \leq K_{U_{D_2}}(x) + c \tag{2.17}$$

where c is a constant that is independent of x. This also shows an upper bound on $K(x)$, by considering the constant c as the length of the program that translates U_{D_2} to U_{D_1}. As U is universal, c is only dependent on the descriptor (Li & Vitanyi, 1993). It follows that $K_D(x) \leq |x| + c$.

The conditional Kolmogorov Complexity gives the minimum descriptor of x, based on information in z.

$$K_D(x|z) = min_y(|y|) : D(y, z) = x) \tag{2.18}$$

And is the complexity remaining in x given information of y. Therefore when x and z are completely independent

$$K_D(x|z) = K_D(x)$$

as no information was saved by knowing z. By the same logic it can also be seen that $K_D(x|z) \leq K_D(x) + c$

Kolmogorov randomness states that for a completely random variable r, it is not possible to describe r in a program smaller than the length of itself (n).

$$K_D(x) \geq n = |x| \tag{2.19}$$

Equation (2.19) is a result of the pigeonhole principle (Herstein, 1964), and shows that the complete randomness of r is expressed in the Kolmogorov complexity. This leads to the non-computability of K.

It can be shown that Kolmogorov complexity is related to Shannon entropy, and by extension mutual information (Grunwald & Vitanyi, 2004). By using the chain rule it can also be shown Kolmogorov complexity is analogous to mutual information. Further discussion on this can be found in section (7.1.2).

2.4.1.3 Non-Computability of the Kolmogorov Complexity

Unfortunately Kolmogorov Complexity is incalculable due to the halting problem, namely the inability to determine if a program will continue to run forever, or finish with an output.

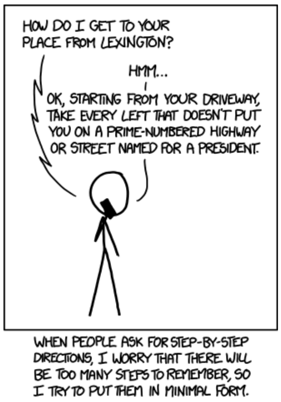


Figure 2.16: A practical application of Kolmogorov complexity! (xkcd.com, 2013)

Let us assume a program computes K so that $x \mapsto K(x)$. Using this fact we design a program P, that calculates if a string x, is Kolmogorov random i.e. that is it satisfies equation (2.19). Here $x \in \{0, 1\}^n$ and requires $log_2(n)$ bits of information to encode.

The program outputs the first value of x it finds satisfies equation (2.19), so that $P(n) = x_n$, and is only dependant on P and n. Here the information of program P is fixed and can be considered a constant, c. This leads to the equation:

$$K(x_n) \leq \log_2(n) + c \tag{2.20}$$

where $\log_2(n) + c$ is the information needed to recover x_n using program P and input n. Therefore it must be a maximum for K. This results in the equality

$$n \leq \log(n) + c \tag{2.21}$$

As c is finite, it can be seen the inequality in equation (2.21) can not be true for all values of n (Trevisan, 2015).

As disorder can be related to the amount of information needed to store an object, this leads to a measure of complexity. Put simply, the more complex a system is the greater the information required to store it, which can be measured using the quantity of entropy (Kaltchenko, 2004; Grunwald & Vitanyi, 2004).

2.4.2 Shannon Entropy

Perhaps the most important quantity in information theory is informational entropy or Shannon entropy, H. Named for Claude Shannon and developed in his seminal paper "A Mathematical Theory of Communication" (Shannon, 1948), it set the groundwork for much of the development of information theory since. It is given by:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{2.22}$$

for discrete variables. In the continuous case the entropy can be estimated to

$$H(X) = -\int_X f(x) \log_2 f(x) \mathrm{d}x \tag{2.23}$$

where $f(x)$ is the probability density function. In both cases using log to the base two means the entropy is expressed in bits. The following convention is also always used, that $0 \log 0 = 0$. In this way entropy can be thought of as a measure of a system's uncertainty, measuring the unknown information in x, based on its probability distribution.

Let us consider a Bernoulli process, where a single variable can take one of two values, let us say zero and one. If we know the value will be, let's say zero, there is no uncertainty and $H = 1 \log_2 1 = 0$. If the probability of it being zero or one is equal, for example like a coin toss, we are most unsure, we could guess heads or tails, it wouldn't matter. Here $H = -2(0.5 \log_2 0.5) = 1bit$. If we generalise this we get the equation

$$H = -p \log_2 p - (1 - p) \log_2 (1 - p) \tag{2.24}$$

where p is the probability of event A (i.e. heads), and is the only variable. The entropy of the second possibility is fully expressed in terms of the first as $(1 - p)$, simply as a consequence that $\sum p(x) = 1$, as with all probability functions. Equation (2.24) is plotted in figure (2.17) for clarity.

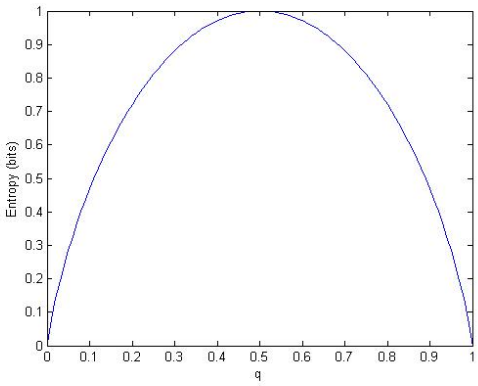


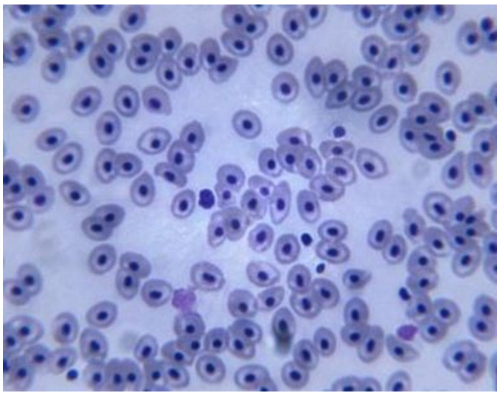
Figure 2.17: Plot of the entropy of a Bernoulli process, with dependant variable being the probability that x takes a particular value, such as in equation (2.24)

A distinction should be drawn with the more commonly known thermodynamic entropy where $\Delta S = \int \frac{dQ_{rev}}{T}$ (where Q is heat and T is temperature), described in statistical physics as the Gibbs entropy $S = -k_B \sum p_x \ln p_x$ (where k_B is the Boltzmann constant). As p_x is also a probability function based on the number of states in the system, it is easy to see comparisons between the two. These two forms of entropy drifted apart, but have seen renewed interest in their joint use in recent years (Mezard & Montanari, 2009). A notable example in their use is maximum entropy techniques used to create distribution functions and models of a variety of physical phenomena as well as construction in image processing (Stern et al., 2002; Jaynes, 1957). Use has also been seen in biology, in particular the mutual information, and as mentioned has been most widespread in the area of computer science with work on error correcting codes and data compression (Mezard & Montanari, 2009). According to Shannon himself, the entropy of the English letter is 4.14 bits (Shannon, 1951), although I fear the entropy of this thesis will be a lot higher!

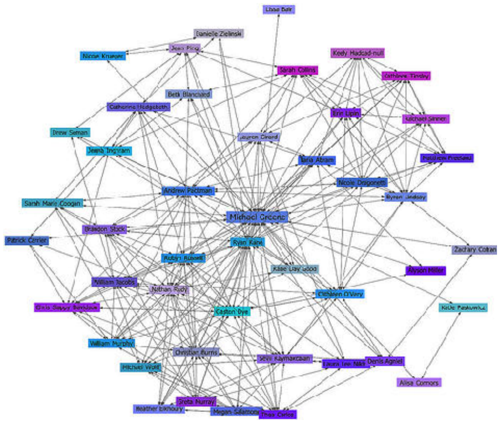
2.5 Complexity Theory

2.5.1 Complex Systems

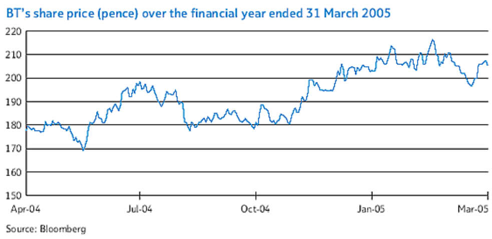
While complex systems have been studied for hundreds of years, the actual science of what a complex system is, and how it behaves, is a relatively new subject. Several examples can be seen in figure (2.18). I like to summarise complex systems with the old saying "the whole is more than the sum of it parts", which is saying that the behaviour of the system cannot be predicted from information of its individual components alone. This is called 'emergence' and is an important property of complex systems.



(a) A group of Human Cells under a microscope



(b) A network comprising of names as vertexes and friendships shown as edges



(c) BT's share price over the course of a year

Figure 2.18: Examples of real world complex systems

Complex systems are often unpredictable and many statistical methods have arisen to attempt to describe these systems. There is particular interest in finance, but applications started in biology and moved on to other areas such as chemistry and physics. The science of complex systems is applied to help understand how the stock market behaves (Aste & Matteo, 2010; Magtegna & Stanley, 2000), how cancer cells divide (Rivier & Lissowski, 1982; Gibson et al., 2011), (in the hope of leading to new cures) and show how real world networks behave (Watts & Strogatz, 1998), (i.e. social networks, to understand how information or disease is passed through a network). Complex systems are made up of many parts and are very sensitive to external and internal fluctuations leading to adaptive and unpredictable behaviour, therefore the methods used are based in probability. While research into the financial markets depends on understanding behaviour at the extremes, such as fat-tailed distributions (Mandelbrot, 1963) (where the probability of extreme events behaves as a power law, and thus are much more likely than a normal distribution), research into granular systems has not shown this behaviour in the Voronoï cell volumes, and so do not well describe them. This is important when considering the behaviour of the probability distribution function when calculating quantities like the mutual information based on the statistics of such systems. Recently there has been increased levels of interest because of their wide range of impact, from physics and biology to finance and mathematics, and particularly chaos theory (Newman, 2010). It is worth noting that it is a common misconception the two are analogous, with all Chaotic systems being complex systems but not vice versa, for example stock prices in the financial market. This occurs most often in systems that exhibit simple macroscopic behaviour, but have complex components. A good example of this is a crowd of people, its motion is far easier to predict than that of a single individual. This is because of the inherent rules which make a complex system, these can include some or all of the following:

- Emergence, as previously stated, as a consequence of this the system shows surprising behaviours and is thus unpredictable.
- History, the way the system looks and will look, is highly dependent on the history of the system. This shows the system is non-convergent and will be very sensitive to initial conditions, even small changes in the system can lead to large changes

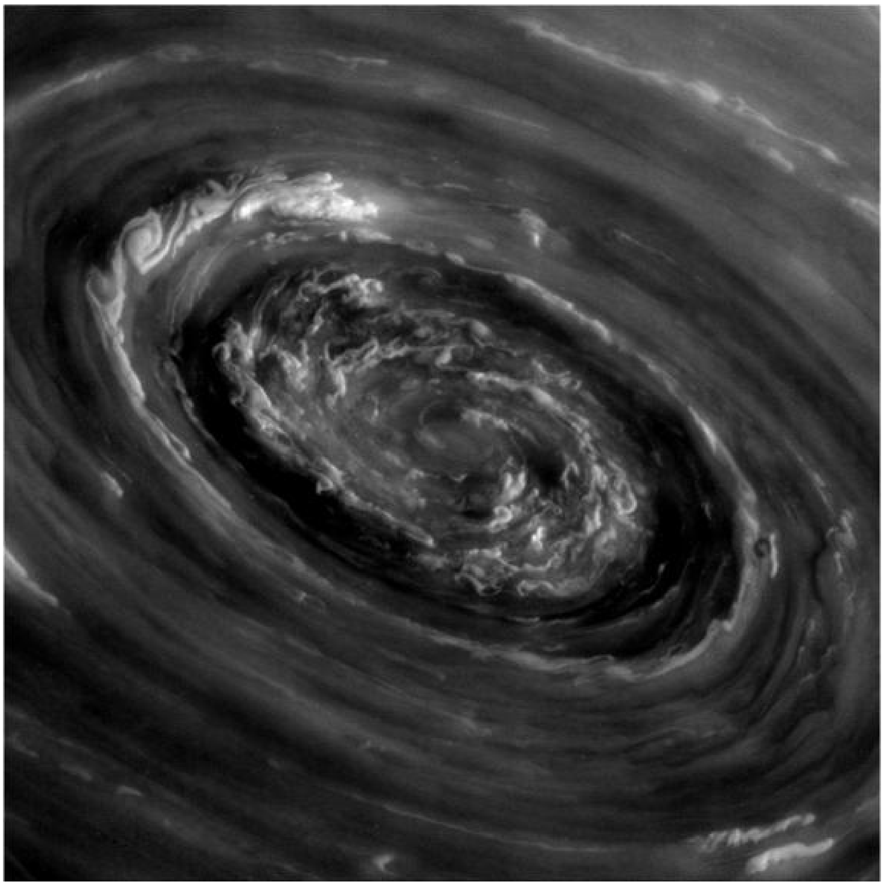


Figure 2.19: A storm on Saturn taken by Cassini, weather patterns are commonly studied as Chaotic systems

in outcome.

- Disorder, special consideration is given to this case in the next section.
- Non-linearity, the effect of small changes are themselves unpredictable, and as stated can have large effects on the system, or none at all!

2.5.2 Networks

Networks are ubiquitous in complex systems. Their study can help to understand how disease is spread through a population, determine the resiliency of computer networks, and how the stock market interacts. It can be helpful to visualise complex systems as networks so I will define some of the terms used in my work here. A network, which can also be referred to as a graph, are a number of objects connected by some relationship. In figure (2.20) the objects, in this case people, are called vertices or

Figure 2.20: An example of a network, in this case a social network with lines showing connections between people

nodes, and the connections between them are called edges. The number of edges connected to a vertex is called its degree. In a directed edge or graph these links can be unidirectional, for example, in financial transactions money usually only goes one way! In more complicated cases, edges can be weighted, meaning there is a probability of movement, in a vertex with only one edge, for example, the traffic can only move down that edge, so its weight or probability is 1. A connected graph is where all vertexes are connected. The distance has the normal meaning, only it is the sum distances of the edges assuming the shortest route possible. A Walk is a graph which comprises a set of nodes connected by a single path of edges. Graph theory contains many other useful definitions for networks but the last one I will mention here is a very useful statistical tool for graphs called the adjacency matrix M_{aj}. This is a V by V matrix (where V is the number of vertexes) and each element corresponds to a potential connection between vertexes, either being a zero for no edge, or a one for a connection. As nodes cannot connect to themselves zeros line the diagonal, and undirected graphs have symmetric adjacency matrices, with the sum of all the elements of matrix M_{aj} being twice the number of edges. Adjacency matrices also have a set of eigenvalues, solutions to the equation $A\bar{X} = \lambda\bar{X}$ where A is a matrix ascribing a linear transformation, and λ is the scalar eigenvalue solutions (Biggs, 1993). The full set of eigenvalues gives the spectrum of the graph, as these are scalars they are usually easier to manipulate than the adjacency matrix itself.

2.6 Programming Languages

As this work has roots in computer science it required some knowledge of programming. Two programming languages were used; Matlab (Mathworks, 1984) and C++. While no knowledge has been assumed on the part of the reader, some information of these languages may be useful when referring to the coding itself, and as such details can be found in the appendix (B.2).

Chapter 3

Methodology - Generating Packings

Generating Packings refers to the techniques used in this thesis to create structures of disks, spheres or hyper-spheres with various properties. Generally packing generation was achieved by computer simulation with one exception shown in section (3.3). There were three main methods used to create packings with their own advantages, which I will outline below.

3.1 Random Sequential Addition (RSA)

The first method used was naturally the easiest and was all accomplished within the Matlab software package itself. The principle was to place a disk randomly in a bounded object and to continue placing disks until there was no more room (as shown in figure 3.1). While this had the advantage of generating simple packings with no need for complex starting conditions, it was a slow method because of the way it looked for new places to put disks, requiring more and more time to find a space for a new disk. The time scaling for a given number of disks N, was approximately $O(N^2)$, although the program was not optimised.

The lack of any real physics made the packings unrealistic, while the theoretical maximum packing fraction for the Random Sequential Addition (RSA) method is the packing maximum $\Phi = 0.92$ (for mono-disperse disks in two dimensions), the probability of finding this randomly is extremely low, as only one configuration exists among a huge number of possible configurations. Practically the RSA algorithm never went much above $\Phi = 0.52$ being that most allowed configurations are expressed within this range. This limitation meant that the algorithm was only used early on. It should be noted that more optimised RSA algorithms can produce higher packing fractions by rejecting sphere placement which maximises the size of unfillable gaps.

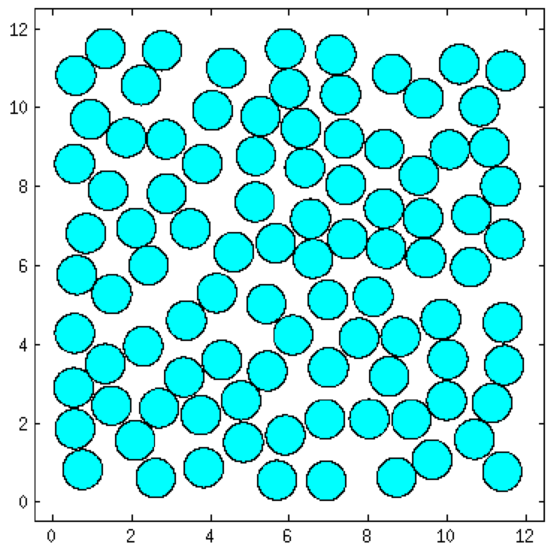


Figure 3.1: A mono-disperse disk packing of around 100 disks using the RSA algorithm

The program itself simply operated on three inputs, the size of the boundary, the number of disks to be placed (at a maximum) and the diameter of those disks. In practice the number of disks (N) would usually be defined as larger than that which would fit to allow maximum saturation of the boundary object. Using these inputs an initial disk was placed by a uniformly distributed pseudo-random number, after which more disks were inserted in the same way, provided they did not overlap with any existing disks. After either N disks are inserted, or a location to place new disks cannot be found given a reasonable number of iterations, the program terminates, outputting the number of disks, packing fraction, and a Nx3 matrix (M) with all disk coordinates $n = (x, y, d)$ where d is the diameter of that disk (important for the case of poly-disperse disks). This can be shown as

$$M = \{n_1, n_2, n_3...n_i\} \text{ assuming } n_{i+1} \notin M \text{ calculated as } \mid \overline{n_{\forall i} n_i} \mid \geq d \tag{3.1}$$

The program can be found in *Appendix* B. While a modified version was created for 3D packings, in practice it was never used.

3.2 Molecular Dynamics Codes

In general terms Molecular Dynamics (MD) codes apply the physical laws to simulate how a time-dependent system of particles will evolve. This involves accounting for interactions between particles as well as forces involved in the environment including pressure, temperature, exchanged momentum, kinetic energy and of course, time. In most cases, including this one, these simulations run according to Newtonian physics taking into account the equations of motion to determine the behaviour of each individual particle.

3.2.1 Lubachevsky-Stillinger algorithm

The code tasked to generate our packings uses the Lubachevsky-Stillinger algorithm (LSA) (Lubaehevsky & Stillinger, 1990), a popular algorithm for sphere packings. The LSA, in general, uses an increase in pressure leading to compression and a subsequent increase in packing fraction as shown in figure (3.2). Under the right conditions (as discussed in section 3.2.2) the LSA leads to a jammed state. The increase in pressure is achieved by either introducing an increasing pressure from the boundary, or, as in

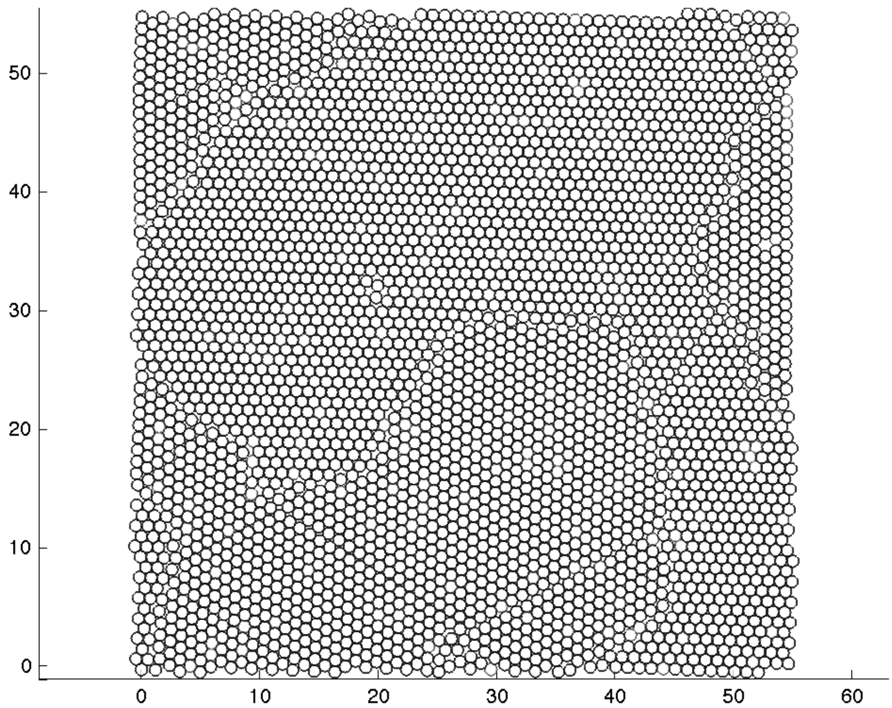


Figure 3.2: An example of a packing of 10,000 disks generated by a LSA, creating a crystallisation with high packing fraction ($\Phi \approx 0.9$)

this case, growing the objects within the boundary giving an increasing density. The boundary used was always a unit cube.

As this algorithm is interested in rigid spheres, a singular interaction potential is applied. This reduces the problem to binary collisions between particles, meaning the code becomes event-driven. Event driven codes are more accurate than time-driven codes (Donev et al., 2005), and advance time by calculating the next event (in this case a collision). That said, the time parameter is still important in predicting the next event, as events are calculated by the equations of motion and growth rates of particles, both being time dependent. To allow calculation of hard sphere packings in MD simulations it is common to give overlapping spheres an infinite repulsive force as used in this code.

To begin, the LSA initially creates some packing using a Poisson distribution by RSA, with a low density of $\Phi \approx 0.35$. Given no termination conditions, the program starts at $t = 0$ with object diameter $D \to 0$, and continues to $t \to \infty$ and $D \to \infty$, governed by the equation $D(t) = r + \gamma t$. This growth rate (γ) is user defined and is critical when creating jammed packings (section 3.2.2). In practice there is of course a maximum diameter, dependent on the size of the particle, the number of particles and the size of the container. In this limit, pressure and collision rate both diverge with little change to the configuration of the packing.

It should be noted that due to the expansion, collisions are not energy conserving, adding a small amount of energy each time. Velocity is therefore rescaled after each cycle (Skoge et al., 2006b), by calculating the average kinetic energy (E_k). The translation and angular velocities are then rescaled by a factor $c = \sqrt{\frac{dkT}{2E_k}}$, (Donev et al., 2005), where T is the desired temperature and is usually user-defined. A cycle is a given number of events, and is also user-defined. Velocity rescaling is not a perfect solution and is not always suitable (Harvey et al., 1998). The initial velocities are taken from a Maxwell-Boltzmann distribution.

In addition to growth, the particles evolve by Newtonian mechanics. Position, momentum, collisions and subsequent transfer of momentum between particles are all governed by Newtons laws. Therefore relative positions are calculated by using a quadratic in t, where collisions are expressed as positive roots. This allows an event list to be created, which gives a list of collisions due to occur at time t_e with partner p_e. The event list contains all the 'impending' events. An impending event is the next event for a given particle and is written as (t_e, p_e). As such the event list is N long, where N is the number of particles in the system. Time is advanced after each event by assigning $t = t_e$. After time is advanced the particles are also moved and the collision is processed. The displacement vectors, exchanged momentum and velocities are calculated, followed by an update to the event list. Events are processed until one of the termination criteria are met, usually the maximum pressure (user-defined) as a consequence of the particles increasing size.

The code also takes advantage of the linked cell method (Figure 3.3) which significantly improves calculation time. By dividing up the volume into cubic cells of equal size we can ensure only the neighbouring particles are used to calculate event possibilities. In the example of figure (3.3), the dark dot is our reference particle, with the lighter grey circle being the interaction distance or cutoff. Any box that falls within that circle is considered as having an effect on the reference particle. This allows collisions to be calculated giving a time scaling of $O(N)$, instead of $O(N^2)$ when predicting collisions for all particles (Sun & Lou, 2008). In the LSA, the boxes are reduced to the minimum size possible, while maintaining the condition that only adjacent cells have any effect.

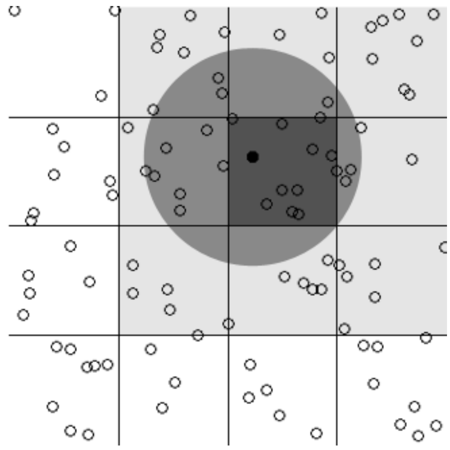


Figure 3.3: Example of linked cell method, only cells within a certain radius of the target particle are considered

One of the most useful features of the program is the ability to use a periodic boundary condition, which significantly reduces interaction of the boundary. This will be discussed in more detail with Mutual Information (chapter 6). The two most common boundaries are hardwall and periodic. A hypercubic cell is used to apply these boundary conditions along each dimension (Skoge et al., 2006b).

Applying a hardwall boundary condition gives the container a solid, rigid edge, from which particles interact directly. In the LSA, hardwall interactions are calculated as collisions, as such they are added to the event list. Hardwall conditions may be considered more realistic, they distort particle cell statistics at the edge.

For investigations concerning sphere packings it is usually far more helpful to limit the effects of the boundary. To this end, a periodic boundary condition (PBC) was usually used. PBCs 'wrap' the packings so that each edge is connected to its opposite side. Opposite sides are defined as being the left and right side faces in each dimension. An

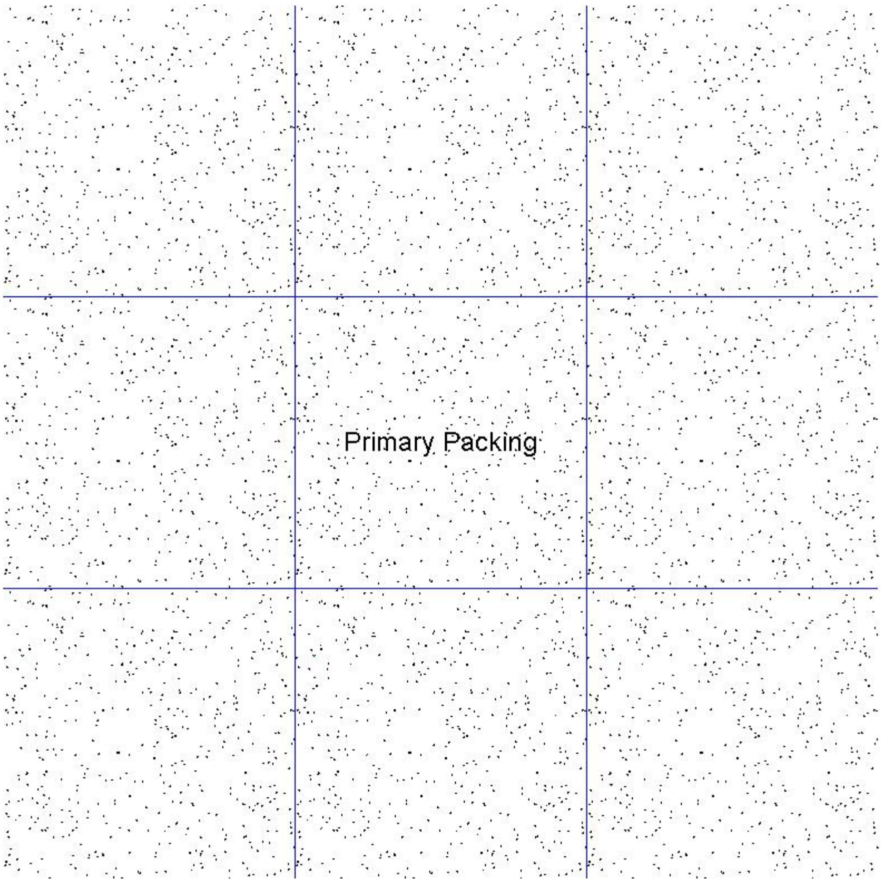


Figure 3.4: Example of how a structure is replicated for a periodic boundary condition

example is given in figure (3.4). In the LSA this is achieved by tiling the packing so that the area under investigation is surrounded by copies of itself. PBCs also introduce another type of event called a transfer, whereby a particle leaves one cell and enters another. As with other events, time is advanced and the new positions and impending events calculated.

3.2.2 Creating Disorder

For my purposes, where we wish to create disordered systems, it is important for jamming to take place while still in the amorphous state. To achieve jamming a slow growth rate must be used, with a high pressure, to allow a sufficiently high packing fraction. In addition the expansion must be initially fast to suppress crystallisation, then slowed to allow a maximally jammed state to be reached (Figure 3.5). The slower the growth rate, the higher the packing fraction, tending to the RCP limit of 0.64 before local crystallisation begins to occur.

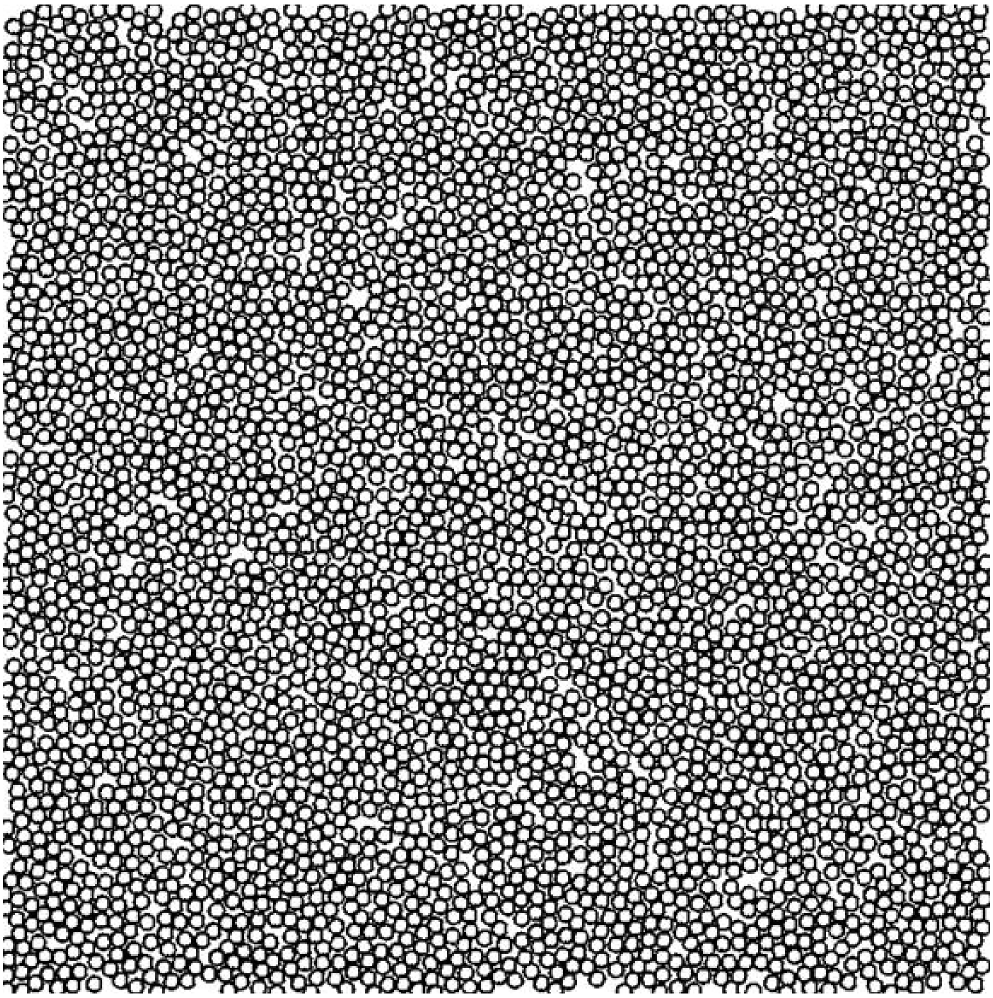


Figure 3.5: An example of a packing of 10,000 disks generated by an LSA, creating a disordered packing at $\Phi \approx 0.6$

The relevant inputs for the code allow for changes to the number of objects inserted, termination pressure and collision rates, and maximum or termination packing fraction. It is important to define some inputs as terminators, in the sense that they are stop commands in the syntax. This is because significantly different properties can arise depending on what parameter has ended the program. For example, if the maximum packing fraction ends the simulation, the system may not have achieved an equilibrium state. Further examples will be investigated in the results, and will, therefore, be discussed in Chapter 6.

As the program is dealing with rigid spheres the temperature has little effect on the final state. To create disordered jammed packings, the expansion rate must be initially high in relation to average thermal velocity (Skoge et al., 2006b), otherwise, the system may crystallise. Therefore, to simplify the parameters, the temperature is kept low and a lower growth rate can be used, giving a range of packing fractions up to and including the packing limit.

The program also allows for the definition of a hardwall boundary condition in addition to the periodic boundary. As mentioned hardwall boundary conditions affect how a packing evolves and can distort statistics at their edge, so the PBC was always used.

In addition, the events per cycle can be modified, which is the events (collisions) processed between each sphere expansion. Increasing the number of events, with an appropriate growth rate, creates higher packing fractions at the cost of time.

Finally the output names can be changed, however naming is not dynamic, so a modified version of the program made allowances to create names based on "in code" collected statistics. This was of particular importance when modifications were made to allow the program to print multiple packings and various packing fractions throughout a simulation. The change did not affect how packings were generated.

A minor rewrite allowed changes to the dimensionality. While the change only required the modification of one number, the program lacked the ability to process it as an input, requiring it to be recompiled.

The output consisted of two main parts. The first part dealt with the statistics, which showed the evolution of the system in terms of its packing fraction, pressure and collision rate. The second was the packing itself, which included the coordinates of each particle in a NxD matrix of values [i.e. $\begin{matrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{matrix}$] as well as the sphere diameters, and some general information about the whole packing.

One further change allowed an additional output for each packing containing the sphere coordinates, but in a slightly different format. This was so that it matched the input characteristics for the Minkowski tensor creation program presented in the next chapter. The change didn't modify any of the original code and only made cosmetic changes to a new output file. Later this modification was expanded to include a loop that printed out the current state of the packing, allowing snapshots to be taken as the system evolved. Again, this change did not affect how the code was processing each cycle, nor could it affect the final configuration. The full changes to the code are shown in the appendix. An unmodified version is available from the link in the reference for (Skoge et al., 2006a). The code was written in the programming language C++.

3.3 Experimental Results

A number of previously experimentally created packings were used, generated by T. Aste, M. Saadatfar, and T. J. Senden at the Australian National University. The packings themselves were composed of mono-sized acrylic beads poured into a container. The container was a cylinder with an internal diameter of 0.075m and the data was collected using x-ray computerised tomography (xCT) with a spatial resolution of 0.03 mm and 0.06mm depending on the sample (Figure 3.6).

Due to the manufacturing process there was poly-dispersity, but only within the bound of 0.05mm. In addition an attempt to mitigate the effects of the hardwall boundary (the cylinder) was taken by random attachment of beads to the inner surface, and consequently when the loose beads are poured a disordered packing results.

The xCT produced some distortion of around 1-2 voxels at the bead boundaries. The bead positions themselves were ascertained from image processing, using convolution to scan the image with a pre-defined sphere. The centres were found when a maximum overlap was detected, this allowed a good estimate of the bead positions (Aste et al.,

2005) Several examples were created with various packing fractions and sizes. The

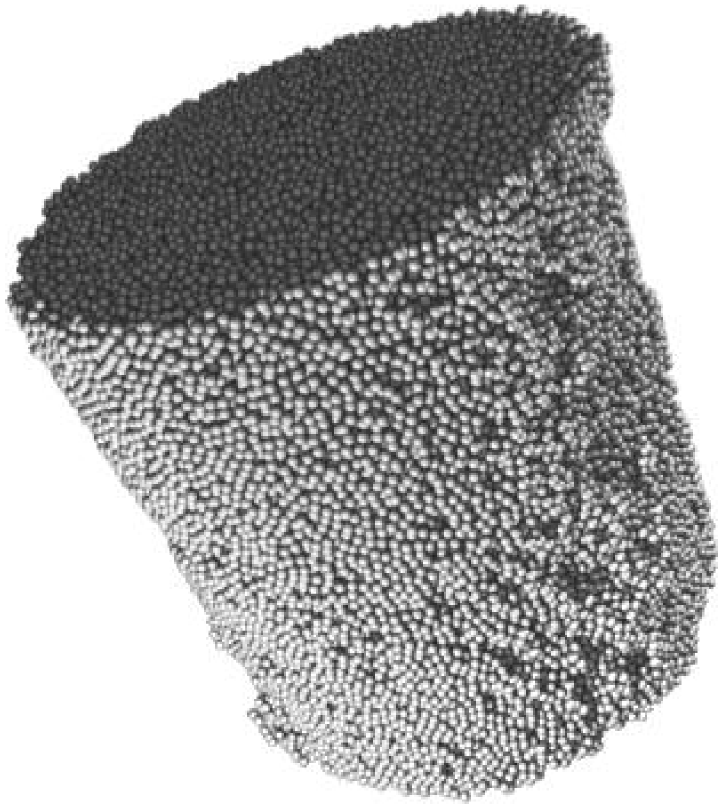


Figure 3.6: X-CT image of $\sim$ 150000 acrylic balls packed in a cylindrical container

larger packings contained $\sim$ 150000 beads and four smaller packings of $\sim$ 35000 with $d = 1.000$mm and $d = 1.59$mm respectively. Given the spatial resolution was 0.03 mm, the precision of the centroids was within 3%, making the results extremely accurate (Aste et al., 2005). The packing fractions ranged from 0.586 to 0.640.

The least dense packings used a technique of inserting a stick into the centre of the container before pouring, then removing it. The other four were created by: simply pouring the beads in slowly, pouring the beads in quickly, gently tapping the cylinder walls after pouring, and a combination of tapping and compression from above after pouring, from lowest to highest packing fraction respectively. It should be noted, where compression was applied, it was only used to alter the packing fraction, and was not continually applied.

3.4 Statistically Driven Data Generation

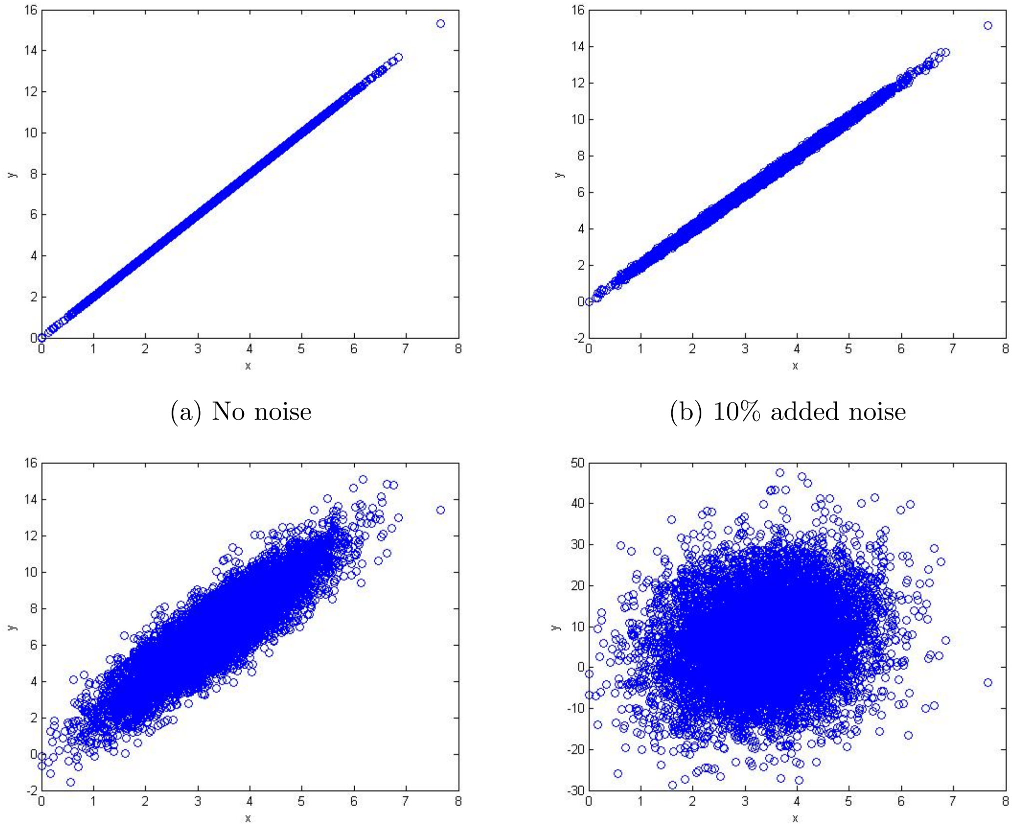
Statistically driven data generation is used to describe a number of methods in which correlated and noisy, semi-correlated data is generated. Most trivially the Matlab functions rand, and randn were used to generate uniformly and normally distributed random numbers, respectively. The Statistically Driven Data Generation method does not create disk packings per se, only creating lists of numbers. I have termed it here so it is clear when referenced in the results.

A set of independent identically distributed (iid or i.i.d.) numbers are created using a pseudo-random number generator. A dependent variable could then be calculated using these numbers. Several equations were used, including linear, quadratic, cubic and exponential relationships, as well as simply generating a second string of iid numbers. Noise was then added to the dependent variable with varying degree and type.

Let x be a string of iid numbers created using a pseudo-random number generator. To allow a comparison, a second string, y, needs to be created with some defined correlation. Lastly a third variable is added, c, which is a random number generated within the same set as x. c acts as noise and is used to test the sensitivity and accuracy of the various correlation measures. n is simply a defined multiplier to control the amount of noise added and its effects are shown in figure (3.7). For contrast of the various correlation measures, six different correlation types were used and are listed below:

- Linear relationship in the form $y = mx + nc$ where m is some fixed constant
- Quadratic relationship in the form $y = x^2 + nc$
- Cubic relationship in the form $y = x^3 + nc$
- Exponential relationship in the form $y = e^x + nc$
- Multivariate Gaussian (MVG), shown in equation (3.2)

- No relationship, where y is another set of iid numbers. In most cases the distribution of the numbers were uniform, however in some specified cases the distribution was normal. In both cases y used the same set as x



(a) No noise

(b) 10% added noise

(c) 100% added noise

(d) 1000% added noise

Figure 3.7: Linearly correlated data with $m = 2$ and various amounts of Gaussian noise added

Scripts were used to generate multivariate Gaussian (Hernadvolgyi, 1998) data (figure 3.8), which allowed the generation of multiple strings of data related by a known correlation coefficient (in this case the co-variance Σ). This created an excellent groundwork for testing how sensitive new methods were to even loosely correlated data, or if they could be used to quantify correlation a-priori.

The Multivariate Gaussian (equation 3.2) was used to create values of x and y.

$$f_{\overline{x}}(x_1, ..., x_k) = \left({}^{1}\!/\!\sqrt{(2\pi)^d|\Sigma|}\right)e^{-\frac{1}{2}(\overline{x}-\overline{\mu})^T\Sigma^{-1}(\overline{x}-\overline{\mu})} \tag{3.2}$$

where $\bar{x}$ is a vector containing $(x_1, ..., x_n)$, $\bar{\mu}$ is the vector containing $(\mu_{x_1}, ..., \mu_{x_n})$ and Σ is the covariance matrix and $|\Sigma|$ is its determinate. $()^T$ denotes a matrix transpose. Σ takes on the size n by n, being the covariance between all elements of $\bar{x}$, so that $\Sigma_{ij} = E[(x_i - \mu_i)(x_j - \mu_j)]$. In the bi-variate case $\bar{x}$ becomes (x, y), $\mu = \binom{\mu_X}{\mu_Y}$ and $\Sigma = \begin{pmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{pmatrix}$. Σ, μ and the length of (x, y) are all user-defined.

While not producing 'real' data, the advantages are the generation of large, easily manipulated data sets very quickly, with a time scaling no more than $O(N)$. This would allow the testing of different correlation methods with pre-defined correlation, increasing confidence that the methods were accurate, robust, or perhaps both.

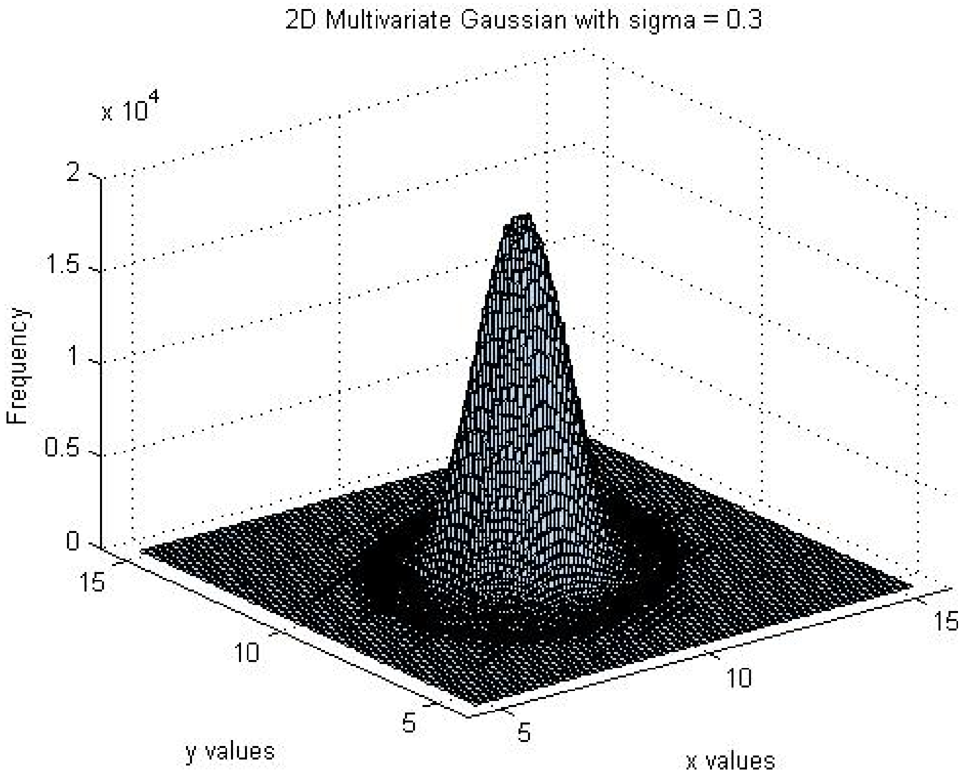


Figure 3.8: A graph of points generated by a multivariate Gaussian function

Chapter 4

Methodology - Framework for Characterisation of Structures

4.1 Description of Packing Structures

Most packings are generated by placing particles in a bounded area (such as figure 4.1), typically a box or a cube, with some boundary conditions. At this point we can start to extract helpful quantities, most importantly the shape, size, and positions of the particles. In a dynamic system other physical quantities become meaningful such as velocity and momentum. The kissing number is the number of spheres a reference sphere is in contact with.

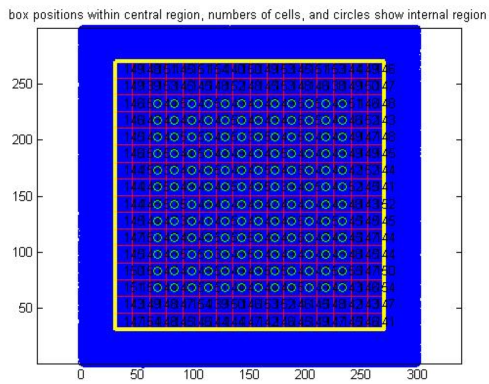


Figure 4.1: A typical two-dimensional packing, blue spots are disk positions with the yellow border outlining the inner edge of the padding. Red lines separate disks into local boxes, green circles show central boxes used as reference samples.

To begin applying a framework to help quantify and understand the disorder some useful statistics need to be extracted. The current section outlines the main techniques and statistics used.

4.1.1 Delaunay Triangulation

Delaunay Triangulation from a set of points in two-dimensions is a space filling triangulation where the circumcircle of each triangle has no other points within its perimeter. In other words only the closest neighbours are used to create the edges from each point, invariably ending with a triangulation (figure 4.2). Delaunay triangulations can be extended to higher dimensions.

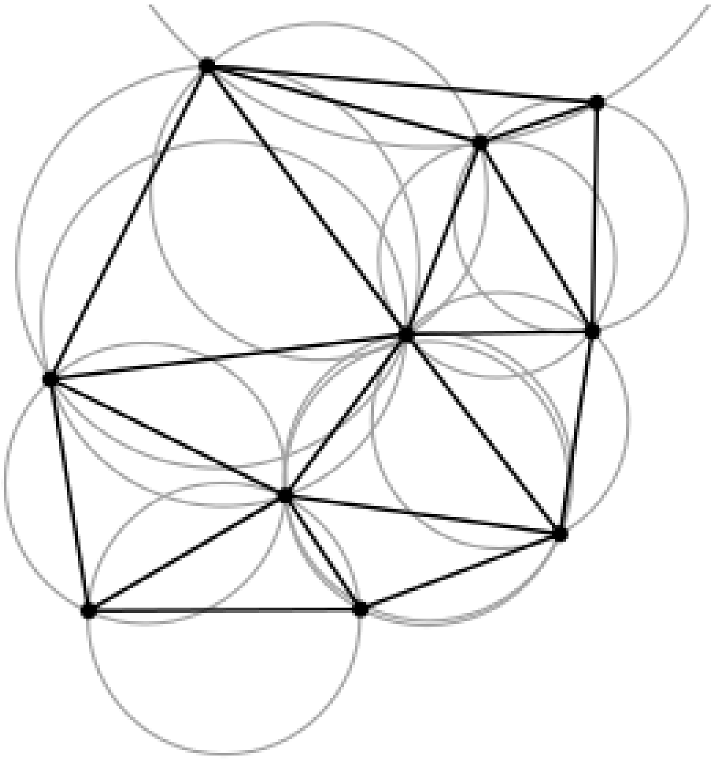


Figure 4.2: A Delaunay Triangulation showing circumcircles for each triangle

While it would of course be possible using these rules to solve a triangulation from brute force (attempting configurations until the conditions are met), it would be a laborious task. The program used, solved the problem by using Qhull. Qhull is a popular c library and computes, amongst other things, convex hulls. Qhull can be found at http://www.qhull.org/, and is also used by Matlab to create Delaunay triangulations and Voronoï diagrams.

By assigning the coordinates a value based on the addition of the squares of the coordinates ($z = x^2 + y^2$ in the 2D case), the program creates a set of points in 3D. These points sit on a paraboloid which is used to calculate the convex hull of lifted sites as shown in figure (4.3). The convex hull becomes a list of facets that enclose a region that contains all the points. The lower convex hull is projected to the input, the facet of which creates the Delaunay triangulation (Qhull, 1995; Mathworks, 2015a). As we have dealt with a parabola, the upper convex hull would simply be the furthest site Delaunay triangulation, or a triangle with no points outside its interior. The Delaunay triangulation is relatively fast with a computational time scaling given a number of points (p) as $O(p \log p)$.

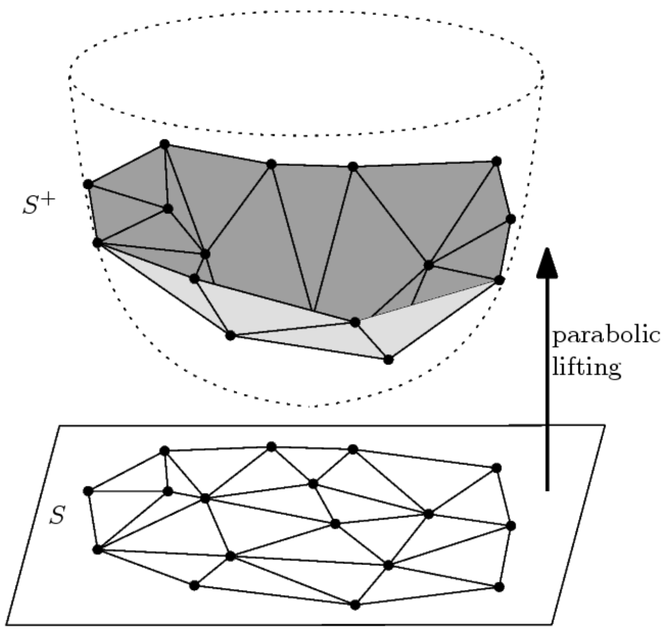


Figure 4.3: An image showing points projected into 3D to allow a convex hull to be created. Source: (Gold, 2006)

4.1.2 Voronoï Space Partition

Voronoï Space Partition, otherwise referred to as Voronoï tessellation or Voronoï analysis, is a powerful tool which can be used to add a shape-filling tessellation of complex polygons to any disk packing. Voronoï partitions can be extended to higher dimensions making them easily adaptable. In order to facilitate creating a Voronoï tessellation it is necessary to reduce a packing to a set of coordinates, achieved by calculating the sphere centroids. A Voronoï polygon or 'cell' can then be constructed around each disk, using the others as a reference to create the edges, such as those shown in figure (4.4). After all the edges have been added a shape-filling tessellation has been created (figure 4.6). Nearest neighbours are defined as any cell in contact with the reference cell. It is worth noting that for a regular arrangement the Voronoï cells will be homogeneous, and in cases such as a HCP arrangement of objects, the cells will be regular polygons (in this case hexagons).

Voronoï tessellations have been widely studied and applied in a number of fields, dating back to Descartes in 1644. Peter Gustav Lejeune Dirichlet was the first to use them in published material in 1850 studying quadratic forms. A famous example of their use was during the Soho cholera epidemic, where the physician John Snow showed the number of infections around the Broad Street pump was much higher than elsewhere, effectively tracking down the source of the infection. Voronoï tessellations have use in many other scientific fields, including geometry, hydrology, materials science, chemistry, biology (including epidemiology) as well as computation and robotics (Aste & Weaire, 2000; Bock et al., 2010).

Let us take four points p_k in space S (as shown in figure 4.4) with coordinates (i, j). For ease p_1 will be taken as the central disk. A Voronoï cell is described as

$$P_1 = [p_1 \in S \mid d(p_1, p_n) \leq d(p_1, p_k)] \tag{4.1}$$

where d is the Euclidean distance between disk centroids. A more complete description

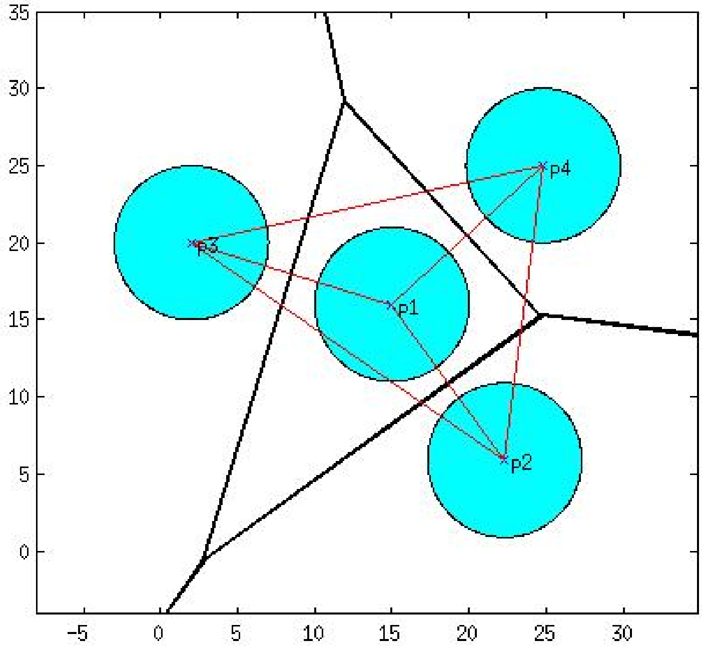


Figure 4.4: A Voronoï Tessellation with four disks and one complete cell, outlined by the thick black lines, other lines connect nearest neighbours creating a Delaunay Triangulation

on constructing Voronoï tessellations can be found in *Appendix* C.

Note in figure (4.4) that cells on the boundary are not closed, giving them a volume $(V) = \infty$. Computationally they are considered closed by the boundary, however, this significantly skews the statistics, especially in smaller data sets. A border must be created for any packing with a boundary to discount cells on or near to the edge, usually several disk diameters. While several disk diameters may seem arbitrary it will be given scrutiny later in the section on Mutual Information.

Voronoï analysis can be used in any number of spatial dimensions, in higher dimensions the method is analogous, giving perpendicular planes slicing a midpoint, then calculating the edges, essentially decomposing it to a two-dimensional problem, then building each cell back up to the n-th dimension. However, higher dimensions require far more computational resources, becoming very difficult in higher dimensions (Mathworks, 2015c).

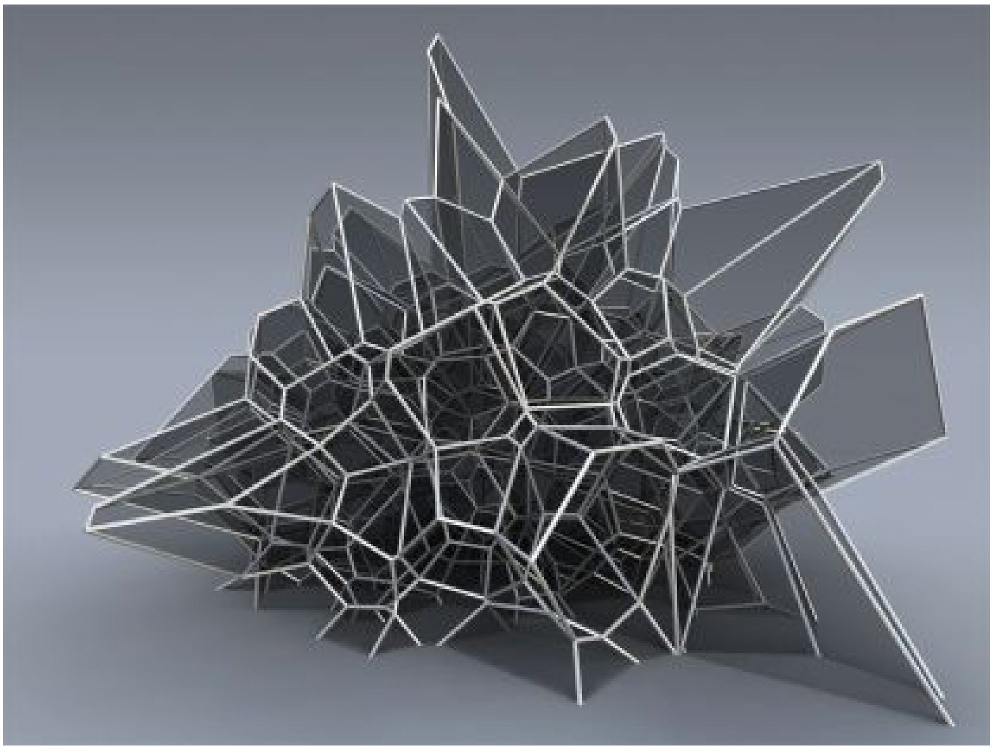


Figure 4.5: A Voronoï tessellation in three dimensions, created by qhull for Knauss Oesterle

4.1.2.1 Properties of Voronoï Tessellations

Voronoï tessellations follow a number of basic rules. Certain values simply scale depending on the size of the particles, such as perimeter, area and volume, however other values such as the number of average edges $< e >$ are fixed.

In two-dimensions $< e > \leq 6$, this can be shown as a consequence of the Euler characteristic, stating that for any polyhedral Tessellation $\chi = v - e + f$, where v is the vertexes and f are the number of faces. For convex polyhedra such as the Voronoï Tessellation, $\chi = 2$. As the minimum number of edges for a polygon is 3 and each edge is shared by 2 cells, it follows

$$\sum_{V \forall v} E_V = 2e \text{ and } \sum_{V \forall v} E_V \geq 3v \therefore 2e \geq 3v \tag{4.2}$$

where $\sum\limits_{V \forall v} E_V$ is the sum of all the vertexes degrees (or their number of edges). It follows that as

$$(v + 1) - e + f = 2 \tag{4.3}$$

(here one has been added to the number of vertexes as a pseudo-vertex to allow all cells on the boundary to be unique to each point as they must for Euler's characteristic (van Kreveld & Loffler, 2015)) that

$$v \geq 2f - 5 \text{ and } e \geq 3f - 6 \tag{4.4}$$

furthermore if we can assume $3v = 2e$, as for a Delaunay triangulation, and use equation (4.3) we can show

$$f = 2 + \frac{e}{3} \tag{4.5}$$

and given

$$2e = f < e > \tag{4.6}$$

where $< e >$ is the average number of sides per cell then

$$< e >= 6 - \frac{12}{f} \text{ so as } f \to \infty, < e >= 6 \qquad \text{Q.E.D.} \tag{4.7}$$

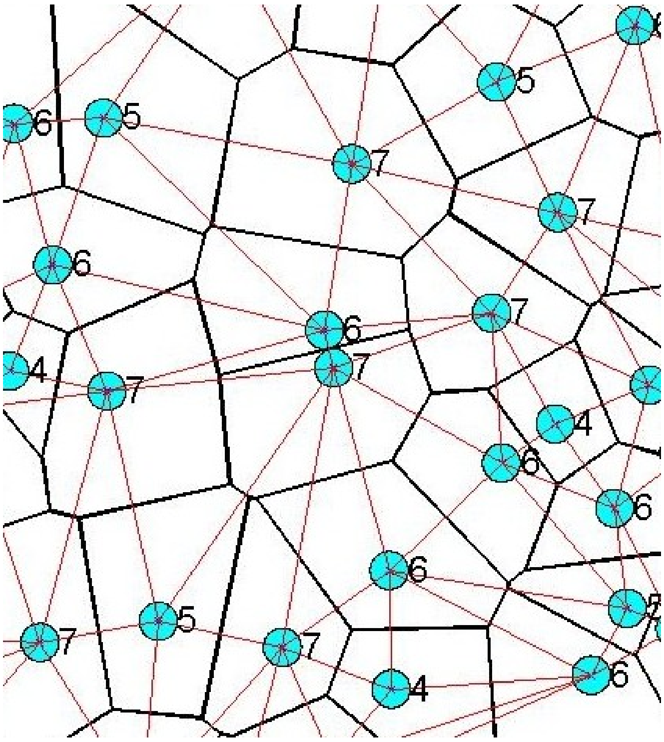


Figure 4.6: A disk packing showing a Voronoï tessellation (bold black lines) with labels showing the number of edges with each cell. The packing was generated using RSA.

As previously stated, all cells are convex. In addition, for $d = 2$ each vertex meets three polygons, with 2 at each edge, for $d = 3$, each vertex meets 4 polyhedra, with 3 meeting at an edge, and can even be extended to non-euclidean spaces (Aste & Weaire, 2000; Isokawa, 2000). For the purposes of programming it is much easier to consider the dual Voronoï tessellation of Delaunay tessellation, whereas the Delaunay triangulation is overlapped on a Voronoï tessellation. Calculating the Voronoï tessellation from Delaunay triangulation tends to give the user more freedom (Mathworks, 2015c). It also allows certain quantities to be obtained more readily, such as the number of edges (figures 4.6 and 4.7).

4.2 Extracting Structural Information

As we have seen a well understood quantity for Voronoï cells is the number of sides, which is typically expressed by a normal distribution with $\mu = 6$ in 2 dimensions. The quantity itself is calculated by counting the number of nearest neighbours, which is computed using the Delaunay triangulation.

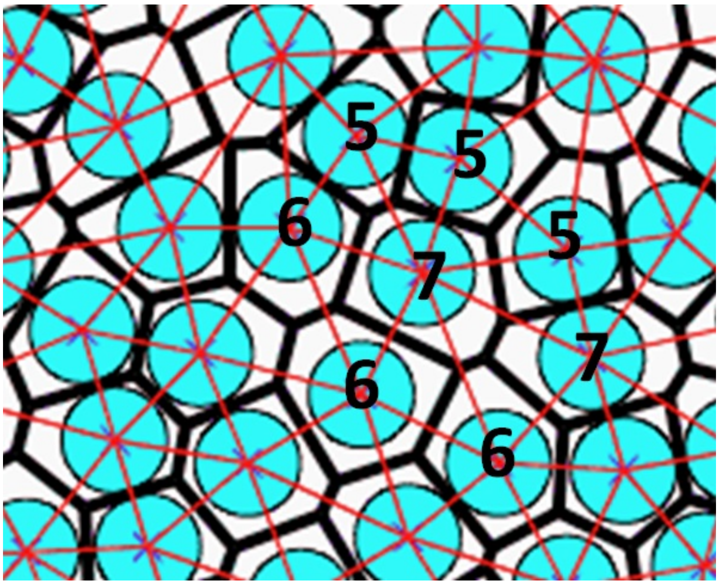


Figure 4.7: A dense packing with Voronoï cells labelled with number of adjacent cells

From raw data several other statistics can be readily extracted such as volume, surface area, and number of faces (analogous to number of sides in three-dimensions). The area (A) is again calculated using Qhull (Qhull, 1995) and can be found by using the coordinates of the vertexes of the Voronoï cell in question. The calculation is then the same for any convex polygon, using a determinant as shown in equation (4.8) with $x_n y_n$ being the coordinates of the vertexes for cell P. Unlike number of faces, which is an integer, area creates continuous statistics that must be binned to create appropriate probabilities for use with entropic measures.

$$A_P = \frac{1}{2}\det\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \\ x_1 & y_1 \end{vmatrix} = \frac{1}{2}[(x_1y_2 + x_2y_3 + \cdots + x_ny_1) - (y_1x_2 + y_2x_3 + \cdots + y_nx_1)] \quad (4.8)$$

In my thesis, global values refer to a quantity obtained using statistics gathered over the whole packing. In the case where a sample is used to create statistics, the convention is to call the quantity local.

4.2.1 Adjacency and Correlation Matrices

In section (2.2) it was shown that matching rules or interaction matrices could be used to create a measure of disorder. This approach inspired a similar use with the extracted structural information, in the form of correlation and adjacency matrices.

In order to create a probability density function from this data I first create a correlation matrix, consisting of $\approx \frac{6N}{2}$ x 2 elements, where N is the number of cells. Each cell is analysed sequentially, and their nearest neighbours added. For p_1, in the example of figure (4.4), the entries are the first three elements of equation (4.9). For M_{cor} I have omitted the symmetrical connections. However, in a true adjacency or correlation matrix all connections are listed twice.

$$M_{cor} = \begin{pmatrix} p_1 & p_2 \\ p_1 & p_3 \\ p_1 & p_4 \\ p_2 & p_3 \\ p_2 & p_4 \\ p_3 & p_4 \end{pmatrix} \tag{4.9}$$

Equation (4.9) shows the completed correlation matrix from figure (4.6), with every connection being expressed. By calculating the mean value and standard deviation of each column, the Pearson correlation coefficient, equation (2.12), can be calculated. In addition the matrix is also used to create probability functions by taking a histogram, which is analogous to a co-occurrences matrix described below.

An adjacency matrix (also referred here as a co-occurrences matrix) uses a correlation matrix and then tallies up all the co-occurrences. To create a co-occurrences matrix for number of sides, for example, the number of times a six sided object occurs next to a five sided object is counted, continuing for all n-sided objects. Over the range of all values this leads to

$$M_{aj}(i,j) = \sum_p \sum_n \begin{cases} +1, & \text{if } f(p) = i \text{ and } f(n) = j \\ +0, & \text{otherwise} \end{cases} \tag{4.10}$$

where $f(p)$ is the number of sides of point p and $f(n)$ is the number of sides of its n^{th} nearest neighbour. In this case, where number of sides is used, $n = [1, 2, ..., f(p)]$.

In cases of continuous data, such as the area, the data must be binned. Therefore bounds are used to tally the data i.e how often a cell with an area in the range A_1 to A_2 lies next to one with an area of A_2 to A_3. The co-occurrences matrix is also square, as the distribution of values in the first set of variables will have the same range as the second (by symmetry). Simply dividing a co-occurrences matrix by its sum the gives the probability distribution.

Calculating the frequency of co-occurrences is done sequentially for each cell over the whole packing. A typical example from two-dimensional packing data would look like:

$$M_{aj} = \begin{pmatrix} 0 & 1 & 17 & 25 & 10 & 0 \\ 1 & 154 & 519 & 458 & 58 & 0 \\ 17 & 519 & 1176 & 700 & 59 & 0 \\ 25 & 458 & 700 & 292 & 23 & 1 \\ 10 & 58 & 59 & 23 & 4 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{4.11}$$

where each row or column corresponds to $i = j = (4, 5, 6, 7, 8, 9)$. It should be noted that as co-occurrences are counted twice, just as adjacency matrices. Therefore, co-occurrence matrices satisfy $M_{aj} = M_{aj}^T$. Matrix (4.11) was taken from early results for the number of sides, the highest value being the $(6, 6)$ occurrence, as one would expect. The data follows a similar shape to the multi-variate Gaussian, this is much clearer for the example of areas in figure (4.8) and the relationship is formally shown in Chapter 6.

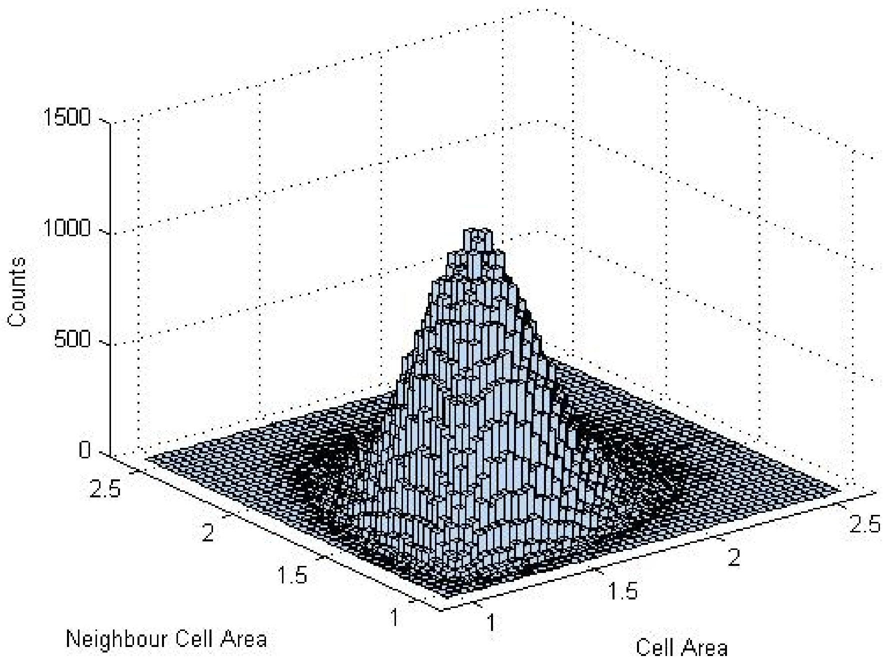


Figure 4.8: A three-dimensional histogram with a bin of [50x50] showing a frequency of cell area co-occurrences taking the form of a multivariate Gaussian distribution. Created using an RSA disk packing of 73076 disks. The disk diameter was set to 1.

4.3 Minkowski Tensors

To allow for the creation of the best statistics possible, a method that completely described the packing would obviously be best. This was found in the Minkowski Tensor (MT or W) approach (Schrder-Turk et al., 2013). Minkowski tensors are a relatively new set of measures used to describe the shape of any three-dimensional object. These tensors are described as generalisations of the scalar Minkowski functionals and allows many characteristics of a generic shape to be extracted as individual scalar, vector and matrix values, building the set that allows a complete description of the morphology. They include values such as surface area and volume, as well as position vectors and the more specific surface integrals and shape descriptors. Some combinations allow for the moment of Inertia to be calculated as well as anisotropy, by taking ratios of eigenvalues for the various surface integrals. Minkowski tensors are often used to describe porous systems, foams, cellular structures and disordered systems, and are very useful in describing Voronoï tessellations.

The tensors were created using the Karambola package (Schaller et al., 2011). The program used a different file format and so results from the RSA and LSA programs had to be modified and a translation program had to be written. Fortunately Matlab had the necessary functions to do this. There are three main sets calculated, including a scalar set (the Minkowski functionals), a set of vectors, and one of tensors. Additionally a set of eigenvalues, calculated from the tensorial measures, formed a fourth set of Minkowski tensors. These values, respectively, were as follows: given a cell K with a boundary δK (in 3 dimensions),

$$\begin{aligned}
W_0^{0,0} &= \int_K \mathrm{d}V \\
W_1^{0,0} &= \frac{1}{3}\int_{\delta K} \mathrm{d}A \\
W_2^{0,0} &= \frac{1}{3}\int_{\delta K} G_2 \mathrm{d}A \\
W_3^{0,0} &= \frac{1}{3}\int_{\delta K} G_3 \mathrm{d}A
\end{aligned} \tag{4.12}$$

These would roughly equate to volume, surface area, mean curvature and Gaussian curvature respectively and take the form of a scalar. V is the space bounded by δK in K, or the volume. A is the surface of the boundary δK or the surface area of K. $G_2 = \frac{1}{2}(\kappa_1 + \kappa_2)$ and $G_3 = \kappa_1 \cdot \kappa_2$ where κ are the principal curvatures. The principal curvatures are the values of maximum and minimum curvature at a point on the boundary as shown in figure (4.9).

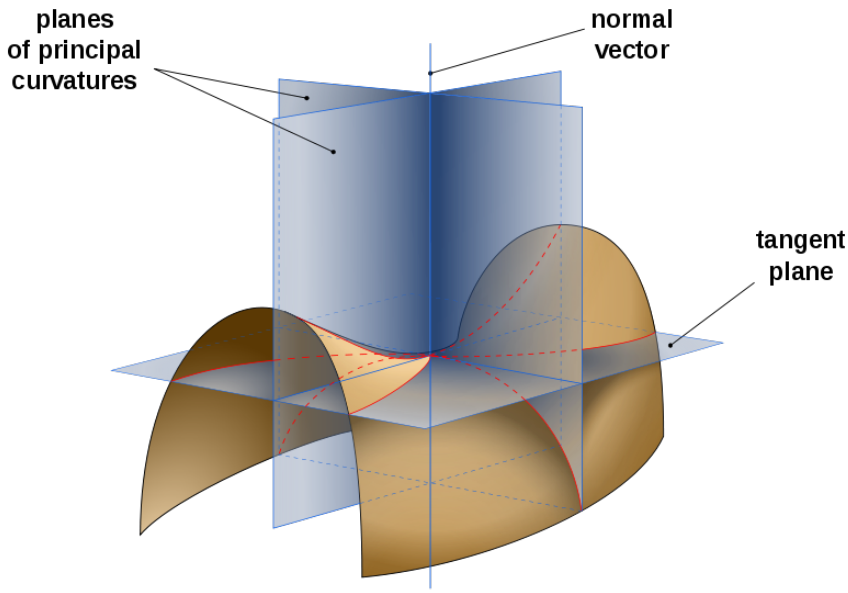


Figure 4.9: Image showing the normal vector for a point on a saddle surface. The normal planes shown also correspond to the planes of principal curvature for that point. Image credit: (Gaba, 2006)

$$\begin{aligned}
(W_0^{1,0})_i &= \int_K \mathbf{x}_i \mathrm{d}V \\
(W_1^{1,0})_i &= \frac{1}{3}\int_{\delta K} \mathbf{x}_i \mathrm{d}A \\
(W_2^{1,0})_i &= \frac{1}{3}\int_{\delta K} G_2 \mathbf{x}_i \mathrm{d}A \\
(W_3^{1,0})_i &= \frac{1}{3}\int_{\delta K} G_3 \mathbf{x}_i \mathrm{d}A
\end{aligned} \tag{4.13}$$

These again would roughly equate to the centre of mass, and three curvature centroids and take the form of a vector. $\mathbf{x}_i$ and $\mathbf{x}_j$ are position vectors of δK where $i, j, k \in \{x, y, z\}$.

$$
\begin{aligned}
(W_0^{2,0})_{ij} &= \int_K \mathbf{x}_i \mathbf{x}_j \mathrm{d}V \\
(W_1^{2,0})_{ij} &= \frac{1}{3} \int_{\delta K} \mathbf{x}_i \mathbf{x}_j \mathrm{d}A \\
(W_2^{2,0})_{ij} &= \frac{1}{3} \int_{\delta K} G_2 \mathbf{x}_i \mathbf{x}_j \mathrm{d}A \\
(W_3^{2,0})_{ij} &= \frac{1}{3} \int_{\delta K} G_3 \mathbf{x}_i \mathbf{x}_j \mathrm{d}A \\
(W_1^{0,2})_{ij} &= \frac{1}{3} \int_{\delta K} \mathbf{n}_i \mathbf{n}_j \mathrm{d}A \\
(W_2^{0,2})_{ij} &= \frac{1}{3} \int_{\delta K} G_2 \mathbf{n}_i \mathbf{n}_j \mathrm{d}A
\end{aligned}
\tag{4.14}
$$

And again these can be roughly equated to the cell volume, three curvature weighted surface integrals and two surface integrals (Schrder-Turk et al., 2013) respectively. These take the form of a set of 3 vectors for which eigenvalues are also calculated. $\mathbf{n}_i$ and $\mathbf{n}_j$ are normal vectors of δK.

The breakdown of the cell description into MTs allow for testing of different metrics and properties of a structure, making it a very useful technique in creating a framework for characterisation of structures.

4.3.1 Radial Distribution Function

The Radial Distribution Function (RDF) measures the distribution of particles from a central reference point (figure 4.10). More specifically it is the likelihood of finding a particle for some given distance from a central point, relative to the overall (global) density.

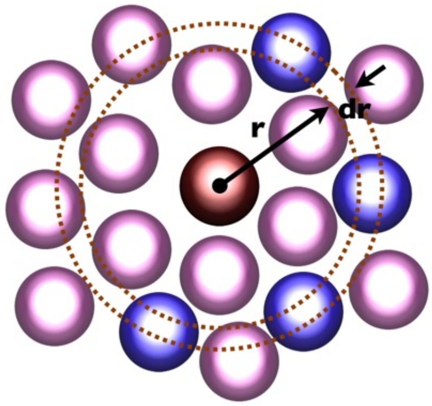


Figure 4.10: Demonstration of the RDF, showing a centre reference sphere and the area dr bounded by two concentric circles. These circles determine n.

The RDF is commonly used in physics to determine the density of points in atomic systems when calculating potentials, molecules in gases and liquids, as well as larger objects such as granular materials. The general form of the RDF $g(r)$ in three-dimensions is

$$g(r) = \frac{n(r)}{\Phi 4\pi r^2 \delta r} \tag{4.15}$$

where δr is the width of the shell bounded by two concentric circles r and $(r + \delta r)$ away, and $n(r)$ is the number of points within the shell. Φ is the density, in most cases the packing fraction. The RDF tends to a value of one, as the global density tends to the local density and the terms cancel out.

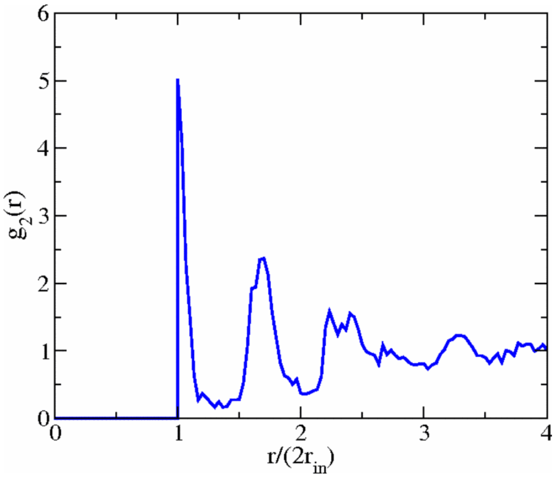


Figure 4.11: Example of an RDF for the densest known packing of tetrahedra, with r_in being the radius of the insphere of a tetrahedron created by Torquato and Jiao (Torquato & Jiao, 2009)

An example of the RDF is shown in figure (4.11) for a dense packing of octahedra. In this case we can see the RDF $g_2(r) \to 1$ as r increases, showing a lack of long-range information being carried through the system (Torquato & Jiao, 2009).

Chapter 5

Methodology - A Statistical Approach

5.1 Entropic Measures of Information

In section (2.4.2) Shannon entropy was discussed as a measure for the information stored in a system. The more configurations there are in a system, the greater the uncertainty, and in turn the greater the entropy.

In order to calculate how information is passed through a system, it becomes necessary to compare disparate parts of that system. In section (2.3.2) it is discussed how knowledge of an event A, may change the probability of an occurrence in event B.

By combining conditional probability and entropy it becomes possible to quantify the information shared by two parts. In essence we are comparing two variables and quantifying the uncertainty left in event B, given event A. Take the example of a packing structure, we can draw a comparison to knowing the unit cell of a crystalline structure, which in turn gives all the information needed about the system. To measure this shared information we introduce conditional and joint entropy.

The joint entropy is the uncertainty for a set of variables. This, like the entropy, can be thought of in terms of Kolmogorov complexity. Therefore, the more well defined the quantities are, the more certain we are about their values and the more compressible the information. The joint entropy is described by the equation:

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log[p(x,y)] \tag{5.1}$$

The conditional entropy is the uncertainty of x when we know y

$$H(X|Y) = -\sum_{y\in Y}\sum_{x\in X} p(x,y)\log\frac{p(y)}{p(x,y)} \tag{5.2}$$

and thus gives us a measure of the information given to x by knowing y. This is the remaining joint entropy, therefore the conditional entropy can be written as:

$$H(X|Y) = H(X,Y) - H(Y) \tag{5.3}$$

If no information is gained by knowing y then $H(X|Y) = H(X)$. This can also be implied following Bayes Theorem giving

$$H(X|Y) = H(Y|X) - H(Y) + H(X) \tag{5.4}$$

which also shows $H(X|Y) \leq H(X)$.

While the Kolmogorov complexity is uncomputable, it is important to state that by using the entropy to quantify complexity we are no longer using a universal language. This is because entropy is a relative measure based on probability theory, and while it is a very useful quantity it is not a perfect descriptor of complexity in the same way Kolmogorov complexity is.

5.2 Mutual Information

To start analysing multi-variate data, approaches beyond Shannon's entropy need to be explored. Mutual Information (MI or I) or Mutual Entropy describes the amount of information contained in one variable that relates to the other or, alternatively, how well variable A describes variable B (figure 5.1). This simple quantity gives us a robust method for quantifying the correlation in a system and by extension its complexity. The MI is robust as it does not assume any apriori information about the type of correlation in the system being used in a number of fields, most notably in Biology and Computing (Steuer et al., 2002; R., 1994). MI measures the difference between the configurational and conditional entropies and thus mathematically quantifies the difference between the joint probability and the conditional probability shown below as:

$$I(X,Y) = \sum_{y \epsilon Y} \sum_{x \epsilon X} p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{5.5}$$

where $p(x, y)$ is the joint probability of x and y. Log to the base 2 has been used so that the MI is always expressed in *bits*. There is also the convention that, as with the entropy, $0 \log 0 = 0$;

Equation (5.6), equation (5.5) and the previously defined Jensen's Inequality (Jensen, 1906), show that $I(X,Y) \geq 0$. This does not hold true for multi-variate cases which are less well defined (de Cruys, 2011). By extension MI is also symmetric $I(x;y) = I(x;y)$, and concave.

$$I(X,Y) \geq -\log \left(\sum_{y \epsilon Y} \sum_{x \epsilon X} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \right) = -\log \left(\sum_{y \epsilon Y} \sum_{x \epsilon X} p(x)p(y) \right) = 0$$

$$\therefore I(x,y) \geq 0 \tag{5.6}$$

MI can be further understood by application of Bayes Theorem which states the joint probability is equal to the product of the two single probabilities if and only if the two variables are independent:

$$p(x, y) = p(x)p(y) \tag{5.7}$$

therefore if two variables are independent

$$\left(\frac{p(x, y)}{p(x)p(y)}\right) = 1 \;\therefore\; I(X, Y) = p(x, y)\log 1 = 0 \tag{5.8}$$

This means if no information about x can be taken from y, $I = 0$.

It is also very useful to define the mutual information in terms of entropies. It can be shown using equation (5.5) that

$$\begin{aligned} I(X, Y) = &\sum_{x\epsilon X, y\epsilon Y} p(x, y)\log_2(p(x)) \\ &+ \sum_{x\epsilon X, y\epsilon Y} p(x, y)\log_2(p(y)) \\ &+ \sum_{x\epsilon X, y\epsilon Y} p(x, y)\log_2\left(\frac{1}{p(x, y)}\right) \\ = &\sum_{x\epsilon X} p(x)\log_2(p(x)) + \sum_{y\epsilon Y} p(y)\log_2(p(y)) - \sum_{y\epsilon Y}\sum_{x\epsilon X} p(x, y)\log_2(p(x, y)) \end{aligned}$$

by substituting in equation (2.22), the Shannon entropy, and equation (5.1) we find

$$I(X, Y) = H(X) + H(Y) - H(X, Y) \tag{5.9}$$

given that

$$H(X|Y) = H(X, Y) - H(Y) \text{ and conversely } H(Y|X) = H(X, Y) - H(X)$$

it can also be shown that

$$I(X, Y) = H(X) - H(X|Y) \text{ or } I(X, Y) = H(Y) - H(Y|X) \tag{5.10}$$

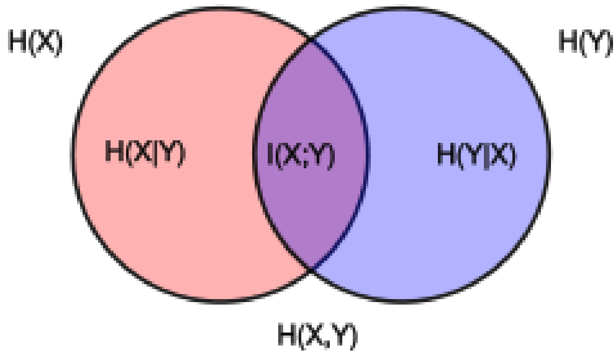


Figure 5.1: Image explaining the concept of information and entropy on a Venn diagram (Voelkel, 1998)

Now some useful bounds can be defined. It has already been shown that for two completely independent variables $I(X,Y) = 0$ (equation 5.8). This is the lower bound for MI previously stated in equation (5.6). Therefore using equation (5.9), $H(X) \geq H(X|Y)$. Extending this to two completely dependent variables, it can be seen that $H(X|Y) = 0$ if and only if the value of x is completely determined by the value of y. Conversely, $H(X|Y) = H(Y)$ if and only if x and y are independent random variables, leading to

$$I(X,Y) \leq H(X) \tag{5.11}$$

and can be thought of as $I(X,X)$ as no variable can give more information than to itself. A kernel density estimator is also used (section 5.2.2). The Kernel method approximates the discrete data to a function by creating a polyfit of the co-occurrences matrix, thus creating a probability density function (pdf). The pdf can then be integrated over as defined by the continuous equation for MI:

$$I(X,Y) = \int\limits_{y\epsilon Y} \int\limits_{x\epsilon X} p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right) \mathrm{d}x\mathrm{d}y \tag{5.12}$$

5.2.1 Statistical Approaches to Quantifying Disorder

To understand how the mutual information behaved for normally distributed variables, I implicitly calculated MI using the well-defined normal probability density function:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu_x)^2}{2\sigma^2}) \tag{5.13}$$

By substituting in to

$$H(X) = -\int_{x \in X} p(x) \log_2 p(x) \tag{5.14}$$

and integrating (by parts) over all values of x, it is shown

$$\begin{aligned} -\int p(x) \log_2 p(x) &= \left[-\frac{1}{2}\ln(2\pi\sigma_x^2) - \frac{\sigma_x^2}{2\sigma_x^2}\right] \\ \therefore H(X) &= -\frac{1}{2}\ln 2\pi - \ln \sigma_x - \frac{1}{2} \end{aligned} \tag{5.15}$$

From here it can be found that the MI relates to the Pearson correlation coefficient directly in the bi-variate case. Usefully this will allow a way to calculate MI independent of any probabilities.

As we have already seen the MI can be expressed as $I(X,Y) = H(X,Y) - H(X) - H(Y)$, leading to

$$\int\int I(X,Y)\mathrm{d}x\mathrm{d}y = \int\int \ln[p(x,y)] - \ln[p(x)] - \ln[p(y)]\mathrm{d}x\mathrm{d}y \tag{5.16}$$

Here we require the additional substitution of the bi-variate normal distribution $p(x,y)$

$$= \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp(-\frac{1}{2(1-\rho^2)}[\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}]) \tag{5.17}$$

Finally, by substituting equation (5.17) in to equation (5.16) we find

$$\int\int \ln[p(x,y)]\mathrm{d}x\mathrm{d}y = \int\int \ln\Big(\frac{1}{2\pi\sigma_x\sigma_y(1-\rho^2)^{\frac{1}{2}}}\Big) + \frac{\frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} - \frac{x-\mu_x}{\sigma_x^2} - \frac{y-\mu_y}{\sigma_y^2}}{-2\rho^2}\mathrm{d}x\mathrm{d}y \tag{5.18}$$

and given that $\rho = \frac{\langle[(x-\mu_x)(y-\mu_y)]\rangle}{\sigma_x\sigma_y}$ and by using the result shown in equation (5.15) it can be shown after simplification that

$$I(X,Y) = -\frac{1}{2}\log_2(1-\rho^2) \tag{5.19}$$

The result is analogous for $H(y)$, and can also be proved in the same way by using equation (5.5) directly.

The result was important, as it showed, for normally distributed data, that the MI was independent of the number of data points (i.e. the number of disks in a packing). The only variable quantity was the correlation coefficient. As entropic measures such as the MI measure uncertainty, and ρ uses standard deviation from a mean value to measure correlation, it was not surprising that the two could be related in some way.

5.2.2 *Probability Binning*

'Binning' is the term used to describe the act of placing data in the intervals of a histogram. A 'bin' refers to one interval. To calculate probabilities it is required to bin data so that the correct frequency of an occurrence can be calculated. In some cases this is implied, such as with integer quantities. In the case of continuous data the interval bounds are defined i.e. the frequency of data lying in the range b_1 and b_2. If n is the total number of data points, k is the total number of bins and m is a discrete histogram then m_i is the total number of points falling in bin i and $n = \sum_{i=1}^{k} m_i$. The bounds are decided on a number of factors including the desired size of the bins, or the number of bins required. In rarer cases the binning may be adaptive, with various sizes of bins that are dynamic.

The choice of binning is important when dealing with entropic measures as it is well documented that this affects the value (Miller, 1955; Paninski, 2003; Schurmann, 2004; Grassberger, 1998). Specifically a characteristic bias can occur as a result of changes to the probability function. Bias can significantly skew the value of the entropy and MI.

Let us take an extreme example for a random string of N values. As the number of bins $\to \infty$ data becomes extremely sparse, with most bins having an occupancy of zero. The average occupancy of occupied bins will $\to 1$. In this case the entropy will tend to

$$H(X) \to \log \frac{1}{N} \tag{5.20}$$

which is not the case for non-uniform data. Conversely taking the number of bins $\to 1$ results in $p(x) \to 1$ therefore $H(X) \to 0$. In both cases the MI becomes invariant and provides no useful information. The goal therefore is to minimise the bias so relevant and reliable information can be extracted. The bias of information measures is well known and there have been a number of proposed solutions that will be given further attention later.

In the same way as the entropy, the MI shows a bias dependent on the size and occupancy of the probability bins, with the added issue of two or more variables. This can be clearly seen in figure (5.2) where a low Φ packing has been used so that $I \to 0$ for larger box sizes.

A number of attempts were made to characterise the bias using n^{th} neighbour boxes. This was because their MI should be close to zero allowing a good model for testing. I began with a polynomial fit in the form $f(x) = g_1 x^n + g_2 x^{n-1} + ... + g_n x + g_{n-1}$. After some trial and error and experimenting with $\log(I)$, the relationship appeared to be exponential in the form $I = e^{c_1 \log(b) + c_2}$ with b the size of the box. c_1 tended to be between 3.5 and 3.9. Polynomial fitting still made assumptions and so a comparison was made for the calculated value of MI for Gaussian data as shown in section (5.2.1).

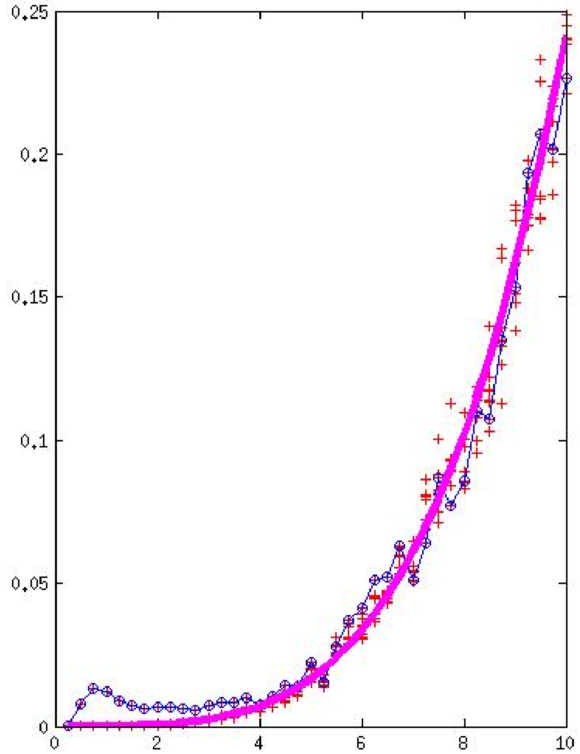


Figure 5.2: A graph of MI (in bits) vs box size (in units of disk diameter) taken to large box sizes to show MI bias. Here a 50,000 disk packing generated by RSA is used. Blue circles represent 1^st neighbour boxes, with red crosses showing boxes further away. The pink line characterises the bias by using a polynomial fit of MI for 4^{th} neighbour boxes

This was useful in cases of statistically driven data, but was not useful in the case of packing, as ρ is not sensitive to non-linear correlation (Cellucci et al., 2005).

The best binning was empirically found to be $\sqrt{\frac{1}{N}}$ in the case of statistically driven data sets, and held true for packing data. MI performed well as a relative measure when binning occupancy remained within an order of magnitude or so. Wherever possible binning was fixed as to minimise any bias. This seemed sufficient in the case of packings as the number of disks remained within an order of magnitude without the need for adjustment. Still, it is important to be aware of information bias when interpreting the MI. In summery the bias is dependent on the statistics and the properties of a packing, meaning the MI could not be relied upon as an absolute value of disorder.

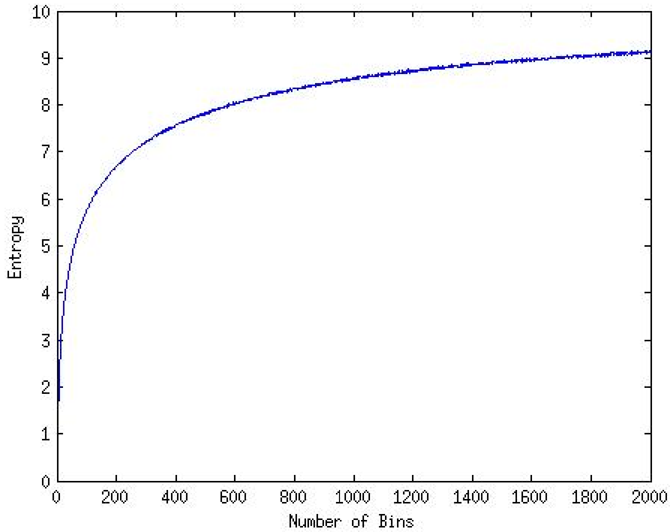


Figure 5.3: Bias of Shannon entropy given a number of bins, with constant N = 1000 for normally distributed random points, the bias tended towards $\log\left(\frac{1}{b}\right)$

A number of other methods were attempted to address the bias issue. The first attempt at quantifying the bias used

$$\frac{\sigma^2}{B^2} \approx \frac{N(\log b)^2}{b^2} \tag{5.21}$$

proposed by (Paninski, 2003), where B is the Bias and b is the number of bins. From equation (5.21) it can be seen that bias is effected when N is too small, or b too large, particularly in the case $\frac{\sigma^2}{B^2} < 1$. This property was important when finding a good binning for packing data. While equation (5.21) is well understood in the range $N \gg b$, it is not as useful in the range $N \approx b$ (Paninski, 2003). Other more general approaches were also tested, including the long standing Miller-Madow Entropy:

$$H_{mm}(x) = H(x) + \frac{b-1}{2N} \tag{5.22}$$

Here a correcting function is added to the Shannon entropy, and is dependent on the number of bins and the number of data points. It would therefore be useful for smaller sample sizes or packing structures.

Clustering MI (Slonim et al., 2005) was another attempted solution, in which information is binned by making as few assumptions as possible. Let us take N data points ($i = 1, 2, ..., N$) and N_c clusters (the bins, $C = 1, 2, ..., N_c$). Also let us take an arbitrary similarity measure for some given elements i_r to be $s(i_1, i_2, ..., i_r)$. Slonim et. al. proposed the average similarity of a cluster as

$$s(C) = \sum_{i_1=1}^{N} \cdots \sum_{i_r=1}^{N} P(i_1|C) \cdots P(i_r|C) s(i_1, ..., i_r) \tag{5.23}$$

where $P(i|C)$ is the probability that element i is in cluster C.

This allows for the formation of an optimisation problem expressed by maximising the average similarity $< s >= \sum_{C=1}^{N_c} P(C)s(C)$ (where $P(C) = \sum_{i=1}^{N} P(C|i)P(i)$ and is the total probability of finding any element in cluster C and $P(i)$ is the probability of element i occurring) and minimising the amount of information taken to store the clusters:

$$I(C, i) = \frac{1}{N} \sum_{i=1}^{N} \sum_{C=1}^{N_c} P(C|i) \log \left(\frac{P(C|i)}{P(C)} \right) \tag{5.24}$$

The cluster MI method seemed promising as it could be likened to Kolmogorov complexity, however it was found to be difficult to implement using packing data giving out of bound results. With some minor adjustment Cluster MI did work for statistically generated results, but bias was still present in the data, similar in magnitude to the unaltered methodology.

An estimator proposed by Grassberger et al. was also considered, specifically one using a Gamma function to estimate the bias $H(x) = \frac{1}{N-1} \sum \log(p(x) + \Gamma_1 - \Gamma_N)$ with Γ_1 being the Euler-Mascheroni Constant (Grassberger, 1998). This was an extension of the Renyi Entropy, which was also used, expressed as

$$H(q) = \frac{1}{1-q} \log \sum_{i=1}^{M} p_i^q \tag{5.25}$$

where q is the order of information. In the linear case $q \to 1$ and the Renyi entropy can be shown to tend to the Shannon entropy (Schurmann, 2004). This worked well over the range $b \gg N$, but was less reliable in the preferred case of $b \sim N$.

Kernel density estimators (KDE) are also used. The technique is not ideal but allows for a more consistent approach (Moon et al., 1995) while minimising any binning bias. The KDE took the place of a histogram, creating a mesh which values fell in. For bi-variate data this mesh was simply a 2-fold matrix defining the bin locations. An identically sized matrix then contained the frequency measures for each bin, effectively making it a 3-dimensional histogram. The KDE data was then passed through a piecewise interpolation to create a smooth function that fit the original data. This function was a probability density function, allowing an integration using the continuous form of MI (equation 5.12).

This being a well understood problem of entropic estimators mean there are a huge number of other proposed solutions. As a summary, these tend to fall into these categories, including those above;

- Correcting the entropy with an added term, in some cases a constant, in others a term dependent on the number of bins or the number of data points, and in some cases both;
- Methods that modify the binning. This may include methods that define a binning size or number, while others use an adaptive partition such as the Fraser Swinney algorithm (Cellucci et al., 2005);
- Substituting terms to eliminate binning altogether, such as probability, as in the case of equation (5.19).

While all these methods had merit, they did not solve the immediate problem and tended to only apply for specific ranges to N, which itself was variable.

As the results will show later, it was possible to compare and contrast these techniques. This allowed an empirical binning to be set which worked well for packing data. By comparing MI calculated with probabilities against MI calculated by equation (5.19), it was possible to create a more dynamic binning with the drawback of being less accurate when using data that was not linearly correlated. One advantage was that generally data sets had the same range, meaning binning could be kept constant over both variables.

Ultimately the binning was kept around $\sim \sqrt{\frac{1}{N}}$, and adjusted as needed. Wherever possible the binning was kept consistent to allow some comparison. Issues with the binning led the research towards more relative measures, such as the Kullback Leibler Divergence discussed next, however work continues on a solution to the bias problem.

5.3 Kullback Leibler Divergence

MI shows how information between two pieces of a system can be quantified. This allowed a relative measure of information, but did not give data regarding the whole system, or even where more information might be found. To this end, a completely relative measure based on statistics for the whole system was used.

Kullback Leibler Divergence (KLD or D_{KL}) proved to be a good tool in this endeavour. The KLD takes two probability distributions and compares them giving a measure of how much information is required to encode one from the other. It is traditionally used as a tool to compare scientific models and experimental data allowing for an absolute value measuring their 'closeness'.

MI and KLD are similar quantities both rooted in information theory, and both requiring the same sort of information, notably probability distributions. It made KLD a good choice as it could be adapted to the current framework without changing how statistics are extracted.

Mathematically the KLD can be described by the equation (in the discrete case)

$$D_{KL}(P|Q) = \sum_{x \epsilon X} P(x) \log_2 \left(\frac{P(x)}{Q(x)} \right) \tag{5.26}$$

giving a mathematical measure of how much extra information is required to code Q from P (shown in figure 5.4). This can be extended to the continuous case but taking the indefinite integral instead of the sum. For two identical distributions where the

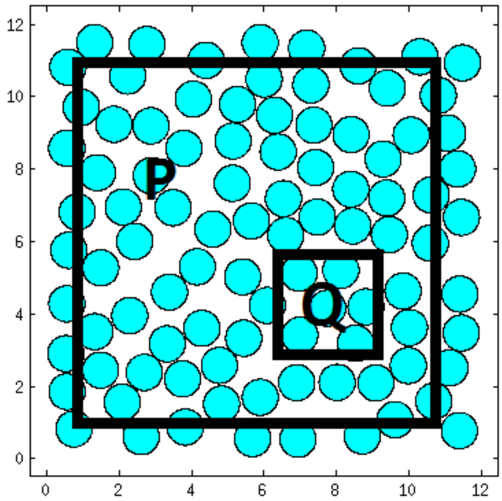


Figure 5.4: Example showing how information from Q can encode information from P, given some uncertainty expressed in the KLD

sample exactly describes the model, $D_{KL}(P|Q) = 0$. It can then be proven to always be greater than zero by applying Jensen's Inequality which states that for a function $f(x)$

$$\langle f(x) \rangle \geq f(\langle x \rangle) \tag{5.27}$$

therefore

$$D_{KL}(P|Q) \geq \log\Big(\int p(x)\frac{Q(x)}{P(x)}\mathrm{d}x\Big) = \log\Big(\int Q(x)\mathrm{d}x\Big) = 0 \ \therefore \ D_{KL}(P|Q) \geq 0 \tag{5.28}$$

this solution is referred to as the Gibb's inequality. For the purpose of computing the KLD it was easier to calculate a sum of entropies, therefore a new quantity must be defined called the cross entropy (equation 5.29). This quantity measures the amount of information in Q, given P (similar to conditional entropy).

$$H(P,Q) = -\sum_{x \in X} P(x)\log[Q(x)] \tag{5.29}$$

Given this and equation (2.22) it is found that

$$D_{KL}(P|Q) = H_x(P, Q) - H_x(P) \tag{5.30}$$

KLD is also non-symmetric and convex, as $\log x$ is convex (Mezard & Montanari, 2009). It is a quantity that can be thought a little as a convolution of images giving a value of their overlap, in this case the overlap of their combined area.

5.3.1 Finding Motifs

The KLD allows a technique of scanning structures looking for areas of high and low information. This is achieved by moving the sample area across the packing given some arbitrary interval (shown in figure 5.5).

By searching for low KLD, it was possible to find areas information is more likely to be stored, that is, areas that have more information about the whole system. By changing the sample size of Q, it was possible to look for motifs. Of course in a disordered system there is no pre-defined size for a motif, but some local configurations give more information about a system than others.

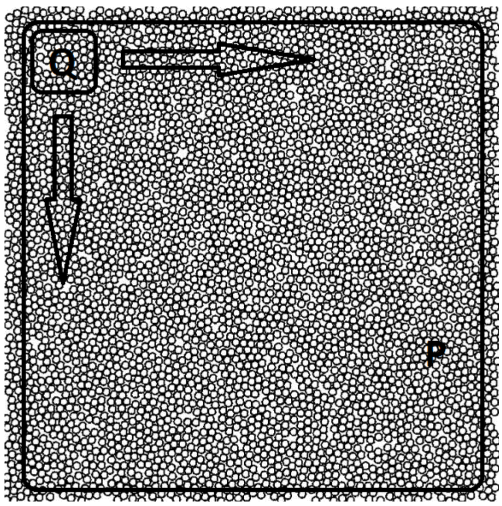


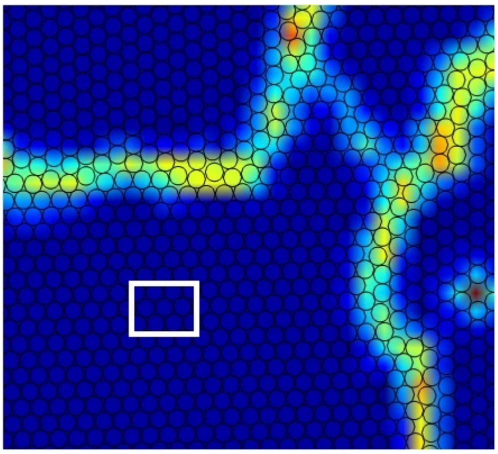
Figure 5.5: A disk packing generated by the LSA, showing where statistics are taken for P, and how the packing is then scanned for local statistics Q. These are then used to calculate the KLD.

Motifs are those parts of the structure that give more information about the rest of the system. This is usually because they are commonly found, or their statistics fit the system well. In the example of a crystal there is only one motif, the unit cell.

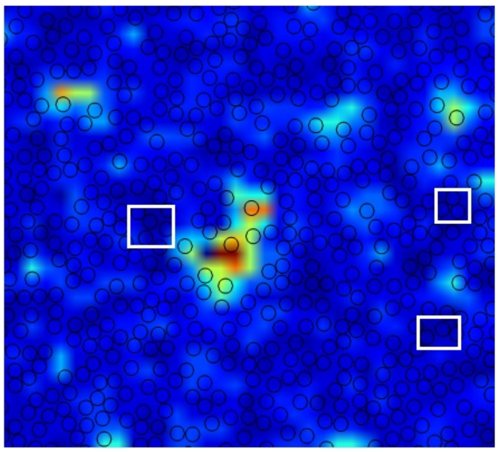
It is possible to build a picture for a given system using the KLD by visualising the information in a colour map, such as in figure (5.6). In this case, instead of scanning, a grid was overlaid on a structure, where $P(x)$ is the statistics for the whole packing, and $Q_n(x)$ describes the statistics in each sample where n denotes the grid samples sequentially. Therefore $Q_n \subset P$ and $\sum_1^n Q_x(n) = P(x)$.

$$D_{KL_n}(P|Q_n) = \sum_{x \epsilon X} p(x) \log_2 \left(\frac{P(x)}{Q_n(x)} \right) D_{KL}(P|Q) = \frac{1}{N} \sum_0^N \sum_{x \epsilon X} p(x) \log_2 \left(\frac{P(x)}{Q_n(x)} \right) \tag{5.31}$$

D_{KL_n} becomes a matrix of n elements to which a colour map can be superposed on to the packing, just as seen in figure (5.6). The white boxes show common motifs in low entropy areas. In these early results they are simply added by inspection.



(a) $\Phi = 0.9$



(b) $\Phi = 0.4$

Figure 5.6: Images showing early work identifying motifs using the KLD, with blue areas having the lowest entropy

By adjusting the sample size, the resolution and sensitivity of the KLD can be changed. For most sample sizes this is a cosmetic change only, however certain ranges are important for finding motifs. Too large a sample describes the system but saves no information, too small and you do not gain specific information about how the system is assembled. Initially the sample size was decided by a user-defined box size. Investigations with the KLD included results showing how sample size affected the KLD. During experiments on structure the size was held between 2-4, as many motifs are tetrahedral this was the most range interesting. Other sizes were used in an attempt to find larger structures that might repeat, but generally no significant motifs were found.

5.3.2 Self Referential Order

While the KLD did identify structures, it did not offer the quantitative information the MI did. This will be seen in the results, where the global MI offers more information about structural changes than the KLD. Work was made towards a new quantity that could combine the best of both the MI and the KLD, leading to the concept of Self Referential Order (SRO):

$$s(X;Y) = 1 - \frac{H(X|Y)}{H(X)} \quad . \tag{5.32}$$

I propose it as a novel measure to quantify the amount of information one part of a structure has about other parts. This can be again thought of trivially in terms of unit cells and crystalline packings. This is achieved by comparing the conditional entropy of a system, given some information about itself, with the overall entropy of a system. This is useful as it depends on no outside information, and Y must be a subset of X (equation 5.32).

Due to the importance of the SRO, it will be given far more scrutiny in chapter 7. I will simply provide some technical data here. The software package Matlab 2012 using the Statistics, parallel computing and image processing toolboxes was used to create the program that calculated the SRO. All the results were carried out using packings created by the LSA and statistics created by the Karambola program.

Chapter 6

Results - Quantifying Disorder

6.1 Entropy and Disorder

The main goal of this chapter was to test the viability of the new framework outlined in the methodology. Disordered structures were chosen due to their similarity with amorphous packings. I will look at how entropy can be used to characterise disorder, and apply it to known quantities in the field of packing problems.

It is important to see how entropic measures behaved for single variables before moving on to quantities such as the MI. Indeed in some cases it was useful to use the quantities either to calculate or create statistics for the MI and KLD. To begin with the global entropy of loose packings generated by RSA was calculated with results being generated by taking Voronoï cell volumes with a constant binning.

Figure (6.1) shows how entropy changed with growing sample size, achieved by taking statistics from a central region and then growing it to include more information. In such cases the results show a decreasing entropy, confirming larger sample sizes provided more information and therefore less uncertainty. Low entropy is expected for global statistics as the volume metric followed a normal distribution with low kurtosis, therefore trends observed shows high entropy only where the statistics are still sparse.

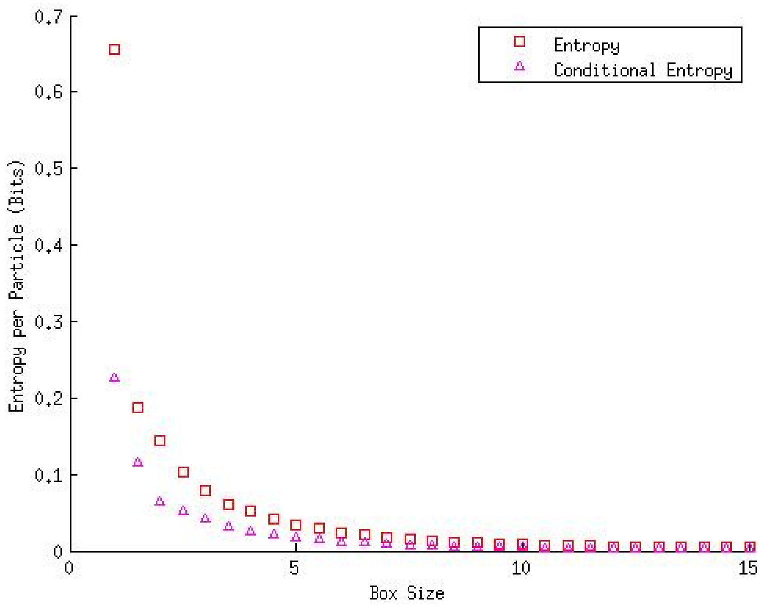


Figure 6.1: Results showing how entropy changed with growing sample size (box size) for a number of samples generated with the RSA

Sparse statistics don't allow enough information to describe the system, and in some cases data could be missing completely, creating a zero probability for common events.

Instead of changing sample size, it was also possible to take several samples in the same system. This was useful for comparison. Figure (6.2) shows a grid that corresponds to the actual sample size on a 2D mono-disperse disk packing. This grid is overlaid but the packing structure is not shown for clarity. For each box in the grid the entropy and number of particles were calculated. The entropy for figures (6.1) and (6.2) is measured using probabilities calculated from the adjacency matrix of volumes for nearest neighbours given a reference Voronoï cell. The adjacency matrix is defined as

$$M_{aj}(i,j) = \sum_p \sum_n \begin{cases} +1, \text{ if } f(p) = i \pm \delta b \text{ and } f(n) = j \pm \delta b \\ +0, \text{ otherwise} \end{cases}$$

where $f(p)$ is the volume of cell p and $f(n)$ is the volume of its n^{th} nearest neighbour. i, j and δb are all determined by the desired binning and range of cell volumes, therefore they define the bin centres and intervals (or rather size) respectively. Note the bin size is equal for all bins, which is the case throughout the results.

The results also showed the entropy did differ between packings, especially for changing packing fractions, however, it was clear for any single packing the magnitude remained close. While this doesn't mean anything in and of itself the similar values of entropy amongst samples taken from the same packing opens the possibility of shared information that could be measured between samples.

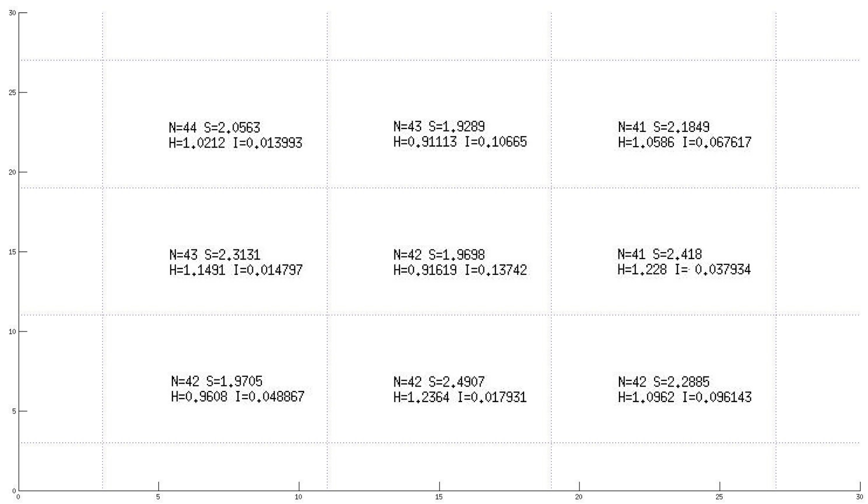


Figure 6.2: An image showing the entropy of various separated statistics of a single loose packing, where S is entropy, H is conditional entropy and N is the number of particles in the box

6.2 Mutual Information

To begin with Mutual Information was used to look at how information was translated through a packing. There has been much interest in the idea of long and short ranged disorder, which determines how the position of one or more particles effects those around it, and to what distance. This was important for two reasons: (1) To see how dependent the structure was on local formations, for example during crystallisation and (2) Gave advice on how packings should be created; how sensitive they are to initial conditions, how the boundary might affect a packing, and how large any 'null' border should be.

6.2.1 MI for Statistically Generated Data

There have been a number of papers demonstrating validation of MI for the use of quantifying correlation (Cellucci et al., 2005). These papers report on several quantities including Pearson's correlation coefficient and the MI. These reports show high values for most measures in linearly correlated data, generally tending to 1 as expected. MI is an exception as it is not bounded between zero and one, nor is it defined as completely correlated at one. Therefore the MI usually tends to around three bits for linear data.

In the normally distributed case these same measures would not detect any correlation, with values tending to zero, the only exception being the MI. The MI remains small compared to its value for linearly correlated data (between 0.1 and 0.2 bits). This, however, is significant, as the statistics used to describe disordered structures follow similar correlations, and so changes in packing structures would remain dectectable.

Finally, parabolically correlated data is presented, produced by applying $y = x^2 + c$ where x is a set of i.i.d. variables and c is some added noise. This is where MI is distinctive, with it remaining high (again close to three bits) but with other correlation values remaining close to zero (Cellucci et al., 2005). I verified that my methodology reproduced these results closely, giving confidence in the overall soundness of my approach.

Equation (6.1) shows how MI can be related to the Pearson correlation coefficient for normally distributed i.i.d. variables. This calculation in itself is interesting as it does not display bias and gives us a second way to calculate the mutual information. In order to calculate the mutual information in both ways efficiently a new code was created.

$$I(x, y) = -\frac{1}{2} \log_2(1 - \rho^2) \tag{6.1}$$

Instead of directly calculating the mutual information from a probability matrix, the new code created a 2x2N correlation matrix showing all pairs of correlated values. From the correlation matrix, the correlation coefficient could be calculated to check the results. Using equation (6.1) had the drawback of working only for normally distributed i.i.d. variables, however, this was accounted for when interpreting the results. A multivariate Gaussian random number generator (MVG) (Hernadvolgyi, 1998) was used to create inputs for the validation of the MI with statistically generated data, as it was far more easily controlled, and as such, verifiable. MVG data was not created for any of the structural data.

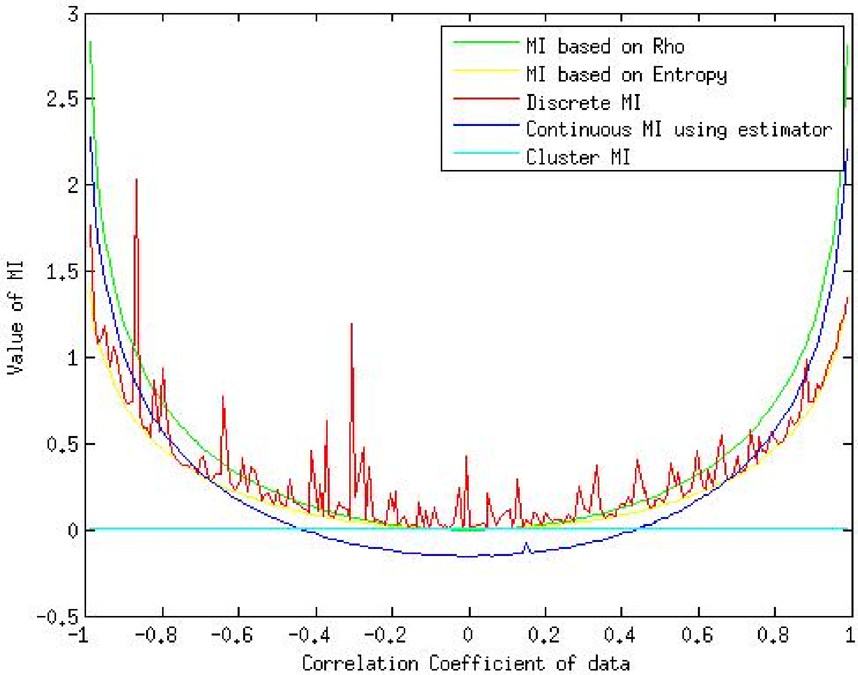


Figure 6.3: This figure shows the MI of statistically generated data for some standard deviation, giving a measured correlation coefficient (shown as the independent variable). The MI is calculated using a number of techniques as shown in the legend

The MVG allowed normally distributed data to be used with different values of covariance. This created two sets of points which were normally distributed, but with a known correlation between them. The correlation was recalculated each time using the correlation matrix as small fluctuations had to be accounted for (caused by the random nature of the variables). A histogram was then created following the general form $n = \sum_{i=1}^{j} M_i$ where k is the number of bins and M_i is the number of occurrences in each bin (i.e. the histogram). This means n is the total number of data points and allows probabilities to be expressed simply as $p(i) = \frac{M_i}{n}$. The probabilities are then used to calculate the MI in 5 different ways for a comprehensive test of various methods. These are listed below:

- Computed using MI estimated for a Gaussian as calculated in equation (6.1)
- Entropy estimated MI as calculated in equation (5.9)
- Discrete MI as shown in equation (5.5)
- Continuous MI using the kernel density estimator to approximate the discrete probabilities as a function, then applying equation (5.12)
- MI calculation based on clustering MI program created by (Slonim et al., 2005)

This list is relative to the legend in figure (6.3).

The results in figure (6.3) show that the MI did not change drastically with each method. That said, there are a few notable differences. The discrete MI, despite being close in values to the entropy based equation, fluctuated away from it. This is probably due to its highly sensitive nature, picking up on even small fluctuations. It suggests particular problems when dealing with fat-tailed distributions, where unlikely events add disproportionally to the MI. The clustering MI calculation didn't change at all remaining close to zero, as the program was not designed for structural data, and so I will dismiss it as an error.

Contrary to the known rules for MI, it can be seen the continuous MI becomes negative, dropping below zero for the range $\rho \approx -0.45 \rightarrow 0.45$. This is because the probabilities are now estimated and do not necessarily sum to one, therefore Jensen's Inequality breaks and the MI can be negative. Practically this means $H(X,Y) \geq H(X) + H(Y)$ and the MI becomes negative, albeit for ranges where the data is less correlated and therefore the estimation error is relatively large. A second normalisation would have prevented such a problem, but was not done in this case for clarity when comparing the methods.

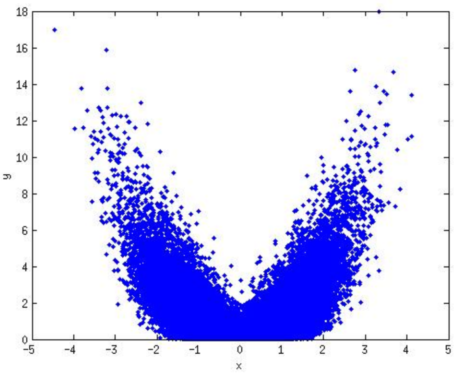


Figure 6.4: Figure showing spread of points generated by the MVG with $\rho = 0.9$ and with y being the second set of related points squared so they are parabolically related

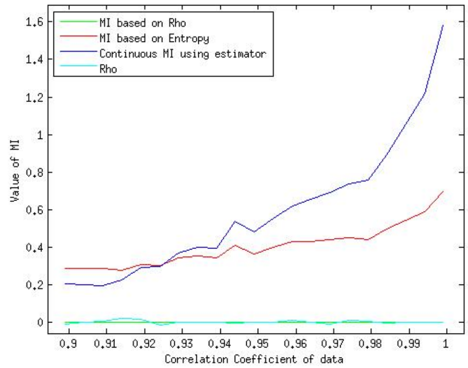


Figure 6.5: This figure shows the MI of statistically generated data for some standard deviation which has been parabolically related (figure 6.4). The MI is calculated using a number of techniques as described in the legend

Results in figure (6.5) show the trend of MI for the parabolically related data in figure (6.4), as described by $y = x^2 + c$, seen earlier in this section. The correlation coefficient is expressed for the independent variable x. However, it has been recalculated for y, defined simply as Rho, and added as a cyan line to figure (6.5). This describes, as one would expect, a very low value of ρ and consequently a very low value for the Guassianly approximated MI. This is also true of the experimental data with the values of ρ being very low compared to the discrete MI. The continuous MI is again shown to be the most sensitive in these results. The binning for these results was set at 15 bins.

It was useful to compare statistically generated results, as it was also possible to change the binning of the probability matrix while keeping the correlation constant, thereby allowing a comparison of the methods outlined in section (5.2.2). This was unfortunately only of limited use as packing data generally exhibits little linear correlation, therefore the correlation measures tended to vary by three orders of magnitude making an exact comparison difficult.

6.2.2 MI for Mono-disperse Hard Disk Packings

This section deals with data generated with structural information extracted from disk packings made with the LSA and RSA programs. Datasets are generated by taking statistics for samples inside boxes overlaid on mono-disperse hard disk packings (example 6.6). The results shown in the following figures use discrete data, and as such the discrete form of the MI is applied:

$$I(x,y) = \sum_{y \epsilon Y} \sum_{x \epsilon X} p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{6.2}$$

As with most results a padding was included to compensate for the fact a hardwall boundary condition had been used in their creation, set at 10% the width of the container (ensuring a minimum of several disk widths were excluded). The results presented were taken from packings made using RSA but were later revised using the LSA. One such set of results is presented for comparison in figure (6.9). The padding was given as $d \geq \frac{a-3}{2}$ with a as the frequency of boxes and d is distance in units of boxes. To allow for a simple case of expressing the boundary in disk diameters all points were normalised to give their radius as 1 so that:

$$\frac{M_i}{2r} \forall i \tag{6.3}$$

where M_i is the entry for the i^{th} point and contains three numbers: two for the coordinates of the disk, and one defining the disk diameter. All three numbers for all i points are divided by $2r$ or two times the disk radius.

6.2.2.1 Mutual Information for n^{th} neighbour samples

In these two-dimensional cases statistics are extracted by splitting up the packing into boxes, this allows a simple grid to be imposed of which the size is also more easily manipulated. Boxes are also trivial to extend to the three-dimensional case. A reference

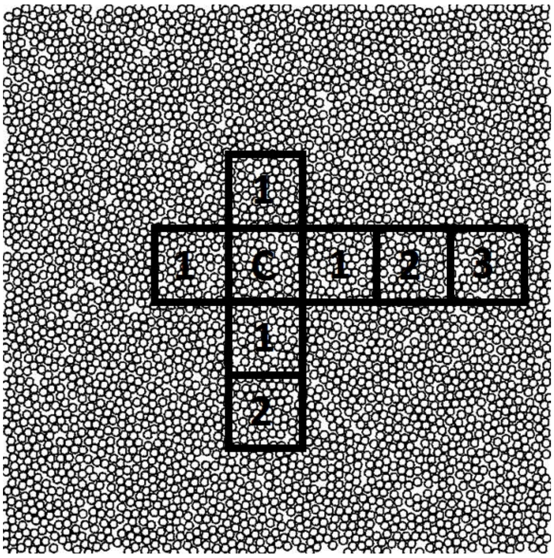
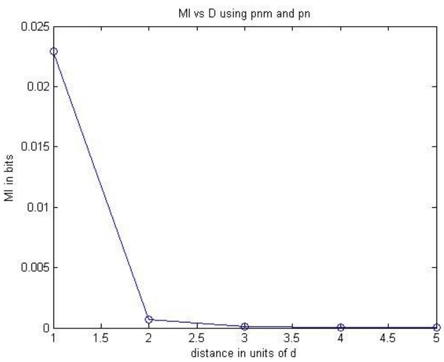
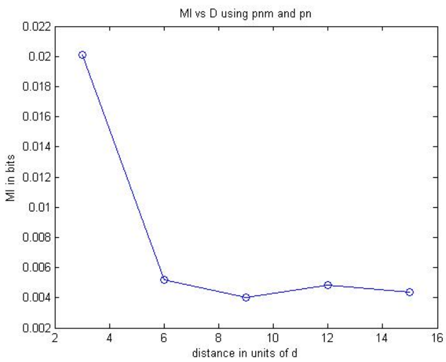


Figure 6.6: Example showing how mutual information is calculated. One variable being taken from the reference box 'C' while the other is taken from n^{th} neighbour boxes.

box is selected (labelled C in figure 6.6), while boxes of the same size are constructed around it. As can be seen in figure (6.6) subsequent boxes are constructed so the distance between their centres is an integer number of box lengths. The boxes labelled one, have centres one box length away, and are the 1^{st} neighbours. Some 2^{nd} and 3^{rd} neighbours have been added as well.



(a) Graph relating MI to distance (in units of disk diameter) for a packing consisting of 58884 disks with a packing fraction of 0.52 where d=1 refers to 1st neighbour boxes d=2 to the second neighbour boxes and so on



(b) Graph relating MI to distance (in units of disk diameter) for a packing consisting of 26457 disks with a packing fraction of 0.52 where d=1 refers to 1st neighbour boxes d=2 to the second neighbour boxes and so on

Figure 6.7: A comparison of MI results using different statistics for the probability function. Figure (a) uses number of disks per sample, while (b) uses the number of neighbours per disk.

Three sets of data were extracted from each box; the number of disks n, the numbers of Voronoï neighbours (calculated as the degree of that disk, based on the Delaunay triangulation), and the Shannon entropy. These all give similar trends in MI as shown in figure (6.7), so unless stated otherwise the initial results are given for the disk occupancy case. The results in figure (6.7) and others show little difference in MI for the given statistics, as well as suggest the shared information between boxes diminishes rapidly. This suggests perhaps sensibly that the positions in one area do not have much influence on any but their closest neighbours. Different statistics will be given further consideration in the next chapter with the use of Minkowski tensors.

After the packing was placed into regions created by a grid the occupation number was calculated as well as the global covariance and covariance distances. This allowed the $N \times N$ adjacency matrix M to be created by adding each cells co-occurrence to the correct line.

$$\begin{aligned}
&C_{i,j} \text{ has neighbours } C_{i+1,j+1}C_{i+1,j-1}C_{i-1,j+1}C_{i-1,j-1} \\
&\text{with occupancy values } n_0, n_1, n_2, n_3 \text{ and } n_4, \text{ then} \\
&M_{n_0,n_m} = M_{n_0,n_m} + 1 \text{ where } m \forall n \text{ given } 0 \leq n \leq N
\end{aligned} \tag{6.4}$$

In addition to this two $N \times 1$ matrices were created for the frequency of occupancy, as the number of boxes in x and y is equal the frequency vector F for each follows $F_x = (F_y)^T$. By dividing through each matrix by the sum of its elements $\sum M = \sum_{i \in I} \sum_{j \in J} M_{i,j}$, the values of $p(x)$ and $p(y)$ can be found from the frequency matrix, and $p(x, y)$ from the co-occurrence matrix.

6.2.2.2 Mutual Information for varying sample size of 1^{st} neighbours

The program was created to allow for different box sizes. This was to see how larger samples effected the entropy (as seen previously). It became apparent however, that it would be more desirable to alter the distance by function of box size. The program ran as before but the dependent value became the box size. Taking figure (6.8) as an example, the distance d remained the same at $d = 0$ (for 1^{st} neighbours). However the distance between the box centres, and by extension the mean distance between the disks in each sample increased with box length b.

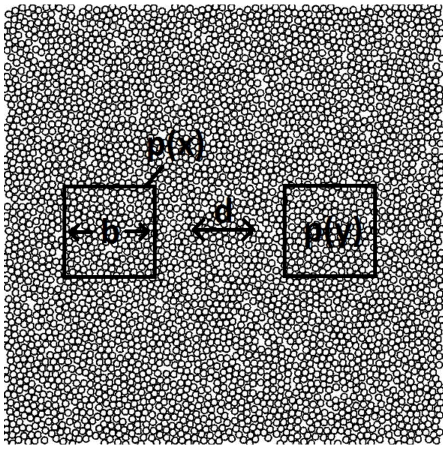
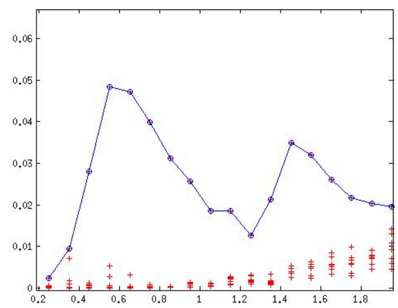
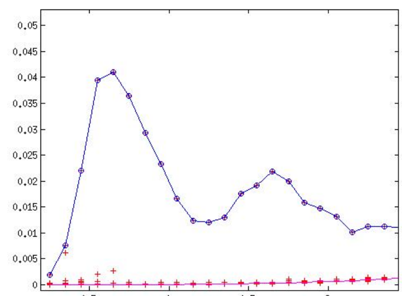


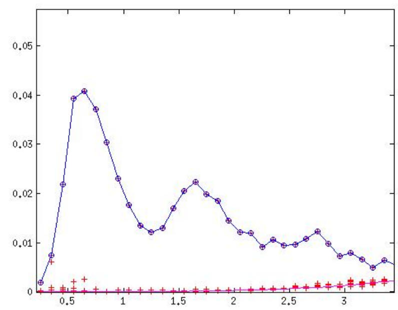
Figure 6.8: Example showing how mutual information is calculated. The two boxes both offer one probability distribution. The distance between them (d), and their size (b), can be adjusted as needed.



(a) N = 10000 disks with $\Phi = 0.60$



(b) N = 50000 disks with $\Phi = 0.53$



(c) N = 100000 disks with $\Phi = 0.53$

Figure 6.9: MI for nearest neighbours (blue points) as a function of box size with various numbers of particles (N) created using the LSA, red crosses show MI for additional n^{th} neighbour boxes

It was apparent that looking at boxes far away
from the reference sample was not significant (figure 6.7), therefore figure (6.9) and subsequent results only shows nearest boxes that have been grown in size (although others were calculated). The smaller box sizes show how the statistics of the reference box rely greatly on those around it (figure 6.7). Indeed when compared to the other neighbours it is shown they offered very little information, for example in figure (6.9) and (6.10).

These results, showing that the information only travels a short distance through a structure, confirms the padding is more than sufficient to remove any skewed cell statistics caused by the container boundary.

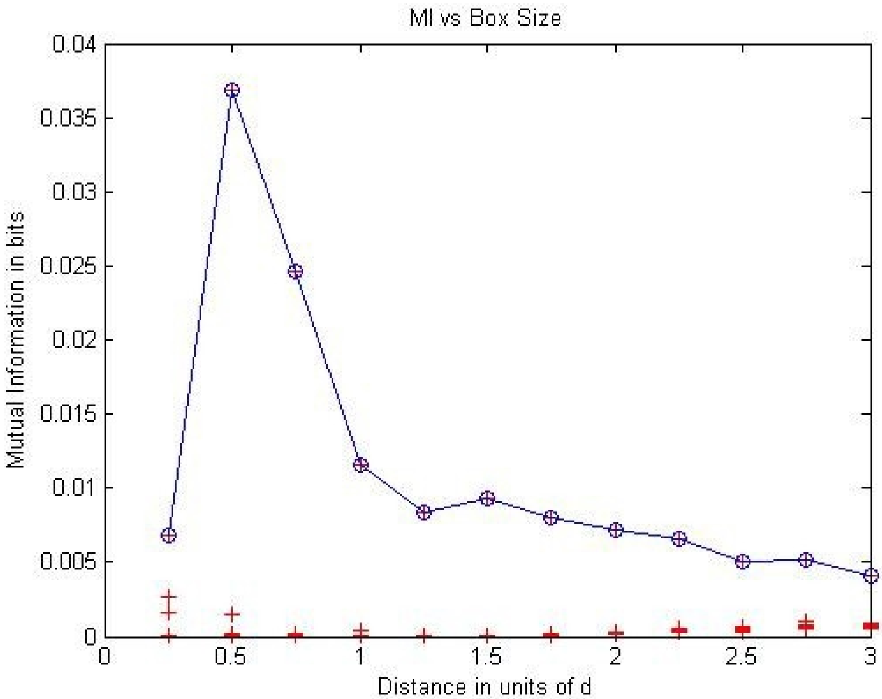
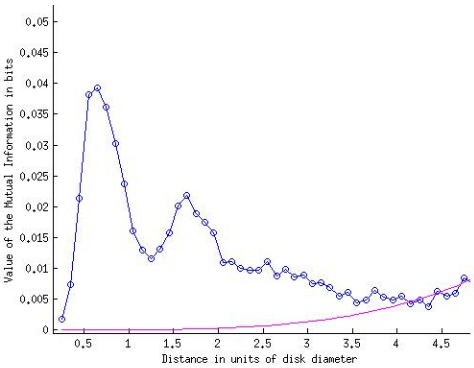
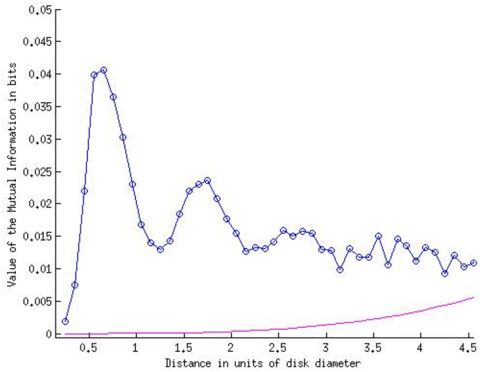


Figure 6.10: MI for nearest neighbours (in blue) as a function of box size for a packing of 58884 disks with a packing fraction of 0.52. Red crosses show 2^{nd}, 3^{rd} and 4^{th} neighbours.

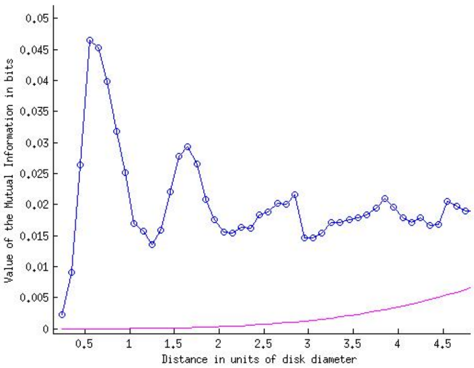
The graphs in figure (6.9) show little change in trend or absolute value of the MI between packing comprised of differing numbers of disks, depending more on the packing fraction as seen in (6.11) and (6.12). The smallest box sizes give a close to binary state, with boxes either occupied by one or no disks. For box sizes ≥ 0.707 this is trivially always true as disks do not overlap and $2a^2 = 2r \therefore a = \sqrt{\frac{1}{2}}$ where r is the radius of the disks set to 0.5. Therefore the optimal method for setting a box size is to define the box occupancy or sample size. This should be generally a minimum of one to avoid a sparse matrix, but not be so large as to allow the sample to describe the whole system, typically a few disks.



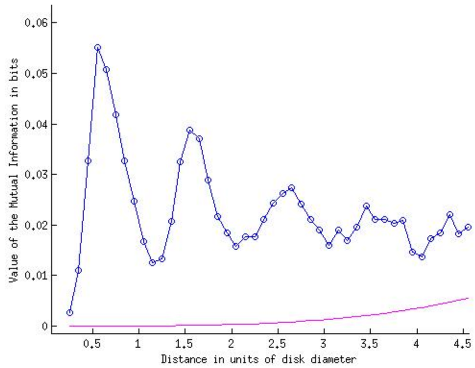
(a) $\Phi = 0.52$



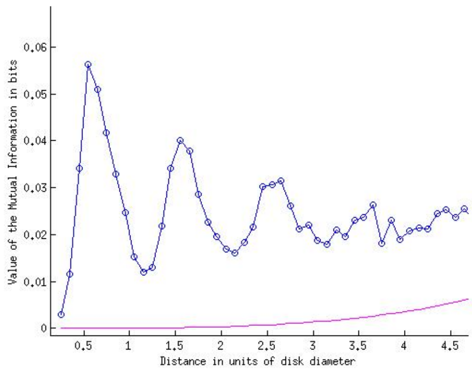
(b) $\Phi = 0.53$



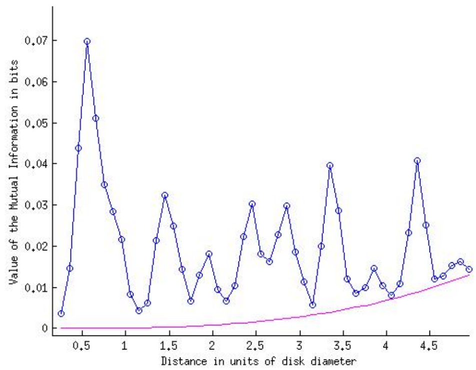
(c) $\Phi = 0.58$



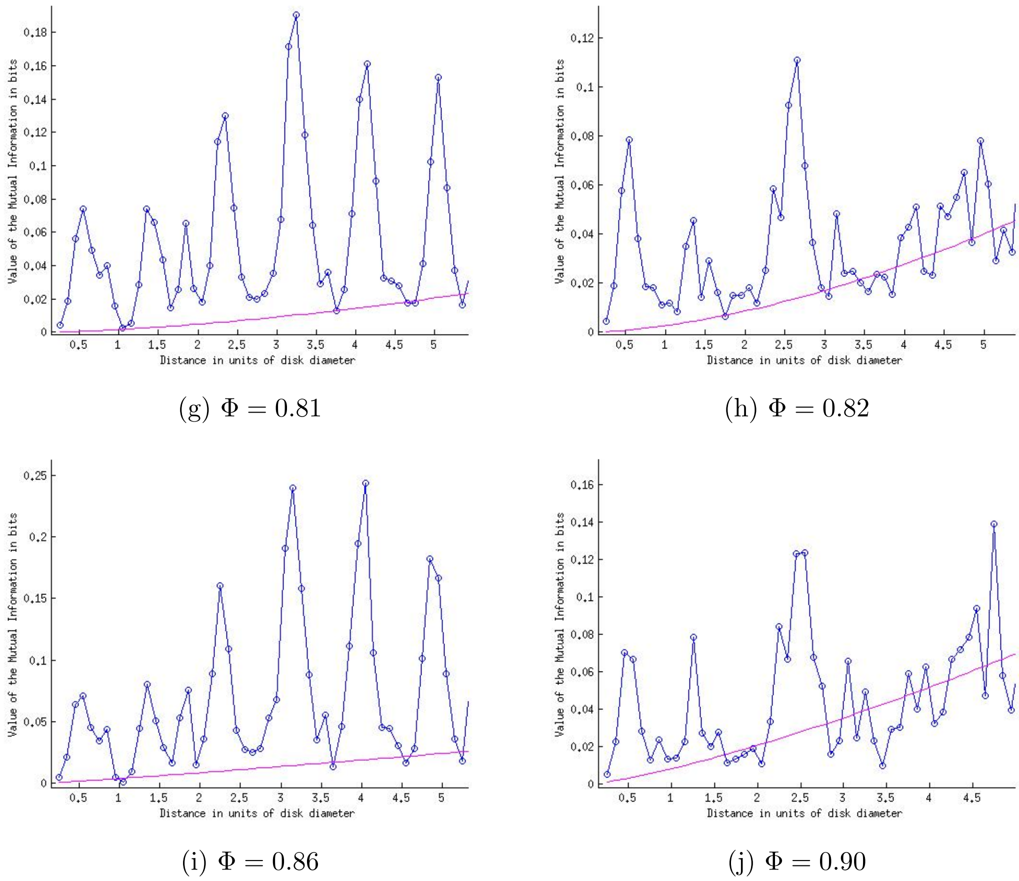
(d) $\Phi = 0.63$



(e) $\Phi = 0.65$



(f) $\Phi = 0.72$



(g) $\Phi = 0.81$

(h) $\Phi = 0.82$

(i) $\Phi = 0.86$

(j) $\Phi = 0.90$

Figure 6.11: Various sets of packings generated by the LSA. Each packing consisted of 100000 disks. Points show box sizes where the MI has been calculated. The smooth line shows how MI increases due to the probability binning

Using the LSA for results in figure (6.11) allowed much higher packing fractions than the RSA. These results look at the relationship of MI given some box size, used as a unit of distance.

Let us look first at the lower packing fractions shown in figure (6.11a) (b) and (c), these confirm the previous results that MI drops off quite quickly with increasing distance. This makes logical sense, the further away something is in a physical system the less effect it should have. What is most interesting is the maximums, with a primary peak ~ 0.5 to $0.7d$ and a secondary peak ~ 1.5.

Looking back at figure (4.11), the radial distribution function, there is a close resemblance in form, in respect to the peak locations. While we could modify the RDF for these packings by taking $\Phi = \frac{N}{V}$ and modify the RDF for two dimensions, giving $g(r) = \frac{n(r)}{\Phi\pi 2r\delta r}$ there is still some disagreement in x values. An explanation can be found in our definition of the shell dr, which would not be a circle (as for the RDF), but a square.

The results in figure (6.11) show us that as the sample box length is grown there is no information before $d \approx 0.5$ as, like for the RDF, no other disks are present. As the border of the sample box encounters new disks there are jumps in the information about the packing. During intermediate expansion phases the boxes are growing into empty space, offering no information about the position of additional disks and most importantly giving no mutual information based on the reference box.

The value of the peaks is also important, as it can be thought of as the true MI for that system. Higher values for the MI maximum means more information is available in that system, therefore the MI can be thought of in relation to the height of these peaks.

The other maximums provide insight into how information is moving through a system. The greater the subsequent peaks are, the more information is being carried through the system. This effectively shows a *range of influence*. In the case of the last four graphs it can be seen that the MI actually grows larger in some peaks. Larger peaks may be the result of increased statistical data and a well defined distribution. Because of dislocations in data beyond the RCP limit, some data describes the crystal structure well and others don't, leading to distorted peaks. This is probably due to the use of a fixed reference cell. As work with the KLD has shown, the reference sample location should not be arbitrary chosen. Combining the two techniques for this purpose could be interesting for future work.

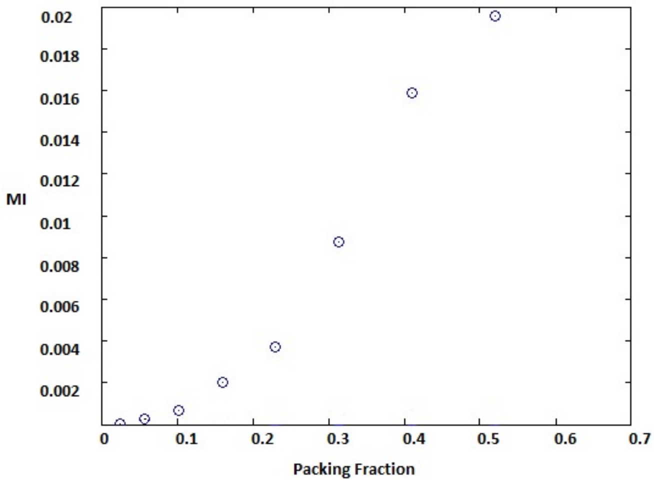


Figure 6.12: Figure showing MI vs Packing fraction, for RSA created packings of 73076 disks with varying disk diameters

6.2.2.3 MI for Mono-disperse Hard Disk Structures with a Static Sample Size

Turning our attention back to the first three results in figures (6.11a), (b) and (c), the same maximums show a different trend, with a well defined first peak, some information from the second peak, and almost none from subsequent peaks.

It is interesting to plot these changes in absolute value over packings with different values of Φ, as shown in figure (6.12). In this figure several packing have been generated using the RSA that comprise the same number of disks. Therefore to achieve changes in packing fraction the size of the whole structure must be reduced. At the same time the sample size is kept static. The MI is calculated as before, by taking statistics from these sample boxes and comparing two neighbours. The results then show how MI changes as a system moves from a gaseous like state to a solid one.

As can be seen in figure (6.12) the value of the MI trends upward as we move towards a less disordered system, revealing more information is carried through the system as it becomes better defined in terms of a single cell. That is to say that a local configuration is affected by, and effects, parts of the system far away. This shows long-range order, just like a unit cell does for a crystal structure.

Figure (6.12) again shows packings with different boundary sizes, allowing the packing fraction to change but leaving the total number of disks static. The graph shows how the MI decreases with less dense structures for 1^{st} neighbours with a constant box size. 1^{st} neighbours refer to the sample boxes touching, with their centres being one box length away. These results were quite limited as they were produced with the RSA. They can be extended by using LSA generated data as shown in figure (6.13).

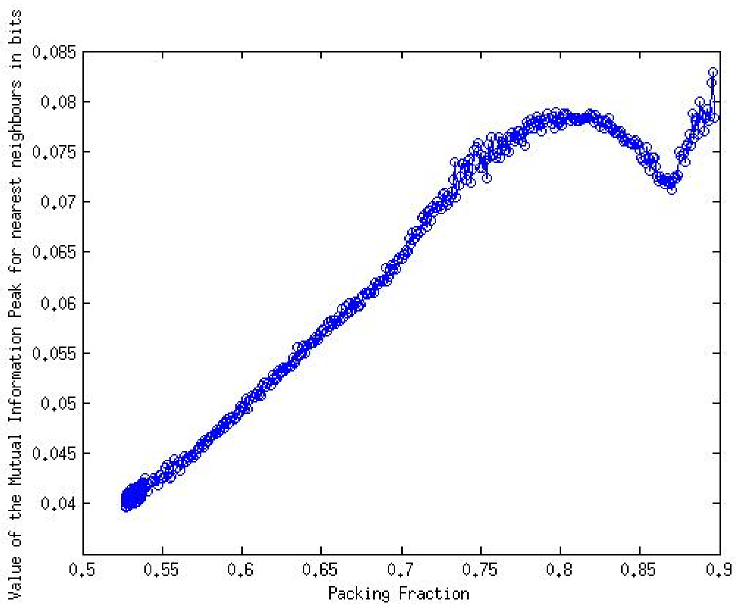


Figure 6.13: Results showing peak MI for various packings generated using the LSA, with increasing values of Φ and a fixed sample size.

Figure (6.13) presents data collected from multiple packings generated with the LSA, which has allowed a much greater range of packing fractions than in figure (6.12). The number of disks has been kept constant. The MI was calculated using Voronoï cell statistics, specifically the number of edges. From these, the co-occurrences for each cells nearest neighbours was calculated. This gave a probability distribution for each sample. The MI was then calculated for a set distance of 1 box length (the diameter of one disk).

As discussed in section (2.1.3.2) the RCP in two dimensions is believed to be close to $\Phi \approx 0.82$ and the RLP limit has not been proven as of yet (Meyer et al., 2005). In figure (6.13) we can see an increasing trend in MI as expected, however a turning point is clearly observed at $\Phi \approx 0.8$, before a slight drop, after which the MIs upward trend resumes. This suggests evidence of the RCP limit in 2D disk packings. As crystallisation begins beyond the RCP limit the system has two distinct phases, a disordered phase and a crystallised phase. While this initially reduces the 'shared' information in the system, once the crystallised phase becomes dominant it quickly and easily describes the system causing a rapid increase in MI. As the system crystallises a drop in the MI is observed because the crystallised phase is seen as an aberration, this is clear evidence of a change in structure.

Given these interesting results the peak MI was also investigated to see how information is passing through the system. As we have already seen in figure (6.11) the MI has several maximums when measured for increasing distance. These peaks follow the form of the RDF, but there value has not been investigated. By measuring the maximum values for MI between $d = 0$ and $d = 1$ we can find the value of the MI for the 1^{st} peak. This was done for several ranges as can be seen in figures (6.14), (6.15), (6.16) and (6.17). Taking the maximum MI in a range does mean the results presented here have slightly differing sample sizes, albeit in a constrained range.

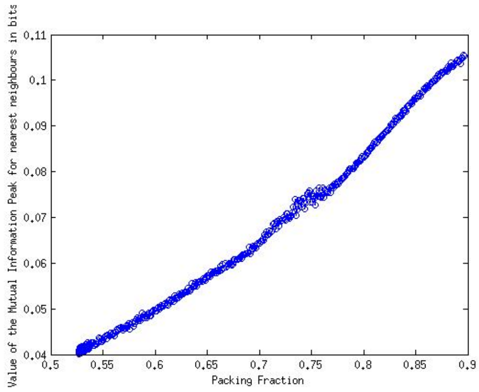


Figure 6.14: Figure showing value for 1^{st} MI peaks, corresponding to a distance of ≈ 1 disk diameter from the reference object

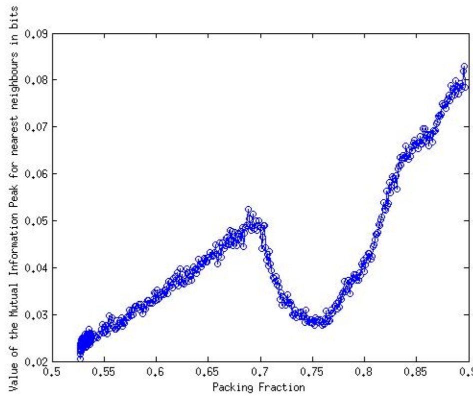


Figure 6.15: Figure showing value for 2^{nd} MI peaks, corresponding to a distance of ≈ 2 disk diameters from the reference object

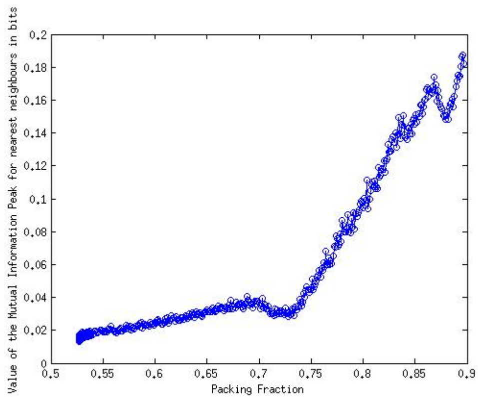


Figure 6.16: Figure showing value for 3^{rd} MI peaks, corresponding to a distance of ≈ 3 disk diameters from the reference object

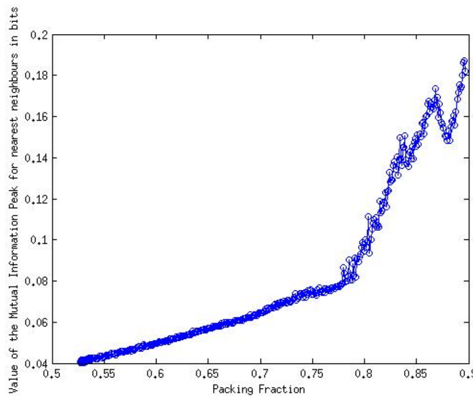


Figure 6.17: Figure showing value of maximum MI for a large range of distance from the centre

In figure (6.17) we can see little information is gained before $\Phi \geq 0.86$ when long range order from crystallisation sets in. That said, if we look at figures (6.14), (6.15) and (6.16) some very interesting results can be seen. The growth in MI is seen in the 1st, 2nd and 3rd MI peaks, the farther away we get from the reference, the shallower the gradient at low packing fractions (as one would expect given the previous results in figures (6.10), (6.11) and (6.12)).

There also seems to be a change in gradient at around $\Phi \sim 0.85$, although this isn't as clear as in figure (6.13) which shows the overall trend in MI. There is, however, a clear change in MI between 0.7 and 0.8. This appears as a broadening of the data in the 1^{st} MI peak (Figure 6.14), but a clear change in figures (6.15) and (6.16). This could be strong evidence of an RLP limit at $\Phi \sim 0.7$, while it is a little outside the scope of this project as a proof of concept, it is something that may warrant further attention in future work.

6.3 Results - Kullback Leibler Divergence

Kullback Leibler divergence (KLD) as discussed in section (5.3) is a relative measure of information. KLD can be used to calculate the information one part of a structure gives to its whole. This makes it very useful in looking for motifs, areas containing high amounts of information about the structure. The KLD also quantifies the remaining information for a given motif.

Another compelling reason for using the KLD over the MI is the absences of bias, discussed earlier in section (5.2.2). As we are dividing one distribution by another (equation 5.26), the binning can be fixed for any given sample size. Unfortunately this presented another issue, as in early results the KLD was found to violate the condition $D_{KL}(P|Q) \geq 0$.

After investigation, it became apparent that binning was again to blame. For small sample sizes of $Q(x)$ gaps could be present in the discrete probability matrix. Because the KLD is only defined if $P(x) \neq 0 \ \forall \ Q(x) \neq 0$ a sparse matrix for $Q(x)$ could cause problems. Put simply you can not compare probabilities if you have holes in your model. Therefore a kernel density operator had to be applied allowing for smooth statistics, and the continuous form of the KLD was used (equation 6.5).

$$D_{KL}(P|Q) = \int_{-\infty}^{\infty} P(x) \log \frac{Q(x)}{P(x)} \mathrm{d}x \tag{6.5}$$

Two types of interpolation where used when creating fit functions to the probability density functions $P(x)$ and $Q(x)$. The first was a piecewise linear interpolation in which

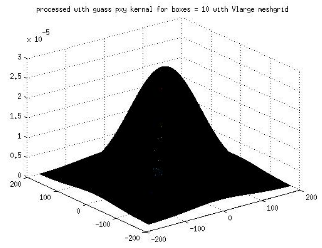


Figure 6.18: Fit of Voronoï cell area data using a Gaussian interpolation with large meshgrid, boxsize = 10

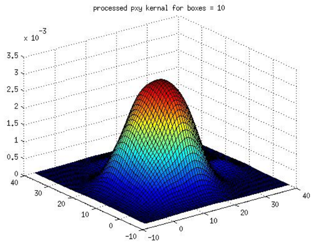


Figure 6.19: Fit of Voronoï cell area data using a piecewise interpolation, boxsize = 10

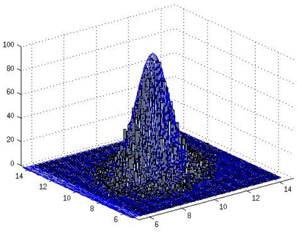


Figure 6.20: Histogram of Voronoï cell area data overlaid with a piecewise interpolated surface, boxsize = 10

the closest known points are taken to evaluate the function using straight lines (linear) between points. The second is a cubic piecewise interpolation which does much the same thing, only using a cubic to fit between points. Cubic interpolation takes much longer, a standard run taking $t = 1522$ seconds compared to $t = 360$ seconds for linear interpolation. As the time taken for the whole program is linear (that is to say, time = number of loops x time taken for one loop), this means cubic interpolation typically takes 4 times longer.

The method was again checked with known values for statistically generated data which shows that normally distributed random data has the least amount of shared information present. For statistically generated polynomial distributions (including linear, quadratic and cubic) a similar value of the KLD is measured, showing the method is sensitive to all polynomial correlation as one would expect.

6.3.1 Motifs

As we have seen (section 5.3.1), motifs are the set of local structures from which the whole structure can be most efficiently encoded. A number of factors affect how descriptive a single motif might be including: frequency of occurrence, fluctuations in that motif, and the number of rules need for self-assembly. The fewer motifs that describe the structure the more efficiently it can be coded, similar to how information is compressed by describing it with the shortest code possible (Kolmogorov complexity).

The amount of disorder can then be quantified based on the number of motifs. This requires identifying common motifs and the rules governing their combination into a space-filling network.

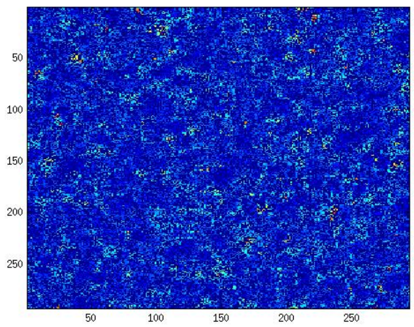


Figure 6.21: Plot mapping the value of the KLD using a colour map, with blue being low KLD, $Q(x)$ taken from a box size of 10. Each box was then compared to the whole packing $P(x)$

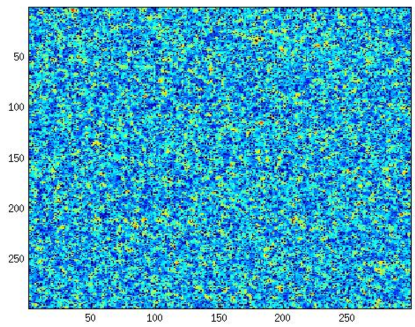
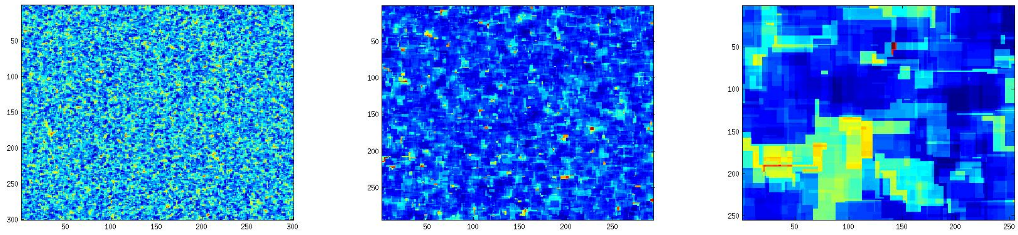


Figure 6.22: Plot mapping the value of the KLD using a colour map, with blue being low KLD, $Q(x)$ taken from a box size of 4. Each box was then compared to the whole packing $P(x)$

The first efforts to find motifs were also done using a kernel density estimator to calculate the KLD. After a correlation matrix was created using Voronoï cell areas two sets of data were extracted, the first was the probability distribution for the whole packing, the second was based on a sample of the packing, serving as P(x) and Q(x) respectively (equation 5.26). While a number of fit types where used, a piecewise linear interpolation was found to be best, although there was not much difference. This created a fit object, which was a surface, that could be used to create a function that was an estimate of the probability density function. Therefore no binning was needed for the probability, and the KLD was computed by evaluating the new function directly. The sensitivity of the kernel was decided by a meshgrid, which effectively changed the resolution of the plot as shown in figure (6.18). It had the disadvantage of being quite slow compared to the discrete methods.



(a) Sample taken from a boxsize of 4

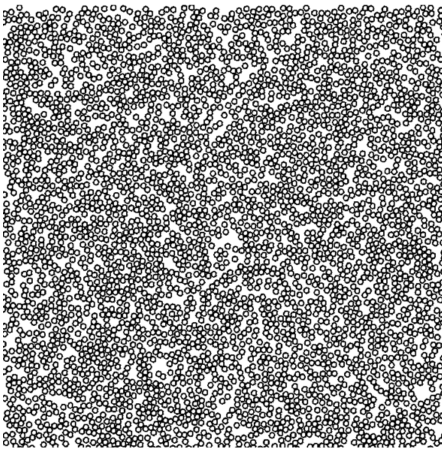
(b) Sample taken from a boxsize of 10

(c) Sample taken from a boxsize of 50

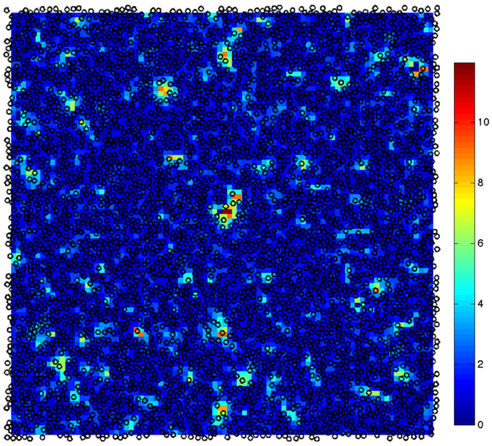
Figure 6.23: KLD colour maps for various sample sizes, all samples are given on the same scales for comparison

As figures (6.21), (6.22) and (6.23) show the smaller sample size increased resolution, and a greater range of values by keeping statistics low. This would be necessary to find useful motifs as large sample sizes would require more information to encode. In other words, as $Q(x) \to P(x)$ no useful information would be gained as the alphabet would be as large as the encoded object.

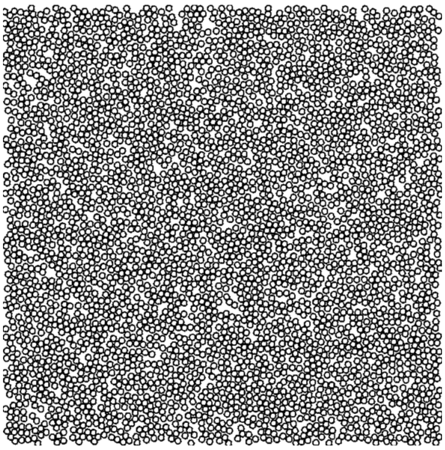
6.3.2 Relative Measures



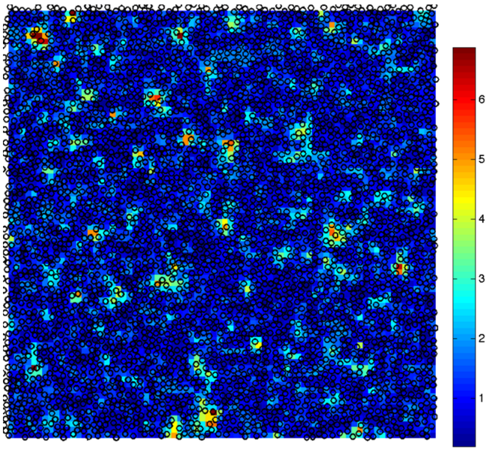
(a) Packing of disks with $\Phi = 0.4$



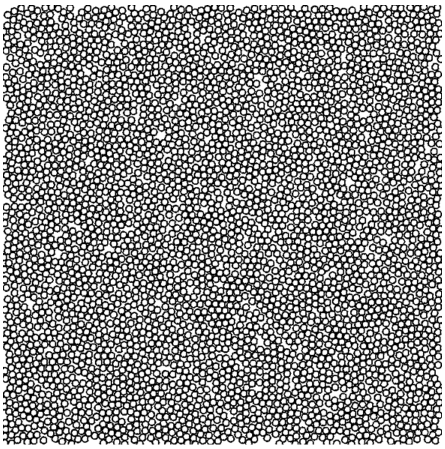
(b) KLD colour map of figure (a)



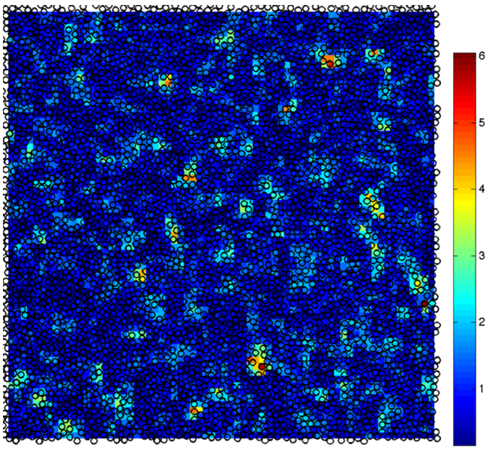
(c) Packing of disks with $\Phi = 0.5$



(d) KLD colour map of figure (c)

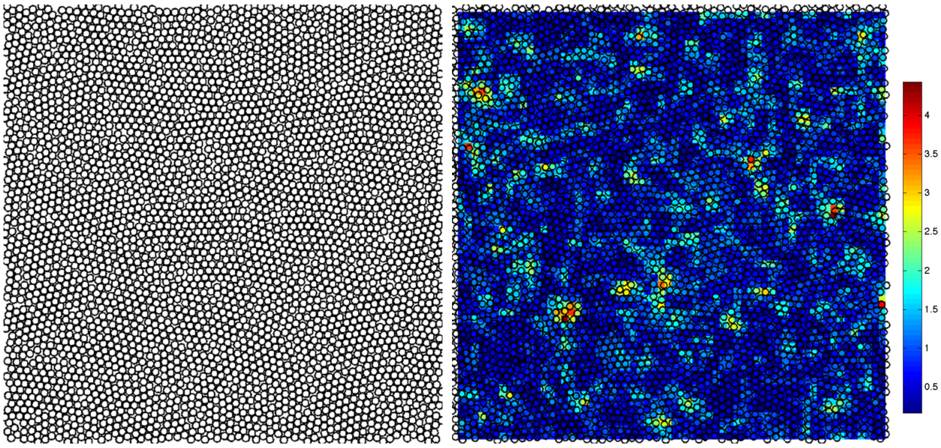


(e) Packing of disks with $\Phi = 0.6$

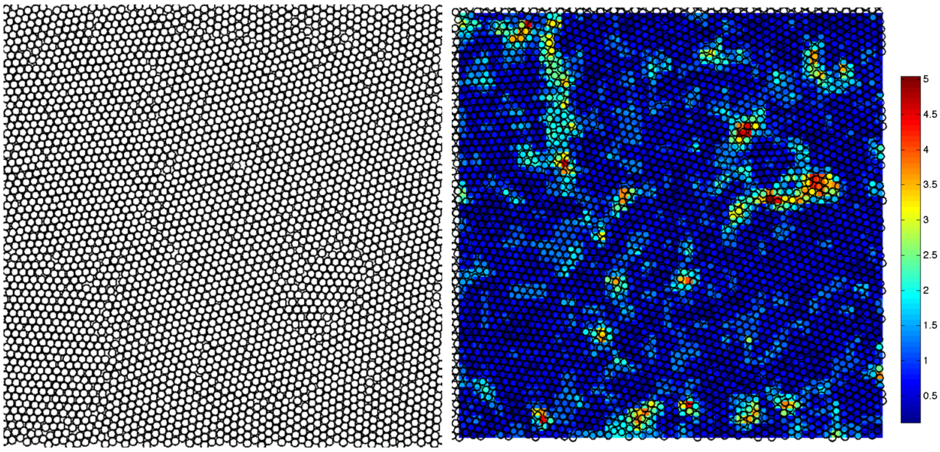


(f) KLD colour map of figure (e)

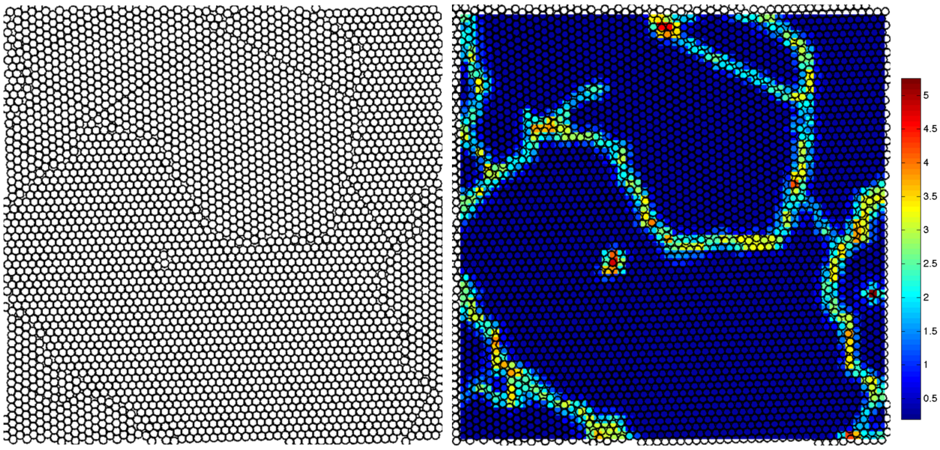
Figure 6.24: Colour maps for various RSA packings of differing packing fractions (Figure 1 of 2)



(a) Packing of disks with $\Phi = 0.7$ (b) KLD colour map of figure (a)



(c) Packing of disks with $\Phi = 0.8$ (d) KLD colour map of figure (c)



(e) Packing of disks with $\Phi = 0.9$ (f) KLD colour map of figure (e)

Figure 6.25: Colour maps for various RSA packings of differing packing fractions (Figure 2 of 2)

To compare packings variables such as the sample size and meshgrid can be held. By constraining these properties it was possible to look at various different packings as shown in figures (6.24) and (6.25). These images paint a picture of how the KLD and the structure change, from the seemingly random fluctuations for low packing fractions to the stark contrast of $\Phi = 0.9$ with clear areas of crystallisation. In these high Φ structures areas of high KLD are located only at the dis-joints.

It can also be seen from the colour map keys how the KLD decreases as the packing fraction is increased. This can be seen much clearer in figure (6.26), and shows more information is found in one sample of the structure if the density higher. The results also show the KLD tending to zero as the comparison sample $Q(x)$ is itself taken from the final packing, showing rapid crystallisation after $\Phi \sim 0.7$ The average KLD could be calculated for the colour maps, thus giving a value of the KLD for the whole packing, using equation (5.31).

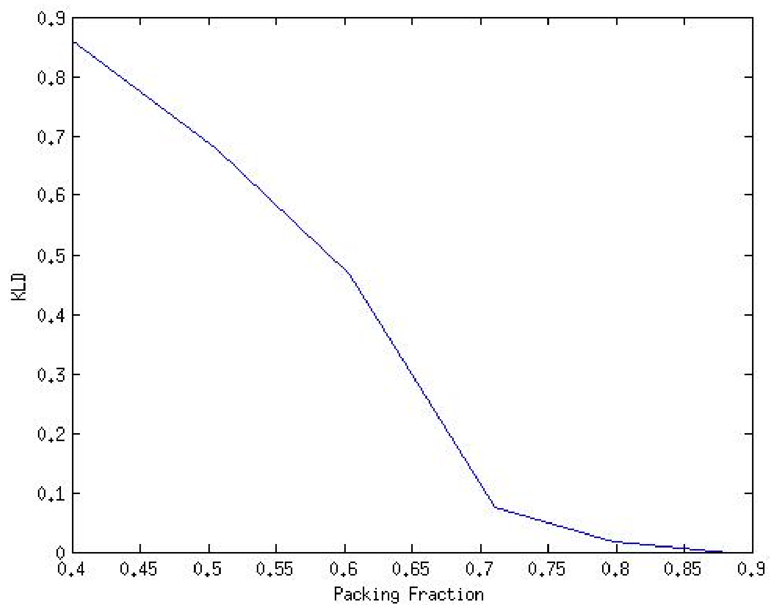


Figure 6.26: The average KLD for several sets of RSA packings with similar numbers of disks, but with different packing fractions where $Q(x)$ is a constant taken from a crystalline packing

Larger sample sizes causes $Q(x) \to P(x)$, this can be seen clearly in figure (6.27). It can be seen as $Q(x)$ goes to 100% of the container size the KLD $\to 0$. Figure (6.27) also demonstrates that the number of disks has little effect on the KLD. Given significant differences, it could skew the statistics, however this is mitigated by use of the kernel density filter.

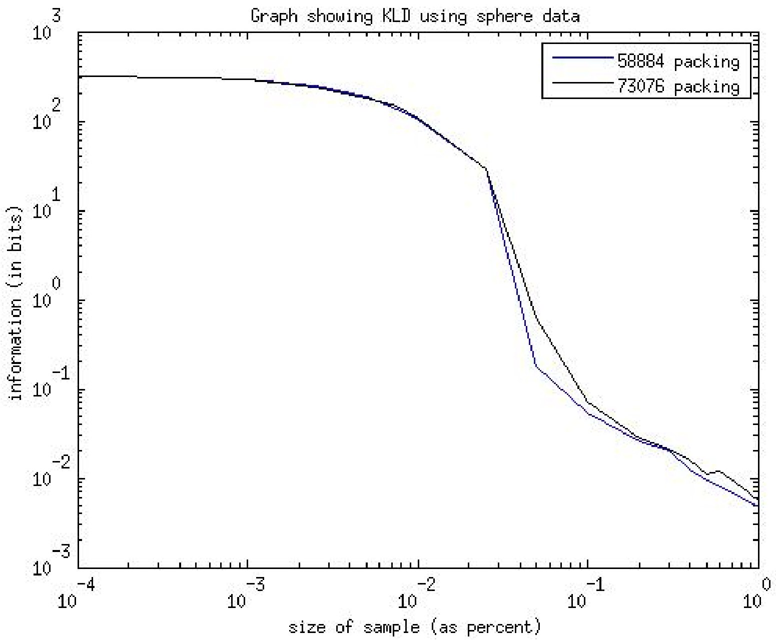


Figure 6.27: A figure of the KLD for two RSA packings with different number of spheres but similar Φ, where the sample size is grown in both packings

Chapter 7

Results - Self Referential Order in 2D

7.1 Proof of Concept

7.1.1 Self Referential Order

Development of information theoretic approaches showed it was possible to quantify disorder in some manner. However it was clear that refinement was needed, in the hopes of combining the most useful elements of all the methods tried so far. Work with the KLD had lead to the idea of self reference, the idea that in the absence of a single descriptive structure (i.e. the unit cell), it was still possible to use part of the material to characterise the rest.

A way to quantify this was required, leading to a definition of *self-referential order*. In the case of a crystalline structure, it can be seen that the unit cell is a very effective descriptor. Instead of requiring data from the whole packing, it can be encoded as a much smaller quantity, the unit cell. The question then becomes is such a compression possible for disordered structures.

A number of paragraphs in the next chapter are excerpts from the paper *Self-Referential Order* (Butler et al., 2013) which can be found in the appendix. This applies to some parts of Sections (7.1), (7.2) and (7.3).

In this chapter a simple case of self-referential description is proposed. The SRO identifies highly descriptive local motifs, akin to the unit cell, which becomes sufficient to describe the whole structure. Even in the absence of any previous knowledge of crystallography it is still rather straightforward to identify the unit cell from the information about the positions of all atoms. Indeed, it is sufficient to take a portion of the structure, translate it in space and see when and where it perfectly overlaps with another part of the structure. The smallest portion of the structure that periodically overlaps with all other parts of the structure is the unit cell.

7.1.2 A Mathematical Description

Let us start as before by taking a structure S composed of hard spheres. By defining X as the information from S, and Y as the information from a small portion of S we have $S = X \cup Y$. To measure the amount of self-referential order we need to be able to quantify how the knowledge about the portion Y reduces the amount of information needed to encode X. Formally we need a measure of information content such as described by the Kolmogorov complexity K (Kolmogorov, 1968; Solomonoff, 1964, 1960; Li & Vitanyi, 1993). This also allows a quantity that still falls within the realm of information theory, allowing similar techniques as those used in chapter 6.

In other words, the quantity $K(X)$ is the amount of information necessary to describe X. Its conditional counterpart, $K(X|Y)$, is the amount of information necessary to describe X, given the full knowledge of Y. When the knowledge about a portion Y of the structure is sufficient to describe the rest of the structure we must have $K(S) = K(Y)$ and $K(X|Y) = 0$. Conversely, when the knowledge about a portion does not add any knowledge about the rest of the structure we must have $K(X|Y) = K(X) > 0$.

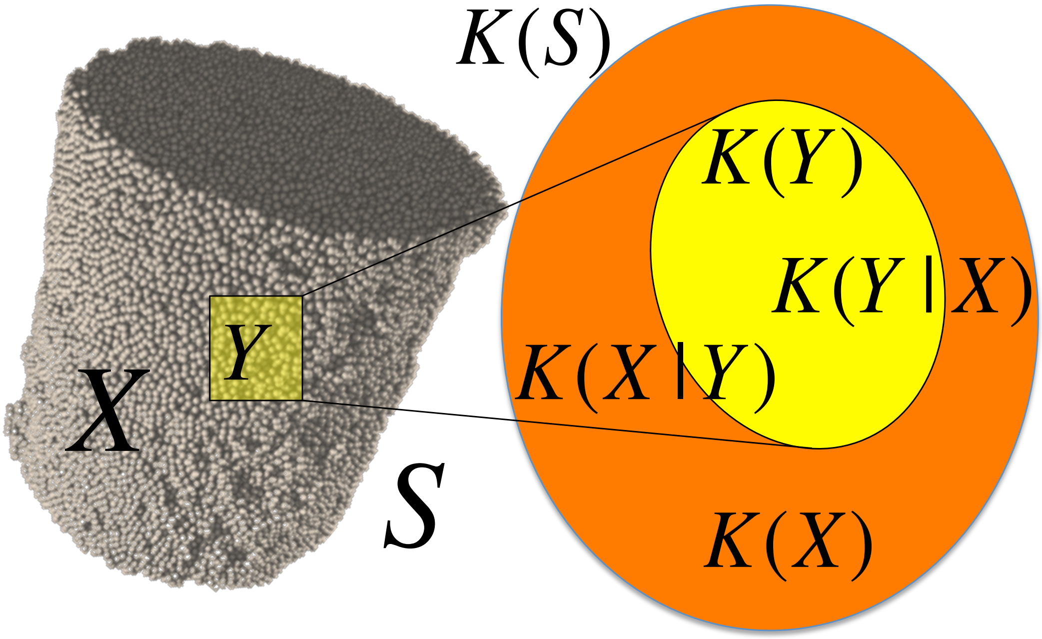


Figure 7.1: In absence of a pre-defined template reference structure, one can use a portion (Y) of the structure to describe the whole structure $S = X \cup Y$. The knowledge about the portion Y can reduce the uncertainty about the rest of the structure X. Kolmogorov complexity, here denoted with $K(X)$ and $K(Y)$, measure the information contained in X and Y respectively. For instance, in the case in which the rest of the structure X is completely determined by the knowledge of the portion Y, we have $K(S) = K(Y)$. In this case, the conditional information about X given Y, $K(X|Y)$, is equal to zero.

While this is similar to entropic measures it is important to point out here the 'perfect' solution of Kolmogorov complexity is used.

Given these quantities it is possible to define a *self-referential order parameter*

$$s(X;Y) = 1 - \left(\frac{K(X|Y)}{K(X)}\right) \quad , \tag{7.1}$$

which is equal to one if the system is fully self-referentially ordered and it is equal to zero if completely random. This approach mathematically defines 'perfect' *self-referential order*. However here we hit the same problem as before because the Kolmogorov complexity is not a computable quantity, as stated in previous chapters (section 2.4.1).

7.1.3 *An Entropic Approach*

As we have seen it is possible to look for the information about X provided by the knowledge of Y using entropic measures. When the portion Y encodes the full information about X then $H(X) = H(Y)$. The remaining entropy of variable X when variable Y is known is quantified by the conditional entropy $H(X|Y)$. Therefore, an entropic measure of self-referential order is:

$$s(X;Y) = 1 - \frac{H(X|Y)}{H(X)} \quad . \tag{7.2}$$

By using the identity $H(X|Y) = H(X,Y) - H(Y)$ and obtaining the equivalent expression

$$s(X;Y) = 1 - \frac{H(X,Y) - H(Y)}{H(X)} \quad , \tag{7.3}$$

which also reads

$$s(X;Y) = \frac{H(X) + H(Y) - H(X,Y)}{H(X)} \quad . \tag{7.4}$$

One may notice that the quantity on the numerator is the mutual information: $I(X;Y) = H(X) + H(Y) - H(X,Y)$, therefore this measure quantifies the relative mutual dependence between structures X and Y.

$$s(X;Y) = \frac{I(X,Y)}{H(X)} \tag{7.5}$$

It can be seen the SRO should always be between zero and one. Let us demonstrate this by taking the bounds for y completely encoding information for x. Therefore $H(x|y) = 0$ i.e there is no uncertainty knowing y. So by equation (7.2) it is found:

$$s(X;Y) = 1 - \frac{0}{H(x)} = 1. \tag{7.6}$$

In the case where no information is given by y to x we find:

$$H(x|y) = H(x) \;\therefore\; s(X;Y) = 1 - \frac{H(x)}{H(x)} = 0 \tag{7.7}$$

As we have already seen both the MI and entropy must be ≥ 0 and therefore $s(X;Y) \geq 0$. As always the convention $0log0 = 0$ is used. To show the SRO must remain in these bounds, we can alternatively take the fact that $0 \leq H(X|Y) \leq H(X)$. Taking equation (7.2) we can see quantity is defined in the interval $0 \leq s(X;Y) \leq 1$ where 0 is associated to a random state and 1 is instead observed for perfect self-referential order.

7.2 Characterisation of Structures

As results in chapter 6 show, parts of the structure can carry larger amounts of information about the whole structure with respect to others. These high information-content portions are repeated similarly in the structure more often than others and therefore they are of particular relevance. We look for local sub-structures containing maximal relative information. We shall call them 'motifs' these are equivalent to the 'patches' used in (Kurchan & Levine, 2011). In general, more than one motif is necessary to encode a disordered structure. Furthermore, these motifs do not repeat perfectly across the structure and therefore they must be described in statistical terms.

7.2.1 Motifs and Self-Referential Order

As we saw with the KLD, it is possible to identify areas giving more information to the whole structure. Here I present similar results generated by the more robust SRO parameter.

Motifs can be identified from equations (7.1) or (7.2) by looking at the local parts that maximally contribute to the information about the whole structure, i.e. the portions Y associated with the largest $s(X;Y)$. Once the motifs are identified, one must quantify their recurrence in the structure. This can be done in three steps:

1. Count the relative frequency of occurrence of each local motif;
2. Compute the probability distribution of its fluctuations;
3. Estimate the entropy.

A number of challenges need to be addressed before completely describing a structure the first being a fast and efficient way to search for structural motifs, particularly in sparse disordered systems. This section will attempt to address the problem, however other challenges exist in terms of rebuilding a structure which must be accounted for in future work. This includes the possible overlaps between motifs that make their unique identification ambiguous and requires the introduction of 'exclusion rules' (i.e. when two motifs overlap, only one must be counted at the time) and statistical ensemble analysis (i.e. all encodings resulting from the different exclusions) must be considered. This would require the use of a more dynamic search algorithm using SRO, rather than the simple overlapping grid presented here.

Some work has already begun looking at motifs in disordered sphere packings (Delaney et al., 2010; Aste & Matteo, 2008; Aste et al., 2006). The work suggests that in these systems the number of motifs m is of the order of 10^2, and the matching rules are of the order of 10^4. This may seem like a large number but it should be pointed out that in terms of information compression we are passing from an information size of the order of 10^{20} (hundreds of billions of gigabytes), which is certainly beyond computable sizes, to a size of 10^4 bytes (tens of kilobytes), which is computationally insignificant.

It should also be stated that for many practical purposes a precise definition of the local geometrical configuration is often irrelevant and the information can therefore be further reduced (Anikeenko et al., 2008). This thesis shows that local tetrahedral motifs are related to the description of a structural transition at the Random Close Packing limit. This can be described in terms of motifs. Given that each tetrahedron has six edges and each edge has two states we can count $m = 2^6 = 64$ relevant motifs. To compute the matching rules we must then consider that tetrahedra match face by face and they have 4 faces giving $(4 \times 2^6)^2/2 \sim 33,000$ matching rules. However, these number can be greatly reduced, for instance, in (Anikeenko et al., 2008), it was shown that the most relevant motifs were only 2: all-short-edges or not. And the relevant matching statistics was given by the chains of all-short-edges tetrahedra.

7.3 Results

In this section the preliminary investigation about the quantification of self-referential order in two-dimensional disks packings generated by the LSA. The results are for 15 samples comprising 5,000 disks representing a range of packing fractions between 0 to ~ 0.9.

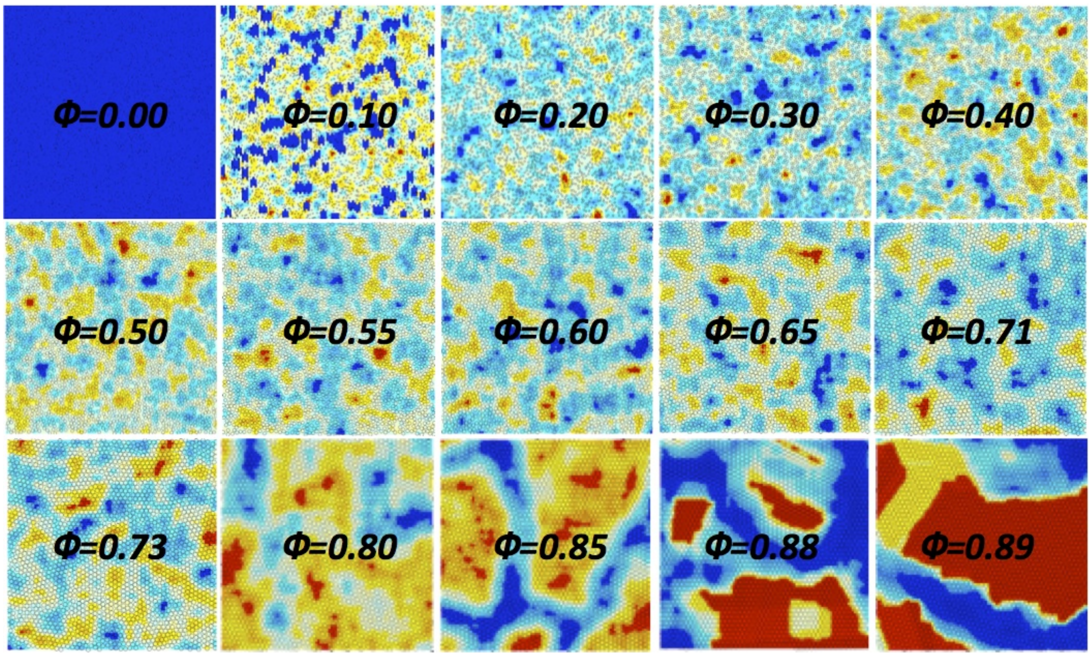


Figure 7.2: Colour maps (where blue is low SRO with red being high) created by scanning a packing using the self-referential order parameter with a boxsize = 5 and packing fraction Φ. The colour map is rescaled for each image.

The SRO parameter in these two-dimensional packings is calculated by looking at the

Voronoï volumes around each disk and identifying a set of $m = 500$ kinds of motifs classified in terms of their different volumes. This is used to verify that the method is robust against this choice with analogous results obtained for $m = 100$ or $m = 2,000$. A local sample Y of the packing is taken and used to compute $s_X(Y)$ by applying equation (7.4). The process is then repeated for 10,000 different samples regularly displaced across each sample. Figure (7.2) shows the SRO values over a grid of samples to build a picture in the same way as was done for the KLD. These images are similar in pattern with the SRO being low (opposite to the KLD) at lower packing fractions, and offering a lot of contrast for high packing fractions where the packing crystallises.

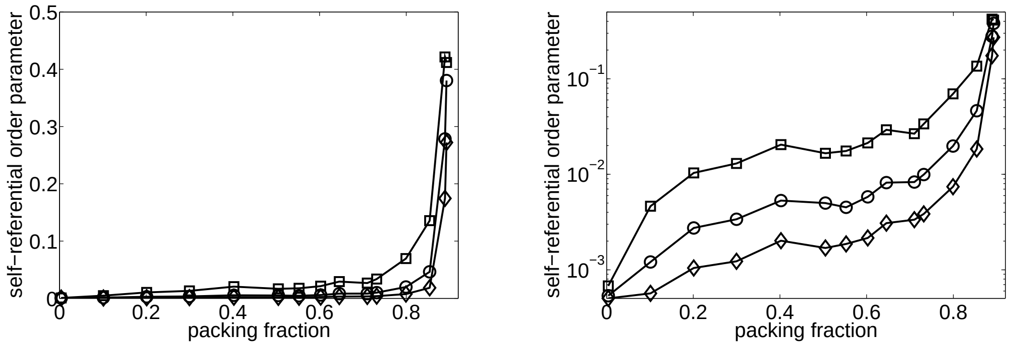


Figure 7.3: Global values of the self-referential order parameter $\hat{s}$ vs. packing fraction displayed in both linear and semi-logarithmic scale. Different curves ($\diamond$, $\circ$ or $\square$ symbols) correspond to different sizes of the local portion Y, which are squares respectively with edges equal to 3, 5 or 10 disk-diameters.

The absolute value for the SRO is shown in figure (7.3). Here the global measure of SRO parameter ($\hat{s}(X;Y)$) is shown by finding the joint probability for given fractions of Voronoï volumes simultaneously present in any of the portions Y and in the rest of the sample $X = S \cap Y$. One can see that the self-referential order parameter increases with packing fraction to reach a maximum at the largest packing of $\Phi \sim 0.9$. From the semi-log plot in figure (7.3) we can note that this parameter ranges over 4 orders of magnitude, with an interesting plateau appearing between packing fractions ~ 0.4 and ~ 0.7. As previously stated the largest packing fraction that can be achieved for hard mono-disperse disks is $\Phi = \pi/\sqrt{12} \simeq 0.907$ (Aste & Weaire, 2000), which corresponds to a perfectly ordered, crystalline packing.

The densest packing has some defects as creating perfect crystalline packings in the LSA takes specific conditions with a very long run-time. For this reason, the highest packing fractions are not the theoretical maximum. This is true for all results presented with the LSA regardless of spatial dimension. These defects are clearly visible in figure (7.2). At these disjoints the structure is skewed, along with the statistics. As these statistics are very different to the rest of the structure the SRO is much lower in these regions. These 'defective' regions are less representative of the sample. Conversely, at lower packing fractions the most representative local portions are not compact configurations with crystalline symmetry but rather more complex and less compact configurations. This leads to an increase in both number and complexity of motifs, so again the SRO is much lower.

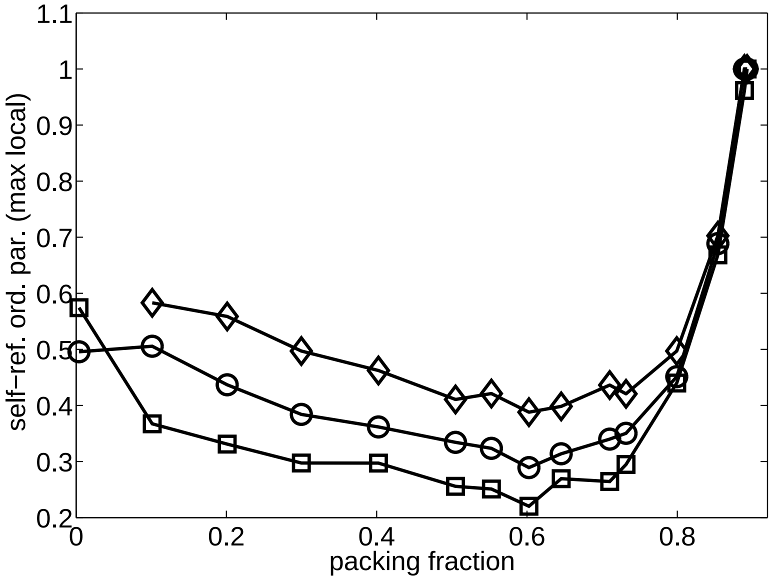


Figure 7.4: Average maximum local values of the self-referential order parameter for each sample. The average is over the 10% largest $s(X;Y)$. Different curves ($\diamond$, $\circ$ or $\square$ symbols) correspond to different sizes of the local portion Y, which are squares respectively with edges equal to 3, 5 or 10 disk-diameters.

In general, at different packing fractions different local configurations carry more or less information about the rest of the sample structure. This can be thought of in terms of different phases of matter. For gases, the placement of a single atom has little or no effect on those around it. As structures become denser displacement can affect local parts of the structure. In the densest cases, small fluctuations can have an effect at long range, such as the disjoints seen in figure (6.25f).

The investigation goes on to show the presence of highly referential motifs by looking at the maximum values of $s(X;Y)$ in each sample. I was not directly responsible for this part of the work, but it is continued in the paper *Self-Referential Order* found in appendix 1. The investigation specifically quantifies the portions of sub structures carrying the largest information by identifying the 10% largest $s(X;Y)$ per each sample. In figure (7.4) we show the values of the average self-referential order parameter $s(X;Y)$ in this top 10% subset of most representative configurations. One can note that at large packing fractions, where the structure is essentially crystalline, only a few configurations carry all structural information. Interestingly, also at very low packing fractions, where the structure is essentially random, again a small part of the most informative configurations characterise well the whole structure. On the other hand, at intermediate packing fractions (around $\Phi \simeq 0.6$) the structure is more complex and even the most informative local configurations carry, in average, a smaller amount of information about the rest of the system.

Chapter 8

Results - Self Referential Order in 3D

8.1 Characterisation of Three Dimensional Structures

Data for three dimensions was generated as before using the LSA. For these results, several sets of packings were used. A set refers to a single run of the LSA program in which many packings are generated from an initial random loose packing. These packings are then processed by the Karambola package (Schaller et al., 2011), and the Minkowski tensors created. As for the two-dimensional case, a grid composing of cubes was overlaid onto the packing and calculations were made for various box sizes and statistics. A binning was created for each statistic, taking into account its range of values and the number of particles. The number of bins went through some trial and error and was eventually set at 20 by 20 bins, then kept constant for all the results.

8.1.1 Characterisation of Structures with Minkowski Tensors

There had already been investigations of hard sphere packings using Minkowski tensors (Schrder-Turk et al., 2010) by calculating Anisotropy. Figure (8.1) shows the anisotropy of a poly-disperse sphere packing, taken from *Disordered spherical bead packs are anisotropic* (Schrder-Turk et al., 2010).

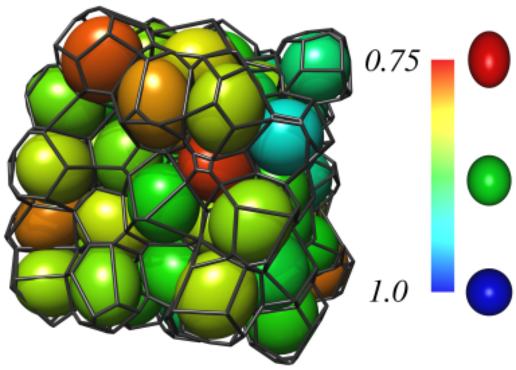


Figure 8.1: A poly-disperse packing of objects with varying anisotropy as shown by a colour map. The black outlines shown the Voronoï cell frames

Anisotropy is a measure of how non-uniform a shape is, most commonly referred to as directional dependence. For example, an isotropic shape, one that is uniform independent of direction, would be a sphere or dodecahedron. Anisotropy of such a shape is equal to one with the magnitude of deviations from one showing the magnitude of anisotropy. By taking the convention of always using the minimum and maximum eigenvalue for each cell, the range of values is confined between zero and one. This leads to a formal definition of anisotropy using Minkowski tensors as:

$$\alpha_W = \langle | \frac{\mu_{min}}{\mu_{max}} | \rangle \tag{8.1}$$

where μ are the eigenvalues of the Minkowski metric W over all cells K. It is important to note not all Minkowski Tensor metrics are symmetric matrices therefore not all have eigenvalues. These results used jammed packings between $\Phi = 0.55$ and 0.72. To validate the code we were using these tests were rerun and show very similar results with a change in gradient at the $\Phi = 0.64$ mark. However, a further step was taken by running the same experiment on non-jammed packings by taking snapshots of an evolving system. This had the added benefit of giving a much greater range of packing fractions with interesting results.

The results presented in figure (8.2) show the results generated by my own program using equation (8.1). Anisotropy changes at the RCP limit are well reported and easily reproduced. However, the results in figure (8.2) show the RLP as a clear disjoint at $\Phi = 0.54$, which is of some interest as it was not reported in the original work. All six eigenvalue sets of the MT are shown, as indicated by the legend. In these results the RCP limit is not present, which is due to the way the packings have been generated. This phenomenon is investigated further in the next section and shows the importance of understanding the evolution of these packings.

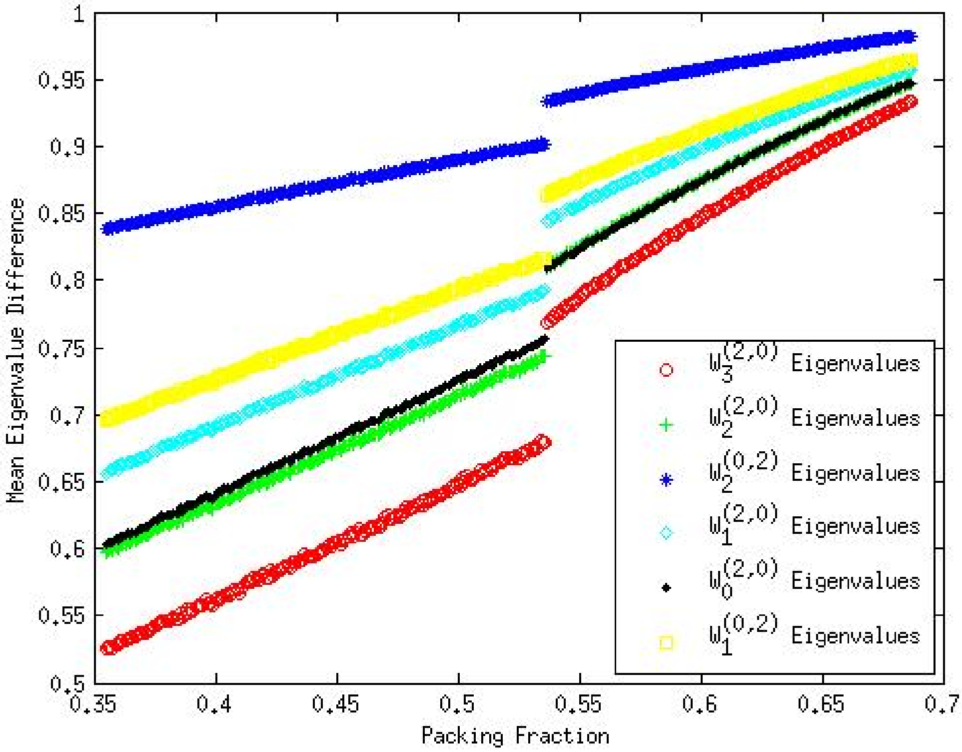
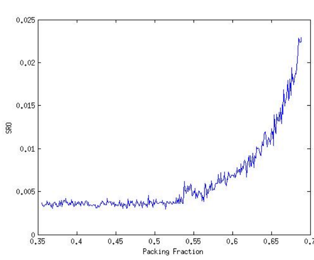
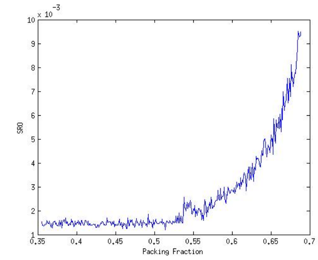


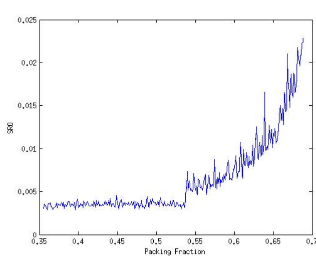
Figure 8.2: Results showing anisotropy for non-jammed snapshots of an evolving system. The results shown are for a packing evolving with increasing pressure



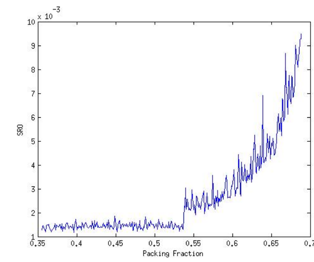
(a) SRO generated by using data for the $W_0^{0,0}$ MT, with a sample size of 10



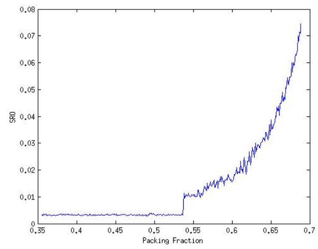
(b) SRO generated by using data for the $W_0^{0,0}$ MT, with a sample size of 4



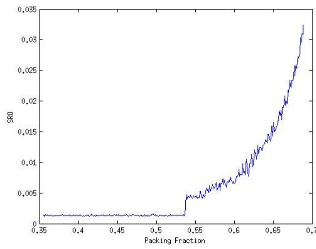
(c) SRO generated by using data for the $W_1^{1,0}$ MT, with a sample size of 10



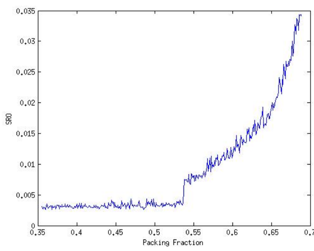
(d) SRO generated by using data for the $W_1^{1,0}$ MT, with a sample size of 4



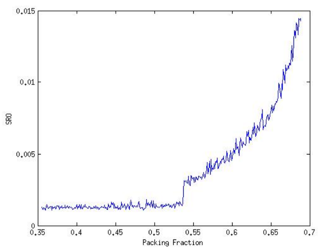
(e) SRO generated by using data for the $W_2^{1,0}$ MT, with a sample size of 10



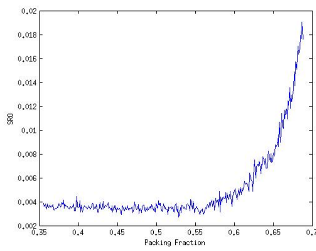
(f) SRO generated by using data for the $W_2^{1,0}$ MT, with a sample size of 4



(g) SRO generated by using data for the $W_1^{0,2}$ MT, with a sample size of 10



(h) SRO generated by using data for the $W_1^{0,2}$ MT, with a sample size of 4



(i) SRO generated by using data for the $W_0^{1,0}$ MT, with a sample size of 10

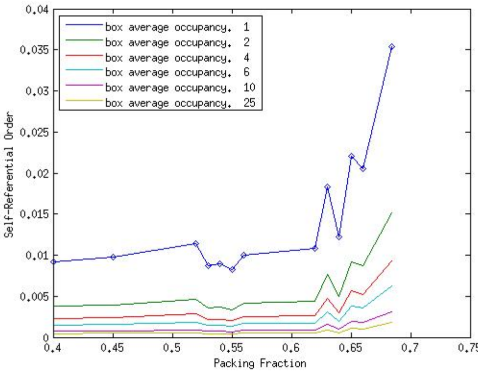
Figure 8.3: Figures show global SRO on structures with packing fractions between $\Phi \sim 0.35$ and ~ 0.7. These structures have been generated using snapshots of an evolving system using the LSA.

8.1.2 Characterisation of Structures with Self Referential Order

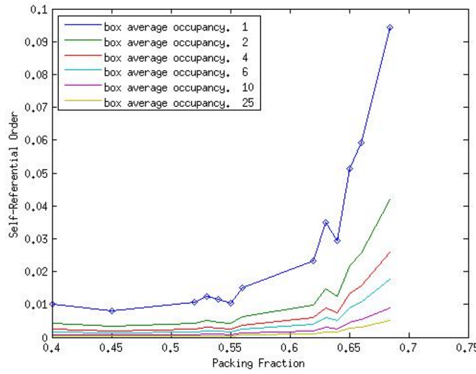
Results in this section were created by the self-referential order parameter (equation 7.4). The results in figure (8.3) show data generated from a variety of statistics. The packings themselves were made individually with a number of settings for the LSA, and given the sensitivity of the SRO can appear quite noisy. They are shown to contrast the information from the various Minkowski Tensors (reported in section 4.3).

Generally results in figure (8.3) show that many of the extracted MT statistics have similar trends, there are some differences such as those shown in figure (8.3[i]). The differences suggest that different Minkowski tensors offer different information, with studies on other structures confirming this (figure 8.6). These differences require further investigation as some MT values may give better information than others in particular circumstances. For example, shape may give more information on poly-disperse packings than volume, and comparing these quantities could produce new data on the nature of certain structures.

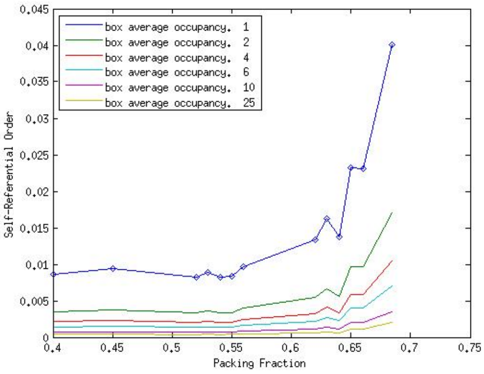
A definite change can be seen in the SRO at 0.54, showing the RLP limit. It is very interesting to also see evidence of a gradient change at $\Phi \approx 0.64$ for data using larger sample sizes. This would be very important, as it shows the SRO is sensitive to both the RLP and RCP limits, most methods are only sensitive to one or the other, relying on a posteriori information, or looking for one particular change in structure. So far as I know, no other method is capable of creating data sensitive to both, particularly a-priori.



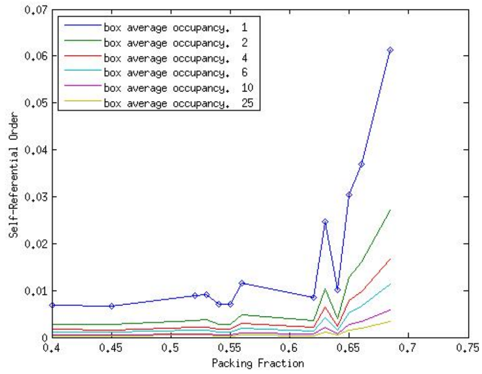
(a) SRO generated by using data for the $W_0^{0,0}$ MT



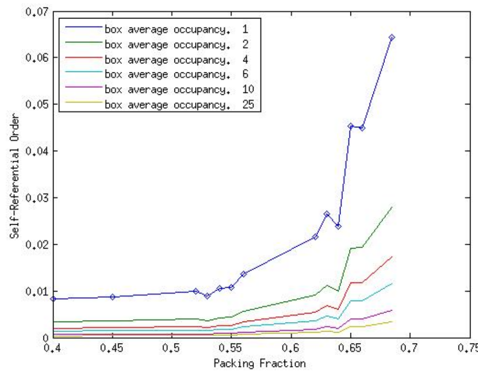
(b) SRO generated by using data for the $W_2^{0,2}$ MT, taking the highest eigenvalue



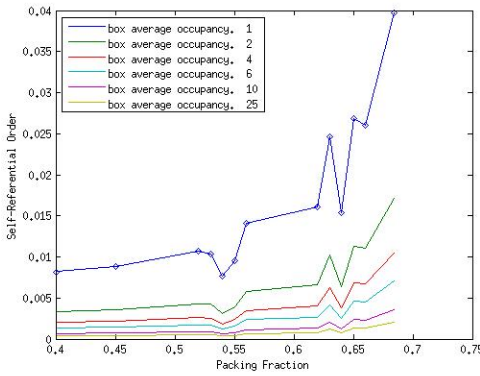
(c) SRO generated by using data for the $W_0^{1,0}$ MT, taking the sum of the square of their values



(d) SRO generated by using data for the $W_2^{2,0}$ MT, taking the ratio of eigenvalues



(e) SRO generated by using data for the $W_3^{2,0}$ MT, taking the middle eigenvalue



(f) SRO generated by using data for the $W_3^{2,0}$ MT, taking the maximum eigenvalue

Figure 8.4: Figures show global SRO on structures with packing fractions between $\Phi \sim 0.35$ and ~ 0.7. These un-jammed packings have been generated using the LSA and are composed of 10000 spheres with modified growth rates. The program terminated when a user-defined packing fraction was reached. A number of samples sizes of between 1 and 25 are also presented and shown in the legend.

It can be seen in figure (8.4) how some MTs detect the RLP limit, but others do not. This is far clearer in the jammed examples in figure (8.6). Jammed examples are generated when the program creating them can no longer compress the structure. In real terms they cannot be compressed as no particles can move enough to affect the global structure. The unjammed results are created when the program terminates due to another condition (for example reaching a particular packing fraction). While the unjammed packings in figure (8.6) are far more sparse than those in figure (8.3), show some of the same noise for unjammed packings. This said it is still clear there is some change in gradient around $\Phi = 0.64$. In figure (8.4[d]) and ([f]) a dip in the SRO can be observed around the RLP limit, however this may be due to noise and the increased density of data points.

One can notice the contrast between unjammed data sets. In the snapshot case the RLP is well defined and the RCP is not. For the case where growth rate is modified and a termination point is defined the opposite appears to be true. As there is a difference in how the structure evolves in these two cases it is perhaps not surprising that the results would show this. The exact reasons for the change require further research into how time-lapsed packings evolve.

In addition the results in figure (8.3), and more clearly in figure (8.5), it is shown how the number of spheres in each sample (sample size) does have an effect on the magnitude of the SRO, however the trend is generally not affected. Due to the resolution in the data the graph looks quite noisy, this is due in part to the sensitivity of the SRO given each packing is unique, regardless of packing fraction.

Figures (8.5) and (8.6) display jammed packings achieved through a changing collision rate. The results show the same SRO change at ~ 0.54, but more intriguingly, the RCP limit is far more clearly defined than in figure (8.3). A drop can be observed, similar to the 2 dimensional MI case, in the SRO around $\Phi \sim 0.56$. However a change can be observed over the range $\Phi \sim 0.54 \rightarrow \sim 0.60$. This could be interpreted in one of two ways. Firstly as one change in the data with increased SRO over the range starting at

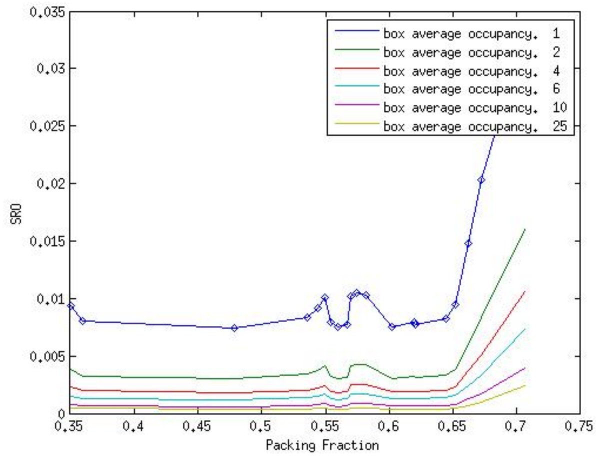
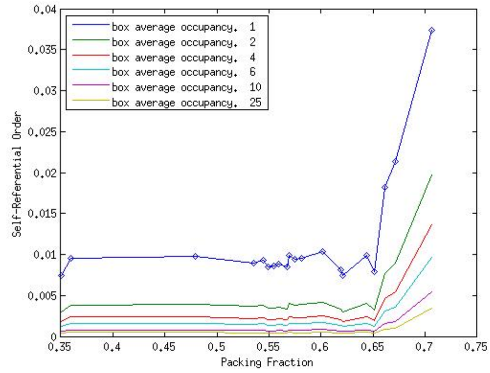


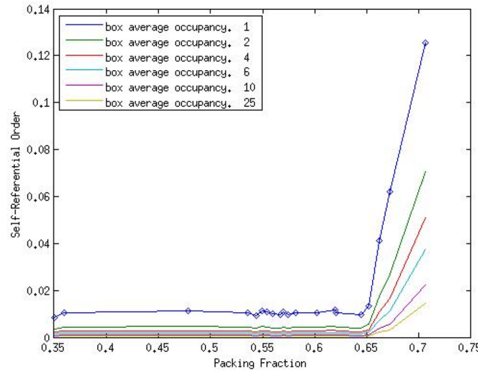
Figure 8.5: SRO generated by using data for the maximum eigenvalue of $W_3^{2,0}$ with a number of jammed packings ($N = 10000$). The samples sizes of between 1 and 25 are also shown in the legend.

$\Phi = 0.54$ and ending midway between $\Phi \sim 0.54$ and 0.64 i.e. the two proposed limits, with a dip in SRO centred at around $\Phi = 0.56$. The second is to consider it as two separate peaks at $\Phi \sim 0.54$ and $\Phi \sim 0.57$. Further investigation is required, with a greater density of points and more packings, but given the data shown in figure (8.3) it could be hypothesised that this trend is due to the RLP limit which as stated would be a very interesting result, given the addition of a clearly defined transition at the RCP limit of $\Phi = 0.64$ shown by a clear change in gradient.

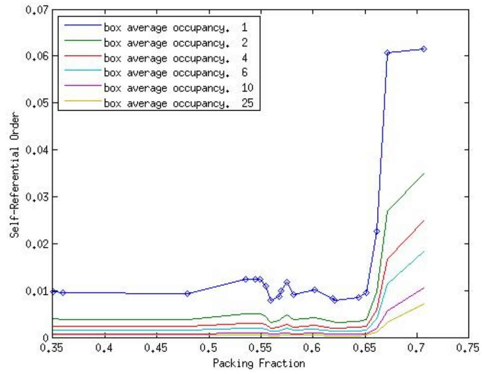
This, however, leads to the question of why we see such a transition in the case of figure (8.3). While both sets of data were generated by the LS algorithm, the sets shown in figure (8.3) are snapshots of an evolving system. Most importantly this means that they are not jammed over the range they are evolving and thus only the last snapshots can be considered jammed. To allow snapshots over a complete range of packing fractions the last snapshots are all at very high values of Φ. In figures (8.5) and (8.6) the data is taken from separate runs which have been allowed to jam, by changing the growth and collision rates to adjust the final packing fraction. Further results must take into account carefully how packings have evolved. This may also have implications for how the RLP and RCP limit can be understood structurally as a quantification has taken place.



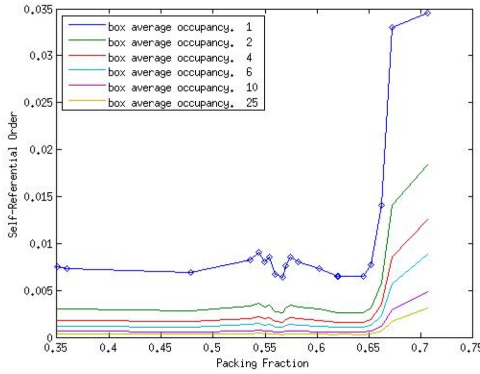
(a) SRO generated by using data for the $W_0^{0,0}$ MT



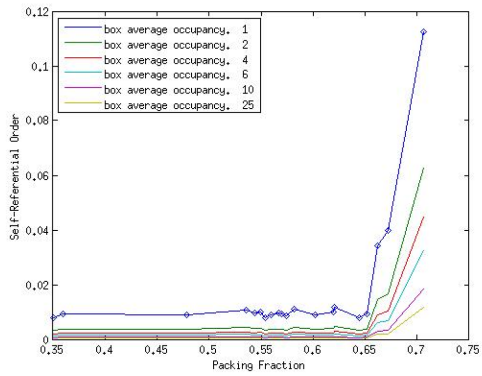
(b) SRO generated by using data for the $W_2^{0,2}$ MT, taking the highest eigenvalue



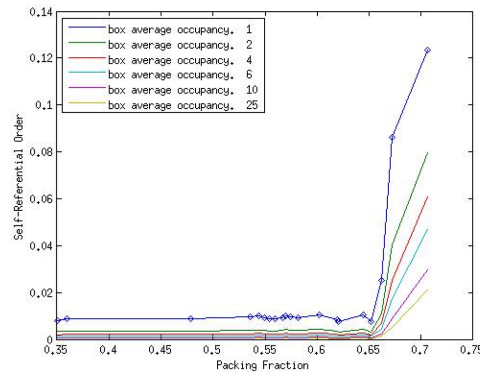
(c) SRO generated by using data for the $W_2^{0,2}$ MT, taking the ratio of eigenvalues



(d) SRO generated by using data for the $W_2^{2,0}$ MT, taking the ratio of eigenvalues



(e) SRO generated by using data for the $W_3^{2,0}$ MT, taking the middle eigenvalue



(f) SRO generated by using data for the $W_3^{2,0}$ MT, taking the minimum eigenvalue

Figure 8.6: Figures show global SRO on structures with packing fractions between $\Phi \sim 0.35$ and ~ 0.7. These jammed packings have been generated using the LSA and are composed of 10000 spheres. A number of samples sizes of between 1 and 25 are also presented and shown in the legend.

Further experiments were carried out on jammed packing sets in figure (8.6). Like in previous results a variety of MTs were used producing some different and interesting results. While all the MTs show a large a distinct change in gradient at the RCP limit but many offer little or no change around the RLP limit.

The most sensitive MT's to the RLP limit tended to be the surface integrals however the most notable results are those that took the eigenvalue ratios. These ratios can be recognised as the cell anisotropies as outlined in section (8.1.1). The difference here being the sensitivity to both changes in jammed packings. The different information offered by the MTs may allow us to probe what structural changes are taking place at the RLP limit. From the perspective of the Voronoï cells, it is clear some change in shape is occurring.

It can also be noted that there is a change in the magnitude of the SRO depending on sample size and MT. While these may seem like independent quantities they are not. The binning number remains constant but the range of the different MTs does not, leading to larger bin sizes. This leads to some data sets having a more or less sensitive resolution leading to a similar change in magnitude as observed by the different values in figure (8.4) and (8.6).

8.1.3 Experimental Data

The SRO data for the experimental data is published here for reference. While the values tally close to the SRO for packings, the range of these results makes it difficult to make statements about the trend, particularly around the RLP and RCP limits. The data is for all twelve packings described in section (3.3) and for six sample sizes. It is displayed in table (8.1) with the average sample occupancy listed along the top row and packing fraction the left column. The scaling is present for all the packings, increasing in a similar trend to each other and the results shown in the previous section.

	Sample Size					
pf	1	2	4	6	10	25
0.56822	0.0011	0.0018	0.0034	0.0050	0.0081	0.0192
0.57072	0.0010	0.0017	0.0033	0.0048	0.0078	0.0184
0.57302	0.0012	0.0020	0.0039	0.0057	0.0093	0.0219
0.56664	0.0010	0.0017	0.0034	0.0050	0.0081	0.0189
0.56616	0.0010	0.0017	0.0033	0.0048	0.0078	0.0183
0.57531	0.0010	0.0017	0.0033	0.0049	0.0079	0.0185
0.57096	0.0011	0.0019	0.0037	0.0054	0.0088	0.0206
0.57923	0.0011	0.0018	0.0034	0.0051	0.0083	0.0195
0.58233	0.0012	0.0021	0.0040	0.0059	0.0097	0.0227
0.59056	0.0012	0.0020	0.0038	0.0057	0.0093	0.0218
0.59955	0.0012	0.0021	0.0040	0.0060	0.0097	0.0227
0.60033	0.0012	0.0021	0.0041	0.0060	0.0098	0.0229

Table 8.1: Table showing SRO for twelve experimental packings generated by using XCT

Chapter 9

Conclusion

9.1 Conclusions

9.1.1 Self Referential Order

Several interesting developments have been reported in this thesis. A new mathematical framework has been proposed for use in characterising disordered structures. This framework introduces a new quantity, the self-referential order, as a way to quantify information in a system and probe important characteristics of the structure.

The self-referential order is a novel and robust quantity as presented in chapters 7 and 8. It has been derived from entropic measures (equation 7.2) making it useful for measuring information.

Investigation of the RCP and RLP limit has shown the quantity to be sensitive and robust enough to detect both transitions without alteration as displayed in figures (8.4) and (8.5). Extensive testing using a variety of statistics shows it capable of distinguishing a number of structural details (Figures 8.4, 8.6).

Through the use of colour maps the SRO has produced striking images (Figure 7.2) of how information is dispersed in a system. These images show how information evolves through the process of jamming and crystallisation. SRO has also allowed the identification of motifs that hold more information about the system, and importantly is capable of quantifying this information and expressing it in a meaningful way.

9.1.2 Mutual Information and the Kullback Leibler Divergence

Results in chapter 6 regarding the MI are currently unpublished. That said they are original and quite interesting. Studies of 2D mono-disperse disk packings have shown clear evidence of structure changes akin to transitions such as the RCP and RLP limits. Several other correlation techniques were examined to validate the MI as shown in section (6.2.1). It should be noted, however, that the entropy bias was never fully resolved. While the entropy bias did lead to the concept of SRO, it also meant MI could not be used as an absolute measure of disorder.

Results shown in figure (6.13) as well as figures (6.14), (6.15), (6.16), (6.17) all show evidence of a structural change with MI peaking at $\Phi \approx 0.79 \sim 0.83$. The RCP limit in 2D is considered to be $\Phi \approx 0.82$, which suggests the MI is providing evidence of the limit. This is an interesting find as the RCP limit can be difficult to obtain in 2D. It is particularly intriguing here due to its clarity, being clear and giving insight in to how the system is evolving around it. Additionally there is possible evidence of the RLP limit at $\Phi \approx 0.7$ seen in figure (6.14), however, due to the lack of literature this would require more research and is a possible direction for future work.

Using MI to quantify information in structures has also led to several other results that show how information is carried through a system. MI showed for loosely packed systems only short range influence but was none the less still present (Figures 6.10 and 6.11). In densely packed structures information propagated through the system to much larger distances. This was imaged by the KLD for crystallising packings such as those seen in figure (6.25).

9.1.3 Information and Disorder

At the beginning of my work I set out to find a measure that could quantify disorder. The SRO gives a measure of order that only requires information about the system it is defining, thus creating an independent measure. Studies involving the MI and KLD show that these 'disordered' systems are not devoid of order and that they have their own rules, requiring a reinterpretation of what we might consider order.

The new mathematical framework for understanding disorder presented in this thesis extracts information using a variety of different methods relating to sphere packings. That information has then been interpreted and encoded to a set of statistics that could describe the system without having to store it in its entirety. Ultimately, the framework has produced results showing what statistics may be useful, and by interpreting this data it can provide knowledge to the understanding and encoding of the structure. By quantifying the information the framework has shown how the properties of the system may be compressed, allowing it to be studied using the techniques and equipment available today.

By finding motifs and measuring self-referential order within a system it has been shown how some parts of the system are more representative of the global statistics. The frequency of some motifs allows a further compression with a reduction of information needed to encode that system. This is quantified in the SRO, in turn quantifying the disorder in a way which has not previously been used in packing problems. The framework does not directly allow for the compression and subsequent decoding of a structure but identifies motifs and structures that may be used to that end.

The method presented both sensitivity and flexibility. This in addition to results at low packing fractions suggested it was not dependent on local crystallisation to detect order. Again lending credence to the idea that disorder is not simply a degenerate form of order as previously believed. It shows sensitivity to structural changes in these packings caused by changes in evolution parameters and packing fraction, going so far as to suggest an RLP limit in 2D, although further research is needed.

By incorporating other methods the application of the framework can be broadened, as achieved in the case of Minkowski tensors with some interesting results (which have not been published as of this conclusion). It can be further developed for uses in computer science (particularly networks), in research into gene co-expression in biology and even be extended to higher dimensions in packing problems (Aste et al., 2010; Pozzi et al., 2013; Song et al., 2012).

The framework developed in this thesis is not limited to structure and could potentially be applied to many complex mathematical systems. For example, the framework could be applied to financial markets, identifying links between companies and showing how stock price fluctuations propagate through a market. One way this might be achieved is by creating conditional probabilities of stock market prices over a given time period, and then quantifying the uncertainty. In a similar way neural activity or gene expression could be explored in biology. Other approaches might include the use of Markov chains to generate probabilities (Markov, 1906).

9.1.4 Future Work

Future work should include first and foremost a further look at the proposed RLP limit in 2D by taking packings created using a variety of methods including experimentally generated results. Further work should include a generalisation to higher dimensions which could be achieved non-trivially through multivariate mutual information estimates. In addition different types of packings could be generated including those with non-spherical particles or poly-disperse spheres. Such an expansion is a logical step as many of the same statistics used in mono-disperse packings could be used to generate new results. This could include Minkowski tensors and Voronoï analysis to explore structural changes, investigating long-range order, changes in local structure, and differences in packing behaviour including changes to the RCP limit (Donev et al., 2004).

It would also be possible to explore interactions of more complex structures using the multi-variate form of MI, and, by extension, self-referential order. For example by building a 3-fold co-occurrence matrix of both 1^{st} and 2^{nd} neighbour cell area co-occurrences, then generating three probability functions based on the distribution of: reference cell areas, 1^{st} neighbour cell areas, and, as the additional quantity, 2^{nd} neighbour cell areas, over the whole packing. This method could potentially be used to investigate more complex behaviours and include more dynamic systems (by selecting appropriate variables to build the co-occurrence matrix). However, such an approach would require significant research.

Many of the techniques seen have been developed not in physics but biology for a number of reasons including epidemiology and histology (Gibson et al., 2011; Bock et al., 2010). It would be of great interest to see how this development may assist in such fields using biological data. Incorporation of an image processing program designed to extract topological data from structures may prove to be a dynamic expansion to the framework, but was not required for my work. This could be achieved using Matlab as it is already optimised for image processing allowing for easier compatibility.

While some potential applications of this framework are discussed, the original aim was to quantify structural disorder using a statistical approach. The methodology lays out the proposed approach, and was tested using applications to disk and sphere packings. The results showed the framework was able to quantify disorder and was sensitive enough to produce interesting results.

Bibliography

Adleman L. M., 1994, Molecular computation of solutions to combinatorial problems, Nature, 369, 40

Ahnert S., Johnston I., Fink T., Doye J., Louis A., 2010, Self-assembly, modularity and physical complexity, Physical review E 82, 82

Anikeenko A. V., Medvedev N. N., Aste T., 2008, Structural and entropic insights into the nature of the random-close-packing limit, Phys. Rev. E, 77

Aste T., Matteo T. D., 2008, Structural transitions in granular packs: statistical mechanics and statistical geometry investigations, The European Physical Journal B, 64, 511

Aste T., Matteo T. D., 2010, Introduction to Complex and Econophysics, Lecture Notes in Complex Systems

Aste T., Matteo T. D., Tordesillas A., 2007, Granular and Complex Materials, World Scientific Lecture Notes in Complex Systems, 8

Aste T., Saadatfar M., Sakellariou A., Senden T., 2004, Investigating the Geometrical Structure of Disordered Sphere Packings, Physica A, 339, 16

Aste T., Saadatfar M., Senden T. J., 2005, Geometrical structure of disordered sphere packings, Physical Review E

Aste T., Saadatfar M., Senden T. J., 2006, Local and global relations between the number of contacts and density in monodisperse sphere packs, J. Stat. Mech., 7

Aste T., Shaw W., Di Matteo T., 2010, Correlation structure and dynamics in volatile markets, New Journal of Physics, 12

Aste T., Weaire D., 2000, The Pursuit of Perfect Packing. Talyor & Francis

Baez J., , 2015, Big data, `https://plus.google.com/117663015413546257905/posts/B16McQB6Vn2`, Accessed Feb 2016

Ball P., 2011, In retrospect: On the Six-Cornered Snowflake, Nature, 480, 455

Berry A. C., 1941, The Accuracy of the Gaussian Approximation to the Sum of Independent Variates, Transactions of the American Mathematical Society, 49

Biggs N., 1993, Algebraic Graph Theory. Cambridge University Press

Biroli G., Garrahan J. P., 2013, Perspective: The glass transition, arXiv preprint arXiv:1303.3542

Bock M., Tyagi A. K., Kreft J.-U., Alt W., 2010, Generalized Voronoi Tessellation as a Model of Two-dimensional Cell Tissue Dynamics, Bulletin of Mathematical Biology, 72, 1696

Bravais A., 1949, Mmoire sur les systmes forms par les points distribus rgulirement sur un plan ou dans l'espace, Crystallographic Society of America

Butler P., Aste T., Matteo T. D., 2013, Self-Referential Order, Philisophical Magazine, 93

Cellucci C., Albano A., Rapp P., 2005, Statistical validation of mutual information calculations: comparisons of alternative numerical algorithms, Physics Review E, 71

Chan H.-K., 2011, Densest columnar structures of hard spheres from sequential deposition, Physical Review E 84

Chang H.-C., Wang L.-C., 2010, A Simple Proof of Thue's Theorem on Circle Packing, Submitted to arXiv on 22 Sept 2010

Coulomb C. A., 1821, Theorie des machines simple, Bachelier

de Cruys T. V., 2011, Two multivariate generalizations of pointwise mutual information, Proceeding DiSCo '11 Proceedings of the Workshop on Distributional Semantics and Compositionality, pp 16–20

Delaney G. W., Matteo T. D., Aste T., 2010, Combining tomographic imaging and dem simulations to investigate the structure of experimental sphere packings, Soft Matter, 6, 2992

Donev A., Cisse I., Sachs D., Variano E. A., Stillinger F. H., Connelly R., Torquato S., Chaikin P. M., 2004, Improving the Density of Jammed Disordered Packings Using Ellipsoids, Science

Donev A., Torquato S., Stillinger F. H., 2005, Neighbor List Collision-Driven Molecular Dynamics Simulation for Nonspherical Hard Particles. I. Algorithmic Details, Journal of Computational Physics, 202, 737

Edwards S. F., Oakeshott R. B., 1989a, The transmission of stress in an aggregate, Physica D, 38, 88

Edwards S. F., Oakeshott R. B., 1989b, Theory of powders, Physica A, 157, 1080

Fejes T. L., 1964, Regular Figures. Pergamon Press

Feller W., 1945, The fundamental limit theorems in probability, Bulletin of the American Mathematical Society, 51, 800

Filmus Y., , 2010, wo Proofs of the Central Limit Theorem, `http://www.cs.toronto.edu/~yuvalf/CLT.pdf`, Accessed Jan 2016

Frank F. C., Kasper J., 1958, Complex Alloy Structures Regarded as Sphere Packings, Acta Cryst., 11, 184

Gaba E., , 2006, View of the planes establishing the main curvatures on a minimal surface, Accessed on March 2017, Permission details: CC-BY-SA

Gibson W. T., Veldhuis J. H., Rubinstein B., Cartwright H. N., Perrimon N., Brodland G. W., Nagpal R., Gibson M. C., 2011, Control of the mitotic cleavage plane by local epithelial topology, Cell, pp 427–438

Gold C., , 2006, The parabolic lifting map for a set S of points in the plane, Accessed on March 2017, Permission details: CC-BY-SA

Grassberger P., 1998, Finite sample corrections to entropy and dimension estimates, Physics Letters A, 128

Grunwald P., Vitanyi P., 2004, Shannon Information and Kolmogorov Complexity, arXiv:cs/0410002

Hales T. C., 2005, A proof of the Kepler conjecture, Annals of Mathematics, 162, 1065

Harvey S. C., Tan R. K., Cheatham T. E., 1998, The flying ice cube: Velocity rescaling in molecular dynamics leads to violation of energy equipartition, Journal of Computational Chemistry, 19, 726

Hernadvolgyi I., , 1998, Generating random vectors from the multivariate normal distribution, `http://www.csi.uottawa.ca/~istvan/work.html`, Accessed Feb 2016

Herstein I. N., 1964, Topics In Algebra. Blaisdell Publishing Company

Huffman D., 1952, A Method for the Construction of Minimum-Redundancy Codes, Proceedings of the I.R.E., pp 1098–1102

ISO, 2015, `http://www.iso.org/iso/home.html`, Accessed Nov 2015

Isokawa Y., 2000, Poisson-Voronoi tessellations in three-dimensional hyperbolic spaces, Advances in Applied Probability, 32, 648

Jaynes E. T., 1957, Information Theory and Statistical Mechanics, Phys. Rev., 106, 620

Jensen J. L. W. V., 1906, Sur les fonctions convexes et les ingalits entre les valeurs moyennes, Acta Mathematica, 30, 175

Kaltchenko A., 2004, Algorithms for Estimating Information Distance with Application to Bioinformatics and Linguistics, Electrical and Computer Engineering, 4, 2255

Kim K., Pak H. K., 2010, Diffusing wave spectroscopy sudy of microscopic dynamics of three dimensional systems, Soft Matter, 6, 2894

Knight R. D., 2008, Physics for Scientists and Engineers. Pearson

Kolmogorov A., 1965, Three Approaches to the Quantitative Definition of Information, Problems of Information Transmission, 1, 1

Kolmogorov A. N., 1968, Three approaches to the quantitative definition of information, International Journal of Computer Mathematics, 2, 157

Krasnogor N., Gustafson S., Pelta D. A., Verdegay J. L., 2011, Systems Self-Assembly: Multidisciplinary Snapshots. Elsevier

Kumar N., Luding S., 2014, Memory of jamming and shear-jamming, Submitted to arXiv on 23 Jul

Kunz G. F., 1918, The Life and Work of Hauy, American Mineralogist, 3, 61

Kurchan J., Levine D., 2011, Order in glassy systems, Journal of Physics A: Mathematical and Theoretical, 44

Li M., Vitanyi P. M., 1993, An introduction to kolmogorov complexity and its applications. Springer

Lord Kelvin W. T., 1887, On the Division of Space with Minimum Partitional Area, Philosophical Magazine 24, 151

Lubaehevsky B. D., Stillinger F. H., 1990, Geometric Properties of Random Disk Packings, Journal of Statistical Physics, 60, 561

Magtegna R., Stanley H., 2000, An Indroduction to Econophysics: Correlations and Complexity in Finance. Cambridge University Press

Malinouskaya V., Mourzenko V., Thovert J. F., Adler P. M., 2009, Random packings of spiky particles: Geometry and transport properties, Physical Review E 80

Mandelbrot B., 1963, The Variation of Certain Speculative Prices, The Journal of Business, 36

Markov A., 1906, Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga, Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete, 2-ya seriya, tom 15, 9 4, 135

Mathworks, 1984, `http://uk.mathworks.com/`, Accessed Nov 2015

Mathworks, 2015a, `http://uk.mathworks.com/help/matlab/ref/delaunay.html`, Accessed Feb 2016

Mathworks, 2015b, `http://uk.mathworks.com/products/`, Accessed Nov 2015

Mathworks, 2015c, Voronoi Diagrams, `http://www.mathworks.co.uk/help/matlab/math/voronoi-diagrams.html`, Accessed Feb 2016

Mehta A., 2010, Spatial, dynamical and spatiotemporal hetrogeneities in granular media, Soft Matter, 6, 2875

Meyer S., Song C., Jin Y., Wang K., an A. Makse H., 2005, Jamming in two-dimensional packings, Phys. Review E, 71

Mezard M., Montanari A., 2009, Information, Physics, and Computation. Oxford University Press

Miller G. A., 1955, A note on the bias of information estimates, Information Theory in Phychology; Problems and Methods, II-B, 95

Moon Y.-I., Rajagopalan B., Lall U., 1995, Estimation of Mutual Information using kernel density estimators, Phys. Review E, 53

Morgan F., 1994, Mathematicians, including Undergraduates, Look at Soap Bubbles, Amer. Math. Monthly, 101, 343351

Newman M., 2003, The Structure and Function of Complex Networks, Siam Review, 45(2), 167256

Newman M., 2010, Networks: An Introduction. Oxford University Press

Paninski L., 2003, Estimation of Entropy and Mutual Information, Neural Computation, 15

Park J.-Y., Han S.-Y., 2013, Application of artificial bee colony algorithm to topology optimization for dynamic stiffness problems, Computers & Mathematics with Applications, 66, 18791891

Pozzi F., Matteo T. D., Aste T., 2013, Spread of risk across financial markets: better to invest in the peripheries, Scientific Reports, 3

Qhull, 1995, `http://www.qhull.org/html/qdelaun.htm`, Accessed Nov 2015

R. B., 1994, Using mutual information for selecting features in supervised neural net learning, IEEE Transactions on Neural Networks, 5, 537

Reynolds O., 1903, On an Inversion of Ideas as to the Structure of the Universe. University Press

Rivier N., Lissowski A., 1982, On the correlation between sizes and shapes of cells in epithelial mosaics, J. Phys. Math. Gen., 15, 143

Rodhes M., 1997, Principles of Powder Technology. John Wiley & Sons

Rothemund P., Winfree E., 2000, STOC '00: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, ACM, New York, pp 459–468

Sausset F. m. c., Levine D., 2011, Characterizing Order in Amorphous Systems, Phys. Rev. Lett., 107, 045501

Schaller F. M., Kapfer S. C., Schrder-Turk G. E., , 2011, Karambola 3D Minkowski Tensor Package, karambola@physik.uni-erlangen.de

Schrder-Turk G. E., Mickel W., Kapfer S. C., Schaller F. M., Breidenbach B., Hug D., Mecke K., 2013, Minkowski Tensors of Anisotropic Spatial Structure, New Journal of Physics, 15

Schrder-Turk G. E., Mickel W., Schrter M., Delaney G. W., Saadatfar M., Senden T. J., Mecke K., Aste T., 2010, Disordered spherical bead packs are anisotropic, EPL, 90

Schurmann T., 2004, Bias analysis in entropy estimation, J. Phys. A: Math. Gen., 37

Shannon C. E., 1948, A Mathematical Theory of Communication, Bell System Technical Journal, 27, 379

Shannon C. E., 1951, Prediction and Entropy of Printed English, The Bell System Technical Journal

Skoge M., Donev A., Stillinger F. H., Torquato S., , 2006a, `http://cherrypit.princeton.edu/Packing/C++/`

Skoge M., Donev A., Stillinger F. H., Torquato S., 2006b, Packing hard spheres in high dimensional Euclidean spaces, Phys. Rev. E, 74

Slonim N., Atwal G. S., Tkaik G., Bialek W., 2005, Information based clustering, PNAS, 51

Solomonoff R. J., , 1960, A preliminary report on a general theory of inductive inference

Solomonoff R. J., 1964, A formal theory of inductive inference. part i, Information and control, 7, 1

Song W.-M., Matteo T. D., Aste T., 2012, Hierarchical information clustering by means of topologically embedded graphs, PLoS ONE, 7

Stern A. S., Li K.-B., Hoch J. C., 2002, Modern Spectrum Analysis in Multidimensional NMR Spectroscopy: Comparison of Linear-Prediction Extrapolation and Maximum-Entropy Reconstruction, J. Am. Chem. Soc., 124, 19821993

Steuer R., Kurths J., Daub C. O., Weise J., Selbig J., 2002, The mutual information: Detecting and evaluating dependencies between variables, Bioinformatics, 18, 231

Sun Y., Lou J., , 2008, Molecular Dynamics: The Art of Computer Simulation, `http://catalyst.blogs.rice.edu/archives/23`

Torquato S., Jiao Y., 2009, Dense packings of polyhedra: Platonic and Archimedean solids, Phys. Rev. E, 80, 041104

Trevisan P. L., , 2015, Notes on Kolmogorov Complexity, `https://people.eecs.berkeley.edu/~luca/cs172/notek.pdf`

Turing A., 1937, On Computable Numbers, With an Application to the Entscheidungsproblem, Proceedings of the London Mathematical Society, 42, 230

UNESCO, 2013, Crystallography matters, `http://www.iycr2014.org/\_\_data/assets/pdf\_file/0010/78544/220914E.pdf`, Accessed Oct 2015

University of Pittsburgh, 2001, Cannonballs and Honeycomb: Three Dimensions - Voronoi, `http://www.math.pitt.edu/articles/voronoi.html`

van Kreveld M., Loffler M., , 2015, Computational Geometry, `http://www.cs.uu.nl/docs/vakken/ga/slides7.pdf`, Accessed Dec 2015

Voelkel K., , 1998, Melting on the Surface of a Sphere, `https://courses.physics.illinois.edu/phys466/sp2013/projects/1998/team5`, Accessed Nov 2015

von Leibniz G. W. F., 1976, Philosophical Papers and Letters: A Selection. Springer Science & Business Media

Voronoi G., 1908, Nouvelles applications des paramtres continus la thorie de formes quadratiques, Journal fr die reine und angewandte Mathematik, 134, 198

Wang X. F., Chen G., 2003, Complex networks: small-world, scale-free and beyond, Circuits and Systems Magazine, IEEE, 3

Watts D., Strogatz S., 1998, Collective Dynamics of small world networks, Nature, 393, 440

Weaire D., Langlois V., Saadatfar M., Hutzler S., 2007, Foam as granular matter, World Scientific Lecture Notes in Complex Systems, 8, 1

Weaire D., Phelan R., 1994, A counter-example to Kelvin's conjecture on minimal surfaces, Philosophical Magazine Letters, 69, 107110

Weisstein E. W., , 2014, Kepler Conjecture, `http:\slash\slashmathworld.wolfram.com\slashKeplerConjecture.html`, Accessed Oct 2015

Weisstein E. W., , 2015, Central Limit Theorem, From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/CentralLimitTheorem.html`, Accessed Oct 2015

Wheeler J., 1998, Geons, Black Holes, and Quantum Foam: A Life in Physics. New York: W.W. Norton & Co

xkcd.com, 2013, `https://xkcd.com/1155/`, Accessed Mar 2016

Appendices

Appendix A

A.1 Introduction

Appendix A shows some of the presented material relevant to this thesis, starting with posters, a paper and two abstracts from talks.

Measures of structural disorder

Abstract:
We investigate the use of an information theory measure to quantify the different degrees of disorder in structures. Specifically we use mutual information (or mutual entropy) to measure the similarity between different parts of a simulated disk packing. We retrieve that the greater the distance between two parts of a packing, the lower the mutual information between the two parts. Furthermore, we observe a meaningful trend with the packing fraction with increasing mutual information for denser packings.

What is the Mutual Information?

Mutual information (MI) measures the difference between the configurational and conditional entropies and it quantifies the difference between the joint probability and the conditional probabilities.

$$I(x,y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)log(\frac{p(x,y)}{p(x)p(y)})$$

|If log to the base 2 is used then the mutual information is given in units of bits

As the Configurational entropy is;

$$H(x,y) = -\sum_{x}\sum_{y} p(x,y)log(p(x,y))$$

And the conditional entropy for y given a value of x is;
It can be shown then that

$$H(y|x) = \sum_{x\in X}\sum_{y\in Y} p(x,y)log\frac{p(x)}{p(x,y)}$$

We can also write

$$I(x,y) = H(x,y) - H(y|x) - H(y|x)$$

Why Mutual Information?

We use mutual information as it shows how the configuration of one part of a system depends on a second. This is because of the relation

$$p(x,y) = p(x)p(y)$$

When and only when x and y are independent quantities. From this it can be seen when two quantities are independent

$$\left(\frac{p(x,y)}{p(x)p(y)}\right) = 1 \text{ and thus } I(x,y) = 0$$

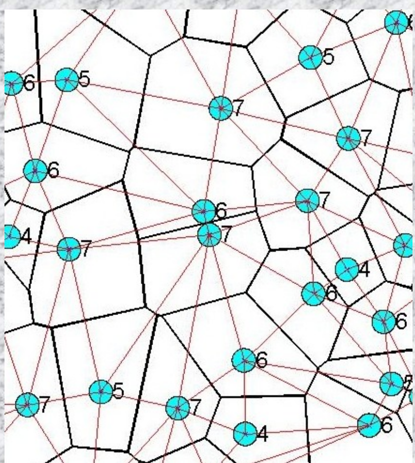


Figure 1—Example of a random packing complete with Voronoi cells and number of sides for each cell.

How we apply the Mutual Information?

First we generate a disordered packing, by placing disks at random positions in a predefined space, after which Voronoi cells statistics is applied (as shown in figure 1). From this, the number of sides for each cell can be calculated and an adjacency matrix can be constructed for the nearest neighbours and a statistical picture of the packing can be created including the entropy, probabilities, occupancy, ect

The packing is then split into squares and the Mutual information is calculated in two ways. In the first case, the occupancy of the disks in each box is used; in the second, the number of Voronoi sides is used. The below graphs were calculated using the first method.

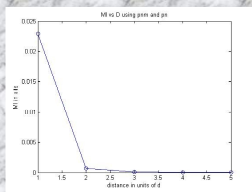


Figure 3—Graph relating MI to distance (in units of disk diameter) for a packing consisting of 58884 disks with a packing fraction of 0.517 where d=1 refers to 1st neighbour boxes d=2 to the second neighbour boxes and so on

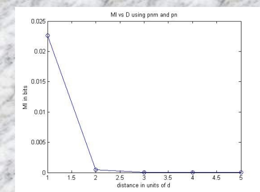


Figure 4—Graph relating MI to distance (in units of disk diameter) for a packing consisting of 26457 disks with a packing fraction of 0.524 where d=1 refers to 1st neighbour boxes d=2 to the second neighbour boxes and so on

Results

The results below show the MI related to distance by varying the box size. These start at 0.25 the size of a single disk and peaks around 0.6 before dropping back off towards zero. Ignoring the smallest sizes (which contain a large number of empty boxes) it shows as the boxes contain more disks, there is less dependency of the structure of the neighbouring boxes due to increasing distance between the disks themselves.

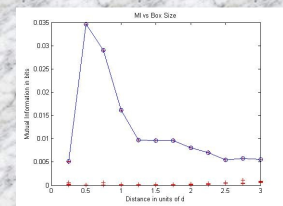


Figure 5—Graph showing the MI for nearest neighbours (in blue) as a function of box size for a packing of 73076 disks with a packing fraction of 0.519

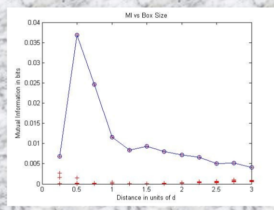


Figure 6—Graph showing the MI for nearest neighbours (in blue) as a function of box size for a packing of 58884 disks with a packing fraction of 0.517

For a crystal lattice simulations showed the MI does not depend on distance and remains close to zero (<0.005)

Packings with different disk sizes and with the same number of disks were used to simulate the MI change with the packing fraction. The graphs below show how the MI decreases for more gaseous like structures with low packing fractions for 1st neighbours for a constant box size. The clustered points with lower MI show the 2nd to 8th neighbours for comparison.

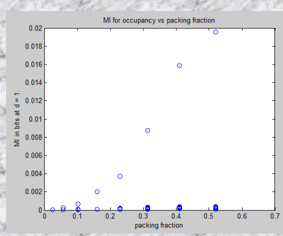


Figure 7—Graph showing MI based on occupancy of boxes for samples of 73076 with varying packing fractions. (Box Size = 2d)

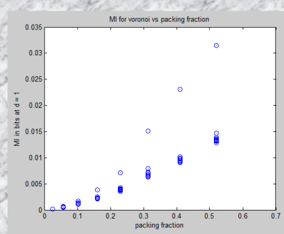


Figure 8—Graph showing MI based on counting Voronoi sides in boxes for samples of 73076 with varying packing fractions.

University of Kent
School of Physical Sciences

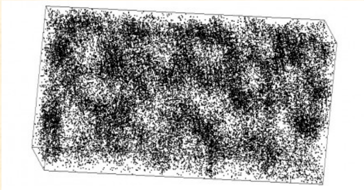
Authors : Paul Butler[1], Tomaso Aste[1]
School of Physical Sciences, Department of Science, Technology and Medical studies, University of Kent

ENCODING COMPLEX STRUCTURES

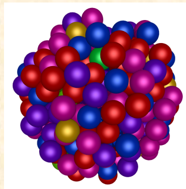
Paul Butler and Tomaso Aste

Functional Materials Group
School of Physical Sciences,
University of Kent,
Canterbury, UK

We investigate the structural organization of very large datasets acquired experimentally by means of tomographic techniques.
We aim to extract the 'genetic code' of the material that determines its functional properties.
We look directly at the positions and interrelations of very large numbers of particles and use an information filtering approach to encode the overwhelming structural information into a space-filling network of local motifs.



Three-dimensional Atom Probe Tomography of phase separated Fe-Cr system (MEMIKA Laboratory)
For the first time, atomic-scale tomography techniques are providing us a way to directly "see" the complex atomic architectures inside materials. In the next few years we will witness to a large production of experimental data concerning large-scale complex atomic aggregates that will demand the development of specific tools and a novel theoretical framework for their interpretation and use.



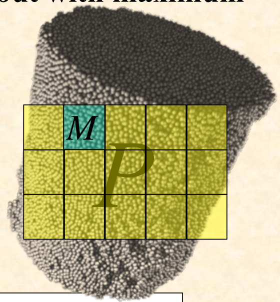
Our approach:

1. **find the local structural motifs;**
2. **measure their statistical recurrences;**
3. **identify matching rules;**
4. **build a space-filling network of motifs;**
5. **link structural encoding with functional properties.**

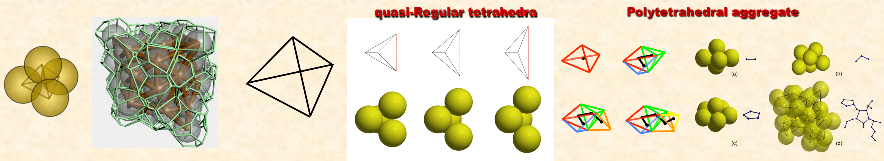
The key-idea is very simple: in absence of a pre-definite template reference structure, *we can use a part of the material as a reference structure for another part*.
Structural motifs are identified as substructures, *M*, with minimum entropy *H(M)* but with maximum *mutual information*:

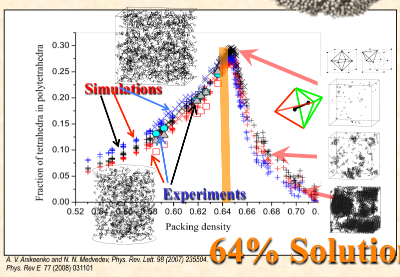
$$\mathrm{I}(P,M) = H(P) - H(P\,|M)$$

Conditional entropy (*remaining uncertainty*)
$$H(P\,|\,M) = \begin{cases} 0 & \text{if } P \text{ is completely determined by } M \\ H(P) & \text{if } M \text{ has no useful information to encode } P \end{cases}$$



Example: jamming at 64%

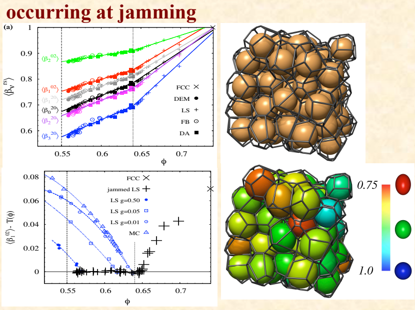




A. V. Anikeenko and N. N. Medvedev, Phys. Rev. Lett. 98 (2007) 235504; Phys. Rev E 77 (2008) 031101

64% Solution

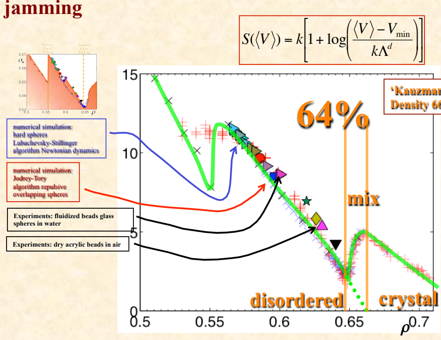
Minkowski measures of structural changes occurring at jamming

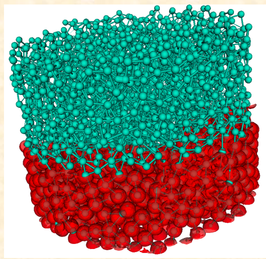


Disordered sphere packings are anisotropic!

Entropic measures of structural changes occurring at jamming

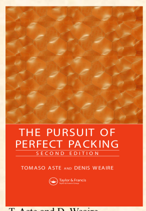
$$S(\langle V\rangle) = k\left[1+\log\left(\frac{\langle V\rangle - V_{min}}{k\Lambda^d}\right)\right]$$





References:

- W.M. Song, T. Di Matteo and T. Aste, "Hierarchical information clustering by means of topologically embedded graphs", PLoS ONE (2012).
- G. E. Schröder-Turk, W. Mickel, M. Schröter, G. W. Delaney, M. Saadatfar, T. J. Senden, K. Mecke and T. Aste, "Disordered spherical bead packs are anisotropic", Eur. Phys. Lett. 90 (2010) 34001-6.
- Gary W. Delaney, T. Di Matteo and Tomaso Aste, "Combining tomographic imaging and DEM simulations to investigate the structure of experimental sphere packings", Soft Matter 6 (2010) 2992–3006.
- Iwan Schenker, Frank T. Filser, Ludwig J. Gauckler, Tomaso Aste, Hans J. Herrmann "Characterization of the Heterogeneity of Particle Packings", Phys. Rev. E 80 (2009) 021302.
- Elsner, A. Wagner, T. Aste, H. Hermann, D. Stoyan, "Specific Surface Area and Volume Fraction of the Cherry-Pit Model with Packed Pits", Journal of Physical Chemistry B 113 (2009) 7780-7784.
- Gary W. Delaney, Stefan Hutzler and Tomaso Aste "Relation Between Grain Shape and Fractal Properties in Random Apollonian Packing with Grain Rotation" Phys. Rev. Lett. 101 (2008) 120602. (4 pages)
- Melissa Jerkins, Matthias Schroter, Harry L. Swinney,Tim J. Senden, Mohammad Saadatfar and Tomaso Aste, "Onset of mechanical stability in random packings of frictional spheres", Phys. Rev. Lett. 101 (2008) 018301. (4 pages)
- T. Aste, T. Di Matteo, "Structural transitions in granular packs: statistical mechanics and statistical geometry investigations", Eur. Phys. J. B 64 (2008) 511-517.
- Iwan Schenker, Frank T. Filser, Tomaso Aste and Ludwig J. Gauckler, "Microstructures and Mechanical Properties of Dense Particle Gels: Microstructural Characterization", J. Eur. Ceram. Soc., 28/7 (2008) 1443-1449.
- T. Aste, T. Di Matteo "Emergence of Gamma distributions in granular materials and packing models", Phys. Rev. E 77 (2008) 021309 (8 pages).
- A.V. Anikeenko and N.N. Medvedev and T. Aste, "Structural and entropic insights into the nature of random close packing limit", Phys. Rev E 77 (2008) 031101 (9 pages).
- T. Aste, T. Di Matteo, M. Saadatfar, T. J. Senden, Matthias Schröter and Harry L. Swinney, "An invariant distribution in static granular media", Euro Phys. Lett. 79 (2007) 24003
- T. Aste and T. Di Matteo "Correlations and aggregate statistics in granular packs" Eur. Phys. J. E 22 (2007), 235–240.
- T. Aste, M. Saadatfar, T.J. Senden, "Local and Global relations between the number of contacts and density in monodisperse sphere packs", J. Stat. Mech. (2006) P07010 (10 pages).
- M. Tumminello, T. Aste, T. Di Matteo, R.N. Mantegna, "A tool for filtering information in complex systems", Proceedings of the National Academy of Sciences of the United States of America (PNAS) 102 (2005) 10421-10426.
- T. Aste, M. Saadatfar ,T.J. Senden, "Geometrical structure of disordered sphere packings", Phys. Rev. E. 71 (2005) 061302.
- T. Aste, "Variations around disordered close packings", Journal of Physics Condensed Matter 17 (2005) S2361--S2390.
- T. Aste and A. Coniglio, "Cell theory for liquid solids and glasses: from local packing configurations to global complex behaviors", Europhys. Lett. 67 (2004) 165-171.



T. Aste and D. Weaire, "The Pursuit of Perfect Packing", - Second Edition- (Taylor and Francis London, 2008). (192 pages)

Philosophical Magazine
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tphm20

Self-referential order

T. Aste [a b c], P. Butler [b] & T. Di Matteo [d]
[a] Department of Computer Science, University College London, UCL Gower Street, WC1E 6BT, London, UK
[b] School of Physical Sciences, University of Kent, Canterbury, UK
[c] Applied Mathematics, Research School of Physics and Engineering, The Australian National University, ACT 0200, Canberra, Australia
[d] Department of Mathematics, King's College London, The Strand, WC2R 2LS, London, UK
Published online: 17 Sep 2013.

To cite this article: T. Aste, P. Butler & T. Di Matteo, Philosophical Magazine (2013): Self-referential order, Philosophical Magazine

To link to this article: http://dx.doi.org/10.1080/14786435.2013.835495

Philosophical Magazine, 2013
http://dx.doi.org/10.1080/14786435.2013.835495

RESEARCH ARTICLE

Self-referential order

T. Aste[a,b,c]*, P. Butler[b] and T. Di Matteo[d]

[a] *Department of Computer Science, University College London, UCL Gower Street, London, WC1E 6BT, UK;* [b] *School of Physical Sciences, University of Kent, Canterbury, UK;* [c] *Applied Mathematics, Research School of Physics and Engineering, The Australian National University, Canberra, ACT 0200, Australia;* [d] *Department of Mathematics, King's College London, The Strand, London, WC2R 2LS, UK*

(*Received 23 May 2013; accepted 12 August 2013*)

We introduce the concept of *self-referential order* which provides a way to quantify structural organization in non-crystalline materials. The key idea consists in the observation that, in a disordered system, where there is no ideal, reference, template structure, each sub-portion of the whole structure can be taken as reference for the rest and the system can be described in terms of its parts in a self-referential way. Some parts carry larger information about the rest of the structure and they are identified as *motifs*. We discuss how this method can efficiently reduce the amount of information required to describe a complex disordered structure by encoding it in a set of motifs and *matching rules*. We propose an information-theoretic approach to define a *self-referential-order-parameter* and we show that, by means of entropic measures, such a parameter can be quantified explicitly. A proof of concept application to equal disk packing is presented and discussed.

Keywords: self-referential order; disordered structure; information theory; order; structural encoding

1. Introduction

Complex, non-crystalline materials are everywhere and the capability of understanding and mastering disordered atomic packings is crucial to enhance properties of materials. The quest for understanding the internal structure of matter has been central to human curiosity since the beginning of science and, despite the remarkable achievements obtained since the Platonic theory of matter (Timaeus $\sim$ 360 BC), still we are only able to describe the structure of a very special class of materials where regular periodic (or quasi-periodic) arrangements of atoms are present. However, disorder is not randomness and nor it is a defective, degenerate form of order, real disordered structures show high degrees of organization that can propagate hierarchically through the material. Nonetheless, these structures do not present any periodic, predictable pattern and the absence of such regularity is precisely what makes disorder difficult to describe and encode in a way that is both accurate and compact. Science is measurement, but disorder is difficult to quantify. For instance, in an ordered, crystalline, system one can introduce a quantity called *'order parameter'*

*Corresponding author. Email: t.aste@ucl.ac.uk

that measures how close the system is to the perfect crystalline reference structure. This parameter is extremely useful to predict the properties of the material. But in a disordered system, there is not a unique, ideal, reference structure and a simple parameter that quantifies the kind and amount of disorder cannot be used. Even common language lacks of terms to define non-ordered structures. Indeed, we are limited to the use of negative identifications: disorder (i.e. disturbance of order) or amorphous (i.e. absence of shape). Such a lack of vocabulary is probably consequence of the fact that the classification of crystalline structures has been one of the great success stories of modern science which has induced us to overlook the evidence that "disordered" and "amorphous" materials are everywhere and the mastering of good techniques to describe atomic disorder is crucial to enhance material performances. It has been recently shown that 'order' in amorphous structures can be identified by looking at 'patches' that repeat more often than typical [1]. This approach reveals diverging correlation lengths at glass transition [2] shading light on the relations between thermal glass transition and athermal jamming of discrete matter [3]. In this paper, we follow a similar approach to [1] using information-theoretic methods to quantify, in a self-referential way, an ordered parameter and identifying the locally most referential structures.

There are two main technical challenges that have so far slowed down the progress in this field. The first has been the lack of experimental data. Indeed, until recently, diffraction techniques have been the main experimental tools to study atomic structures inside the bulk of materials. However, diffraction gives insights only on the average relative positions of the constituents and the reconstruction of the structure from diffraction data becomes very hard in absence of regularly repeated local units. Now, for the first time, atomic-scale tomography techniques are providing us a way to directly "see" the complex atomic architectures inside materials. Indeed, in the last few years, techniques such as Atom Probe Tomography and Electron Tomography have started to provide direct information about the position of millions of atoms in the bulk of materials [4–9]. In the next few years, we will witness a large production of experimental data concerning large-scale complex atomic aggregates. However, this brings up the second technical challenge concerning the huge size of data to process demanding the development of specific tools and a novel theoretical framework for their interpretation and use. Indeed, in absence of a compact way to encode structural complexity, the processing of this amount of information is still beyond the capability of the world's largest supercomputers. The total world information storage capacity, currently estimated 10^{20} bits, would not be enough to encode the structure of a gram of disordered matter. There is therefore a demand to develop a general approach to encode complex structures and reduce the amount of information to the relevant part related to the material's functional properties. In principle, in a disordered material positions, properties and the interactions of every atom must be recorded independently. In some special cases, when the structure is a regular periodic repetition of identical parts (i.e. crystals), the problem can be reduced to the study of the unit cell: a local sub-structure that repeats periodically in space, however this cannot be directly extended to non-crystalline materials. Nonetheless, even in these 'disordered' materials, geometrical, physical and chemical laws impose local regularities that spontaneously develop into a structural organization spanning the whole system. In this paper, we show that these regularities can be identified as a set of local motifs that combine together into a hierarchically organized space-filling complex network in a analogous way as an alphabet combines into words which assemble into phrases forming the whole text. Retrieving the 'alphabet', identifying the 'words', uncovering the

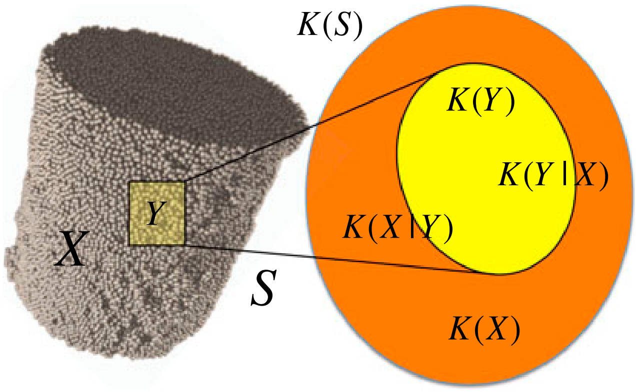


Figure 1. (colour online) In the absence of a pre-defined template reference structure, one can use a portion (Y) of the structure to describe the whole structure $S = X \cup Y$. The knowledge about the portion Y can reduce the uncertainty about the rest of the structure X. Kolmogorov complexity, here denoted with $K(X)$ and $K(Y)$, measure the information contained in X and Y, respectivelly. For instance, in the case in which the rest of the structure X is completely determined by the knowledge of the portion Y, we have $K(S) = K(Y)$. In this case, the conditional information about X given Y, $K(X|Y)$, is equal to zero.

'grammatical' rules and ultimately, decoding the 'syntax' is the key to describe the structure of non-crystalline matter.

2. Describing the structure in terms of itself: self-referential order

The key-idea at the basis of the present work is very simple: in the absence of a pre-definite template reference structure, we can use a part of the material as a reference structure for another part. The structure is consequently encoded with a *self-referential* description. For instance, from a general information-theoretic perspective, we can re-interpret the identification of the unit cell of a crystalline structure has a very efficient way to encode a structure with the amount of data required to encode the structure passing from order n to order 1. Even in the absence of any previous knowledge of crystallography it is still rather straightforward to identify the unit cell from the information about the positions of all atoms. Indeed, it is sufficient to take a portion of the structure, translate it in space and see when and where it perfectly overlaps with another part of the structure. The smallest portion of the structure that periodically overlaps with all other parts of the structure is the unit cell. In the context of this paper, this is the simplest case of self-referential description where only one local motif -the unit cell- is sufficient to entirely describe the whole crystal.

2.1. *An ideal approach*

Let us consider a structure S and let us consider it as composed of a large portion X and a smaller portion Y, so that $S = X \cup Y$. To measure the amount of self-referential order, we need to be able to quantify how the knowledge about the portion Y reduces the amount of information needed to encode X. Formally, we need a measure of information content such as the Kolmogorov complexity K [10–13]. In simple terms, the quantity $K(X)$ is the amount of information necessary to describe X. Its conditional counterpart, $K(X|Y)$, is the amount of information necessary to describe X, given the full knowledge of Y. When the knowledge about a portion Y of the structure is sufficient to describe the rest of the structure, we must have $K(S) = K(Y)$ and $K(X|Y) = 0$. Conversely, when the knowledge about a portion does not add any knowledge about the rest of the structure we must have $K(X|Y) = K(X) > 0$.

We could therefore introduce the *self-referential order parameter*

$$s_X(Y) = 1 - \left(\frac{K(X|Y)}{K(X)}\right), \tag{1}$$

which is equal to one if the system is fully self-referentially ordered and it is equal to zero if completely random. This approach formally defines *self-referential order* and it would solve the problem. However, –unfortunately– Kolmogorov complexity is not a computable quantity.

2.2. *The entropic way*

A computable quantity that measures information content is the *entropy* that, in the Shannon formulation [14], can be written as:

$$H(X) = -\sum_{r_X} p_X(r_X) \log_2 p_X(r_X), \tag{2}$$

where $p_X(r_X)$ is the probability of occurrence, in X, of a configuration with a given set of structural properties, denoted with r_X. Entropy is everywhere in physics; it is a thermodynamic state variable and the Shannon formula coincides with the Gibbs derivation (with base-e log and multiplied by k_B [15]) of the entropy for the canonical ensemble. Here, we shall use entropy for its information significance: $H(X)$ is the amount of information encoded into a structure X when its properties r_X are considered. We shall therefore use entropic measure of information instead of the Kolmogorov complexity. Let us note that Kolmogorov complexity of X is the size of the smallest programme that generates X, instead Shannon entropy measures smallest number of bits required, *on average*, to describe X [13]. The two measures can coincide in some special cases (signals computable by a Turing machine) but not in general, though they are related [16].

In analogy with the previous section, we can therefore look for the information about X provided by the knowledge of Y. The remaining entropy of variable X when variable Y is known is quantified by the conditional entropy $H(X|Y)$. Therefore, an entropic measure of self-referential order is:

$$s_X(Y) = 1 - \frac{H(X|Y)}{H(X)}. \tag{3}$$

We have $0 \leq H(X|Y) \leq H(X)$, therefore this quantity is defined in the interval $0 \leq s_X(Y) \leq 1$ where 0 is associated to a random state and 1 is instead observed for perfect self-referential order. We can use the identity $H(X|Y) = H(X,Y) - H(Y)$ obtaining the equivalent expression

$$s_X(Y) = 1 - \frac{H(X,Y) - H(Y)}{H(X)}, \tag{4}$$

which also reads

$$s_X(Y) = \frac{H(X) + H(Y) - H(X,Y)}{H(X)}. \tag{5}$$

One may notice that the quantity on the numerator is the mutual information: $I(X;Y) = H(X) + H(Y) - H(X,Y)$, therefore this measure quantifies the relative mutual dependence between structures X and Y.

3. Motifs

There must be parts of the structure that carry larger amount of information about the whole structure with respect to others. These high information-content portions are repeated similarly in the structure more often that others and therefore they are of particular relevance. We look for local sub-structures containing maximal relative information. We shall call them 'motifs' these are equivalent to the 'patches' used in [1]. In general, more than one motif is necessary to encode a disordered structure. Furthermore, these motifs do not repeat perfectly across the structure and therefore they must be described in statistical terms. Motifs are the set of local structures from which the whole structure can be most efficiently encoded. Frequency of occurrence, fluctuations and relations between motifs characterize and quantify the kind and amount of disorder in the structure. We then use these motifs as an encoding alphabet and we search for an efficient description of the entire structure with the shortest code-length. By identifying the recurrent structural motifs and by uncovering the rules governing their combination into a space-filling network, we can encode the structure of complex materials into a compressed format.

Motifs can be identified from Equations 1 or 3 by looking at the local parts that maximally contribute to the information about the whole structure, i.e. the portions Y associated with the largest $s_X(Y)$. Once the motifs are identified, one must quantify their recurrence in the structure. This can be done in three steps: (i) count the relative frequency of occurrence of each local motif; (ii) compute the probability distribution of its fluctuations; and (iii) estimate the entropy. A computationally fast identification of the motifs in presence of structural fluctuations is a very challenging task. Another challenge is associated with possible overlaps between motifs that makes their unique identification ambiguous and requires the introduction of "exclusion rules" (i.e. when two motifs overlap, only one must be counted at the time) and statistical ensamble analysis (i.e. all encodings resulting from the different exclusions) must be considered.

Motifs are building blocks that connect to each-other forming a space-filling three-dimensional structure. When described in terms of motifs, the structure is characterized by two aspects: (1) topology – a network of interconnected motifs; and (2) geometry, where position and orientation of each motif is specified. Due to the possible overlaps between motifs, there can be more than one network for a given structure, the ensamble all these networks must be considered. For a given network, the matching rules can be identified from a statistical study of local co-occurrences. Matching rules are both topological and geometrical. Indeed, motifs can join together only in specific relative positions and orientations.

The description of a structure in terms of the network of motifs and their matching rules provides a compact encoding of the structure. For example, a crystal is reduced to only one motif (a parallelepipedal unit cell), one topological matching rule (6 neighbors) and one geometrical matching rule (unit cells join by opposite faces). In general, for a complex structure we have a large – but finite and non-scaling – number of motifs m and a order $O(m^2)$ of matching rules. Therefore, the amount of information required to encode the structure is of the order $O(m^2)$. A-priori it is quite hard guesswork to estimate the size of m,

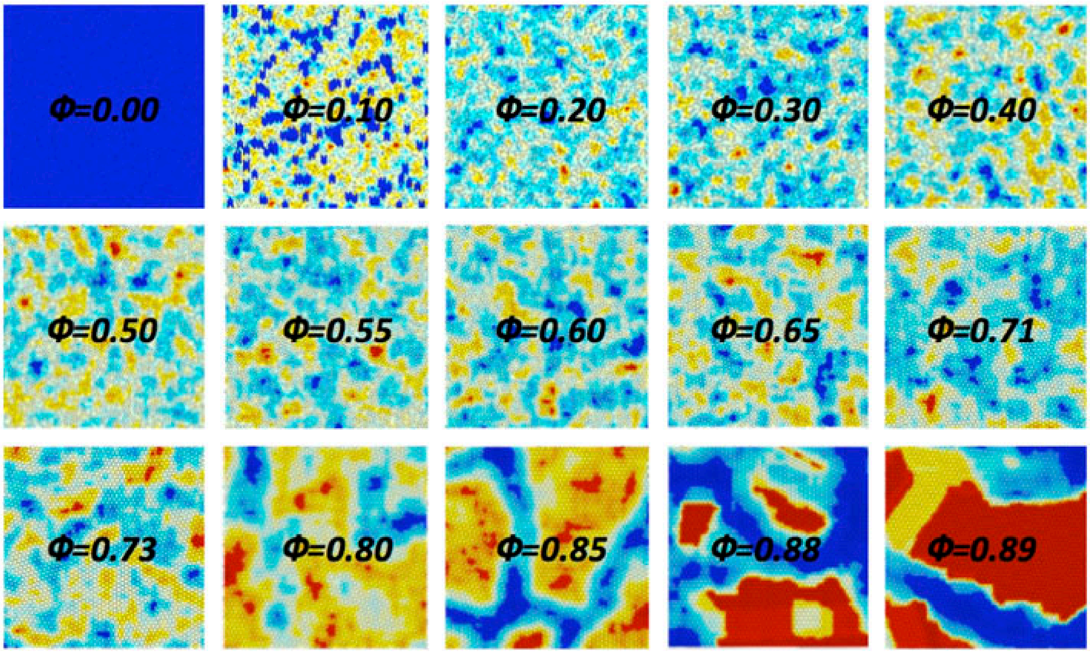


Figure 2. (colour online) Snapshots of the local self-referential order parameter $s_X(Y)$. The local portion Y is a square of edge 5 disc diameters. The pictures are a heat map (blue low red high, color online) representing the relative values of $s_X(Y)$ for a portion centred in each given part of the packing. Φ indicates the packing fraction of each sample. Colourmap is rescaled for each image.

which -of course- varies from system to system. The experience acquired with disordered sphere packings [17–19] suggests us that in these systems m is of the order of 10^2, and the matching rules are of the order of 10^4 (note that resolving all the reciprocal orientations can be demanding). This may seem a large number but it must be pointed out that in terms of information compression, we are passing from an information size of the order of 10^{20} (hundreds of billions of gigabytes), which is certainly beyond computable sizes, to a size of 10^4 bytes (tens of kilobytes), which is computationally insignificant. Furthermore, for many practical purposes, a precise definition of the local geometrical configuration and its orientation is often irrelevant and the information can therefore be further reduced. Let us here explain this point with an example from the results on sphere packings in [20] where it was shown that local tetrahedral motifs are related to the description of a structural transition at the Random Close Packing limit. In that paper, it was shown that the controlling parameter were the length of the tetrahedral edges with an effective differentiation between “short” or “long” edges. In terms of motifs, given that each tetrahedron has six edges and each edge has two states, in this special case we can count $m = 2^6 = 64$ relevant motifs. To compute the matching rules, we must then consider that tetrahedra match face by face and they have 4 faces giving $(4 \times 2^6)^2/2 \sim 33,000$ matching rules. However, these numbers can be greatly reduced, for instance, in [20], it was shown that the most relevant motifs were only 2: all-short-edges or not. And the relevant matching statistics was given by the chains of all-short-edges tetrahedra.

4. Results

In this paper, we report a preliminary investigation about the quantification of self-referential order in two-dimensional disks packings generated via molecular dynamic simulations.

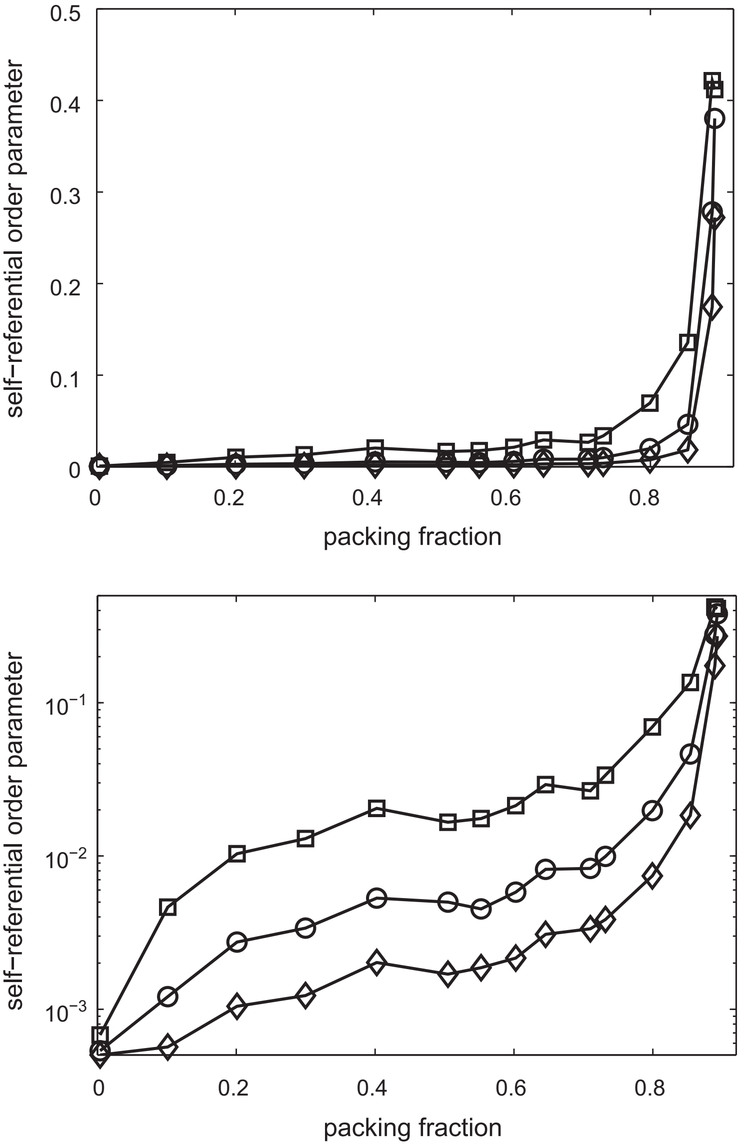


Figure 3. Global values of the self referential order parameter $\hat{s}$ vs. packing fraction displayed in both linear and semi-logarithmic scale. Different curves (◇, ○ or □ symbols) correspond to different sizes of the local portion Y, which are squares, respectively, with edges equal to 3, 5 or 10 disk-diameters.

The results presented here are a 'proof of concept' demonstrating that this method can be used quantitatively. Extended applications to three-dimensional structures from simulations and experiments are under investigation.

We generate several packings of disks at various packing fractions by using the algorithm proposed by [21], which is a molecular dynamic simulation with constant compression rate. We terminate the simulation when a desired packing fraction is reached, before the reach of (local) jamming. We report results for 15 samples comprising 5,000 disks representing a range of packing fractions between 0 to ~ 0.9.

We compute the self referential order parameter by looking at the Voronoï volumes around each disk and identifying a set of $m = 500$ kinds of motifs classified in terms of their different volumes. We verify that the method is robust against this choice with analogous results obtained for $m = 100$ or $m = 2,000$. We then take a local square portion

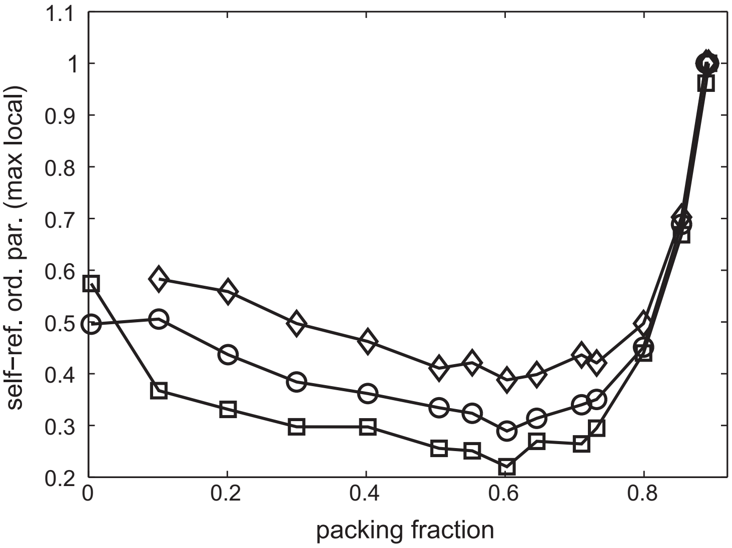


Figure 4. Average maximum local values of the self referential order parameter for each sample. The average is over the 10% largest $s_X(Y)$. Different curves ($\diamond$, $\circ$ or $\square$ symbols) correspond to different sizes of the local portion Y, which are squares, respectively, with edges equal to 3, 5 or 10 disk-diameters.

Y of the sample and compute $s_X(Y)$ by applying Equation 5. We repeat the process in 10,000 different portions regularly displaced across each sample.

In Figure 2, distributions of the local self-referential order parameter $s_X(Y)$ inside each sample and across the samples are shown. One can note that the values are low at low packing fractions where the system is essentially in a random state. Conversely, they are large at high packing factions where the system starts nucleating crystalline regions. This is quantified and shown in Figure 3 where we report a global measure of self referential order parameter ($\hat{s}_X(Y)$) computed by estimating the joint probability to have given fractions of Voronoï volumes simultaneously present in any of the portions Y and in the rest of the sample $X = S \cap Y$. One can see that the self referential order parameter increases with packing fraction to reach a maximum at the largest packing of $\Phi \sim 0.9$. From the semi-log plot in Figure 3, we can note that this parameter ranges over 4 order of magnitude, with an interesting plateau appearing between packing fractions ~ 0.4 and ~ 0.7. Let us note that the largest packing fraction attainable for equal disks is $\Phi = \pi/\sqrt{12} \simeq 0.907$ [22], which corresponds to a perfectly ordered, crystalline, triangular packing. Our densest packing has still some defects that lower slightly its packing fraction. These defects are clearly visible in Figure 2 where one can appreciate that in correspondence with miss-alignment of the crystalline order we observe lower values of $s_X(Y)$. Indeed, these defective regions are less representative of the sample. We can also note that, conversely, at lower packing fractions the most representative local portions are not compact configurations with crystalline symmetry but rather more complex and less compact configurations. In general, at different packing fractions different local configurations carry more or less information about the rest of the sample structure. We investigated the presence of highly referential motifs by looking at the maximum values of $s_X(Y)$ in each sample. Specifically, we quantified the portions of sub structures carrying the largest information by identifying the 10% largest $s_X(Y)$ per each sample. In Figure 4, we show the values of the average self referential order parameter $s_X(Y)$ in this top 10% subset of most representative configurations. One can note that at large packing fractions, where the structure is essentially crystalline, only few configurations carry all structural information. Interestingly, also at very low packing fractions, where the structure is essentially random, again a small part of the most informative configurations characterize well the whole structure. On the other hand, at intermediate packing fractions -around $\Phi \simeq 0.6$ – the structure is more complex and even the most informative local configurations carry, in average, a smaller amount of information about the rest of the system.

5. Conclusion

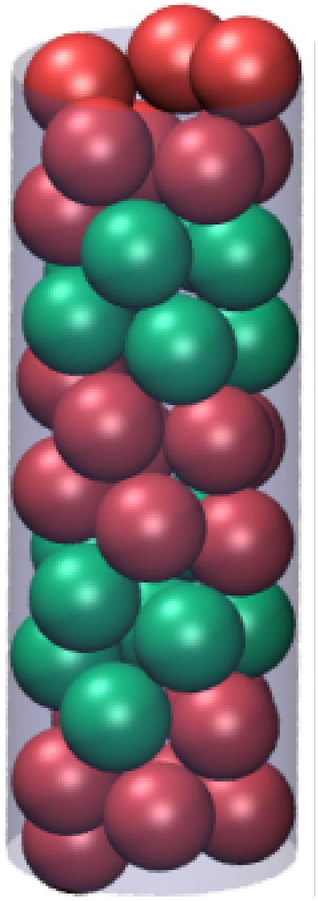
We addressed the intriguing question concerning how atoms organize themselves inside non-crystalline, complex materials and how to extract, filter and encode this information in an efficient and meaningful way. To this purpose we introduced the concept of self-referential-order and we proposed a method to quantify it from entropic measures. There are over one billion trillion atoms in a gram of matter, and in the absence of a regular, ordered arrangement, the characterization of an amorphous structure would require accounting for the position of every atom. This is an impossible task that would require over a billion terabytes. However, the material functional properties are associated with a much smaller amount of information. In this paper, we have illustrated a general approach to encode complex structures and to reduce this overwhelming amount of information to the relevant part related to the material's functional properties. Our method can be used to select the most informative portions of the material, the 'motifs', and encode the complex structure in a set of motifs and matching rules reducing dramatically the amount of information required. In this paper, we present a 'proof of concept' with application to equal disk packing at different packing fractions. We found that the self-referential-order parameter well characterizes globally the transition towards crystallization, but also it identifies locally the emergence of an increasing complexity at intermediate packing fractions. Future studies will be dedicated to the analysis of three-dimensional structures from experiments and large scale simulations. Our information filtering and encoding techniques can be directly applied to very different kinds of complex structures which are defined in high-dimensional phase-spaces: the study of the structure of dependency in financial systems [23,24] or the structure of gene co-expressions in biological systems [25].

References

[1] J. Kurchan and D. Levine, J. Phys. A: Math. Theor. 44 (2011) p.035001.
[2] François Sausset and D. Levine, Phys. Rev. Lett. 107 (2011) p.045501.
[3] G. Biroli and J.P. Garrahan, *Perspective: The glass transition*. arXiv, preprint arXiv:1303.3542, 2013.
[4] P. Ronhovde, S. Chakrabarty, D. Hu, M. Sahu, K.K. Sahu, K.F. Kelton, N.A. Mauro and Z. Nussinov, Scientific reports 2 (2012). Article number 329.
[5] S. Van Aert, K.J. Batenburg, M.D. Rossell, R. Erni and G. Van Tendeloo, Nature 470 (2011) p.374.
[6] R.K.W. Marceau, L.T. Stephenson, C.R. Hutchinson and S.P. Ringer, Ultramicroscopy 111 (2011) p.738.
[7] B. Gault, M.P. Moody, F. De Geuser, A. La Fontaine, L.T. Stephenson, D. Haley and S.P. Ringer, Microsc. Microanal. 16 (2010) p.99.

[8] M.K. Miller and R.G. Forbes, Mater. Char. 60 (2009) p.461.
[9] M.K. Miller, *Atom probe tomography*, in *Handbook of Microscopy for Nanotechnology*, N. Yao and Z.L. Wang, eds., Springer, Berlin. 2005, p. 227
[10] A. Nikolaevich Kolmogorov, Int. J. Comput. Math. 2 (1968) p.157.
[11] R.J. Solomonoff, Information control. 7 (1964) p.1.
[12] R.J. Solomonoff, *A preliminary report on a general theory of inductive inference*. United States Air Force, Office of Scientific Research, 1960.
[13] M. Li and Paul MB. Vitanyi, An introduction to kolmogorov complexity and its applications, Springer, 1993.
[14] C.E. Shannon and W. Weaver, A mathematical theory of communication, The University of Illinois Press, Urbana, IL, 1949.
[15] D. Chandler, Introduction to modern statistical mechanics, in Introduction to Modern Statistical Mechanics, D. Chandler, eds., Foreword by David Chandler. 1 Sep 1987, Oxford University Press, 1987, p. 288. ISBN-10: 0195042778. ISBN-13: 9780195042771.
[16] S. Leung-Yan-Cheong and T. Cover, IEEE Trans. Information Theor. 24 (1978) p.331.
[17] G.W. Delaney, T. Di Matteo and T. Aste, Soft Matter 6 (2010) p.2992.
[18] T. Aste and T. Di Matteo, The: Eur. Phys. J. B. 64 (2008) p.511.
[19] T. Aste, M. Saadatfar and T.J. Senden, J. Stat. Mech. P07010 (2006).
[20] A.V. Anikeenko, N.N. Medvedev and T. Aste, Phys. Rev. E 77 (2008) p.031101.
[21] M. Skoge, A.D.F.H. Stillinger and S. Torquato, Phys. Rev. E 74 (2006) p.041127.
[22] T. Aste and D. Weaire, *The Pursuit of Perfect Packing*, Institute of Physics, Bristol, 2000.
[23] T. Aste, W. Shaw and T. Di Matteo, New J. Phys. 12 (2010) p.085009.
[24] F. Pozzi, T. Di Matteo and T. Aste, Scientific Reports 3 (2013) p.1665.
[25] W.-M. Song, T. Di Matteo and T. Aste, PLoS ONE 7 (2012) p.e31929.

International Workshop on PACKING PROBLEMS,
2-5 September 2012, Trinity College Dublin, Ireland.



Organizers: Prof. Stefan Hutzler (Trinity College Dublin)
Dr. Adil Mughal (Aberystwyth University)
Dr. Ho-Kei Chan, *Workshop Secretary* (Trinity College Dublin)

Financial support:

sfi science foundation ireland fondúireacht eolaíochta éireann

Fáilte Ireland National Tourism Development Authority

meetinireland.com

Host:

School of Physics

TRINITY COLLEGE DUBLIN COLÁISTE NA TRÍONÓIDE, BAILE ÁTHA CLIATH

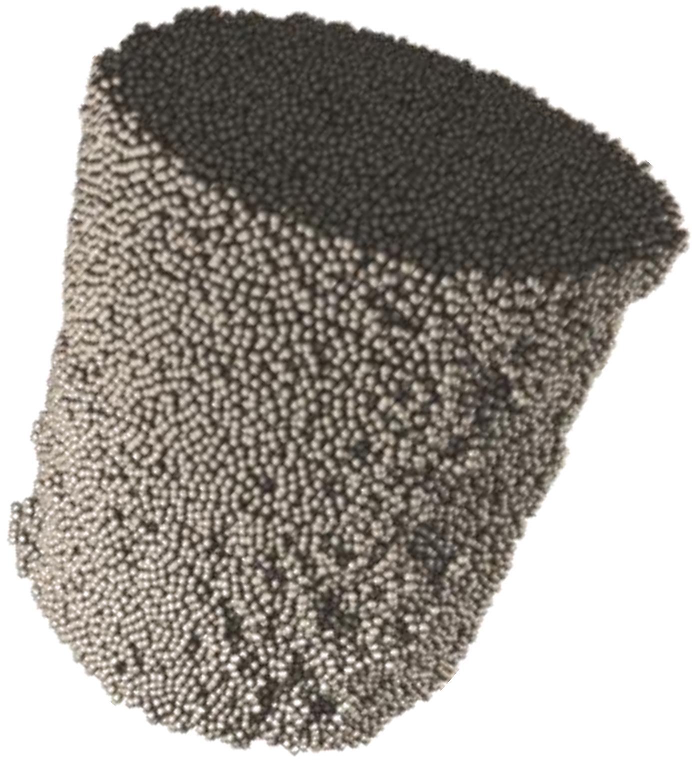
THE UNIVERSITY OF DUBLIN

Imaging Order in Complex Structures

Paul Butler

In this talk I will discuss the application of information theory to the study of complex structures by looking at random monodisperse sphere packings. I will demonstrate how using entropy driven measurements like the Kullback Leibler divergence we can show local motifs in these structures. The talk will include methodology on how we extract statistics by Voronoi analysis, and how to calculate the redundant information in a structure so we can show areas of low and high order. This will be a step forward in understanding how we can characterise disorder, and how we might encode disordered structures in the most efficient way.

I will show colour maps showing relative information in random disk packings and use them to visualise local motifs for disordered loose packing up to highly ordered crystal structures with various packing fractions. We will finish with how the absolute value depends on the information in the system, and how this might improve our definition of disorder.



The use of Information Theory to describe disordered structures

In this talk we will look at an information theoretic approach to characterising disordered systems, including the use of computer models and granular systems as an approximation of amorphous atomic systems. We will discuss extracting statistical information from said systems after which the entropy can be calculated, and how this leads to a formal description in the amount of information in a system. From this other quantities can be calculated, such as the mutual information, which can be quantified, to give a description of the dependency of one part of the system to the rest, which allows us to find how much shared statistical information is held in the system (how much redundant information there is). For example in a crystalline packing, almost all the information is contained in the unit cell. We shall explore mutual information in the Gaussian case, and how this can be applied to atomic packings.

Appendix B

B.1 Introduction

Appendix B shows a number of codes that were used for calculation of various results shown in this thesis. All Matlab codes were written by myself and Tomaso Aste except where stated otherwise. C++ code shows modification of original code which can be found in the Bibliography. Descriptions can be found in the table of contents. These codes are modified depending on the experiments and so may not function correctly when applied to different data. If there are any questions about the code or how to modify it please feel free to contact me.

B.2 Programming Languages

A programming language, broadly speaking, is a library of terms which can be compiled into instructions a computer can use and carry out. While ultimately this means it is converted into binary code, only "1st level" programming languages directly do this, programs built only on these libraries are called "2nd level" languages, programs built on the first two are called "3rd level" and so on. Compiling refers to the 'translation' of a program into Binary. Errors in libraries on lower levels can cause major issues, for example, in one function in Matlab that was used, there existed a memory leak, meaning memory was not cleared properly, filling up more and more space. After so many loops the program fails, due to insufficient room.

B.2.1 Matlab

Despite these issues, Matlab (Mathworks, 1984), short for matrix laboratory, is a widely used and reliable program based on codes from C, Java, Fortran and a number of others, making it a fourth level language. Matlab is created and maintained by Mathworks. It was created in the 1980's, with its latest version being released this year as Matlab R2015a. Mainly used in the sciences for visualising and manipulating large sets of data and functions, it is programmed with many high-level commands, such as being able to create Voronoï tessellations and random data sets as seen in section 4.1.2. It is capable of using multiple CPUs and GPU arrays for parallel computing. There are a number of toolboxes to add functions for Curve fitting, optimisation, networks, image processing, data acquisition, finance, bio-informatics as well as many others (Mathworks, 2015b). All of the codes used to calculate entropy were created in Matlab. Hybrid codes also ran from the Matlab environment. Hybrid codes simply refer to where multiple programming languages are used. In this case, codes are called individually after being compiled in their respective languages, and run using a shell command. This way quantities are available in Matlab, and the programs can be loops and modified as required.

B.2.2 C++

C++ is a far more general programming language. While used to manipulate and present data, it is used to program device firmware, runs a variety of computer programs and many others. It is even a standard by the International Organisation for Standardisation (ISO) (ISO, 2015). It was based upon the language C and began use in 1983 and is used on a variety of operating systems. Many of the packing programs found are based on C++ including the Molecular Dynamics Codes used in this work. The Minkowski tensor programs were also written in C++.

```
%calc ent of eig values and anistropy (whole packings not cells and boxes!)

clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PROGRR = ('/spheres'); %sphere creation exe %3D @ moment
PROGMT = ('/karambola'); %MT creation exe

path = ('/media/LinuxExtension/Anisotropy');
%('/home/paul/Work/MyFiles/C++Code'); %path for files to be used in
Rpath = ('/ComplexSpheres'); %additional folders for the sphere and MT
programs
Mpath = ('/karambola-1.5');
Vpath = ('/karambola-1.5/demo/pointpattern2voronoi3d');

PROGV = ('/pointpattern2voronoi3d');

%parameters for sphere generation
numberofspheres = 10000; %default is 100
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%creates voronio polygons for MT to read using demo karambola program
system('find write*.xyz > filenamesxyz');
system('find write*.dat > filenamesdat');
namesdat = importdata('filenamesdat');
namesxyz = importdata('filenamesxyz');
for ii = 1:1:length(namesxyz)
    name = namesxyz{ii};

    OUT = [' -o ','hs-
',num2str(numberofspheres),'_',name(6),'.',name(7:12),'.xyz2']; %don't define
output to keep same name with .poly
    INOUT = [' -i ',name, OUT]; %options and input
    VVpath = [path Vpath PROGV INOUT];

    system(VVpath);
end

system('rename s/\.xyz2$/.poly/ *.xyz2 '); %for use when defining output
system('mv *.poly ./polys');


system('./run_karambola.sh');
system('find ./polys/hs*val -prune > folders')
MTfolders = importdata('folders');

MTnames = cell(6);
MTnames{1} = 'w020_eigsys'; %%%%
MTnames{2} = 'w102_eigsys'; %%%%
MTnames{3} = 'w120_eigsys';
MTnames{4} = 'w202_eigsys';
MTnames{5} = 'w220_eigsys';
MTnames{6} = 'w320_eigsys';
```

```
w020e = zeros(numberofspheres,3);
w102e = zeros(numberofspheres,3);
w120e = zeros(numberofspheres,3);
w202e = zeros(numberofspheres,3);
w220e = zeros(numberofspheres,3);
w320e = zeros(numberofspheres,3);

TMT = cell(length(MTfolders),1);
for ii = 1:1:length(MTfolders)
    folder = MTfolders{ii};
    DELIMITER = ' ';
    HEADERLINES = 100000;
    xw020e = importdata([folder, '/', MTnames{1}], DELIMITER, HEADERLINES);
    xw102e = importdata([folder, '/', MTnames{2}], DELIMITER, HEADERLINES);
    xw120e = importdata([folder, '/', MTnames{3}], DELIMITER, HEADERLINES);
    xw202e = importdata([folder, '/', MTnames{4}], DELIMITER, HEADERLINES);
    xw220e = importdata([folder, '/', MTnames{5}], DELIMITER, HEADERLINES);
    xw320e = importdata([folder, '/', MTnames{6}], DELIMITER, HEADERLINES);
    for jj = 1:numberofspheres
        w020e(jj,1) = str2num(xw020e{jj+1}(22:41));
        w020e(jj,2) = str2num(xw020e{jj+1}(102:122));
        w020e(jj,3) = str2num(xw020e{jj+1}(183:202));
        w102e(jj,1) = str2num(xw102e{jj+1}(22:41));
        w102e(jj,2) = str2num(xw102e{jj+1}(102:122));
        w102e(jj,3) = str2num(xw102e{jj+1}(183:202));
        w120e(jj,1) = str2num(xw120e{jj+1}(22:41));
        w120e(jj,2) = str2num(xw120e{jj+1}(102:122));
        w120e(jj,3) = str2num(xw120e{jj+1}(183:202));
        w202e(jj,1) = str2num(xw202e{jj+1}(22:41));
        w202e(jj,2) = str2num(xw202e{jj+1}(102:122));
        w202e(jj,3) = str2num(xw202e{jj+1}(183:202));
        w220e(jj,1) = str2num(xw220e{jj+1}(22:41));
        w220e(jj,2) = str2num(xw220e{jj+1}(102:122));
        w220e(jj,3) = str2num(xw220e{jj+1}(183:202));
        w320e(jj,1) = str2num(xw320e{jj+1}(22:41));
        w320e(jj,2) = str2num(xw320e{jj+1}(102:122));
        w320e(jj,3) = str2num(xw320e{jj+1}(183:202));
    end
pf = folder(18:25);

save([folder(9:24),'.mat'],'w220e','pf','w320e','w120e','w202e','w020e','w102
e');
TMT{ii} = load([folder(9:24),'.mat']);
end

clear pf Z2 Y2 U2 D2 P2 Q2

for f = 1:length(TMT)

    Z = TMT{f}.w320e; %curve weighted surface intergral
    Z1 = abs(Z(:,1)./Z(:,3));
    Z2(f) = mean(Z1);
    Y = TMT{f}.w220e;
    Y1 = abs(Y(:,1)./Y(:,3));
    Y2(f) = mean(Y1);
```

```
    U = TMT{f}.w202e;
    U1 = abs(U(:,1)./U(:,3));
    U2(f) = mean(U1);
    D = TMT{f}.w120e;
    D1 = abs(D(:,1)./D(:,3));
    D2(f) = mean(D1);
    P = TMT{f}.w020e;
    P1 = abs(P(:,1)./P(:,3));
    P2(f) = mean(P1);
    Q = TMT{f}.w102e;
    Q1 = abs(Q(:,1)./Q(:,3));
    Q2(f) = mean(Q1);

    pf(f) = str2double(TMT{f}.pf);

end
%
% figure
% plot(pf,Z2)
% figure
% plot(pf,Y2)
% figure
% plot(pf,U2)
% figure
% plot(pf,D2)
% figure
% plot(pf,Q2)
% figure
% plot(pf,P2)

figure(10)
plot(pf,Z2,'ro')
hold on
plot(pf,Y2,'g+')
plot(pf,U2,'b*')
plot(pf,D2,'cdiamond')
plot(pf,P2,'k.')
plot(pf,Q2,'ysquare')
xlabel('Packing Fraction')
ylabel('Mean Eigenvalue Difference')
legend('W_{3}^{(2,0)} Eigenvalues','W_{2}^{(2,0)} Eigenvalues','W_{2}^{(0,2)}
Eigenvalues','W_{1}^{(2,0)} Eigenvalues','W_{0}^{(2,0)}
Eigenvalues','W_{1}^{(0,2)} Eigenvalues','Location','northwest')
```

```
clear all
close all

ii=0;
Nt = 50000;
    bestbin = 15;
for a = [0.899:0.1:0.999]; %then make a constant and change no of elements
each method
    ii = ii+1;
    %bestbin = 10;

    %Calculte Packing
    RR = MVG([0 0]',[1 a; a 1],Nt)';
    %RR = MVG([0 0]',[1 a; a 1],Nt)';
    %RR(:,2) = b.*RR(:,1) + (RR(:,1)); %linear
    RR(:,2) = (RR(:,2)).^2;
    %RR(:,2) = b.*RR(:,2) + log2(RR(:,1)); %log

    %plot(RR(:,1),RR(:,2),'.')

    roe1t = corrcoef(RR);
    roe1(ii) = roe1t(1,2);

    %Calculate best binning
    %bestbin = ceil(sqrt(sqrt(Nt).*0.3))+1;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %calc stats
    bin = [bestbin, bestbin];
    [fxy,bxby] = hist3(RR,bin);
    fxy2 = reshape(fxy,[],1);
    hisN = histc(fxy2,[min(fxy2):max(fxy2)]); %histc(A,[min(A):max(A)])
%%prob of a number occuring

    px = sum(fxy)./sum(sum(fxy));
    py = sum(fxy')./sum(sum(fxy));
    pxy = fxy./sum(sum(fxy));
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %calc MI
    Sxy = sum(pxy(pxy>0).*log(pxy(pxy>0)));
    Sx  = sum(px(px>0).*log(px(px>0)));
    Sy  = sum(py(py>0).*log(py(py>0)));
    MIdiscrete(ii) = Sxy-Sx-Sy;

    x = pxy.*(log((pxy./(px'*py))));
    MIdiscrete2(ii) = sum(x(isfinite(x))); %check using MI eq.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %kernal MI
    [bandwidth,F,X,Y] = kde2d(RR,64);
    F(F<=0) = 10^(-99);
    x = RR(:,1);
```

```
    y = RR(:,2);
    hxy_fun = @(xm,ym)(interp2(X,Y,F,xm,ym)).*(log2(interp2(X,Y,F,xm,ym)));
    hxy(ii) = -(dblquad(hxy_fun,min(X(1,:)),max(X(1,:)), min(Y(:,1)),
max(Y(:,1))));

    dy = ksdensity(y,Y(:,1));
    dy(dy<=0) = 10^(-99);
    dy_fun = @(ym)interp1((Y(:,1)),dy,ym).*(log2(interp1((Y(:,1)),dy,ym)));
    hy(ii) = -(quad(dy_fun,min(Y(:,1)),max(Y(:,1))));

    dx = ksdensity(x,X(1,:));
    dx(dx<=0) = 10^(-99);
    dx_fun = @(xm)interp1((X(1,:)),dx,xm).*(log2(interp1((X(1,:)),dx,xm)));
    hx(ii) = -(quad(dx_fun,min(X(1,:)),max(X(1,:))));

%      dy = ksdensity(y,Y(:,1));
%      dy(dy<=0) = 10^(-99);
%      dx = ksdensity(x,X(:,1));
%      dx(dx<=0) = 10^(-99);
%      [bandwidth,F,X,Y] = kde2d(RR,64);
%      F(F<=0) = 10^(-99);
%      x = RR(:,1);
%      y = RR(:,2);
%      MI_fun =
@(xm,ym)(interp2(X,Y,F,xm,ym)).*(log2(interp2(X,Y,F,xm,ym)./(dx.*dy)));
%      MIkernal(ii) = (dblquad(MI_fun,min(X(1,:)),max(X(1,:)), min(Y(:,1)),
max(Y(:,1))));
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %cluster MI
    [Res] = Find_MI_relationsoriginal(RR');
    MIcluster(ii) = Res.I(2,1);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Grassberger MI
   %[MIg] = reGrassberger(RR);
   %MIgrass(ii) = MIg;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
A = [0.899:0.005:0.999]; %roe1;
%A = 30.*(1./A);
MIkernal2 = hxy - hy - hx;
MIroe = -0.5.*log2(1-(roe1.^2));

figure(2)
plot(A,MIroe,'g')
hold on
plot(A,MIdiscrete,'r')
%plot(A,MIdiscrete2,'r')
plot(A,(-MIkernal2))
plot(A,roe1,'c')
%plot(A,MIgrass)
%title('MI vs Rho')
xlabel('Correlation Coefficient of data')
```

```
ylabel('Value of MI')
legend('MI based on Rho','MI based on Entropy','Continuous MI using
estimator','Rho','Location','NorthWest')
```

```matlab
function [y,varargout] = mvg(mu,Sigma,N)
%   MVG    Multivariate Gaussian random number generator.
%
%   y = mvg(mu,Sigma,N), where mu is mx1 and Sigma is mxm and SPD, produces
%   an mxN matrix y whose columns are samples from the multivariate
%   Gaussian distribution parameterized by mean mu and covariance Sigma.
%
%   [y,R] = mvg(mu,Sigma,N) also returns the Cholesky factor of the
%   covariance matrix Sigma such that Sigma = R'*R.
%
%   See also RAND, RANDN, SPRANDN, SPRANDN, RANDPERM.

%   Chad Lieberman, MIT 2008.
%   Questions/Comments:  celieber@mit.edu
%   $Revision: 1.0.0 $  $Date: 2008/09/01 $
%   $Revision: 1.0.1 $  $Date: 2008/09/03 $

%   References:
%   [1] I.T. Hernadvolgyi (1998) "Generating random vectors from the
%       multivariate normal distribution."
%   Available on-line at http://www.csi.uottawa.ca/~istvan/work.html.
%
%   Acknowledgements:
%       I would like to acknowledge John D'Errico for his helpful comments
%       and suggestions.

if nargin<3
  error('MVG must be called with three arguments.');
elseif nargin>4
  error('MVG called with too many arguments.');
end
if length(mu)~=size(Sigma,1)
  error('Length(mu) must equal size(Sigma,1).');
end
if size(Sigma,1)~=size(Sigma,2)
  error('Sigma must be square.');
end
if norm(Sigma-Sigma')>1e-15
  error('Sigma must be symmetric.');
end
try
  R = chol(Sigma);
catch
  error('Sigma must be positive definite.');
end
if (N<1 || mod(N,1)~=0)
  error('A positive integer number of samples must be requested.');
end
m = length(mu);
y = R'*randn(m,N) + repmat(mu,1,N);
if nargout>1
  varargout{1} = R;
end
```

```matlab
%function [RR,incidence] = EqualSizedCirclePackS2

%clear

RR =[];
A = [];

Nt = 1000; % number of circles that will be attempted to insert
L = 30;   % size of the box
d = 1;   % diameter of the circle
if isempty(RR)
    RR = [rand(1,2)*(L-d) d] + [1/2 1/2 0]*d;
%     fnplt(fncmb(rsmak('circle',RR(1,3)/2,[RR(1,1) RR(1,2)])))
%     hold on
else
%     for ii = 1:size(RR,1)
%         fnplt(fncmb(rsmak('circle',RR(ii,3)/2,[RR(ii,1) RR(ii,2)])))
%         hold on
%     end
end
%drawnow
kk =0;
while(size(RR,1)<=Nt & kk < Nt*100)
    kk =kk+1;
    r = [rand(1,2)*(L-d) d] + [1/2 1/2 0]*d ;
    if sum( ( (RR(:,1) - r(1)).^2 + (RR(:,2) - r(2)).^2 ) < (d +
RR(:,3)).^2/4 ) == 0
       RR(size(RR,1)+1,:) = [r];
%         fnplt(fncmb(rsmak('circle',d/2,[r(1) r(2)])))
%         drawnow
         if (size(RR,1)/1000-floor(size(RR,1)/1000))==0
               fprintf('Pack S2 %d  spheres inserted \n',size(RR,1))
               % save_on_file(RR,'RRS2.dat','w');
         end
    end
end
size(RR)
fprintf('Packing of equal disk done with %d disks, packing fraction
%f\n',size(RR,1),size(RR,1)*pi*d^2/4/(L-d).^2)
```

```matlab
function drawVoronoi(RR)
%
% draws a set of N circles
% R must be a 3xN vector containing X Y r
%
circles(RR(:,1),RR(:,2),RR(:,3)/2);
hold on
TRI = delaunay(RR(:,1),RR(:,2));
[vx, vy] = voronoi(RR(:,1),RR(:,2),TRI);
plot(RR(:,1),RR(:,2),'bx');
plot(vx,vy,'k-','LineWidth',2);
triplot(TRI,RR(:,1),RR(:,2),'-r')
axis([min(RR(:,1))-max(RR(:,3)),max(RR(:,1))+max(RR(:,3)),min(RR(:,2))-
max(RR(:,3)),max(RR(:,2))+max(RR(:,3))])


function circle(x,y,r)

t = 0:pi/200:2*pi;
for ii = 1:length(x)
    patch(x(ii)+r(ii)*sin(t),y(ii)+r(ii)*cos(t),'c')
end
box
axis equal
hold off
```

```
clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load('58884 30 01 (1)')
padding = 15;
boxsize = [7 8 9 10 12 15 20 25 50 70 100 150 200];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a = 0;
RR = RR(:,1:2)./repmat(RR(:,3),1,2);

% calculate Voronoi cells properties
[v, C] = voronoin([RR(:,1),RR(:,2)]); % v = verticies
TRI = delaunay(RR(:,1),RR(:,2));
for ii = 1:length(C) %ii = number of cells
    n_n(ii) = length(C{ii}); %Number of Voronoi neighbours
    [i,j]=find(TRI == ii); %coordinates in TRI
    incidence{ii}=setdiff(unique(TRI(i,:)),ii); %all neighbours of ii
    A(ii) = polyarea(v(C{ii},1),v(C{ii},2)); % computes Voronoi areas
end

%create boundry (ignore anythingthing with RR(:,4) = 0;
xm = min(RR(:,1))+padding;
xM = max(RR(:,1))-padding;
ym = min(RR(:,2))+padding;
yM = max(RR(:,2))-padding;
intervals = find(RR(:,1) > xm & RR(:,1) < xM & RR(:,2) > ym & RR(:,2) < yM);
RR(:,3) = 0;
RR(intervals,3)=1;  %if x or y coord is less than the padding away from box
edge set RR(;,4) = 0

%create stats for whole sample
cor = [];

for ii = 1:1:length(C);
    k = incidence{ii};
    k = k(RR(k,3)==1);
    c = A(k);
    d = repmat((A(ii)),1,length(c));
    cor = [cor ; d'  c']; %slow
end

fprintf('number of disks included in large box \n')
size((find(RR(:,3)==1)),1)

%discrete KLD

%KLD stats
a = min(min(cor));
b = max(max(cor));
d = ((b-a)/10);
for ii = 1:1:10
    c(ii) = a-(d/2) + ii*d;
end
```

```
C = [c; c];
D = cell(2,1);
D{1} = C(1,:);
D{2} = C(2,:);

pxh = hist3(cor,D);
px = pxh./sum(sum(pxh));
p1 = corrcoef(cor);

% %Continuous KLD
% [bandwidth,F1,X1,Y1] = kde2d(cor,128);
% F1(F1<=0) = 10^(-99);
%
% Ent_fun =
@(xm,ym)((interp2(X1,Y1,F1,xm,ym)).*(log2(interp2(X1,Y1,F1,xm,ym))));
% EntX = -(dblquad(Ent_fun,min(X1(1,:)),max(X1(1,:)), min(Y1(:,1)),
max(Y1(:,1))));

%create box
b = 0;
for BSa = boxsize;
    BS = BSa + padding;
    b=b+1;
    ind = find(RR(:,1) < BS & RR(:,2) < BS & RR(:,1) > padding & RR(:,2) >
padding);
    cc = C(ind);    %replaces C

    %create stats

    cor = [];
    for ii = ind';
        k = incidence{ii};
        k = k(RR(k,3)==1);
    c = A(k);
    d = repmat((A(ii)),1,length(c));
    cor = [cor ; d'  c']; %slow
    end

    pyh = hist3(cor,D);
    py = pyh./sum(sum(pyh));
    p2(:,:,b) = corrcoef(cor);

    x = px./py;
    kl = px.*log2(x);
    KLDdis(b) = sum(sum(kl(isfinite(kl))));

    %cont case
    [bandwidth,F2,X2,Y2] = kde2d(cor,128);
    F2(F2<=0) = 10^(-99);

    %xx = rr(:,1);
    %yy = rr(:,2);
    CrossEnt_fun =
@(xm,ym)((interp2(X1,Y1,F1,xm,ym)).*(log2(interp2(X2,Y2,F2,xm,ym))));
```

```
    KLDkernal2(b) = -(dblquad(CrossEnt_fun,min(X2(1,:)),max(X2(1,:)),
min(Y2(:,1)), max(Y2(:,1))));

    fprintf('number of disks included in small box \n')
    size((find(RR(ind,3)==1)),1)
end

KLDkernal = KLDkernal2 - EntX;

figure(1)
hold on
plot(boxsize,KLDdis,'bl')
figure(2)
hold on
plot(boxsize,KLDkernal,'bl')
```

```matlab
clear all
close all

load('73076 30 009')
RR = RR(:,1:2)./RR(1,3);
padding = 15;

% calculate Voronoi cells properties
[v, C] = voronoin([RR(:,1),RR(:,2)]); % v = verticies
TRI = delaunay(RR(:,1),RR(:,2));
vormax = 1:1:length(C);
for ii = vormax
    n_n(ii) = length(C{ii}); %Number of Voronoi neighbours
    [i,j]=find(TRI == ii); %coordinates in TRI
    incidence{ii}=setdiff(unique(TRI(i,:)),ii); %all neighbours of ii
    A(ii) = polyarea(v(C{ii},1),v(C{ii},2)); % computes Voronoi areas
end

%create boundry (ignore anythingthing with RR(:,4) = 0;
xm = min(RR(:,1))+padding;
xM = max(RR(:,1))-padding;
ym = min(RR(:,2))+padding;
yM = max(RR(:,2))-padding;
intervals = find(RR(:,1) > xm & RR(:,1) < xM & RR(:,2) > ym & RR(:,2) < yM);
RR(:,3) = 0;
RR(intervals,3)=1;  %if x or y coord is less than the padding away from box
edge set RR(;,4) = 0

RR2 = RR(intervals,1:2);

cor = [];
for ii = 1:1:length(C);
    k = incidence{ii};
    k = k(RR(k,3)==1);
    c = A(k);
    d = repmat((A(ii)),1,length(c));
    %indexx = repmat(ii,1,length(c));
    cor = [cor ; d'  c']; %indexx']; %slow
end

a = min(min(cor));
b = max(max(cor));
f = ((b-a)/10);
e = zeros(10,1);
for ii = 1:1:10
    e(ii) = a-(f/2) + ii*f;
end
E = [e e];
F = cell(2,1);
F{1} = E(:,1);
F{2} = E(:,2);

pxh = hist3(cor,F);
px = pxh./sum(sum(pxh));
p1 = corrcoef(cor);
```

```
boxsize2 = 10;
sampsize = boxsize2 + padding; %only changes endpoint
maxoffset = ceil(max(max(RR))) - sampsize - padding;
offset = 0:1:maxoffset;
KLDd = zeros(maxoffset + 1,1);
KLDstore = zeros(maxoffset + 1,maxoffset + 1)
for xoffsets = offset;
    xoffsets
    for yoffsets = offset;
        %if in 10's use yoffset.*10 and offset./10
        ind = find((RR2(:,1) > padding + xoffsets) & (RR2(:,1) < sampsize +
xoffsets) & (RR2(:,2) > padding + yoffsets) & (RR2(:,2) < sampsize +
yoffsets));
        cor1 = [];
        for ii = ind';
            k = incidence{ii};
            k = k(RR(k,3)==1);
            c = A(k);
            d = repmat((A(ii)),1,length(c));
            cor1 = [cor1 ; d'  c'];
        end
        pyh = hist3(cor1,F);
        py = pyh./sum(sum(pyh));
        [KLD] = createFitDis(px, py);
        KLDd(yoffsets + 1) = KLD;
    end
    KLDstore(:,xoffsets + 1) = KLDd;
end
%do 1 for 10!
save('10')

imagesc(real(KLDstore)) %KLDstore(16:end,16:end)
colormap('jet') %gray %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
find(KLDstore == 0)
```

```matlab
% function autoreferentialOrder_KLD

clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
boxsizes  = [2 2.5 3 3.5 4]; %[1 3 5] %[1 1.5 2 2.5 3 4 5];
padding   = max(boxsizes)+3;
step      = 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%files = {'2D_Pack_0.9_5k.mat'};

% files = {'2D_Pack_0.4_5k.mat',...
%          '2D_Pack_0.5_5k.mat',...
%          '2D_Pack_0.6_5k.mat',...
%          '2D_Pack_0.7_5k.mat',...
%          '2D_Pack_0.8_5k.mat',...
%          '2D_Pack_0.9_5k.mat'};

files = {'2D_Pack_0.4_5k.mat',...
         '2D_Pack_0.5_5k.mat',...
         '2D_Pack_0.55_5k.mat',...
         '2D_Pack_0.6_5k.mat',...
         '2D_Pack_0.65_5k.mat',...
         '2D_Pack_0.7_5k.mat',...
         '2D_Pack_0.75_5k.mat',...
         '2D_Pack_0.8_5k.mat',...
         '2D_Pack_0.85_5k.mat',...
         '2D_Pack_0.9_5k.mat'};

for b = 1:length(boxsizes)
    boxsize = boxsizes(b);
    lgd{b} = ['box sz.  ' num2str(boxsize)];
    for f=1:length(files)
        load(files{f})
        RR = RR(:,1:2);
        % calculate Voronoi cells properties
        [v, C] = voronoin([RR(:,1),RR(:,2)]); % v = verticies
        TRI = delaunay(RR(:,1),RR(:,2));
        for ii = 1:length(C)
            n_n(ii) = length(C{ii}); %Number of Voronoi neighbours
            [i,j]=find(TRI == ii); %coordinates in TRI
            incidence{ii}=setdiff(unique(TRI(i,:)),ii); %all neighbours of ii
            A(ii) = polyarea(v(C{ii},1),v(C{ii},2)); % computes Voronoi areas
        end

        %create boundry (ignore anythingthing with RR(:,4) = 0;
        xm = min(RR(:,1))+padding;
        xM = max(RR(:,1))-padding;
        ym = min(RR(:,2))+padding;
        yM = max(RR(:,2))-padding;
        internals = find(RR(:,1) > xm & RR(:,1) < xM & RR(:,2) > ym & RR(:,2)
< yM);
        RR(:,3) = 0;
        RR(internals,3)=1;  %if x or y coord is less than the padding away
from box edge set RR(:,3) = 0
```

```
        max_xy = ceil(max(RR)) -padding-boxsize;
        min_xy = floor(min(RR))+padding;
        xx = min_xy(1):step:max_xy(1);
        yy = min_xy(1):step:max_xy(1);

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%% Kullback-Leibler divergences   %%%
        [a,bin_v]=hist(A(internals),10);
        q_v = a/sum(a);
        %
[a,bin_n]=hist(n_n(internals),[min(n_n(internals)):max(n_n(internals))]);
        q_n = a/sum(a);
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        KLD_v = zeros(length(xx),length(yy));
        iKLD_v= zeros(length(xx),length(yy));
        H_v    = zeros(length(xx),length(yy));
        %KLD_n = KLD_v;
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        ix = 0;
        for x = xx
            ix=ix+1;
            iy =0;
            for y = yy
                iy=iy+1;
                box = find((RR(:,1) >= x) & (RR(:,1) < (x+boxsize)) &
(RR(:,2) >= y) & (RR(:,2) < (y+boxsize)));
                % length(box)
                a      = hist(A(box),bin_v);
                p_v    = a/sum(a);
                if sum(p_v>0&q_v>0)>0
                    KLD_v(ix,iy) =
sum(p_v(p_v>0&q_v>0).*log2(p_v(p_v>0&q_v>0)./q_v(p_v>0&q_v>0)));
                end
    %            H_v(ix,iy)    = -sum(p_v(p_v>0).*log2(p_v(p_v>0)));
    %             a    = hist(n_n(box),bin_n);
    %             p_n    = a/sum(a);
    %             KLD_n(ix,iy) =
sum(p_n(p_n>0&q_n>0).*log2(p_n(p_n>0&q_n>0)./q_n(p_n>0&q_n>0)));
            end
        end
        pf(f,b) = pi/4/mean(A(internals));
        meK_v(f,b) = mean(KLD_v(KLD_v>0))
        moK_v(f,b) = mode(KLD_v(KLD_v>0))
        a = hist(KLD_v(KLD_v>0),40);
        maK_v(f,b) = max(a)/size(KLD_v,1)/size(KLD_v,2)
    %     sR_v(f) = sum(KLD_v(KLD_v>0 &
H_v>0)./H_v(H_v>0&H_v>0))/sum(KLD_v(:)>0&H_v(:)>0)
    %     sK_n(f) = sum(KLD_n(KLD_n>0))/sum(KLD_n(:)>0)
%          figure
%          imagesc(iKLD_v)
%          colorbar
%          axis image
%          set(gca,'visible','off')
%          print(['KLD_image',num2str(pf(f)),'.eps'],'-depsc2')
```

```
%         figure
%         imagesc(KLD_v)
%         hold on
%         colorbar
%         axis image
%         drawCircles([RR(internals,2)-min_xy(2)-boxsize/4,RR(internals,1)-
min_xy(2)-boxsize/4],0.5,10,2,[1 1 1]*0)
%         axis('equal')
%         set(gca,'visible','off')
%         print(['packs',num2str(pf(f)),'.png'],'-dpng')
    end
end

figure
plot(pf,meK_v,'o-')
% plot(pf,meK_v-moK_v,'o-')
% hold on
% axis([min(pf) 1 0 max(sK_v)])
% hold on
% plot(pf,sK_n,'s-r')
% axis([min(pf) 1 0 max([sK_v,sK_n])])
set(gca,'linewidth',1,'fontsize',24)
legend(lgd,'fontsize',18)

figure
plot(pf,1-maK_v,'o-')
set(gca,'linewidth',1,'fontsize',24)
legend(lgd,'fontsize',18)


% figure
% plot(pf,sR_v,'o-b')
% axis([min(pf) 1 0 max(sR_v)])

return

figure
hist(KLD_v(KLD_v>0),40)

figure
imagesc(KLD_v)
colorbar
axis image
%axis('equal')
set(gca,'visible','off')
hold on
patchCircles([RR(internals,2)-min_xy(2)-boxsize/4,RR(internals,1)-min_xy(2)-
boxsize/4],0.5,20,[1 1 1],0.5,[1 1 1]*.3,0.7,1)


figure
patchCircles(RR(internals,1:2),0.5,20,[0 0 1],1,[0 0 1],1,1)
hold on
patchCircles(RR(setdiff([1:length(RR)],internals),1:2),0.5,20,[0 0 1],0.5,[0
0 1],0.7,1)
```

```
plot([min_xy(1) min_xy(1)],[min_xy(2) max_xy(2)],'r')
plot([max_xy(1)+boxsize max_xy(1)+boxsize],[min_xy(2) max_xy(2)+boxsize],'r')
plot([min_xy(1) max_xy(1)+boxsize],[min_xy(2) min_xy(2)],'r')
plot([min_xy(1) max_xy(1)+boxsize],[max_xy(2)+boxsize max_xy(2)+boxsize],'r')


% figure
% imagesc(KLD_n)
% colorbar
% axis image
% %axis('equal')
% set(gca,'visible','off')
% hold on
% patchCircles([RR(internals,2)-min_xy(2)-boxsize/4,RR(internals,1)-
min_xy(2)-boxsize/4],0.5,20,[1 1 1],0.5,[1 1 1]*.3,0.7,1)
```

```matlab
function [KLD] = createFitDis(px, py)

x1 = (1:length(px))';
y = x1;

[xData, yData, zData] = prepareSurfaceData( x1, y, px );
[xData2, yData2, zData2] = prepareSurfaceData( x1, y, py );

% Set up fittype and options.
ft = 'linearinterp'; %'cubicinterp'; %'linearinterp'; %'cubicinterp';
opts = fitoptions( ft );
opts.Normalize = 'on';

% Fit model to data.
[fitresult, gof] = fit( [xData, yData], zData, ft, opts );

[fitresult2, gof2] = fit( [xData2, yData2], zData2, ft, opts );


fun1 = @(p,q)(fitresult(p,q)).*(log2((fitresult(p,q) + 10.^(-
99))./(fitresult2(p,q)+ 10.^(-99))));
%fun1 = @(p,q)(((fitresult(p,q)).*(log2(fitresult(p,q) + 10.^(-99))))) -
(((fitresult(p,q)).*(log2(fitresult2(p,q) + 10.^(-99)))));

% for eval = 1:1:max(xData)
%     for pq = 1:1:max(yData)
%         px(eval,pq) = fitresult(eval,pq);
%         py(eval,pq) = fitresult2(eval,pq);
%     end
% end

for eval = 1:1:max(xData)
    for pq = 1:1:max(yData)
        KLD(eval,pq) = fun1(eval,pq);
    end
end
KLD = sum(sum(KLD));
```

```
%function [RR,incidence] = EqualSizedCirclePackS2

clear all
close all
%return
bb=0;
%%%%%%% parameters to change %%%%%%%%%%%%%%%%%%%%%%%
maxDistance = 8; % distance in unit of boxes
BS = 0.25:0.1:10; % dimension of the box in units of d
BSS = BS;%.*(1./0.707106781);
for boxSize = BS
    bb=bb+1;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    clear pnm mi mi1 covarDist
    load('082.mat');
    d = RR(1,3);    % diameter of the circle
    RR(:,1:2) = RR(:,1:2)/d;  %scale all sizes in unit of d
    N = length(RR);
    %
    % two choices for the padding
    padding = 30; % layer of padding to take off the boundaryes (in units of
d)

    %boundares
    xm = min(RR(:,1))+padding;
    xM = max(RR(:,1))-padding;
    ym = min(RR(:,2))+padding;
    yM = max(RR(:,2))-padding;

    Dx = (ceil((xM-xm)/boxSize)-2*maxDistance)*boxSize;
    Dy = (ceil((yM-ym)/boxSize)-2*maxDistance)*boxSize;
    internalRegion = [xm,xm+Dx,ym,ym+Dy]+maxDistance*boxSize;

    figure(1)
    clf
    plot(RR(:,1),RR(:,2),'o')
    hold on
    plot([xm xm],[ym yM],'-k')
    plot([xM xM],[ym yM],'-k')
    plot([xm xM],[ym ym],'-k')
    plot([xm xM],[yM yM],'-k')

    if ((padding-maxDistance) < 2) | (xm >xM) | (ym >yM)
        fprintf('change parameters!\n')
        return
    end
    fprintf('Statistics over a central region of %d cells containaing %d
disks\n',floor((xM-xm)/boxSize)*floor((xM-
xm)/boxSize),length(find(RR(:,1)>=xm & RR(:,1)<=xM & RR(:,2)>=ym &
RR(:,2)<=yM)) )

    iii=0;
    RRR = [];
    freqN = sparse(length(RR),1);
    X = (internalRegion(1)-
maxDistance*boxSize):boxSize:(internalRegion(2)+maxDistance*boxSize);
```

```
    Y = (internalRegion(3)-
maxDistance*boxSize):boxSize:(internalRegion(4)+maxDistance*boxSize);
    % creates the cell-list giving to each disk the cell address
    for i = 1:(length(X)-1)
        RR( RR(:,1)>=X(i), 4 ) = i;
        plot([X(i),X(i)],[min(Y),max(Y)],'-r')
    end
    RR( RR(:,1)>=X(end), 4 ) = 0;
    for i = 1:(length(X)-1)
        RR( RR(:,2)>=Y(i), 5 ) = i;
        plot([min(X),max(X)],[Y(i),Y(i)],'-r')
    end
    RR( RR(:,2)>=Y(end), 5 ) = 0;
    axis equal
    % ij = find(RR(:,4)==1 & RR(:,5)==1)
    % plot(RR(ij,1),RR(ij,2),'+k','MarkerSize',100)

    plot([min(X),min(X)],[min(Y),max(Y)],'-y','LineWidth',3)
    plot([max(X),max(X)],[min(Y),max(Y)],'-y','LineWidth',3)
    plot([min(X),max(X)],[min(Y),min(Y)],'-y','LineWidth',3)
    plot([min(X),max(X)],[max(Y),max(Y)],'-y','LineWidth',3)

    for i = 1:(length(X)-1)
        x = find(RR(:,4)==i);
        for j=1:(length(Y)-1)
            cij = find(RR(x,5)==j); % cell i,j
            n(i,j) = length(cij); %occupation number of cell i,j
            %plot(X(i)+boxSize/2,Y(j)+boxSize/2,'+g')
        end
    end
    fprintf('The internal part contains %d cells\n',(length(X)-maxDistance-
1)*(length(Y)-maxDistance-1))

    for dis =1:maxDistance
        nn = n(1:(end-dis),1:end);
        mm = n((dis+1):end,1:end);
        cx=cov(nn(:),mm(:));
        nn = n(1:end,1:(end-dis));
        mm = n(1:end,(dis+1):end);
        cy=cov(nn(:),mm(:));
        covarDist(:,:,dis)=(cx +cy )/2;
    end;

    freqN = zeros(max(max(n))+1,1);
    coOccurences = zeros(max(max(n))+1,max(max(n))+1,maxDistance);
    for i = 1:(length(X)-maxDistance-1)
        for j=1:(length(Y)-maxDistance-1)
            nij = n(i,j);
            freqN(nij+1) =  freqN(nij+1) +1;
            for dis =1:maxDistance
                nij1= n(i,j+dis);
                ni1j= n(i+dis,j);

coOccurences(nij+1,nij1+1,dis)=coOccurences(nij+1,nij1+1,dis)+1;

coOccurences(nij+1,ni1j+1,dis)=coOccurences(nij+1,ni1j+1,dis)+1;
```

```
            end
            %plot(X(i)+boxSize/2,Y(j)+boxSize/2,'og')
        end
    end
    pn = freqN/sum(freqN);
    distance = (1:maxDistance)*boxSize;

    % mutual information
    for ii =1:length(distance)
        pnm(:,:,ii) =  coOccurences(:,:,ii)/sum(sum(coOccurences(:,:,ii)));
        x = pnm(:,:,ii).*log(pnm(:,:,ii)./(pn*pn'));
        mi(ii) = sum(x(~isnan(x))); %% this gives often numerical
problems!!!!!
        x =
pnm(:,:,ii).*log(pnm(:,:,ii)./(sum(pnm(:,:,ii),2)*sum(pnm(:,:,ii),1)));
        mi1(ii) = sum(x(~isnan(x))); %% this works well!!!!
        % entropy
        x = pnm(:,:,ii).*log(sum(pnm(:,:,ii),2)*sum(pnm(:,:,ii),1));
        s(ii) = -sum(x(~isnan(x)));
        % entropy ratio
        rmi(ii) = mi1(ii)/s(ii);
    end
    probn = [find(pn>0),pn(pn>0)];
    probn = [[probn(1,1)-2;probn(:,1)-1;probn(end,1)] , [0;probn(:,2);0]];
    probn = probn(probn(:,1)>=0,:);

    figure(2)
    bar(find(freqN>0)-1,full(freqN(freqN>0)))

    figure(3)
    plot(probn(:,1),probn(:,2),'ok-')

    figure(4)
    plot(distance,mi,'ob-')

    figure(5)
    plot(distance,mi1,'sr-')

    figure(6)
    plot(distance,-reshape(covarDist(1,2,:),1,size(covarDist,3)),'sb-')

    % entropy ratio
    fnrmi(bb)     = rmi(1);
    allrmi(bb,:) = rmi;
    figure(7)
    clf
    plot(BS(1:bb),allrmi,'+r')
    hold on
    plot(BS(1:bb),fnrmi,'-ob')
    % average Mutual info
    ami(bb)=mean(mi1(2:end))

    %fist neighbours Mutual info
    fnmi(bb)     = mi1(1);
    allmi(bb,:) = mi1;
```

```
    figure(8)
    clf
    %plot(BSS(1:bb),allmi,'-c')
    hold on
    plot(BSS(1:bb),fnmi,'-ob')
    xlabel('Distance in units of disk diameter')
    ylabel('Value of the Mutual Information in bits')
end
cc=polyfit(log(BS(BS>3)),log(ami(BS>3)),1)
plot(BS,exp(cc(1)*log(BS)+cc(2)),'-m')
```

```
clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PROGRR = ('/spheres'); %sphere creation exe %3D @ moment
PROGMT = ('/karambola'); %MT creation exe
PROGV = ('/pointpattern2voronoi3d'); % voronoi creation exe

path = ('/home/paul/Desktop/3D_Packs_and_Codes');
%('/home/paul/Work/MyFiles/C++Code'); %path for files to be used in
Rpath = ('/ComplexSpheres'); %additional folders for the sphere and MT
programs
Mpath = ('/karambola-1.5');
Vpath = ('/karambola-1.5/demo/pointpattern2voronoi3d');

%parameters for sphere generation
growthrate = 0.16; %default is 0.001
numberofspheres = 10000; %default is 100
eventspercycle = 20; %default is 20
maxcollisions = 5000000; %default is 100000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% IN = ('input.txt');
% input = [' ./',IN];
% RRpath = [path Rpath PROGRR input]; %forms linux command for spheres
%
% for growthrate = 0.001; %loops for varible to create multiple packings
%    %creates input file
%    fid = fopen(IN,'w');
%    fprintf(fid,['int eventspercycle = ',num2str(eventspercycle),'; \nint N
= ',num2str(numberofspheres),'; \ndouble initialpf = 0.01; \ndouble maxpf =
0.99;v\ndouble temp = 0;\ndouble growthrate = ',num2str(growthrate),';
\ndouble maxpressure = 100000; \ndouble maxcollisionrate =
',num2str(maxcollisions),'; \ndouble bidispersityratio = 1;\ndouble
bidispersityfraction = 1; \ndouble massratio = 1.; \nint hardwallBC = 0;
\nchar* readfile = new \nchar* writefile = write.dat \nchar* datafile =
stats.dat \n']);
%    fclose(fid);
%    %creates output in 2 forms (1 for MT program)
%    system(RRpath);
% end

%creates voronio polygons for MT to read using demo karambola program
system('find write*.xyz > filenamesxyz');
system('find write*.dat > filenamesdat');
namesdat = importdata('filenamesdat');
namesxyz = importdata('filenamesxyz');

for ii = 1:1:length(namesxyz)
    name = namesxyz{ii};

    OUT = [' -o ','hs-
',num2str(numberofspheres),'_',name(6),'.',name(7:12),'.xyz2']; %don't define
output to keep same name with .poly
    INOUT = [' -i ',name, OUT]; %options and input
```

```
    VVpath = [path Vpath PROGV INOUT];

    system(VVpath);
end

system('rename s/\.xyz2$/.poly/ *.xyz2 '); %for use when defining output
system('mv *.poly ./polys');

%reads in file names for MT exe
% system('find *.poly > filenames'); %finds .poly files
% names = importdata('filenames'); %and reads the in
%
% for ii = 1:1:length(names)
%     name = names{ii};
%
%     OUT = ['./polys/hs-
',num2str(numberofspheres),'_',num2str(name(9:16)),'_mink_val']; %temp for
cheat minkval and dir
%     INOUT = [' -i ',name,' --labels -o ',OUT]; %options and input
%     MMpath = [path Mpath PROGMT INOUT];
%
%     system(MMpath);
% end
%

system('./run_karambola.sh');
system('find ./polys/hs*val -prune > folders')
MTfolders = importdata('folders');

MTnames = cell(14);
MTnames{1} = 'w000_w100_w200_w300';
MTnames{2} = 'w010_w110_w210_w310';
MTnames{3} = 'w020';
MTnames{4} = 'w020_eigsys';
MTnames{5} = 'w102';
MTnames{6} = 'w102_eigsys';
MTnames{7} = 'w120';
MTnames{8} = 'w120_eigsys';
MTnames{9} = 'w202';
MTnames{10} = 'w202_eigsys';
MTnames{11} = 'w220';
MTnames{12} = 'w220_eigsys';
MTnames{13} = 'w320';
MTnames{14} = 'w320_eigsys';

w000 = zeros(numberofspheres,1); %volume per cell
w100 = w000; %surface area per cell (all facets)
w200 = w000; %mean curvature
w300 = w000; %guass mean curvature
w010 = zeros(numberofspheres,3); % moment of inertia
w020 = zeros(numberofspheres,9); % volume intergral
w120e = zeros(numberofspheres,3);
w202e = zeros(numberofspheres,3);
w220e = zeros(numberofspheres,3);
w320e = zeros(numberofspheres,3);
```

```
TMT = cell(length(MTfolders),1);
for ii = 1:1:length(MTfolders)
    ii
    folder = MTfolders{ii};
    DELIMITER = ' ';
    HEADERLINES = 100000;
    wx00 = importdata([folder, '/', MTnames{1}], DELIMITER, HEADERLINES);
    wx10 = importdata([folder, '/', MTnames{2}], DELIMITER, HEADERLINES);
    w020x = importdata([folder, '/', MTnames{3}], DELIMITER, HEADERLINES);
    xw120e = importdata([folder, '/', MTnames{8}], DELIMITER, HEADERLINES);
    xw202e = importdata([folder, '/', MTnames{10}], DELIMITER, HEADERLINES);
    xw220e = importdata([folder, '/', MTnames{12}], DELIMITER, HEADERLINES);
    w320x = importdata([folder, '/', MTnames{13}], DELIMITER, HEADERLINES);
    xw320e = importdata([folder, '/', MTnames{14}], DELIMITER, HEADERLINES);

    for jj = 1:numberofspheres
        w000(jj) = str2num(wx00{jj+1}(22:41)); %infact all character lie
between 23 and 40
        w100(jj) = str2num(wx00{jj+1+numberofspheres}(22:41));
        w200(jj) = str2num(wx00{jj+1+(2*numberofspheres)}(22:41));
        w300(jj) = str2num(wx00{jj+1+(3*numberofspheres)}(22:41));
        w010(jj,1) = str2num(wx10{jj+1}(22:41));
        w010(jj,2) = str2num(wx10{jj+1}(42:61));
        w010(jj,3) = str2num(wx10{jj+1}(62:81));
        w120e(jj,1) = str2num(xw120e{jj+1}(22:41));
        w120e(jj,2) = str2num(xw120e{jj+1}(102:122));
        w120e(jj,3) = str2num(xw120e{jj+1}(183:202));
        w202e(jj,1) = str2num(xw202e{jj+1}(22:41));
        w202e(jj,2) = str2num(xw202e{jj+1}(102:122));
        w202e(jj,3) = str2num(xw202e{jj+1}(183:202));
        w220e(jj,1) = str2num(xw220e{jj+1}(22:41));
        w220e(jj,2) = str2num(xw220e{jj+1}(102:122));
        w220e(jj,3) = str2num(xw220e{jj+1}(183:202));
        w320e(jj,1) = str2num(xw320e{jj+1}(22:41));
        w320e(jj,2) = str2num(xw320e{jj+1}(102:122));
        w320e(jj,3) = str2num(xw320e{jj+1}(183:202));
    end
pf = folder(19:25);
%read in postions(RR)
DELIMITER = ' ';
HEADERLINES = 2;
newData10 = importdata(namesxyz{ii}, DELIMITER, HEADERLINES);
RR = newData10.data;
RRd = importdata(namesdat{ii});
RR = RR./(RRd(6)); %normalizes sphere diameters to be 1

save([folder(9:24),'.mat'],'w120e','w202e','w220e','w000','w100','w200','w300
','w010','pf','RR','w320e');
TMT{ii} = load([folder(9:24),'.mat']);
end

%calculate Self-Referencial Order
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
boxoccupancy  = [1 2 4 6 10 25];% 100 500 2000 5000];%[0.05 0.1 0.2 0.3];
%LARGE BOXSIZE IS ONE
```

```
ngrid = [30000 30000 30000 25000 23000 20000]; %30000 (1000 equivilant in
3D); % number of positions in which the cube is placed
bin_p = 20;
bin = 500;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

grid2 = ngrid.^(1/3); %grid in 1D
resultsMV = cell(length(TMT),length(boxoccupancy));
resultsPF = cell(length(TMT),length(boxoccupancy));

clear pf
for f = 1:length(TMT)

    RR = TMT{f}.RR;
    Z = TMT{f}.w320e; %curve weighted surface intergral
    Z2 = abs(Z(:,3)-Z(:,1));
    Z3 = abs(Z(:,3)+Z(:,1));
    Z4 = Z2./Z3;
    Z1 = abs(Z(:,1)./Z(:,3));
    Y = TMT{f}.w010;
    Y = sqrt((Y(:,1).^2)+(Y(:,2).^2)+(Y(:,3).^2));
    AAA = TMT{f}.w000; %volume
    X = TMT{f}.w120e; %curve weighted surface intergral
    X = abs(X(:,1)./X(:,3));
    W = TMT{f}.w202e; %surface intergral
    W2 = abs(W(:,3)-W(:,1));
    W3 = abs(W(:,3)+W(:,1));
    W4 = W2./W3;
    W1 = abs(W(:,1)./W(:,3));
    V = TMT{f}.w220e; %curve weighted surface intergral
    V = abs(V(:,1)./V(:,3));

    T =
{Z(:,1),Z2,Z3,Z4,Y,AAA,X,W(:,1),W1,W2,W3,W4,V,Z(:,2),Z(:,3),W(:,2),W(:,3)};

    max_xyz = ceil(max(RR));
    min_xyz = floor(min(RR));

    for b = 1:length(boxoccupancy) % loops over different box sizes
        lgd{b} = ['box average occupancy.  ' num2str(boxoccupancy(b))];
        %%%  packing fraction
        grid = grid2(b);
        pf(f,b) = str2double(TMT{f}.pf);
        [b f]
        %load
        boxsize = ((boxoccupancy(b)*pi)/(6*pf(f,b)))^(1/3);
        lgd2{f} = ['pf. ' num2str(pf(f,b))];
        %%%% sqrt to cube root
        stepX = (max_xyz(1)-min_xyz(1))/grid;
        stepY = (max_xyz(2)-min_xyz(2))/grid;
        stepZ = (max_xyz(2)-min_xyz(2))/grid;
        xx = min_xyz(1):stepX:max_xyz(1)-boxsize;
        yy = min_xyz(2):stepY:max_xyz(2)-boxsize;
        zz = min_xyz(3):stepZ:max_xyz(3)-boxsize;
```

```
        % right binning for v

        edges = cell(1,length(T));
        m12=quantile(Z(:,1),[0,1-1/bin]);
        edges{1}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(Z2,[0,1-1/bin]);
        edges{2}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(Z3,[0,1-1/bin]);
        edges{3}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(Z4,[0,1-1/bin]);
        edges{4}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(Y,[0,1-1/bin]);
        edges{5}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(AAA,[0,1-1/bin]);
        edges{6}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(X,[0,1-1/bin]);
        edges{7}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(W(:,1),[0,1-1/bin]);
        edges{8}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(W1,[0,1-1/bin]);
        edges{9}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(W2,[0,1-1/bin]);
        edges{10}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(W3,[0,1-1/bin]);
        edges{11}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(W4,[0,1-1/bin]);
        edges{12}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(V,[0,1-1/bin]);
        edges{13}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(Z(:,2),[0,1-1/bin]);
        edges{14}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(Z(:,3),[0,1-1/bin]);
        edges{15}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(W(:,2),[0,1-1/bin]);
        edges{16}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m12=quantile(W(:,3),[0,1-1/bin]);
        edges{17}=[m12(1):(m12(2)-m12(1))/(bin-2):m12(2),Inf];
        m_v  = nan(length(xx),length(yy),length(zz),length(T));
        pfLoc = nan(length(xx),length(yy),length(zz));
        FXY = cell(1,length(T));
        for i = 1:length(T)
            FXY{1,i} = zeros(bin_p,bin_p);
        end

        ix = 0;
        for x = xx
            ix=ix+1;
            iy =0;
            for y = yy
                iy=iy+1;
                iz = 0;
                for z = zz %3D parameter
                    iz = iz+1;
                    box = find((RR(:,1) >= x) & (RR(:,1) < (x+boxsize)) &
(RR(:,2) >= y) & (RR(:,2) < (y+boxsize)) & (RR(:,3) >= z) & (RR(:,3) <
(z+boxsize)));
```

```
                    if ~isempty(box)
                        for ii = 1:length(T)
                            A = T{ii}; %A is everything is whole sample
                            AA = A; %AA is everything outside of box
                            AA(box) = [];
                            %matching every x to every y (why are we matching
                            %them like this?
                            fx = histc(A,edges{ii});
                            ppx = fx/sum(fx(:)); %probability of all
                            fy  = histc(A(box),edges{ii});
                            if size(fy,1) == 1
                                fy = fy';
                            else
                            end
                            ppy = fy/sum(fy(:));
                            fxy =  hist3([ppx,ppy],[bin_p,bin_p]);
                            FXY{1,ii} = FXY{1,ii} + fxy;
                            n = sum(fxy(:));
                            pxy = fxy/n;
                            px  = sum(fxy,1)/n;
                            py  = sum(fxy,2)/n;
                            Sxy = -sum(pxy(pxy>0).*log2(pxy(pxy>0)));
                            Sx  = -sum(px(px>0).*log2(px(px>0)));
                            Sy  = -sum(py(py>0).*log2(py(py>0)));
                            m_v(ix,iy,iz,ii)= (Sx+Sy-Sxy)/Sx;
                            pfLoc(ix,iy,iz,ii) = pi/4/mean(A(box));
                        end
                    end
                end
            end
        end
        resultsMV{f,b} = m_v; %local
        resultsPF{f,b} = pfLoc;
        for iii = 1:length(T)
            FXYs = FXY{1,iii};
            n = sum(FXYs(:));
            pxy = FXYs/n;
            px  = sum(FXYs,2)/n;
            py  = sum(FXYs,1)/n;
            Sxy = -sum(pxy(pxy>0).*log2(pxy(pxy>0)));
            Sx  = -sum(px(px>0).*log2(px(px>0)));
            Sy  = -sum(py(py>0).*log2(py(py>0)));
            I(iii,f,b) =  Sx+Sy-Sxy;
            m(iii,f,b) = (Sx+Sy-Sxy)/Sx; %global
            %save
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

diversity_v1 = zeros(length(TMT),length(boxoccupancy),length(T));
bins = 50;
for b=1:length(boxoccupancy)
    for f=1:length(TMT)
```

```
        m_v = (results{f,b}); %abs
        for Mts = 1:length(T)
            m_v2 = m_v(:,:,:,Mts);
            a = hist(m_v2(:),bins);
            pxx = a./sum(a);
            cc = log2(pxx);
            diversity_v1(f,b,Mts) = -
sum(sum(pxx(isfinite(cc)).*cc(isfinite(cc))));
        end
    end
end
div3 = div3./(log2(bins)); %normalisation

for gg=1:17;
figure
plot(pf,m(gg,:,1),pf,m(gg,:,2),pf,m(gg,:,3),pf,m(gg,:,4),pf,m(gg,:,5),pf,m(gg
,:,6),'d-') %plots 1st tensor
set(gca,'linewidth',1,'fontsize',24)
legend(lgd,'fontsize',18,'location','northwest')
ylabel('Self-Referential Order')
xlabel('Packing Fraction')
end

% figure
%
plot(pf,m(2,:,1),pf,m(3,:,1),pf,m(4,:,1),pf,m(5,:,1),pf,m(6,:,1),pf,m(1,:,1),
'd-') %plots 1st tensor
% set(gca,'linewidth',1,'fontsize',24)
%
% figure
% plot(pf,diversity_v1(:,:,1),'s-') %plots 1st tensor diversity(ent)
% set(gca,'linewidth',1,'fontsize',24)
% legend(lgd,'fontsize',18)
%
% hold on
%
% plot(pf,diversity_v1(:,:,2),'gs-') %plots 1st tensor diversity(ent)
% set(gca,'linewidth',1,'fontsize',24)
% legend(lgd,'fontsize',18)
%
% plot(pf,diversity_v1(:,:,1),'ks-') %plots 1st tensor diversity(ent)
% set(gca,'linewidth',1,'fontsize',24)
% legend(lgd,'fontsize',18)
%
% plot(pf,diversity_v1(:,:,1),'rs-') %plots 1st tensor diversity(ent)
% set(gca,'linewidth',1,'fontsize',24)
% legend(lgd,'fontsize',18)
```

```
//==========================================================
//==========================================================
//==========================================================
//
//  Molecular dynamics simulation of hardspheres
//
//==========================================================
//==========================================================
//==========================================================

#include <iomanip>
#include <locale>
#include <sstream>
#include <string>

#include <iostream>
#include <math.h>
#include <fstream>
#include <vector>
#include <time.h>
#include <string.h>

#include <stdlib.h>

#include "box.h"
#include "sphere.h"
#include "event.h"
#include "heap.h"
#include "read_input.h"


int main(int argc, char **argv)
{
  read_input input;
  int error = input.read(argc, argv);
  if (error) return error;

  double d, r;   // initial diameter and radius of spheres

  if(strcasecmp(input.readfile, "new")==0)
    input.readfile[0]=0;

  if (input.readfile[0]) // read in existing configuration
    {
      // read the header
      std::ifstream infile(input.readfile);
      if (!infile)
      {
        std::cout << "error, can't open " << input.readfile  <<
std::endl;
        exit(-1);
      }
```

```
      else
      {
        int dim;
        infile >> dim; infile.ignore(256, '\n');
        if (dim != DIM)  // quit if dimensions don't match
          {
            std::cout << "error, dimensions don't match" << std::endl;
            exit(-1);
          }
        infile.ignore(256, '\n');  // ignore the N 1 line
        infile >> input.N; infile.ignore(256, '\n');
        std::cout << "N = " << input.N << std::endl;
        infile >> d; infile.ignore(256, '\n');
        std::cout << "d = " << d << std::endl;
        r = d/2.;
        std::cout << "r = " << r << std::endl;
      }
    }
  else // create a new configuration
    {
      r = pow(input.initialpf*pow(SIZE, DIM)/(input.N*VOLUMESPHERE),
1.0/((double)(DIM)));
    }

//me messing around
int loopers;
for(loopers=1;loopers<=1;loopers++)
{
input.N = (loopers + 2);
//norm

  box b(input.N, r, input.growthrate, input.maxpf,
input.bidispersityratio,
      input.bidispersityfraction, input.massratio, input.hardwallBC);

  std::cout << "ngrids = " << b.ngrids << std::endl;
  std::cout << "DIM = " << DIM << std::endl;

  if(input.readfile[0])
    {
      std::cout << "Reading in positions of spheres" << std::endl;
      b.RecreateSpheres(input.readfile, input.temp);
    }
  else
    {
      std::cout << "Creating new positions of spheres" << std::endl;
      b.CreateSpheres(input.temp);
    }

  std::ofstream output(input.datafile);
  output.precision(16);
```

```
  while ((b.collisionrate < input.maxcollisionrate) && (b.pf <
input.maxpf) && (b.pressure < input.maxpressure))
    {
      b.Process(input.eventspercycle*input.N);
      output << b.pf << " " << b.pressure << " " <<
     b.collisionrate << " " << b.neventstot << " " << std::endl;

      b.Synchronize(true);
    }

  output.close();

//end norm

std::string Result1;
std::ostringstream convert;
int pf2 = (b.pf)*1000000;
convert << pf2;
Result1 = convert.str();

std::string Result0 = "write0";
std::string Result2 = ".dat";
std::string overall = Result0 + Result1 + Result2;
//std::cout << overall << std::endl;

//sleep(10);
//const char* writefile = "write.dat";
//const char* writefile = Result0 + Result1 + Result2;
const char* writefile;
writefile = overall.c_str();

  b.WriteConfiguration(writefile);//(input.writefile);
  std::cout << "b.pf = " << b.pf << std::endl;
  std::cout << "b.pressure = " << b.pressure << std::endl;
  std::cout << "b.collisionrate = " << b.collisionrate << std::endl;
//////////////////
std::string Result3 = ".poly";
std::string overall2 = Result0 + Result1 + Result3;
const char* writefile2;
writefile2 = overall2.c_str();
//////////////////
std::ofstream output54;
output54.open (writefile2);
std::ifstream input1 (writefile);
output54 << "POINTS" << std::endl;
std::string line;
input1.ignore(256, '\n');  // ignore the dim line
input1.ignore(256, '\n');  // ignore the #sphere 1 line
input1.ignore(256, '\n');  // ignore the #sphere line
input1.ignore(256, '\n');  // ignore the diameter line
input1.ignore(1000, '\n'); // ignore the 100 010 001 line
input1.ignore(256, '\n');  // ignore the T T T line
```

```
int looper = 0;
while ( input1.good() )
{
looper++;
std::getline (input1,line);
if (line.length() > 1)
{
output54 << looper << ": " << line << std::endl;
}
}
///////////////////

//end norm
std::cout << input.N << std::endl;
std::cout << loopers << std::endl;
sleep(2);
}
  return 0;
}
```

Appendix C

C.1 Creating Voronoï Tessellations

To construct a Voronoï tessellation a minimum of 4 points are required when working in two-dimensions, as the minimum number of sides to any polygon is 3. Here I will take this simple case of four points to demonstrate the construction of a Voronoï cell (please refer to figure C.1). I will label the four points p_k in space S with coordinates (i, j). For ease p_1 will be taken as the centre disk, that is the only disk that will form a complete cell. Lines will connect all the disk centroids to the others, defined $\overline{p_n p_m}$, with euclidean distances

$$d(p_n, p_m) = \sqrt{(p_n(i) - p_m(i))^2 + (p_n(j) - p_m(j))^2} \tag{C.1}$$

These lines, for all intents, gives the Delaunay triangulation.

Only the nearest neighbours affect the shape of the relevant cell, as no points should lie in the circumcircle of any Delaunay triangle (see figure 4.2). Consequently the Voronoï cell boundaries will always encompass only one disk, as seen for all three Delaunay triangles (p_1, p_2, p_3), (p_1, p_2, p_4) and (p_1, p_3, p_4) in figure (C.1). The boundary will never overlap, as the distance away from the centroids is defined by their distance from each other.

To obtain the edges of the cell, the bisections (E) of these lines are taken. To obtain

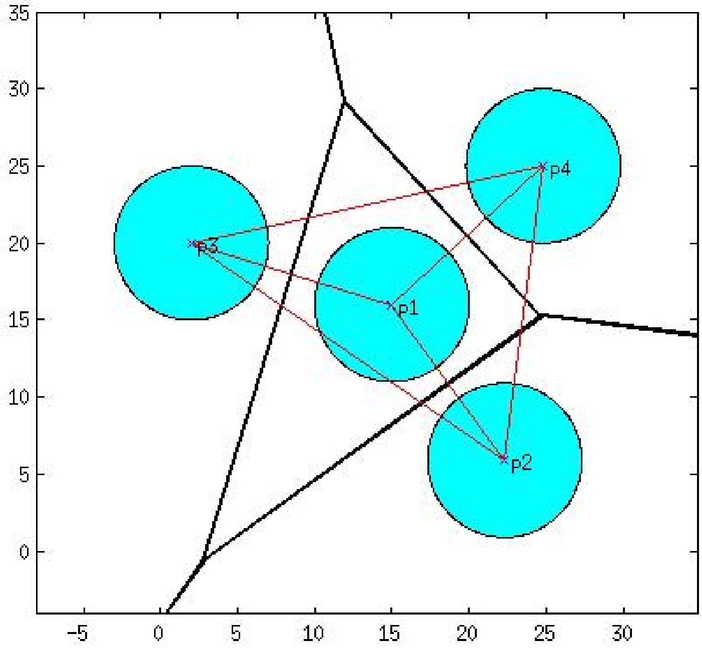


Figure C.1: A Voronoï Tessellation with four disks and one complete cell, outlined by the thick black lines, other lines connect nearest neighbours creating a Delaunay Triangulation

the bisection the midpoint must be calculated

$$(\overline{p_1p_n})_m = [\frac{p_1(i) + p_2(i))}{2}, \frac{p_1(j) + p_2(j))}{2}] \tag{C.2}$$

The edge can then be found by calculating the negative reciprocal of the two points, giving a perpendicular line segment, to ensure the midpoint is cut, the point is substituted into the linear equation, giving the perpendicular line section

$$E(\overline{p_1p_n}) = \frac{p_1(j) + p_n(j))}{2} + [\frac{p_n(i) - p_1(i)}{p_n(j) - p_1(j)} * \frac{p_1(i) + p_n(i))}{2}] - \frac{p_n(i) - p_1(i)}{p_n(j) - p_1(j)}x \tag{C.3}$$

This is rather inelegant and can be simplified to a cell description:

$$P_1 = [p_1 \in S \mid d(p_1, p_n) \leq d(p_1, p_k)] \tag{C.4}$$

which effectively 'sketches' out the area of S which is closer to p_1 than any other centroid

(Voronoi, 1908).