Proximity

How to represent reality in sketches

A graduation thesis by David Rutten [TUDelft, 2006]

A graduation project by David Rutten

Technical University of Delft, The Netherlands

Faculty of Architecture
Berlageweg 1, 2628 CR Delft

Department Urban Design
9th and 10th floor

Chair Technical Ecology And Methodology (TEAM)
http://team.bk.tudelft.nl/
Office 10.01

Many thanks to

Taeke de Jong

without whom this project would never have gotten off the ground

Egbert Stolk

without whom I would have drowned before the halfway point

Peter de Bois

for knowing what must be done

Ali Guney *and* Martijn Stellingwerff

for taking me seriously

Also many thanks to: Steve Baer, Dale Fugier, Jess Maertterer, Jelle Feringa, Philippe Morel, Tim Hemmelman, Karen Buurmans, Paul Rutten, Maria Rutten-Kubicarova, Jared Tarbell, Ross Shannon, Paul Bourke, Takashi Ohyama, Eric Weisstein, Dave Brubeck, Ethan and Joel Coen, Ibanez, Tommy Emmanuel, Gerardo Nunez, Richard Dawkins, Norman Cook, Alex Gifford, Will White, Ludovic Navarre, Luis Bonfa, Eden Kirin, J. S. Bach, Keith Wilkinson, Alan Turing, Rudi Stouffs, Nick Franglen, Fred Deakin, Arturo Bandini, Euclid, Ronald Graham, Olivier Suire, Dan Sunday, Robert Sedgewick, Kevin Wayne, Jono Podborsek, Daniel Rutter, Bill Bryson, Douglas Adams, Audrey Karperien, Thomas R. Roy, The .NET Framework, Reinhard König, Christian Bauriedel, Fransisco Colombo, Han Meyer, Marvin Minsky, Nikos Salingaros, Meta Berghauser-Pont, Per Haupt, Christopher Alexander, Thomas Wingate, Sigrid Brell-Cokçan, H. Edelsbrunner, H. Samet, Mick Underwood, D. Coetzee, Piet Mondriaan, Pieter Bruegel de Oudere, Zaha Hadid, Oliver Bertram, Maarten van den Heuvel and everyone else who contributed to this project.

It is my profound regret that I shall never meet most of you...

Table of contents

Introduction

When we look at the history of architecture and urbanism, we tend to base the story on great achievements and ingenious people. Palladio's designs are still used extensively in lectures today, paradigms by Le Corbusier are mandatory chapters of the curriculum[1]. Of course this is the same throughout human culture; we tend to revere those who excelled and to forget those who did not. This study has no intention of changing this, rather it attempts to examine a possible answer to the question that flows from this observation. What makes people excel in urbanism?

I believe the purpose of education is to help people excel through the means of teaching, stimulating and training. Yet we see that 'uneducated' people sometimes outperform those with proper schooling. Frank Lloyd Wright[2], Charles-Edouard Jeanneret[3], Tadao Ando[4] and Will Bruder[5] to name just a few. So are we then to conclude that education is a luxury or overkill for the 'genius'? That education is only worthwhile for the average student?

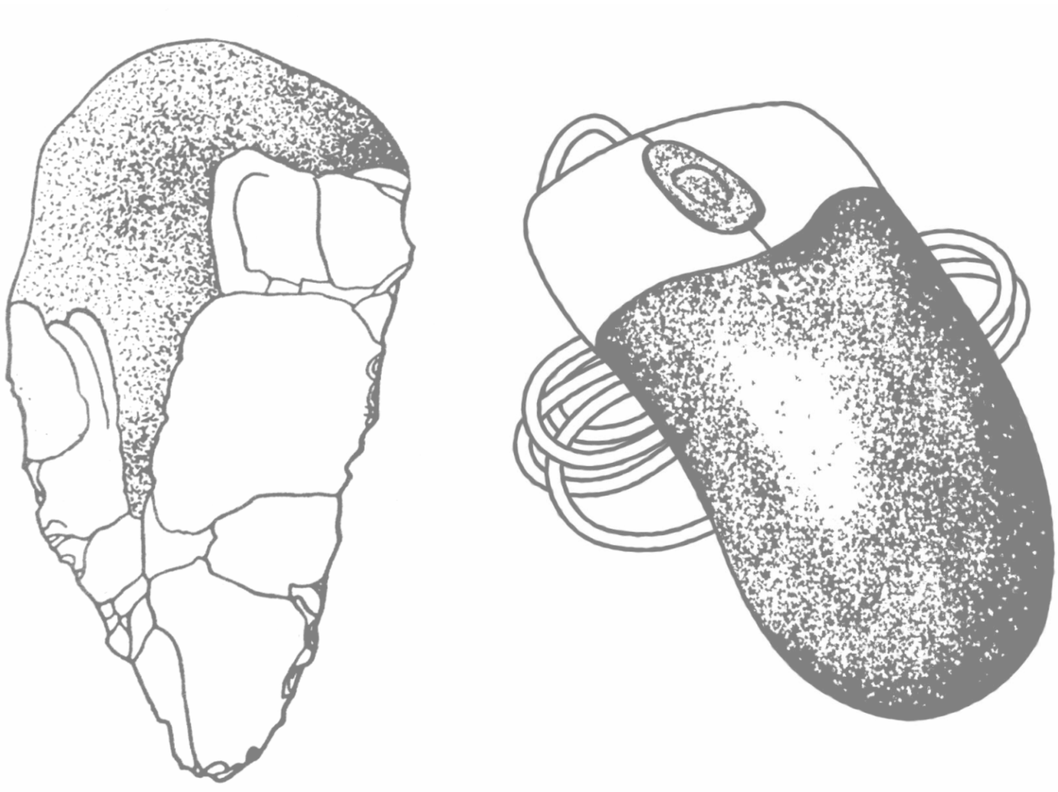
Obviously, things are far more complicated than this, and the achievements we hold in awe always arise in a context that would not have existed without the trade we have created as a culture. If we discuss the subject of urbanism (and the level of cultural quality) from a top-down point of view, the issue seems a vast and complicated one. Perhaps it would be better to start at the building blocks of urbanism; the urbanist himself.

An urbanist is only human. And along with humanity come certain limitations. When we plot the history of human cities onto the history of human kind, it becomes such an infinitesimally small part that we can safely state that -from an evolutionary point of view- there is no such thing as urbanism[6]. As a result of this, most people are totally incapable of imagining the orders of scale that are involved in urban design. We simply haven't evolved the ability to grasp the vast complexity of urban systems, in much the same way as most of us haven't evolved a tolerance for dairy products yet. As a consequence we have developed rules of thumb, measurement tables and mathematical models to aid us when we need to make decisions on such a scale. But the usage of these means is usually cumbersome and time-intensive and thus we tend to put them off until we have narrowed down our design. Unfortunately, by then, our plan may be littered with fundamental mistakes.

Fortunately, humans do posses other qualities which offset the above stated culprits. Inspiration allows us to find solutions in appeearently unrelated sources, something no computer can do. Our capacity for abstract thought enables us to simulate solutions in our imagination. Our sense of (inverse) reduction[7] keeps us comfortable when using models, and our capacity for judgement by comparison[8] makes sure we can pick solutions from underconstrained problems. These traits have kept the profession afloat for centuries and there is no reason to believe we shall ever be able to live without them.

1. Of course many out-dated ("archaic", "antiquated", "ancient" or however we prefer to label them) paradigms taught, are not meant to be taken to heart. Rather they are provided as bad examples. This does not in any way diminish their importance to the professional field.
2. Frank Lloyd Wright (1890–1978) was briefly attached to the University of Wisconsin at the faculty of Civil Engineering, but learned the trade not at school, but in the field. Mostly while working for J. Lyman Silsbee, Louis Sullivan and Dankmar Adler.
3. Le Corbusier (1887-1965) acquired the bulk of his early knowledge through practical application as well. Among his teachers are Auguste Perret and Peter Behrens.
4. Tadao Ando (1941-present) really appears to be completely auto-didactic. The only architectural training he participated in prior to starting his own firm in 1969, appears to be the tracing of sketches in a book by Corbusier. One of the finest examples of the superfluity of education?
5. William P. Bruder (1946-present), a frequent but shadowy player in the field, founded a company based on nothing more than a degree in sculpture and apprenticeships with Gunnar Birkerts and Paolo Soleri. So why then does he consistently outperform his competitors?
6. Evolution of modern man started about six million years ago. Earliest spatial planning projects were about fifteen thousand years ago. This equals 0.0025% of the species time line.
7. The ability to imagine the simple system underlying a complex one, and to imagine the complex system as implied by a simple reduction.
8. Regardless of primary-school teachers desperate attempts to suppress it, human beings are perfectly capable of comparing cheese to chalk.

Still, the world we inhabit is changing rapidly[1] and urbanism must adapt equally quickly if it is to remain a major component of human culture. Sooner or later, the very essentials of our craft as we know them today will fail to pass the test, and if we are to survive we need original solutions. In any case, -from that day on- urbanism will never be the same again.



1. According to the modern evolutionary paradigm, the world we inhabit is not changing with a constant pace:

> *"...in addition if you look at the timescales that's involved here; two billion years for life, six million years for the Hominid, a hundred thousand years for mankind as we know it. You're beginning to see the telescoping nature of the evolutionary paradigm. Then when you get to agriculture, when you get to scientific revolution and industrial revolution you're looking at ten thousand years, four hundred years, a hundred and fifty years... you're seeing a further telescoping of this evolutionary time. What that means is that as we go through the new evolution it's going to telescope to the point that we should be able to see it manifest within our lifetime, within a generation."*
>
> *-Eamonn F. Healy, Waking Life-*

Urbanism has been greatly affected (and has affected) all of these revolutions. Indeed, cities and human culture are one. They cannot be imagined separately. At some point in the near future the rapidity of contextual change will disallow the continuation of our current planning system.

A. Problems

To those of you who concluded the introduction with a feeling of impending doom, I wish to apologize. Death is never a pleasant event for those involved, be it of biological nature or the demise of idea(l)s. Still, death -and in a more paramount role, extinction- is a vital part of life and evolution[1], and urbanism-as-we-know-it is only here because it managed to overtake its predecessor.

> *The only alternative to extinction is stagnation, and stagnation is seldom a good thing in any field.*
>
> *-Ian Tattersall-*

But we should only dispose of existing theorems if we have a valid reason to do so. In this section I shall try to highlight several problems with the practice and theory of contemporary urban design. Some of these problems are addressed in the second section called "Solutions", where I propose a new method of drawing (and thinking about) spatial systems.

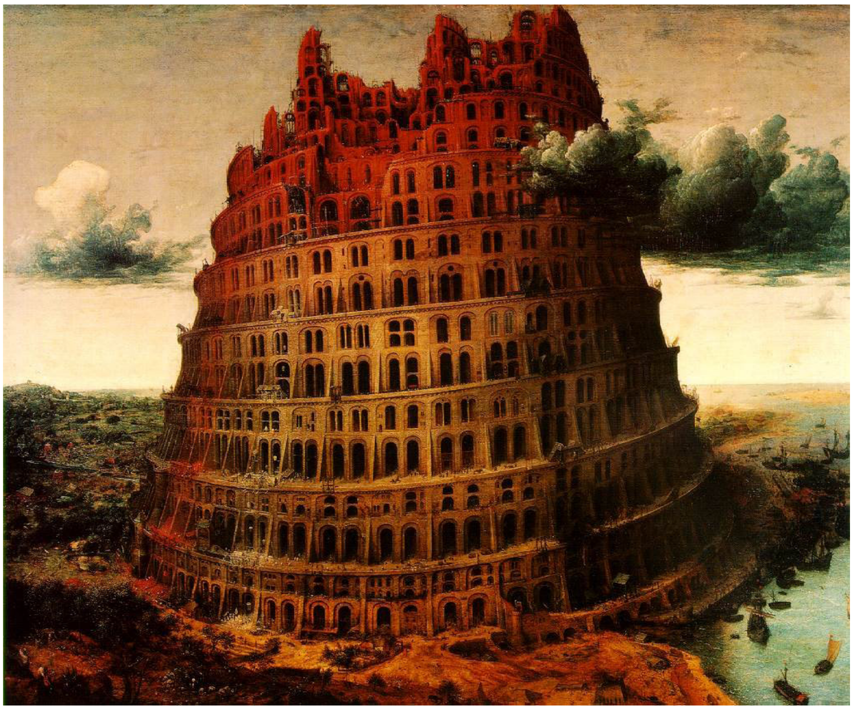
Over the course of this project, I have talked to many people about its paradigms and details, and by far the most common response was doubt. Perhaps it was my manner of presentation, but I cannot rid myself of the feeling that there exists a powerful aversion of digital techniques in the field of urbanism. I understand it is quite natural for humans to fear being overtaken by machinery, and I would like to stress that this is not in any way the aim of this study. I am a strong believer in the power of human creativity and only wish to remove a few obstacles in its path.

Indeed, this essay is not intended to be a holistic solution of any kind, and many existing problems will remain unsolved and many new ones will arise. Nor is this essay archetypal in the urbanistic field. Any good ideas I have come up with are in some manner related to the 1992 publication of Prof. T.M. de Jongs "Abbreviated methodology for research by design". A strong candidate for the least read (or perhaps the least valued) important publication in the field.

1. According to Richard Dawkins in "The Extended Phenotype", primary evolutionary speed depends directly on survival pressure. Evolution space is to be imagined as a large surface where the summits and peaks represent local optima. Selection pressure scales the surface vertically so as to speed up movement towards the nearest peak. However, high pressure also limits the phenomenon known as 'evolutionary drift', where processes are allowed to wander away from local optima towards farther away, perhaps higher summits. It would appear as though selection pressure for urbanism is at an all time high at the moment, as opposed to architecture which seems to be in a drifting-period.

A.1 Problems with terminology

The most unenticing issue when discussing urbanism, is terminology. In order to talk about something in an intelligible manner, we must adhere to the same dictionary. The most famous major urban project of all time -the Tower Of Babel- ran aground exactly because of a lack of common nomenclature[1].



The Tower Of Babel

I am not a follower of Karl Popper in that I reject the value of words, simply because they point only to other words[2]. Nor do I openly side with Wittgensteins rigid semantic definitions:

> *"The limits of my language mean the limits of my world"*.
>
> *-Ludwig Wittgenstein-*

The assumptions I have made concerning my readers, allow me to treat language in general -and the urban jargon in specific- with a certain amount of license. I shall only call focused attention to important, frequent or unclear terminology. In the following paragraphs, along with the definition of certain terms, I will also highlight some of the problems that are relevant to this particular study.

1. Genesis 11:4-7
"Come, let us build ourselves a city and a tower with its top in the heavens, and let us make ourselves a name, lest we be scattered upon the face of the entire earth". And the Lord descended to see the city and the tower that the sons of man had built. And The Lord said, "Lo! one people, and they all have one language, and this is what they have commenced to do. Now, will it not be withheld from them, all that they have planned to do? Come, let us descend and confuse their language, so that one will not understand the language of his companion".
2. If you think about it, a dictionary is probably the worlds worst repository of knowledge. You need to know a major amount of the dictionary content in advance or the whole thing doesn't make sense. It explains things only by referencing other entries in the same book. If you needed to look up the words used in a definition, you would very quickly find yourself in endless loops.

A.2 What is Urbanism?

It may strike you as ironic, that the first term I would like to dispose of, is "Urbanism" itself. There is no agreement on what exactly urbanism is. Indeed, the definitions vary wildly;

> *"The culture or way of life of city dwellers."*
> *-The American Heritage Dictionary-*

> *"The study and practice of creating humane communities for living, work, and play."*
> *-Wikipedia, on-line encyclopedia-*

> *"The study of the characteristics of a town or city."*
> *-Oxford Dictionary-*

Three major repositories of the English language differ substantially in their definition of the subject. Either omitting the human component, the functional component or the spatial component. But my main reason for relinquishing the word, derives from its Latin heritage. "Urbanus" meaning "City", and since cities are such a key component of human culture, we are helplessly predisposed when using the word. Furthermore the term "urban" or "city" conveys additional qualifications such as a certain size, a certain density and a certain renown. Instead of urbanism, I propose to use the term "spatial planning". Not that it covers the subject flawlessly, but it is at the very least free of prejudice and assumption. Also it caters for rural and regional design, which -as will become clear- cannot be treated separately from urban planning. Also, "spatial planning" enforces the link with the field of planology.

Since Architecture and Urbanism are taught at the same faculty in Delft, the fields tend to permeate a bit in everyday life. It is of vital importance that we draw a line between spatial and geometrical planning here. Geometric planning derives from Architecture and it deals with the usage and function of three-dimensional shapes (or at least shapes in three-dimensional space). Spatial planning deals with the distribution of usage and function over $\mathbb{N}$-dimensional space[1].

The immediate benefit of this approach, is that there are no more overlaps between geometrical and spatial planning. Every aspect of every design is unambiguously one or the other. Under the classical characterizations derived from the terminology "Architecture" and "Urbanism", there are vague areas where both apply to a design at the same time, or -even worse- neither apply:

> *"Urbanism reaches a meter into a building.*
> *Architecture extends a meter beyond a building."*
> *-Common proverb-*

This is an unacceptably ill-defined approach. In this study, the subject is spatial planning. Thus keywords that apply to geometric planning, will be of no importance here (shape, form, volume, colour and material to name a few). The upshot of these new definitions is that spatial planning is no longer restricted to the large scale. In a way this is a good thing, since "large" is a grade, not a measurable threshold and it can serve no purpose in a scientific definition.

By dismantling the scale limitations we run into one of the most pressing issues with spatial planning; methodology. Things on different scales are designed using different methods. Architects would be ill-advised to design buildings in the same way as technicians design microchips. Urbanists are indeed frequently ill-advised to design cities in the same way as architects design buildings.

---

1. The urban space is $\mathbb{N}$-dimensional since connections between two objects in it no longer need to exist within the visible urban surface. Before the advent of electricity and digital networks, the urban space was a two-dimensional surface, but now it features hyperlinks, short-cuts and intersections. If you send an e-mail to your neighbour, the data might pass through Fiji before it reaches him. For the purpose of readability we still project the $\mathbb{N}$-dimensional space onto a two-dimensional surface reducing the complex volume to symbols and dotted lines.

A.3 What is an urbanist?

It would follow from paragraph 1.1 that the term “urbanist” should be cast away in favour of “spatial planner”. Although it would be both consistent and logical in the light of this paper, I am hesitant to make this step. Firstly, the word “urbanist” does not suffer from the same ambiguities as ”urbanism” and if it’s not broken I’d rather not fix it. The easy way out is to simply define an urbanist as one who is involved in spatial planning. Luckily this section is not about urbanists as a professional status, but about urbanists as human beings.

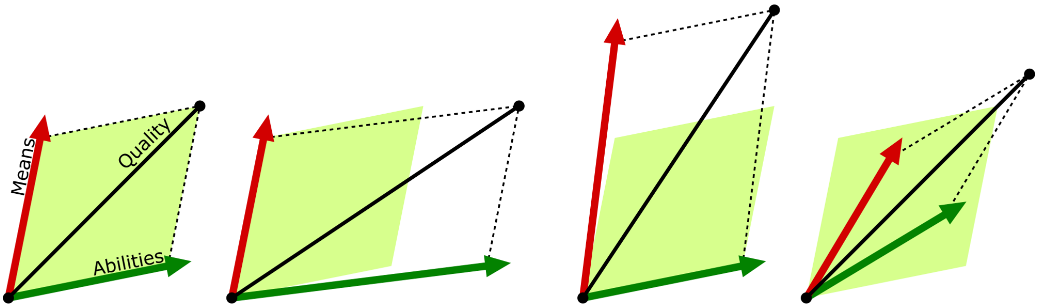
A.3.1 What is professional quality?

When we look at the average urbanist, the first thing we notice is that he is average. In order to become a better -above average- urbanist, he must improve at least one of the two vectors called ‘abilities’ and ‘means’ [T.M. de Jong, H. Priemus, 2002]. According to de Jong and Priemus the quality (or fitness) of a certain individual is the vector-sum of these two properties.

Under abilities we group all personal properties such as knowledge, talent, swiftness, intelligence and imagination. This is the territory of education. By learning we enlarge the vector called ‘abilities’ which will result in a larger quality sum.

Means on the other hand are non-personal properties and can thus vary on a moment-to-moment basis. You might have a lot of means to your disposal at the office, but these cease to exist once you get fired. Means can also differ from project to project and from time to time. Means are thus very similar to tools, both physical (a drawing board) and untouchable (the guy sitting next to you who knows everything about anything).

The amplitudes of these vectors are a measure for the relative quality-in-part of an individual. The direction in which they point is meaningless, but the angle between them indicates the amount of discrepancy between the two. ‘Experience’ is related to the vector alignment, which means you can improve at something all by yourself, without buying means and without attending classes. So in order to become a better urbanist, one can either get more schooling, buy more tools or practise his craft.



From left to right: The default situation (since this diagram is purely relative, any situation can be chosen as the default); increased abilities (education); increased means; improved alignment (practise).

We cannot completely separate means and abilities, since you need certain abilities to use certain means, and -vice versa- these abilities do not make much sense without those means. But we can expect that an increase in any of these two components will result in a better urbanist. If we plan to tackle the problem not from an educational starting point -this is after all the field of politics and management- we should examine how to **create new means that help designers achieve better results**. That is the aim of this study.

A.3.2 What is design quality?

The obvious answer of course is: "A design which performs well in respect to expectations". But in the case of urbanism[1] (and to a lesser extend architecture), we cannot simply build any design to see if it performs well. Indeed, even **if** a design is finally realized according to plan, it might take years or even decades before it can be evaluated properly. Becoming an urbanist therefore is a bit like learning to ride the bicycle by reading the manual.

Clearly we need other criteria for judging design quality, especially in the case of urbanism students whose designs are seldom picked up by society. Typically, a design will be judged primarily by the designer himself, using his evaluation vocabulary[2] to reach a personal verdict, and subsequently by others who might have more experience; his teachers or colleagues. This is a highly unscientific method of judging a design and everyone who has ever lost a design competition will agree with me.

In order to define design quality, the design must be a measurable entity. Then and only then can we apply typical quality constructs to urbanism. The design as a whole can contain any amount of subjective elements as well, but we should be careful not to judge these in the same way as we might evaluate objective (measurable) properties. 'Design quality' then, breaks up into two separate qualities; that which can be measured and that which can be (dis)agreed upon. The first is a scientific property, the second an artistic one.

> *"As a scientist you always think you know what you're doing, so you tend to distrust the artist who says, "It's great," or "It's no good," and then is not able to explain to you why..."*
>
> *-Richard P. Feynman-*

The definition given in the first sentence of this paragraph does not stop at "well". If it did then quality would be absolute. This is not the case. Quality is a relative property which is measured in terms of expectation ('expectation' is usually referred to as 'programme' or 'assignment' in urbanism). This in turn means that a design cannot be scientifically evaluated when the programme does not contain statements which can be put in numeric form. A simple example of a programme which does not[3] feature numeric statements would be:

- *The design must be part of the natural landscape*
- *The design must fit within the local cultural framework*
- *The design must be safe for playing children*

Some of these properties might be measured after the plan has been realised[4], but while the design is still virtual they are a matter of opinion. A programme which aspires to be measurable must provide measurable statements such as a minimum number of dwellings, parking space coverage, access routes or playground surface areas.

It is not necessary for a programme to contain numbers in order to be measurable. Since quality is a relative property, some design aspects can be evaluated against the programme, while others can be evaluated against similar aspects of rival designs. The safety of playing children is a typical example of a comparative aspect. Of course if the design forces children to play hide-and-seek in the middle of a train depot then it will be absolutely unsafe, but this is an unrealistic scenario. Typically, if two designers come up with competing solutions for child-safety, it might be very difficult to determine which one is the best. If 'safety of playing children' is to be treated as a measurable property, then there must exist a numeric defini-

1. Since we have defined "urbanism" to be in fact spatial planning, we can now happily use both terms to indicate the same thing. For the sake of readability I prefer to use "urbanism" instead of "spatial planning", but remember what urbanism is about in the context of this paper.
2. What is commonly referred to as gut-feeling or intuition.
3. This is an over-simplification for the purposes of narrative. Eventually we will see that some apparently artistic statements have a numeric counterpart.
4. Measured for instance in the amount of opposition by locals, or the amount of toddler-fatalities per year.

tion of this property and this definition must be agreed upon by all parties prior to design comparison. By defining numeric counterparts for 'soft' properties, a programme can remain vague without becoming less scientific.

A.4 What is methodology?

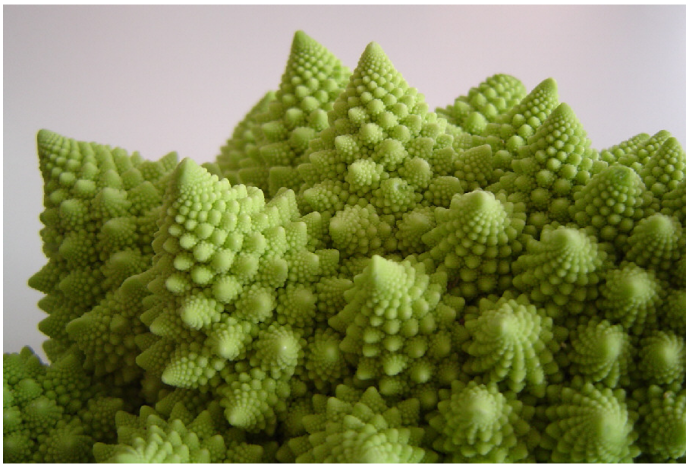
Spatial planning marks the start of modern man. From cavemen digging bear-traps, via the ancient Greek cities to Disney-world. The conscious alteration of the environment is one of the trademarks of the human species[1]. It would be natural to view modern spatial planning as an improved version of the planning that was. Bigger, higher, faster. This is in fact not correct. Spatial planning went through three distinct periods (and did so more than once), each a result of a specific administrative context.

- Bottom-up planning
- Top-down planning
- Emergent planning

Every type requires a different approach of the planner. This approach is methodology.

A.4.1 Bottom-up planning

Bottom-up planning is a ruleset which applies only to the building blocks of the whole system. By limiting or guiding individual buildings, the city as a whole emerges. Bottom-up construction is natures favourite solution for spatial planning issues. Bee-hives, coral reefs and foliage are common examples.



Romanesco broccoli. The final shape of the vegetable is a result of the growth pattern of the individual cells. The ruleset in this case is stored in the DNA. I.e. at the bottom-most level of the system.

The earliest congregations of human culture were based on bottom-up planning and even today it is still a widely used system in many parts of the world where there is no centralized planning office. Bottom-up planning can be emulated by using so called 'intelligent agent systems' in software design. Here a large group of agents is equipped with a specific ruleset and then inserted into an environment. Every agent has the ability to measure and alter its surroundings by applying the rules it has been given. The agents are not genuinely intelligent. We call them intelligent because they react to their environment and as a result of this, their behaviour is complex and maybe even chaotic[2]. Note that intelligent agent systems *can* contain genuine intelligence, but so far this only applies to biological or extremely advanced digital systems.

1. Other building animals such as beavers, bees and termites are not conscious about their actions. Their behaviour is genetically determined, ours is memetic. We can choose to build something different for a change, bees cannot.
2. Typical ant behaviour is dominated by less than ten variables. Yet the motion and behaviour of an entire colony exhibits signs of intelligence. A strong parallel exists between ants and brain cells. Only when active in the context of each other, does intelligence emerge.

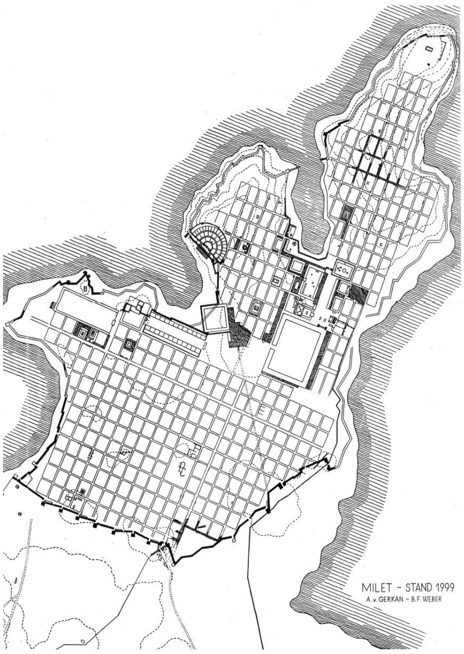
From a spatial planning point of view, one such agent could represent an individual over the course of a day, a building over the course of a century or a million things more. The fundamental advantage of intelligent agent systems is that they tend to find local optima. Every agent (or 'particle') attempts to maximize it's own level of satisfaction, meaning the system as a whole can always find an equilibrium. This stability makes it a popular method in simulation science.

By introducing varying rulesets into an environment at the same time, an economic component is added to the simulation. By making agents affect each other though a common environment, different units can be compared and waged.

The methodology which accompanies bottom-up planning is not a very abstract one. Problems are always dealt with in situ and -as a rule- the system as a whole is never taken into consideration.

A.4.2 Top-down planning

Top-down planning is in fact the opposite of bottom-up planning. Instead of creating rules for the smallest blocks, rules apply to top-level objects. Ancient Greek and Roman cities for example are often layout on a grid. This grid is a top-down constraint, dictating the behaviour of the smaller building blocks. Note that grids can also arise from bottom-up planning, as we can see in bee-hives. It is the way in which grids arise that is important here.



Top-down planning requires a strong enforcement of law and since it cannot solve problems locally, it is generally less suited to take position in an existing environment. Since top-down systems occupy the reality in which they exist, they are harder to compare than bottom-up systems. Two top-down systems cannot coexist in the same space at the same scale at the same time, since that could lead to ambiguous constraints, but bottom-up systems are capable of interacting with each other through the reality they share.

Top-down systems are however the best example available of structured planning. The advantage of top-down planning is that the system is intuitive and understandable for the human psyche. It appears less chaotic and less arbitrary. The methodology involved in top-down planning is typically a very abstract one which uses very platonic solutions.

An example of top-down planning in nature can be found in non-growing structures such as birds nests. Although the shape of a bird's nest is stored in the DNA of the birds cells, the ruleset -from the point of view of the nest- is imposed onto the system from above.

A.4.3 Emergent planning

Emergent planning is more than just a blend of the preceding two systems. In a world that is changing with an ever increasing pace, long-term planning can no longer originate from a single top-level source. Rules and limitations are spawn, changed and retracted at all levels of scale. Planning now becomes a chaotic system, where any actor can only affect a limited amount of layers for a limited amount of time. Hierarchy disappears as local, regional and global aims conflict. Due to the overlapping weave of do's and don'ts it may in fact no longer be possible to satisfy all constraints all the time. It is here where emergent planning differentiates itself from its predecessors. Emergent planning revels in a chaotic system because it is chaotic itself. Not 'chaotic' in the out-of-control sense of bottom-up planning, but chaotic in a calculating, plotting manner. It is also here, that the creativity of the urbanist can play a paramount role in the process. The absence of rigid rules makes room for interpretation. Emergent planning is not a checklist.

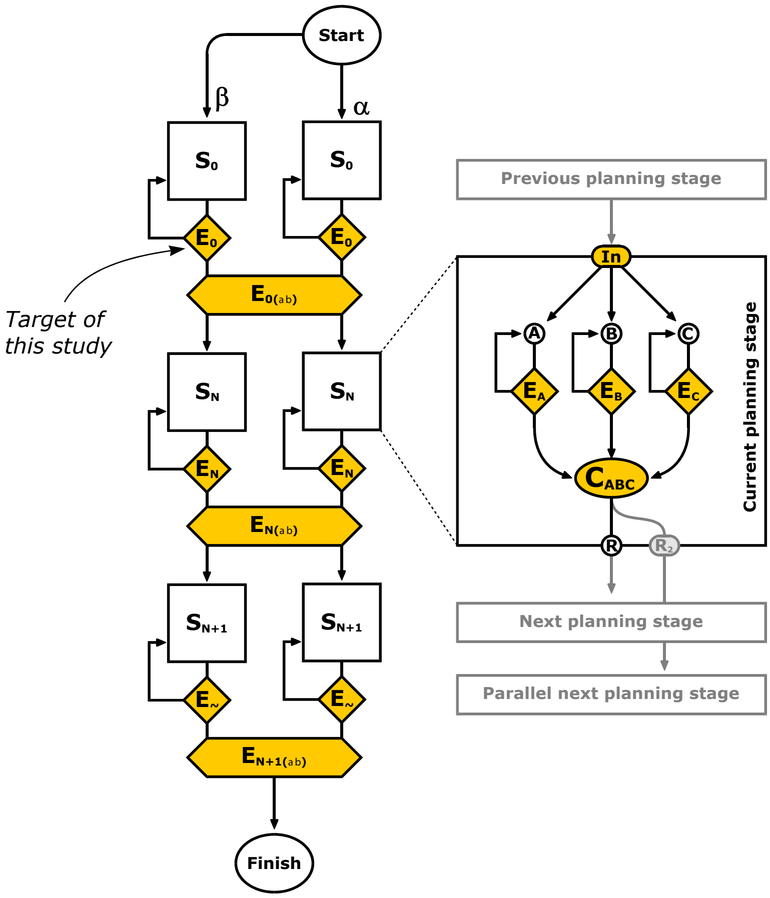
So top-down planning is not an acceptable methodology in a chaotic environment. The obvious solution that does not require a wholesale revision of planning paradigms, would be to reduce the refinement of the top-level plan and to make it serve only a frame-work function, the details being tackled in situ. This divide-and-conquer approach is in fact a very popular method among contemporary planners. The down side is that it works by diffusing control and responsibility. Bottom-up planning stems from anarchy; the very lack of a supervising entity is part of the definition. As occupational urbanists we must protect our profession which means we should not endorse out-of-control systems with uncertain outcomes.

Still, most modern spatial planning projects are rooted on (usually a combination of) these antiquated paradigms. True emergent planning is not an easy solution. The myriad of constraints and legislation make the system hard to analyse and simplify. In effect, an emergent planning system might be one of the few examples of genuine irreducible complexity[1].

This complexity is especially disobliging in the design phases where limited amounts of information are available, such as during the sketch-phases. Most of the restraints that apply to a design cannot be evaluated during early design stages, because they haven't been designed yet. The emergent methodology now becomes an evolutionary cycle divided into several stages, which usually -but not always- occur in the same order. Definition of programme, concept-formation, sketch-designs and blueprints are four of such stages. Since every one of them deals with either different subjects or different levels of detail, some 'mistakes' in one stage are bound to be fixed in the next one, while others have a nasty habit of perpetuating themselves. Let us call these mistakes "overwritable" and "progressive" respectively. Of course there is no clear line between the two. Every mistake starts out as being overwritable, and if it isn't dealt with in time, every mistake will become progressive. The most logical place to draw the line would be between stages. If problems can be solved in upcoming stages of the design process, they are overwritable; we can afford to overlook them for the time being. Examples of overwritable mistakes are 'lack of accuracy' and 'incomplete legends' since these require further elaboration on lower scales rather than adjustments on higher scales. Examples of progressive mistakes are 'errors of scale' and 'faulty networks', since these contain errors on the highest inherent scale.

A creative process has many feedback loops where designs are evaluated and compared. Many publications about methodology feature a flow-chart representing the ideal design process as imagined by the author, and no two authors seem to agree. The reason for this is to be found in the inherent subjectivity of any creative process. Designing is one part science, one part craft and two parts art. The upcoming chart is thus not to be treated as my vision on perfect planning. Rather, it is a reduction of what I believe to be the common design process.

1. An irreducably complex system cannot be simplified by removing or reducing a single element in it. It is a bit of a logical oddity since it is very difficult for such systems to arise. Still, reducable systems can theoretically become irreducable when elements continuously adapt to their changing context. Something which undoubtedly happens all the time in human culture.



The Start field supplies the ruleset around which the design will be wrapped. This has been referred to earlier as the programme. Additionally, a designer might have a signature of rules which he appends to the Start field in order to make a trademark design.

S_0 then covers the first planning stage. This could be a textual, schematic or illustrative stage. As mentioned before, every planning process might have different amounts and different sequences of stages.

Whenever a Stage S_N is completed, it is evaluated at E_N and if it does not pass the test (as supplied by Start), the planner has to revert to the beginning of S_N. This is an overwritable mistake which has been caught just in time. This is a relatively expensive setback. The only worse setback would be a return to S_{N-1}, which would constitute a progressive mistake. If the temporary design evaluates correctly, it will be compared with the parallel designs at $E_{N(\alpha\beta)}$. At this point the decision to abandon a design strain can be made. The remaining strains will enter the following Stage S_{N+1} and the process repeats itself until the desired or required level of elaboration is achieved.

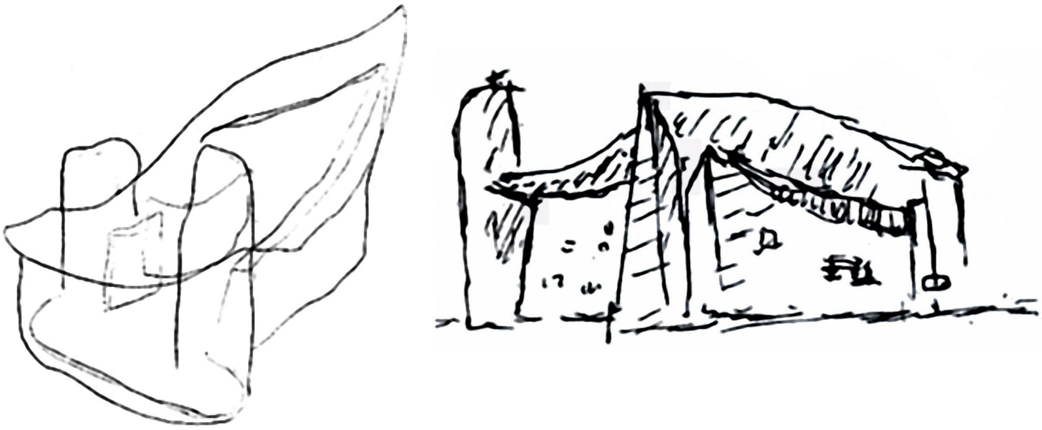
A similar process exists within individual Stages. A single design with the accuracy of S_{N-1} is inputted and elaborated into higher accuracy designs (A, B & C). These are in turn evaluated at $E_{(A,B,C)}$ and finally compared with each other. Since two or more designs may be evaluated as valid, the output of a Stage may be more than one design. I.e. every Stage has the ability to instigate a new parallel design strain. R_2 is an example of this.

So far so good. The process appears to be logical, clear and flexible. But drawing a yellow diamond and actually evaluating a design are two different things altogether. The chart may be clear-cut, the process it describes is anything but. The target of this study is represented by the little arrow from E_N pointing backwards to S_N. The **immediate and accurate** evaluation of a design phase. The currently available methods of performing this evaluation are few and cumbersome; nobody wants to spend five hours evaluating a sketch that took five seconds to make.

The evaluation of a design-in-part can be performed by a person using either intuition or calculatory methods as described in paragraph [*A.3.2 What is design quality?*]. This type of assessment is part of the ability-vector. No matter how fluent one becomes at evaluating a design, it will either take a significant amount of energy and time in order to calculate and analyse the plan, or the resulting judgement is purely emotional, in which case you might have problems convincing others of the correctness of your deductions (assuming for the time being they are actually correct). This may not be a big problem when it comes to blueprint evaluation, but -as mentioned earlier- it is something we tend to put off during the early stages. This is a profound snag in emergent methodology. Since every level of scale overlaps with the adjacent levels, any change made to a certain plan element, will cause the planner to revise previous stages all over again.

A.5 What is a sketch?

When we talk about sketches, we are not necessarily discussing the same subject. For the word "sketch" is used to describe two different kinds of drawings[1]. One is the representation of thought such as used in the design process. The other is the representation of perception. We are only interested in the first type here. The drawings below clearly show the separation of thought and perception in sketches.



Two sketches of the Ronchamp chapel. The one on the left was made by Le Corbusier himself prior to construction and the one on the right was made by a visitor.

In practical terms, wherein do these two kinds of drawing differ? Sticking with the Ronchamp example, the sketch on the right is very clearly a representation of its subject matter. No architect would have problems recognizing it. The sketch on the left on the other hand is not such a clear-cut case. However, once we are **told** what the sketch is about, it is suddenly much more informative than drawing #2. The essence of the building has been captured much more vividly. The sketch on the left tells you things you could not know simply by looking at the chapel.

The representation of perception is a translation from sensory input to strokes on paper. By 'reading' the paper afterwards one can reconstruct the artists image of reality at the time the drawing was made. However, when thoughts are transferred into strokes, only the artist is able to read them back, everyone else will have to come up with their own thoughts to fit the gap. This is simultaneously the greatest strength and the greatest weakness of sketches.

1. It could also mean "A brief, light, or informal literary composition, such as an essay or a short story.", "A short, often satirical scene or play in a revue or variety show" or "An amusing person"...

A.5.1 The strength of sketches

Sketches are made very early in the design process. In many cases we might even refer to the very first drawings as 'sketches'. The thing which separates sketches from other drawings is the way in which they are made. Sketches are a hard-copy (a back-up if you like) of a certain train of thought that occurs to the planner while designing. Sketches then, allow the designer to work on many parallel designs and concepts at the same time, since the amount of paper to hold his thoughts is infinitely larger than the amount of memory[1].

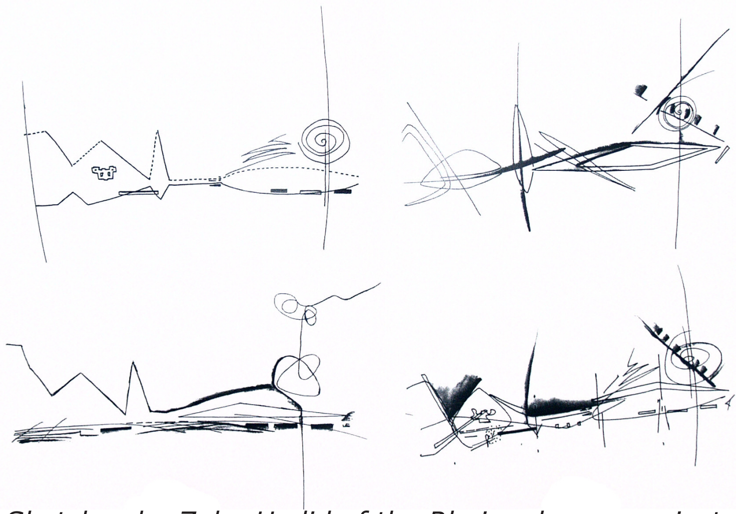
But a sketch is not necessarily drawn after the thought has occurred. In many cases the very making of a sketch facilitates the thinking baout its subject matter; this effect is usually referred to as "the paper dialogue". The upshot is that the design which exists in the planners mind when he begins the sketch is probably very different by the time he completes it. Sketches are thus not only a record of thought, they are a record of changing thought.

To the designer, sketches are of major importance because they allow him to cope with a much higher level of abstraction. To others involved in the same project, sketches might be of major importance since they allow designers to communicate tacit knowledge. To outsiders, sketches are probably meaningless.

> *A picture says more than a thousand words, but which words are these?*
> *-T.M. de Jong, Ways To Study-*

A.5.2 The weakness of sketches

So how does one create a scientific evaluation for thoughts on paper? I offer the modest proposal that this would be impossible[2] with traditional sketches. A line in a sketch drawing could represent just about anything. The only person who is capable of evaluating the design a sketch represents, is the person who made it.



Sketches by Zaha Hadid of the Rheinauhaven project.

The drawings above were made with decisive strokes. Hadid knew what she was going to draw. Yet, nobody in the world except Hadid herself can truly understand what is drawn here because the sketch components have no meaning. That which has no meaning, cannot be measured. That which cannot be measured cannot be evaluated.

1. We're not talking about the total memory of a human being. Von Neumann estimated that a human can store somewhere in the vicinity of 10^{20} bits over a lifetime. We're talking about active memory which can only store seven (plus or minus two) discrete items of information.
2. In fact, many researchers are concerned with the conversion from thought to mathematics. The Sandbox project [Rob Harris, MIT 2005], Talk aloud protocol [Nancy Yen-wen Cheng, 2003], [Koichi Matsuda et. al. 1997], Egaku research [Jennifer Yoon et. al., 2004] to name just a sampling. Process analysis is not the same as Meaning analysis though and it is the latter we are interested in.

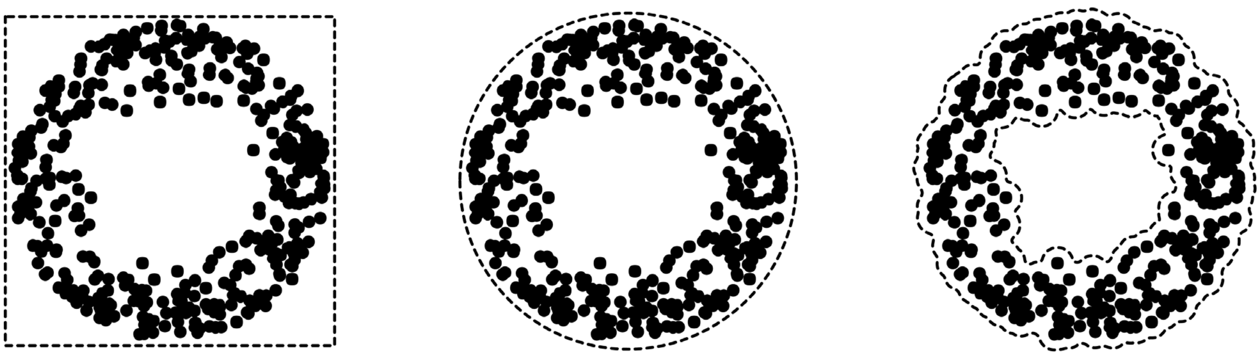
This is a profound problem. If we cannot judge early sketch designs on programmatic properties, how can we choose among different sketches? Really the only way we have is to elaborate the sketches into more accurate designs until they can be evaluated. This is a time and labour intensive process (see [3.4 The creative process]) which is likely to be omitted.

Another problem with sketches for spatial planning stems from their architectural heritage. A geometrical design can be captured in line-drawings since lines can be used to indicate sharp boundaries, such as when concrete ends and glass begins. Representing distribution -the underlying property of all *spatial* designs- using only pencil strokes is not so straightforward. Still, since we have almost no other means to our disposal[1] we still tend to use this unpleasant method.

> *Most of our drawing equipment forces our thoughts directly into lines, producing directions, surfaces and bodies you cannot forget so easily [once] they are drawn. They limit your imagination more than strictly necessary by an existing site and a preliminary programme of requirements. They do not challenge [the] imagination as much as still possible within these constraints, like stellar constellations or clouds do, evoking many different shapes to be recognized.*
>
> *-T.M. de Jong, private correspondence-*

In order to represent spatial properties using geometrical tools, we have invented for ourselves a dictionary of symbols; green areas with little triangles on maps tell us that there's a forest in that particular location. It doesn't provide any additional information such as the average height of trees, or the average spacing between them. Of course specialized maps will have additional legend-units that do convey additional information, but again it is based on the same dictionary-of-symbols principle and thus bound to the same limitations. In contrast, if the map is accurate enough to portray buildings, the contour of these buildings can be drawn very accurately. The exact shape of a contour is not a legend unit, the information is stored in the drawing itself; it does not require a reference table.

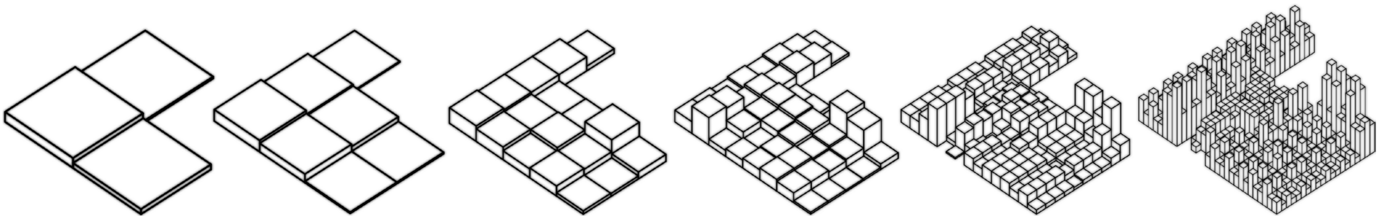
Apart from having to use a legend to describe the properties of a region in a map/sketch, it is also very difficult to determine the exact outline of the regions. Often there is no such thing as a clear border. City doesn't suddenly turn into Nature any more than a person suddenly turns old. Since distribution of objects is the base property in urbanism, there is rarely a clear boundary to be found.



Given the distribution of black dots, where to draw the boundary? These three options are all mathematically valid and are known respectively as: Bounding-Box, Spanning-Circle and Connelly-Boundary.

And it does matter a great deal where the region boundaries end up being drawn. Since a region is assumed to be uniform on the inside, the three regions in the drawings above will all end up with a different density value.

1. Or, since we are rarely taught that other means exist...



A density bar graph of a certain distribution of points. The model remains unchanged, but the region size is decreased from left to right. Truth depends on how you look at it.

Traditional maps then, are very well suited to represent the geometrical reality, but we need abstraction to make them work on spatial reality. This layer of abstraction prevents numeric analysis of spatial sketches and severely hinders a scientific approach. The obvious solution would be to dispose of the limiting sketching tools and instead start to draw that what it is all about; spread.

How do we draw spread? How do we evaluate such a drawing? How do we protect creativity?

A.6 What is creativity?

> *Creativity is defined as the tendency to generate or recognize ideas, alternatives, or possibilities that may be useful in solving problems, communicating with others, and entertaining ourselves and others.*
>
> *-Robert E. Franken-*

The above definition is probably no longer considered to be 'valid' by modern day philosophers and psychologists, but I find it describes the creativity-which-is-sketching very well. On the one hand we sketch because it's often easier to solve problems while visualizing them. A sketch is also a useful tool when explaining your idea to someone else. And finally, sketching is fun. If we are to abolish traditional sketching techniques, we must pay attention not to jeopardize any of these three cornerstones.

A.7 How do we solve problems while sketching?

Human creativity thrives in an underconstrained environment. Whenever a decision cannot be made on logical terms, creativity steps in to fill the gap. A design assignment is always heavily underconstrained[1] in the beginning. By sketching we create temporary constraints which allow us to explore the flexibility and quality of certain ideas. As the sketch progresses, the ideas pile up into an 'idea-pool'. Some ideas in the pool will occlude others while some will appear to be compatible. In this way we can construct groups of ideas that work well together. These different groups will eventually become separate design strains (see A.4.3 Emergent planning).

The formation of new ideas and the sketching of new ideas happens within the same time-span. The elements in the sketch must therefore be drawn with sufficient clarity to reference the ideas[2], but also with sufficient vagueness to reflect the uncertainty. This ambiguity is a key property of sketches.

1. A constraint can be anything that limits the further development of a plan. Creating constraints is our job, we always want to end up with a design that makes sense rather than a design which appears to be arbitrary.
2. According to A Framework Representing Knowledge [M. Minsky, 1974], memory and perception work by dynamically linking and unlinking preset frame constructs in a semantic fashion. This process of trial and error is likely to be the origin of creativity. If predictable frames fail to provide a satisfactory solution for a problem, unrelated frames will be tested in a hit and miss process. This, combined with the psychological school of phenomenology provides a few footholds for creativity-compatible-feedback.

A.7.1 What is visual ambiguity?

The balance between clarity and uncertainty in sketches can be explained by Gestalt psychology (GP) or "phenomenology". GP touches upon the subject of sensory recognition (purely visual in the case of sketches) and explains how certain effects can be used to increase or decrease image clarity. Sketches must have a certain polyvalence to prevent saturation of the creative mechanism. This polyvalence or ambiguity is the result of a hierarchy conflict between phenomenological sketch characteristics. GP defines five such character-principles, each of which can be used to adjust the clarity/vagueness ratio of an image:

- Proximity
- Similarity
- Closure
- Simplicity
- Figure-ground separation

A.7.1.1 Proximity

Depending on how close objects are to one another, we tend to view the ones closest to each other as a group. Proximity can thus be used to suggest a hierarchy of importance.

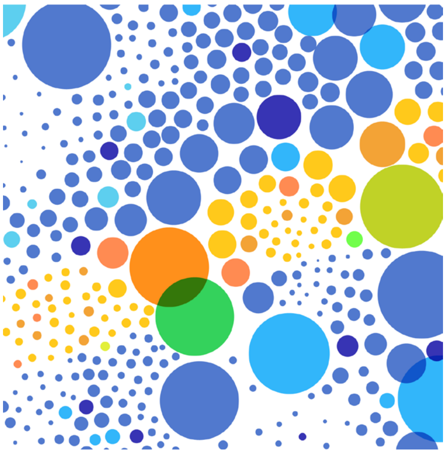
> *...appears to be confirmed by experiments conducted by Staats and Staats (1958) in which words were presented auditorially to subjects immediately after the visual presentation of a name of a nationality. The words presented auditorially had either positive or negative connotations (e.g. vacation, gift, bitter, failure). Dutch was systematically paired with positive words, Swedish with negative ones. When tested afterwards, subjects rated Dutch more positively than Swedish.*
>
> *-Mick Underwood-*

Proximity, thus, does not only apply to distance, but also to time. If we want to de-group close objects, competitive grouping properties (see upcoming paragraphs) should be enforced.

A.7.1.2 Similarity

If objects are similar in appearance to one another we tend to group them together. Again, this system can be used to invoke a sense of hierarchy and clustering. By picking colours, shapes and sizes intelligibly, a complex system can be more easily 'read'. In fact, similarity is just another implementation of proximity, where the 'distance' between objects is expressed as a custom unit (colour-hue, direction, texture, vibration-frequency etc. etc.) instead of space or time. In terms of inspiring imaging, this means we can use similarity characteristics to draw attentions to different, conflicting groupings.

The image on the right shows two competitive groupings. Do we see a curl of large dots, or a strip of yellow ones? More groupings could easily be added without changing the geometry (black outlines, transparency, feather edges). More groupings result in a higher ambiguity; a conflict of hierarchy.

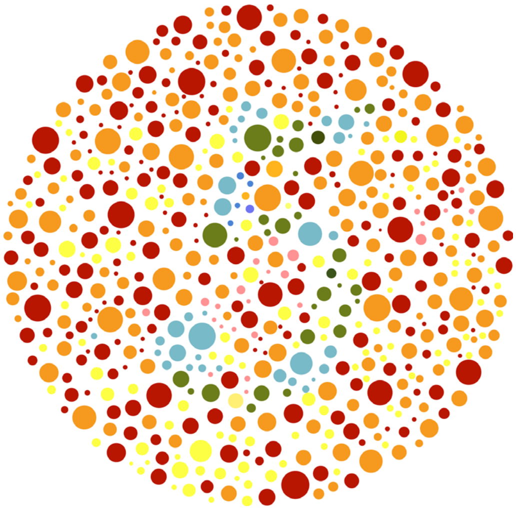


A.7.1.3 Closure

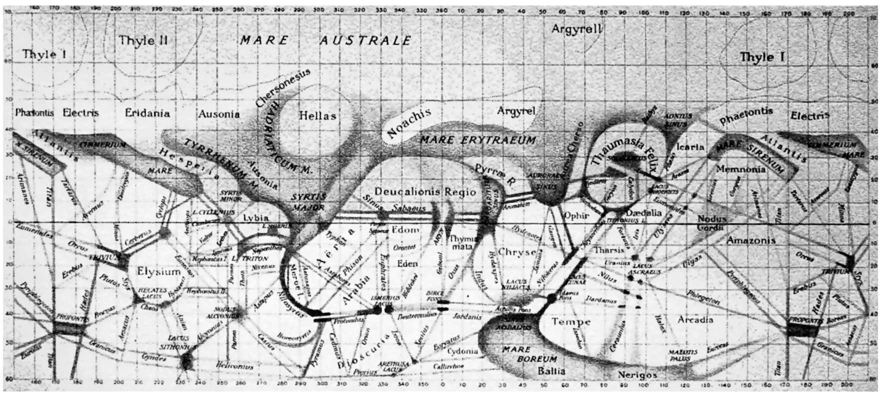
This is one of the most creative of human visual systems. Closure refers to the automatic completion of partial patterns. A famous example is shadowed text, which is -provided you're a human being- perfectly readable[1]:

Phenomenology

Biological intelligence has always been far better at recognizing (or imagining) hidden patterns than artificial intelligence. Unfortunately this ability is not unfaltering. Apart from recognizing incomplete patterns, we sometimes see things that just aren't there and sometimes we fail to spot a complete pattern.



Ishihara circle. Those of us who suffer from Red-Green colour blindness perceive a 2 which doesn't exist, and those of us without colour-blindness perceive a 5 which doesn't exist.



Drawing of the Mars Canals in 1888. In that same year Mars' two moons Phobos and Deimos were discovered which tells us something about the quality of telescopic images. In short: People just want to see structure in noisy images.

A.7.1.4 Simplicity

There is little difference between the modi operandi 'Similarity' and 'Simplicity'. Basically, 'Simplicity' is the result of grouping objects by external similarities. So instead of grouping all red objects (red is an individual property), all objects to the left of a certain line are grouped. Other simplicity characteristics are symmetry, regularity and continuation. One must realize that these are not necessarily mathematical properties. Symmetry and regularity are products of the human imagination.

1. Closure also works on non-familiar shapes. It is not bound to a familiar alphabet.

A.7.1.5 Figure/Ground separation

Figure/Ground separation is the process of mental highlighting. When examining an image, certain areas 'draw attention'. These conspicuous parts together are called the foreground. The remainder is called the background. Grouping by emphasis is important when processing image data, since the foreground layer probably tells us what the image is about. Once the framework of subject is in place, the background layer can be analysed at leisure and fitted into the foreground backbone.

There are many problems with foreground assessment, most of which occur when filtering text rather than images. Often people remember only the funny bits of a story instead of the important facts. The most famous visual example is the Rubin vase, in which one can either see a vase or two human faces. There is no clear cue on which of these is more important, thus we often switch between them, unable to decide.

A.8 The discrepancy between mind and software

It is an interesting fact that practically all software CAD packages aim to maximize image clarity. This -by definition- renders them useless for sketching. Still, we need digital means to accomplish automated sketch evaluation. Before we can expect to use digital tools in the sketching process, we must first make sure they are compatible with the human mind, not the other way around. Sketching software must adhere to the ambiguity paradigm in both in- and output.

> *...We misperceive. We misunderstand. We misremember. This, coupled with our natural tendency to see faces in oatmeal and personality in tiny electronic pets, can lead us to severely misshapen ideas about what's actually going on [in the computer]...*
>
> *-Daniel Rutter, Atomic: Maximum Power Computing-*

A.9 Problems with development

It was clear from the beginning that the development of software that incorporated the ideas for sketch evaluation would be a very big task, probably too big for a single thesis project. Therefore it was decided to develop a proof-of-concept application which focused on workability rather than usability. The sole purpose of this application was to test the algorithms.

The decision was made eventually to write the application as a plugin to an existing CAD program in order to cut overhead development. All the images and models that are shown in this paper were made using tools that are part of the plugin, not the native CAD platform.

As the project -and the software- progressed, it became clear that some initial ideas were faulty or extremely difficult to integrate. At the same time other ideas came into being as a result of finished routines. The application is not to be treated as an ad hoc attempt to test the project paradigms. Rather, the software *is* the project. It represents by far the largest amount of investment of any kind. But there is a problem with evaluating software as opposed to evaluating a written paper or a graphical design. Small algorithms pose no problem, but the plugin as a whole contains many thousands lines of code and many dozens of classes, written in a programming language which none of my teachers can read[1]. The output can be criticized, but not the executable.

1. The plugin was written in VB.NET version 2003 for the DotNET 1.1 framework. Some of my teachers do have experience with other programming languages such as VB6, java and C++.

B. Solutions

Up until here the story has been fairly straightforward and you probably -hopefully- agree with me on the majority of issues. Or if not, then I hope you have at least good reasons to disagree. Now however, we have reached the end of my criticism on the practise of contemporary urbanism and the following section -handsomely entitled “Solutions”- is where I propose a few answers to the questions raised in “Problems”. All my answers are based on a new method of drawing spatial designs in general and spatial sketches in particular. “New” in this case does not mean most-recent but fairly-recent. The underlying theorem of Proximity sketching was developed by T.M. de Jong even before I went to college. This project features -to the extend of my knowledge- the first digital implementation of this paradigm, which in turn opened the door to sketch-analysis.

This new digital method, which I have christened **Proximity sketching**, solves a bunch of scientific issues with sketch evaluation. That is not the difficult part. The difficult part is to make the evaluation scientific while keeping the sketching process responsive, interactive and unrestricting.

B.1 The new sketching

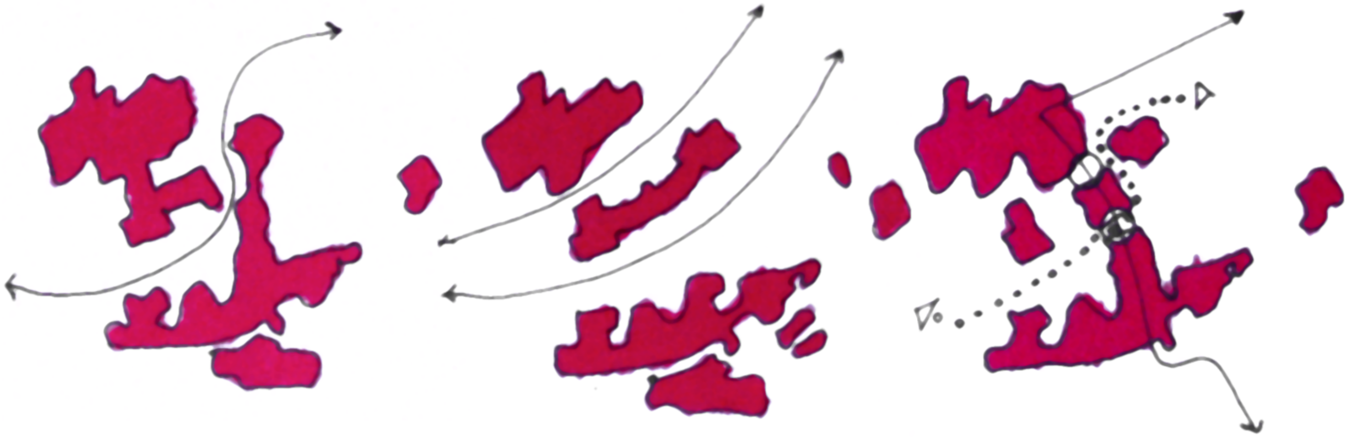
Regular sketching is a typical reduction process. An area with 1.000 homes is reduced to a greasy spot. An ecology ribbon is reduced to a hatched bar. Lines and shapes such as you would find in a regional sketch design suggests a degree of sharpness and contrast that just does not exist in reality, or in the mind of the planner for that matter. There is often no reason for specific shapes which are drawn in sketches. All they are meant to convey is a general feeling: “Something about this big, about there, oriented roughly along the river...”. In the chapter on terminology, I’ve made a distinction between geometrical and spatial planning. One of the major differences was the lack of form in spatial planning. From this flows forth the irrationality of sketching on paper since it is nearly impossible to draw anything using paper and pencil that has no shape. Luckily, the ‘shapes’ we see in sketch drawing are a result of the reduction process. So once we find a way of making unreduced drawings, we solve both issues at once.

In the end, everything comes down to spread. Not just on the large scale level. Lines and shapes only exist because we imagine them. Even a pencil stroke representing a road is merely a spread of carbon particles representing a spread of asphalt grit. Such a reductio ad absurdum will not do us any good in reality, but the point remains valid; spread is at the basis of all reality [T.M. de Jong, 2002].

B.1.1 Sketch characteristics

Since sketches are the first illustrational stage in the design process, it is likely that we reach it within days or maybe even hours after reading the project assignment and programme[1]. At this point we thus know very little about the project, and we’re probably only focused on the major programme issues. I.e. early sketches use severely simplified legends, sometimes even binary legends (new housing area vs. no new housing area). At this point there will be only a thin line between a sketch design and a concept.

1. That is if we ignore the analysis stage. Sometimes extensive research is performed prior to the design process. During this research there is of course plenty of time for ideas to ferment in the brain.



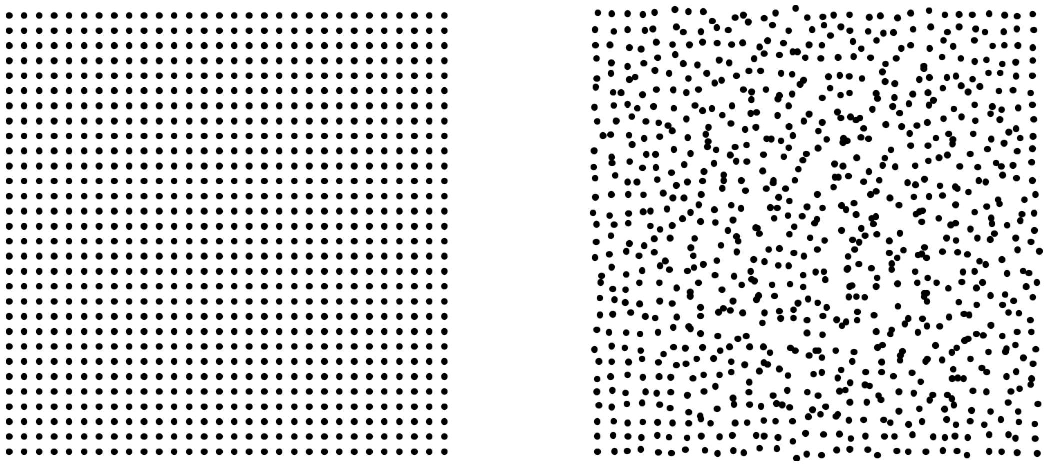
Design concept sketches with binary legends and symbols. Such drawings can be encountered both during early design stages and analytical stages.

Then, as the design progresses, accuracy and legend units are added. Most of these additions are a mixture of design and necessity; "for every N households, R shops are needed". But the precise location of the shop is up to the planner. It is important that we divide the properties of a design into 'aim' and 'result' heaps. A clear hierarchy of functions is needed if drawings are to be elaborated instead of redrawn at each stage. Without this hierarchy the planning process would quickly descend into chaos. (The problem is that this hierarchy may not exist in such a profound clarity in reality. In that case we'll have to invent it and use an iterative method to make it all fit together in the end.)

But the increase in legend units does not mean we have to alter the paradigm of spatial planning. Regardless of scale and accuracy, spatial planning is never about form. The only direct property is spread. From spread a myriad of related properties can be deduced, some of which I will elaborate on in the upcoming chapters.

B.1.2 Reality approximation

The main purpose of Proximity sketching is evaluation. In order to make a sketch measurable it cannot remain an artistic jumble of lines and spots; somehow the drawing must be translated into an approximation of reality. Proximity sketching solves this problem by using the one and only property of reality (again: spread) as the sketching medium. Thus Proximity sketching is *always* an approximation of reality. No translations to and from the model are required.

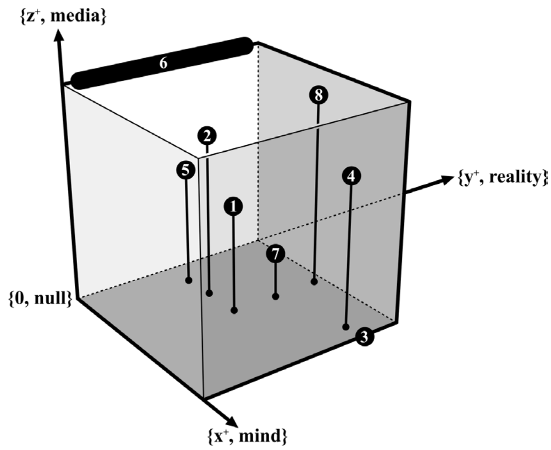


The spreads above cover the same area with the same density, yet they represent wholly different realities. This distinction cannot easily be mimicked using traditional sketching techniques.

B.1.3 Method classification

According to Stellingwerff [M. C. Stellingwerff, 2005] design methods can be classified in a three-dimensional parameter space. The axes of the space indicate the relative position along Mind, Reality and Media. Classical design is used as a reference coordinate, around which other methods are inserted:

1. Classical design (traditional media)
2. Virtual reality (traditional CAD)
3. Contemplative/philosophical
4. Augmented interaction
5. Maximization method
6. Simulation (software)
7. Proximity sketching (sand)
8. Proximity sketching (digital)



Classification of methodology according to Stellingwerff.

Mind indicates the amount of thought that goes into the design. Simulation is a purely digital process and thus has no mind-component[1]. The Maximization method also has a low mind value, since it is mostly a process of averaging and not problem-solving. Augmented interaction is -in some ways- fairly similar to Proximity sketching. For small scale projects it might even be completely coincident. On the large scale however, 'reality' and 'reality-as-we-experience-it' is no longer the same thing. To place Proximity sketching on the same mind-component as Classical design, may be a bit overzealous. Ideally, Proximity sketching should not limit the mind any more than Classical design does, but the lack of symbolism in Proximity sketching may prove to be an obstacle.

1. Artificial intelligence is grouped under Media and not Mind.

B.2 Software implementation

A platform which supports Proximity sketching must be capable of simulating both very simple and very complex plans. The design often begins with binary legends and then slowly progresses towards the final plan by adding legend units and detail. But all detail and legend types manifest themselves through the property of spread and thus the system will require no knowledge of geometry[1].

The following chapter discusses an implementation of the Proximity paradigm I have written for purposes of experimentation. This is by no means the only, nor the best implementation imaginable. I have explored the algorithmic key components of the platform: the system, the sketching tools and the analysis tools.

B.2.1 Agents and Particles

The smallest unit of simulation consists of a location C (coordinate) accompanied by a value T (character or type). Every unit acts as a single object and can be used to represent whatever is needed; a person, a home, a parking place, a postal box, a tree, a PT-stop... etc. Depending on implementation, these units are either referred to as 'particles' or 'agents'. Mindless units that exhibit no 'conscious' behaviour are called particles. They can be likened to sand-grains which are at the whim of wind, water and gravity.

Agents on the other hand have in-build behavioural rules, which means they respond to their environment. The most common analogy is that of ants. Ants are not intelligent beings; their behaviour is preprogrammed by genetics, yet there appears to be emergent intelligent behaviour when we look at an entire colony. When particles are equipped with a behavioural ruleset, they start to act for themselves and they become agents. The ruleset can be immensely complex in order to simulate settling behaviour in human beings, or it could be a single rule causing all agents to flock downhill[2].

Particles are stored in lists called "clouds" and since they will be numerous throughout the model, it pays to make them optimized. I have therefore moved the character property of particles from the particle class to the cloud class, where it has to occur only once. This inadvertently means that every cloud can only contain one type of particle. Thus the only remaining unique property of a particle is it's location C. C can be accessed both by the designer and by the system. It is by changing particle locations and amounts that we sketch in the computer. This is the only purely manual tool Proximity sketching provides. It's counterparts in conventional sketching are pencil and eraser. With the salient difference that a line drawn on paper cannot be adjusted, it must instead be erased and redrawn.

A particle cloud.

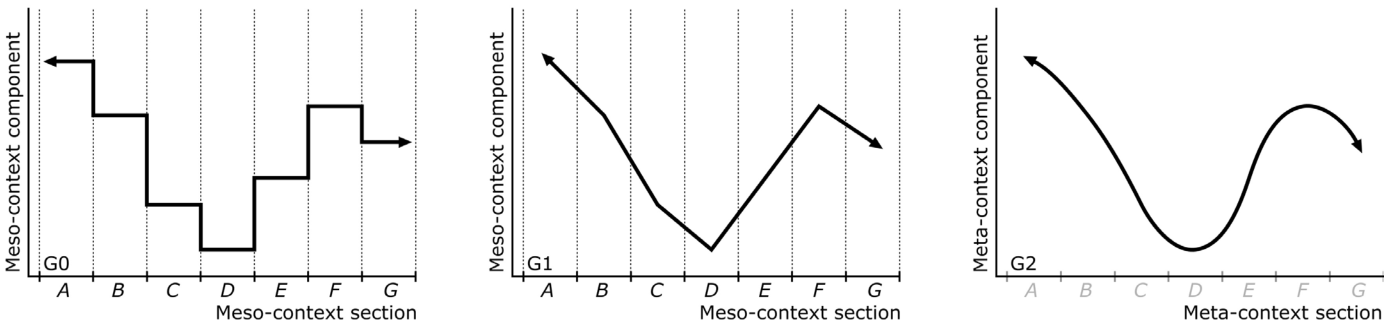
1. Although shape and geometry is not part of the sketching process, the analysis algorithms do require a solid understanding of geometry and topology. The designer is typically not confronted with these mathematical elaborations.
2. Apart from the programmatical difficulties of implementing agents, it also limits to a great extent the flexibility of the simulation and it poses a great impact on system resources, making the whole simulation run much slower. Agents are more suited to solve specific problems. This implementation of Proximity sketching aims at a high flexibility and responsiveness and thus uses the particle approach.

B.2.2 Context

Apart from being responsive to the designer, a particle cloud can at the same time be responsive towards it's environment. Every design arises in a context, and this context imposes certain limitations and suggestions onto the design. Existing roads or rivers, the very shape of the site, existent housing etc. Thus, even though particles will have no mind of their own, they can be affected by the complex of context and are thus not completely under direct control of the planner. This context must also be inputted by the designer, and thus functions as a delayed drawing tool. The context in Proximity sketching can be compared by the paper in regular sketching. It can be blank, it can have an impression of another design on it, or perhaps the map of the actual site.

A context provides limitations and suggestions. These are known respectively as hard and soft constraints. A hard constraint might for example be the boundary of the site; whatever the planner does, he is not allowed to cross this border. A soft constraint might be the attractive affect a certain location has on particles. Particles could be programmed to move towards a water-front for example, even though the designer can locally force them into another direction. The complex of context is potentially a vast and diverse collection of constraints, each of which affects a particle at every stage of the simulation. Such an iterative process would quickly flood the computer resources, making the system sluggish and unresponsive.

A solution is to place all the constraint generators into a meta-context, which does not immediately affect the particle clouds. Instead they only affect a meso-context layer which stores a two-dimensional grid of forces, each of which is the sum-force of all the constraints in the meta-context. This means that every particle only has to calculate its location in the meso-context and to retrieve the force vector in question. The use of a meta/meso context removes an entire iteration from the simulation, but unfortunately also reduces the accuracy of the context. By treating the context as a gridded instead of a parametric space all the coordinates that lie inside a meso-cell, will be affected in the same way. Because there are sudden jumps of behaviour on the transitions between meso-cells, this is a discontinuous system (G_0). Still, by sampling the particle coordinate not with a 'nearest-neighbour' filter but with a more advanced 'linear' or 'cubic interpolation' filter, the discrepancy can be reduced to a G_1 continuity. The original parametric space is designated G_2 since it contains only smooth iso-force curves.

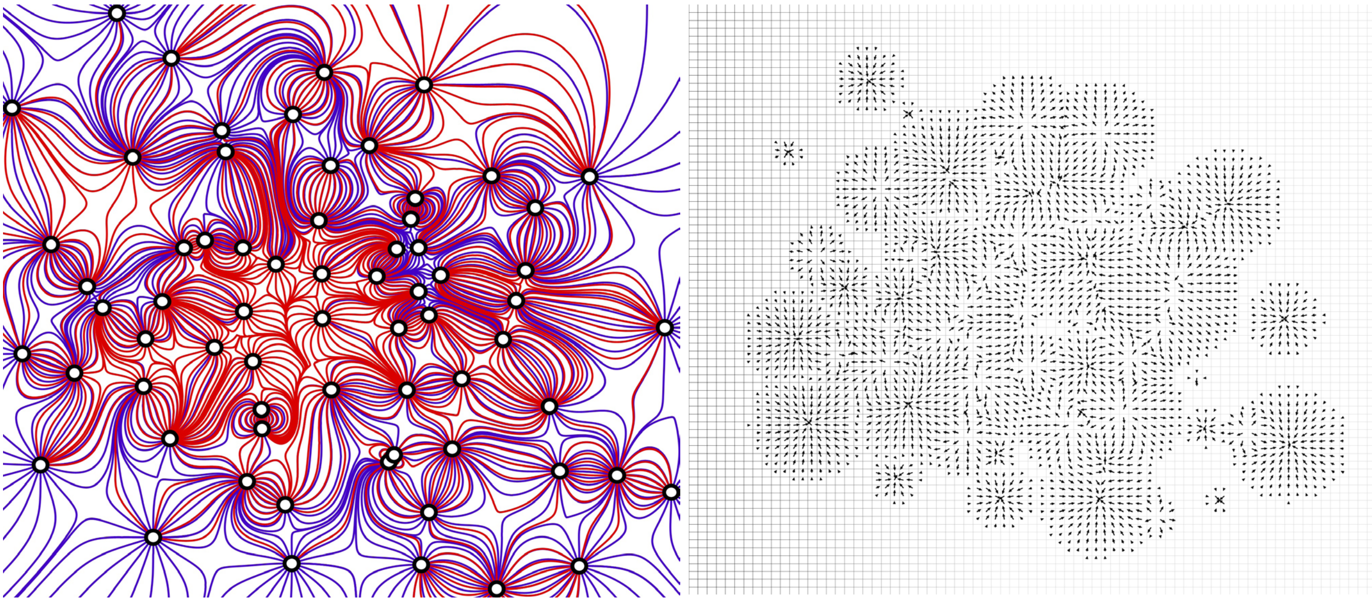


In the above graphs you see three identical sections though the same context. The left most section shows the discontinuous meso-context, the middle section uses linear interpolation to create a continuous sampling space. The right most section is a direct representation of the parametric, completely continuous meta-context. The meta-context does not use cells since the values can be computed on every possible location with any desired accuracy:

$$v = \overrightarrow{PP'}$$

$$F = \sum_{k \in \mathbb{C}} f\left(\frac{P - k}{|\boldsymbol{v}|} \cdot \frac{k_S}{|\boldsymbol{v}|^2}\right)$$

Where P is the particle coordinate, P' is the projection of P onto constraint k, v is the vector from P to P', $\boldsymbol{F}$ is the resulting force of the complex of context on point $\boldsymbol{P}$, $\boldsymbol{k}$ represents any constraint out of the set of all constraints $\mathbb{C}$ and $\boldsymbol{k_S}$ is the strength of constraint $\boldsymbol{k}$.



Metacontext vs. Mesocontext. The parametric metacontext (on the left) consists of 67 soft constraint of two types each with a different strength and radius. The visible actor-potential curves are but a cross-section of the whole parametric space. The resulting mesocontext on the right is a simple grid where each cell contains the sum effect of all actors. The resulting diagram is not reduced.

B.2.3 Actors

The context is a static construct, containing only pre-existing constraints. Since we're treating particles and the context layers as inactive objects, it is not possible to perform a simulation. In order to overcome this limitation a new class of objects needs to be introduced to the platform. Actors will only be needed for simulation purposes where the direct, manual intervention of the designer is no longer the most important factor. Actor classes can reside on all levels and all layers of the platform. Some actors might influence the context, others might affect particles directly. At this time however, the platform is aimed at the sketch phase and will therefore not feature actor classes.

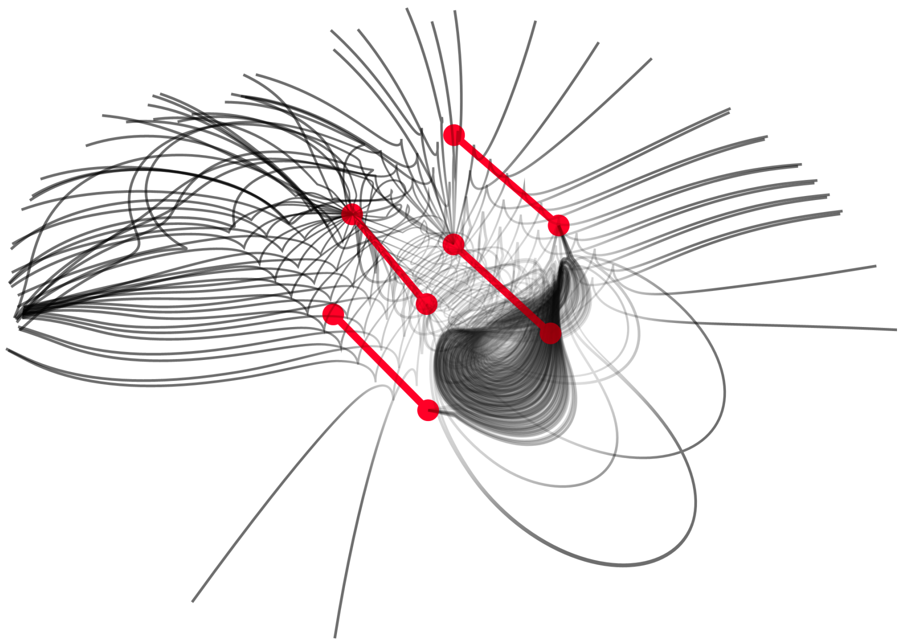
B.2.4 Controls

Proximity sketching has to be an interactive and responsive process. The development of 'drawing' tools is of paramount importance. The user must be able to manipulate the particles and the context constraints as intuitively as possible. To this end I have chosen to implement the Proximity platform into an existing CAD application, this does not in any way affect the theory or Proximity sketching.

B.2.4.1 Manipulating constraints

Before sketching will take place, it will probably be necessary to set up the contextual constraints. These will consist of standard point, curve and region objects each representing borders or punctures in the meta-context. The meso-context only exists in the computer memory for the duration of the sketching process and the user is never directly confronted with it.

Constraints are potentially diverse objects with both geometry and an attribute database. Contextual objects are not limited to the same degree of reduction as the particle(cloud)s and can thus be manifested through a wide array of geometry types; points, curves, regions, even scalar fields. In the image below, a very simple collection of four actors (the red segments), each of them with a very simple rule set (attract/repel according to inverse square of distance) already results in a very complex meta-context (the black curves). The complexity of the meta-context will increase drastically when more constraints are added, the meso-context will always be simply a grid of vectors.

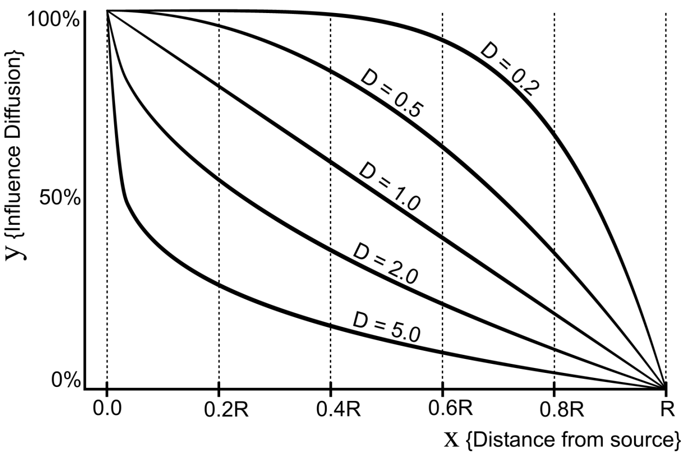


B.2.4.2 Manipulating particles

Particle manipulation is at the very core of the proximity paradigm. The user must be able to append, remove and adjust particles at will, using both freehand and exact methods. I have devised a set of tools to this end, most of which are based on the 'Influence Diffusion' equation. Influence Diffusion (ID) is based on the natural decay of influence from a force. It is used to compute the left-over influence at a certain distance from the source. It uses three variables which can be combined to simulate many different types of decay (linear, square, inverse square et.). The equation for ID is:

$$I_P = F_S \cdot \left(1 - \left(\frac{|S-P|}{r_S}\right)\right)^{D_S}$$

Where I_p is the left-over influence (factor value; 0.0 <= I_p <= 1.0), F_s is the strength of the source, $|S\text{-}P|$ is the distance between the source and the particle, r_s is the radius of influence as specified by S and D_s is the decay constant as specified by the source. If $|S\text{-}P|$ is larger than r_s, then this equation is circumvented and the left-over influence is zero. D_s can be any value in $\mathbb{R}^+$ and it controls the diffusion falloff. Basically ID means that each source has an area of influence around it, in which particles will be affected by the source. Within this (circular) area the amount of influence the source conveys depends on the decay factor and the distance from the particle to the source. However, the amount of influence always decreases as $|S\text{-}P|$ increases and the amount always approaches zero near the outer boundary of the area.



By multiplying every change to the total model with an ID filter, global deformations will be localized in an intuitive fashion. Just as we expect a spray-can to only cover the nearby part of the wall with paint instead of the whole wall.

B.2.4.2.1 Adding particles

Particles must first be inserted before they can be manipulated. This can be done using native CAD tools such as manually placing individual points or adding rectangular pointgrids, but such an approach does not fit the Proximity paradigm very well. I have written two tools for adding particles which adhere to the sketching principle.

When we abandon the sketching component of Proximity, we should be able to completely automate the insertion process. This comes very close to simulation or even artificial intelligence and is not the subject of this study. However, a very 'dumb' insertion algorithm which still requires a human component can be grouped under 'sketching tools'.

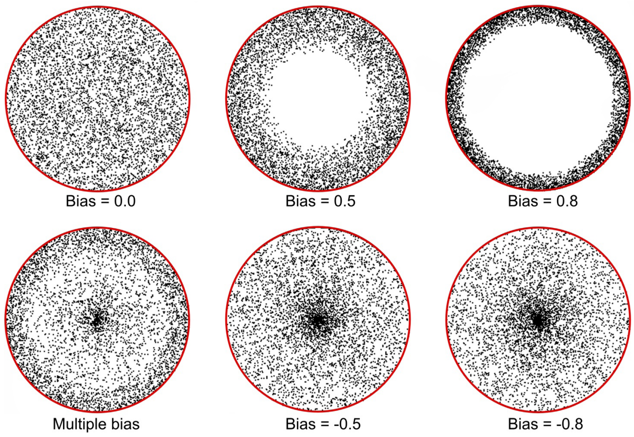
B.2.4.2.1.1 Automatic addition

Automatically adding particles is useful in situations where a region contains a known density, or when the user wants to distribute particles in a balanced way. This tool requires the existence of a closed region ***R*** into which the particles will be inserted and two variables ***N*** (amount, can be written as density, since the surface area of the region is a given) and ***B*** (bias). Particles are then inserted randomly into the bounding rectangle of ***R***. If a particle is evaluated to be on or inside the space bounded by ***R***, it will be added to a cloud. The iteration completes when the list contains ***N*** particles.

In addition, for every particle ***P*** in the list the projection ***P'*** of ***P*** onto ***R*** is computed. A vector ***T*** [***P P'***] is then multiplied by ***B*** and added to the coordinates of ***P*** in order to bias every particle towards or away from the edges of ***R***:

$$P_{new} = P_{old} + B \bullet \overrightarrow{P_{old}\, P'_{old}}$$

By changing the bias value, a more natural distribution of content can be accomplished:



This method fails to produce proper results on non-convex regions. When complex bounded regions are involved a better method of automatically adding particles is to treat the boundary as a repulsing/attracting context object (see [2.1.2 Context]). A temporary meso-context can then be constructed which will affect particles during placement.

Another method of automatically adding particles involves the use of particle generators. These are objects that create (or destroy) particles in their vicinity. A city centre for example tends to generate dwellings, while an industrial zone repels the inhabitants. This approach would be suited for a simulator but not for a sketch evaluator, which is why I have not implemented it.

B.2.4.2.1.2 Manual addition

Manual addition of particles is similar to using a paint spray can. A preset amount of particles ***N*** is added randomly around the source ***S*** using an adapted version of ID where the F_s factor is replaced by ***N***. If the spray nozzle shape is circular then the position of every added particle ***P*** is computed by:

$$P_x = S_x + \left(r_S \bullet K_{Rnd}^{D_S}\right) \bullet \cos\left(A_{Rnd}\right)$$

$$P_y = S_y + \left(r_S \bullet K_{Rnd}^{D_S}\right) \bullet \sin\left(A_{Rnd}\right)$$

Where P_x and P_y are the x and y coordinates of P respectively, S_x and S_y are the x and y coordinates of S respectively, r_S is the radius as specified by S, K_{Rnd} is a random value ($0.0 <= K_{Rnd} < 1.0$) which is computed in place and thus differs for all components, A_{Rnd} is a random value ($0.0 <= A_{Rnd} < 2\pi$) which is computed in advance and is thus identical for all components.

For non-circular sprays these functions do not apply. I have added support for square and cross sprays but not for freeform outlines which are often found in pixel editors. Optionally a closed region can be specified to mask the sprayable area.

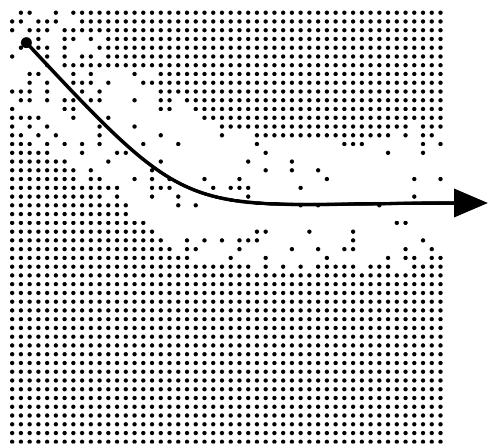
B.2.4.2.2 Adjusting particles

Once one or multiple particle clouds have been added to the model, the user must adjust them in order to generate multiple design strains (see [A.4.3 Emergent planning]). Typically, all variations should contain the same amount of particles. The only property which is allowed to differ is the spread of the particles. Comparison between two models that are not topologically identical[1] is not possible. However, sometimes removing a subset of particles and adding them in a different location may be less work that deforming the entire set.

A collection of tools has been written to cater for the above stated needs. This is a relatively small collection featuring only simple controls, but for the purpose of experimentation they will suffice. Also note that all the tools described in this paragraph branch, are subject to the ID layer as described in paragraph [B.2.4.2 Manipulating particles] and the formulas stated here thus need to be multiplied by the corresponding ID factor M_f. The equation for M_f has been omitted since it is always identical.

B.2.4.2.2.1 Removing particles

Particle removal is a fairly straightforward algorithm, which simply assigns a random value V to every particle P which resides within the active area of the removal tool. V is then compared with a threshold value T which is based on the distance from P to the centre of the active area and the preset strength of the removal tool. If $V < T$ then the particle is removed from the set.



Deleting particles is a useful tool since it allows for the creation of strong contrast and sharp corners in a particle cloud. This is something which is hard to accomplish with other tools.
However, global reduction might also be a useful feature. A simple random reduction algorithm[2] has been implemented in the platform. Other -possibly useful- algorithms that have not been implemented are:

- Delete by distance to actor
- Delete by distance to nearest neighbour
- Delete by local density value

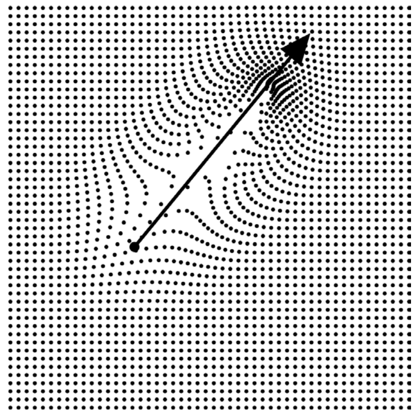
1. The only topological constraint for particle clouds in a 2D space, is the amount of particles. Still, it is possible to deform a spread in accordance with topological laws, while breaking others. Affine and homeomorphic deformations impose more restrictions on spread changes.
2. The random reduction algorithm implemented here is a simple sort and cull method. First, the particle cloud is sorted using an array of random floating point values. Subsequently the array is redimmed to match the specified size.

B.2.4.2.2.2 Drag

The most basic method of deformation is translation. In the Drag tool a particle P is moved along a vector which is parallel to the mouse trail vector T_0T_1. The distance moved is a percentage of the length of T_0T_1 depending on M_f:

$$P_{new} = P_{old} + \left(M_f \bullet \overrightarrow{T_0T_1}\right)$$

Where P_{new} is the new location of P and P_{old} is the current location of P. Dragging particles is the only undoable tool in the collection. I.e. a drag from point A to point B will be completely undone by a drag back onto A. This behaviour makes the drag tool especially suited for accurate and subtle deformations. Continued dragging is not going to cumulatively deform the particle cloud.

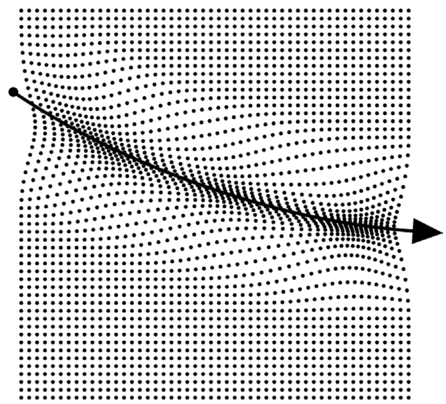


B.2.4.2.2.3 Flock/Scare

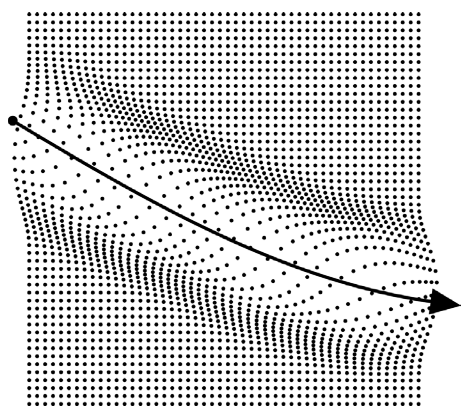
Flocking is the behaviour where particles congregate around the mouse trail T_0T_1. Every particle P is attracted to its projection P' onto T_0T_1:

$$P_{new} = P_{old} + \left(M_f \bullet \overrightarrow{PP'}\right)$$

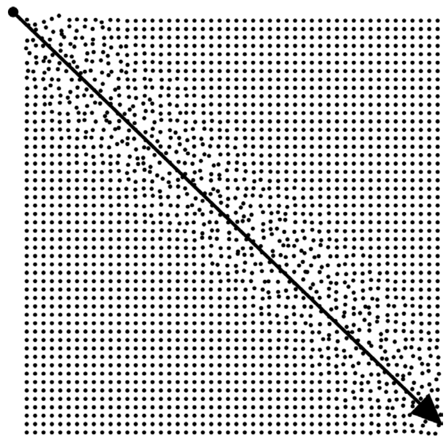
If $|PP'|$ is very small, i.e. if P is near the mouse trail, chances are P will overshoot and turn up on the other size of T_0T_1, perhaps even beyond the reach of the flock tool if the translation distance exceeds the preset radius. By limiting the translation of P to its initial distance to T_0T_1 this can be avoided. In the stated equation C represents a constant value. For particle motion towards the mouse trail C needs to be a positive number, for motion away from the mouse trail C needs to be negative. Alternatively the vector PP' can be inverted to avoid a multiplication.



Flocking can be used to intensify certain areas or to move particles away from an adjacent area. Especially since the direction of motion can be inverted flocking gives great control over particle clustering. Flocking tends to smooth particle clouds and is therefore less usable for subtle touch-ups.



B.2.4.2.2.4 Panic/Run



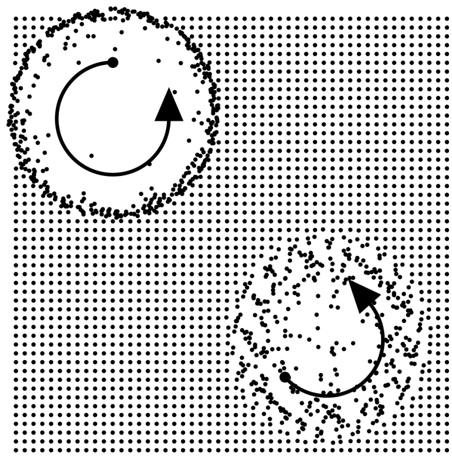
Flocking also tends to create extreme high densities, either by collecting particles on the mouse trail, or by pushing them towards the boundary of the active region. This is often an unwanted property and some tools are needed to disperse high density particle clusters. Dispersing multiple particles that share the same coordinates (or nearly share), cannot be done with a homeomorphic deformation; a random element must be introduced into the equation. The easiest solution is to translate every particle a random distance in each direction, where the translation distance depends on M_f. This behaviour can be compared to panicking and it will very quickly shuffle a particle set. But its measurable effect also decreases very rapidly, since shuffling an already shuffled particle cloud will not increase the randomness.

Panicking can be used to increase the entropy of a particle cloud, but it is of no use in a random set. Another approach is to translate every particle along a fixed random vector:

$$D_i = \left(\mathbf{X}_{\text{Rnd}}, \mathbf{Y}_{\text{Rnd}}, \ldots, \mathbf{N}_{\text{Rnd}}\right)$$

$$P_{\text{new}} = P_{\text{old}} + \left(M_{\text{f}} \bullet \overrightarrow{\mathbf{W}_0\mathbf{D}_i}\right)$$

Where D_i represents the head of a fixed random vector, W_0 represents the world coordinate system origin and is thus a constant. D_i is computed by using a pseudo-random generator which always starts at the same seed value[1]. This seed value must be unique for every particle in the cloud. The particle index into the cloud list would therefore be an ideal candidate. This running behaviour has the interesting peculiarity that at first it resembles a regular random panic but then as the particle motion vectors start to point away from the mouse trail, it moves towards an inverse flocking deformation.



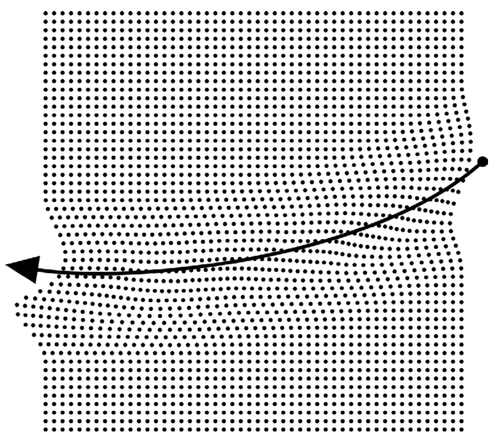
Another effective way to disperse clusters of high density particles, would be to treat the whole particle cloud as a charged particle system, wherein every individual particle repels every other particle. Such an approach requires an additional iteration per mouse-movement.

1. A computer is a logical machine that cannot be used to generate truly random numbers. Instead computers can generate pseudo-random numbers. Pseudo random numbers always occur in series where every new number depends only on the previous one. The first number in a series depends on a seed value. Thus, identical seed values always result in identical series.

B.2.4.2.2.5 Rotate/Twirl

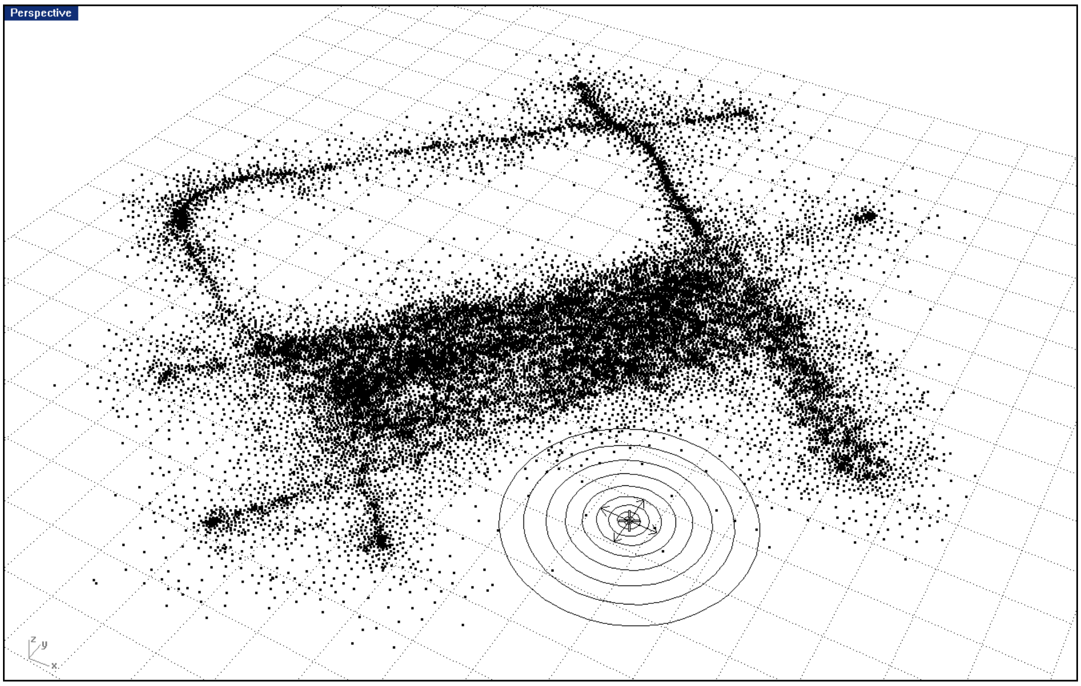
Twirling is probably not a very practical deformation tool. All the previous tools are either random or colinear deformations; particle motion vectors always point in the same way. Twirling makes a particle P rotate around the projection P' of P onto T_0T_1:

$$P_x = P'_x + \left(\left(P_x - P'_x\right) \cdot \cos\left(M_f\right) + \left(P_y - P'_y\right) \cdot \sin\left(M_f\right)\right)$$
$$P_y = P'_y + \left(-\left(P_x - P'_x\right) \cdot \sin\left(M_f\right) + \left(P_y - P'_y\right) \cdot \cos\left(M_f\right)\right)$$



B.2.4.2.2.6 Further manipulative tools

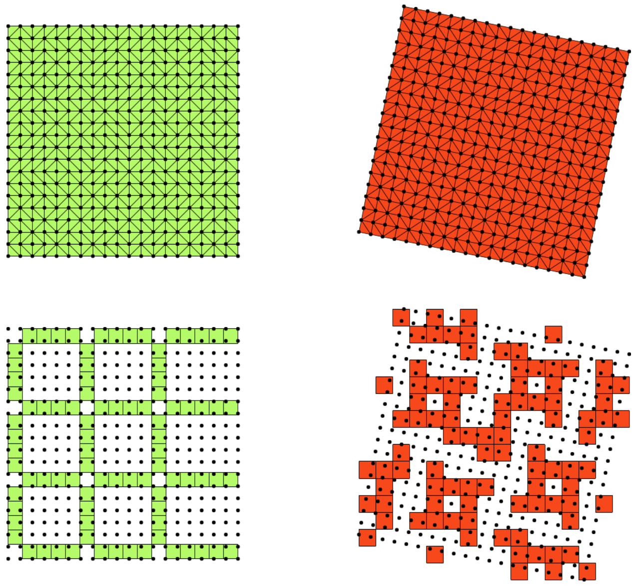
The tools discussed here are just a sampling of useful manipulations. In order to gain the same amount of control over digital particles as one does have over tubes of graphite crystals wrapped in wood, more tools are needed. Not simply other ways of deforming a particle cloud, but mostly much smarter ways. Particles should not be allowed to flock together indefinitely, particles should not be allowed to drift apart indefinitely, particles should not be allowed to approach some object or other etc. etc. As the tool becomes more intricate, it will be harder and harder to classify it; at what point does artificial intelligence take over?



A screenshot of the software in action. The particle cloud is superimposed on a hectare-grid, and a scare-tool is active. The circles around the mouse pointer indicate the radius and the influence diffusion decay of the active tool.

B.2.5 Particle cloud derivatives

A particle cloud is nothing more than a list of coordinates, essentially unrelated points that only make sense when seen as a whole. In other words, a particle cloud consists solely of unrelated, meaningless objects. In order to tell anything about the properties of particle clouds, we must first translate them into something measurable. These entities are called derivatives. Ideally this translation results in an **unbiased representation**, because that will ensure the scientific validity of the analysis. Biased representations carry within them a preset choice made by the programmer, and they do not guarantee identical outcomes of homeomorphic[1] particle clouds. Any linear transformation applied to the particle cloud, must be matched by the derivative constructor. This is more easily explained with a diagram:



Biased vs. unbiased derivatives. The Delaunay derivative (upper images) adopts affine transformations of the particle cloud (a simple rotation in this case) while the Quad Tree derivative changes drastically.

As you can see the manifestation on the left looks different after a linear scaling transformation was applied to the particle cloud. This happened because a grid must have a certain direction, size and offset. These properties are not a result of the particle cloud, they are preset by the programmer. Delaunay triangulations do not use preset values and thus we can rotate, move and scale the particle cloud at will without distorting the resulting diagram.

Delaunay meshes are not the only unbiased manifestations of particle clouds. I have added five different types of derivatives to this implementation of Proximity, each of which allows us to analyze different properties of a particle clouds spread. Non-homeomorphic derivatives are italicized:

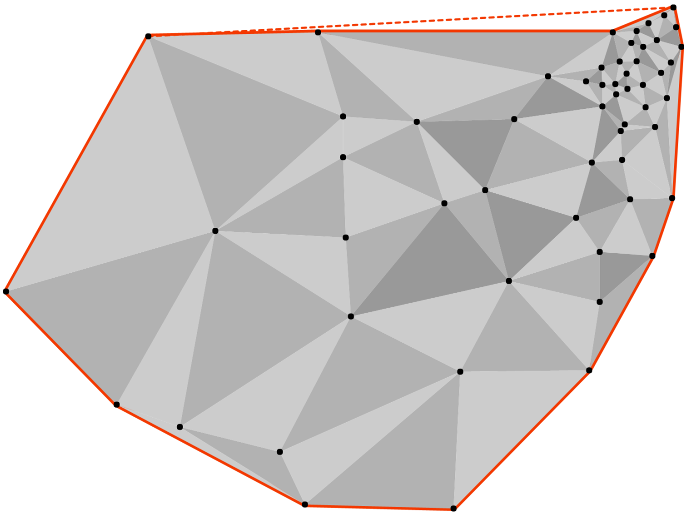
- Delaunay triangulation
- Voronoi diagrams
- *Search grids*
- Connectivity network
- *Quad Trees*

Ideally, many more derivatives should be available in order to maximize the choice when measuring certain properties (see [B.2.6 Derived properties]).

1. Homeomorphism: Possessing similarity of form [Eric W. Weisstein, 2006]. Linear (affine) transformations of geometry maintain homeomorphism. The transformation can occur in higher dimensions than the geometry occupies, i.e. mirror images are homeomorphic.

B.2.5.1 Delaunay triangulations

> *In mathematics, and computational geometry, the Delaunay triangulation (...) for a set **P** of points in the plane is the triangulation **DT(P)** of **P** such that no point in P is inside the circumcircle of any triangle in **DT(P)**. Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation...*
>
> *-Wikipedia, on-line encyclopedia-*



A typical Delaunay mesh. The red dotted line indicates the discrepancy between the Delaunay border and the convex hull.

The delaunay triangulation of a pointset in a two-dimensional space is a collection of triangles without gaps that connect all points (and only all points) in such a way that the circumcircle of every triangle does not contain other points. The total delaunay triangulation covers a finite area which is approximately identical to the convex hull of the pointset.[1]

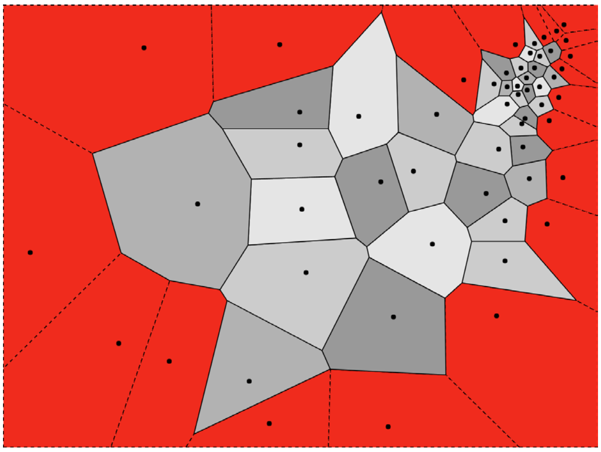
In urbanistic terms this means that the triangles (or 'faces') of the delaunay triangulation connect direct neighbours. The relationships between adjacent points are limited to the first neighbour in a general direction. These properties make delaunay triangulations very suitable to represent the social aspects of a particle cloud since every face represents the potentially shared space between three neighbours. This is not an exact measurement since the sum total of all faces equals the area of the entire particle cloud and we also need area for private and public functions. But there exists a strong relationship between the delaunay faces and the semi-private characteristics of the spread.

We can extract more information from the delaunay triangulation than just the faces. Every triangulation is a mesh instance which means it contains vertices (the coordinates of the original particles), edges (connections between direct neighbours) and faces (areas enclosed by three adjacent edges). The collection of all edges provides a simplistic network model which can be used to both search and adjust the original particle cloud. For example, the edges can be used to equalize average neighbour distance, resulting in a more balanced (but less diverse) spread. A delaunay triangulation is a topological representation of a particle spread. It does not change with scale and orientation. This property makes it exceptionally well suited for unambiguous analysis.

1. Here I have implemented the delaunay algorithm as described by Paul Bourke et. al. First a super-triangle is added which encompasses all points in the cloud. Then, for every point ***P*** in the cloud, all triangles are located that contain that point within their circumcircle. These triangles (set ***L***) are removed from the mesh and new triangles are drawn from the active point to all the vertices on the outer boundary of ***L***. Finally, all triangles that contain vertices coincident with the initial super-triangle are removed. To speed up the search, the initial particle cloud is sorted by ascending x-coordinates. Every 110 iterations (trial and error optimization) all triangles to the left of P are moved into a static array as to avoid redundant inclusion tests.
This algorithm cannot deal with organized input very well, so points are shuffled a small amount prior to computation. These translations are undone again upon completion.

B.2.5.2 Voronoi diagrams

*In mathematics, and computation geometry, a Voronoi diagram for a set **P** of points in the space W_0 is a collection of convex cells which divide the infinite volume of W_0 into non-overlapping regions where all the coordinates contained in cell C_i are closer to P_i than to any other point in **P**.*



A typical Voronoi diagram. Cells that are along the preset edge of the diagram are coloured red.

A Voronoi diagram of a pointset in a two-dimensional space is a collection of convex cells without gaps that divide the infinite XY-plane into distinct regions. Every cell **C** is linked to a particle **P** and contains all the coordinates that are closest to **P**. Many cells along the edges of the diagram will have an infinite area and will thus be ignored. Instead of the infinite world XY-plane, the bounding box of the original particle cloud is used to clip the Voronoi cells.[1]

Voronoi diagrams provide a very natural way of partitioning space and they allow us to examine many particle properties. A Voronoi cell can be thought of as a localization of immediate context. By connecting cells that meet certain characteristics, clusters and backbones start to emerge. Voronoi diagrams can also be used to find the largest empty circle in a particle set (this is known as the "toxic waste problem").

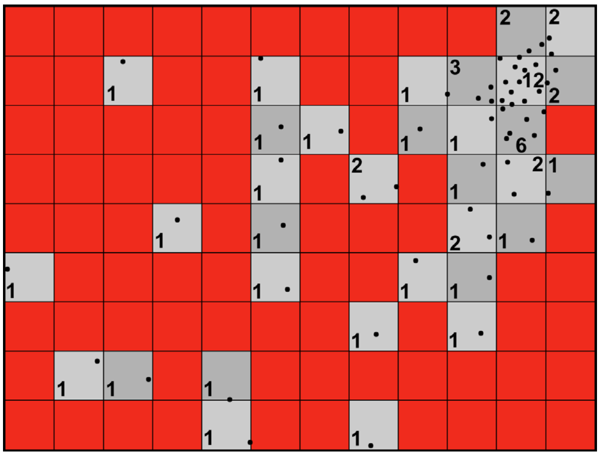
Voronoi diagrams come in several flavours, the definition above describing only one of them. A fairly complete cross section of available Voronoi algorithms[2] is provided by Takashi Ohyama [Ohyama 2002]:

- Higher Order ($\mathbb{N}^+$)
- Farthest point
- Elliptic
- Multiplicatively weighted
- Additive weighted
- Compound weighted
- Manhattan
- Farthest point Manhattan
- Karlsruhe

1. Voronoi diagrams are computed on a per-cell basis. Every cell begins with the assumption it is coincident with the infinite world-xy plane. Then, for every other particle **P** in the spread, the cell area is sliced using the medial axis of the line connecting the current, active particle and **P**. This approach results in many rogue intersection operations, since a Voronoi cell can only be affected by direct neighbours. Since Voronoi diagrams are duals of Delaunay triangulations -if a Delaunay mesh exists- the Delaunay edges can be used to locate all neighbouring particles and to thus minimize rogue intersections. In addition, the infinite base area of the diagram is reduced to a rectangle 10% larger than the bounding box of the original particle cloud. In most cases cells that share an edge with this rectangle will be ignored during computation.
This algorithm is not the fastest method of computing Voronoi diagrams. The Sweepline algorithm devised by Steven Fortune runs in $O(n \log n)$ while the implementation here is $O(n^2)$. These algorithms are only suited for Order_0 diagrams.
2. All stated algorithms have a hyperbolic counterpart as well.

B.2.5.3 Search Grids

*In computational geometry, a Search Grid is an **N**-dimensional grid (where **N** matches the number of dimensions of the base particle cloud) where every cell stores references to the particles contained in that cell.*



A typical Search grid. Grid cells that do not reference particles are coloured red.

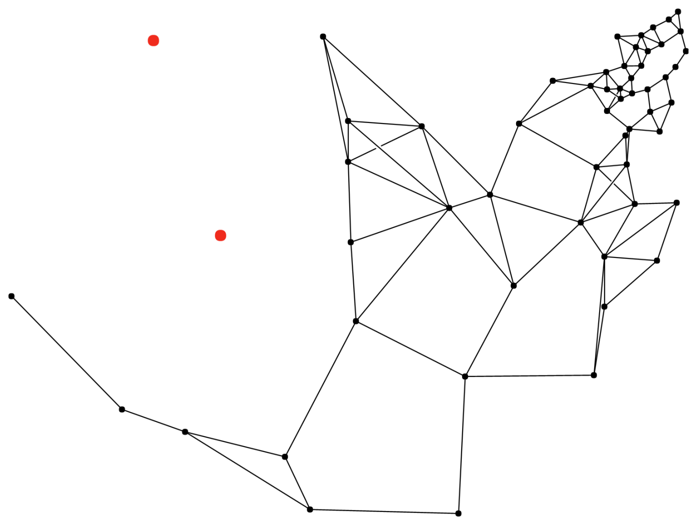
A search grid isn't so much a geometric manifestation of a particle cloud, as a database which enables us to search the particle cloud in an efficient manner. Search grids provide ways to perform non-topological neighbour searches.[1] However, search grids can also be used for other types of analysis, such as density and contrast. A Search grid is not homeomorphic and should therefore be used with caution. Hexagonal search grids would be better, but this is much harder to implement.

The advantage of Search Grids is, that they can be updated locally. Small changes in the original particle spread are likely to break the topology of other derivatives. Search grids can be updated relatively effortless. Unfortunately the platform I have written does not cater for movement and is thus unable to capitalize on this advantage.

1. A Search grid consists of a rectangular array of square cells, that approximate the bounding box rectangle of the original particle cloud. Every particle is then added to the cell which contains its coordinates. Since all cell dimensions are known beforehand, this method minimizes computations and iterations.

B.2.5.4 Connectivity networks

> *The topological description of a network that specifies, in terms of circuit termination locations and quantities, the interconnection of the nodes.*
>
> *-ATIS committee-*



A typical Connectivity network. Isolated particles are coloured red.

A Connectivity network is a topological representation of similar (within tolerance) particles. The tolerance is expressed as a distance threshold value T, meaning that particles which are closer together than T, are treated as similar[1]. Note that when particle A evaluates as similar to particle B, and particle B evaluates as similar to particle C, this does not mean that particle A is similar to particle C. Because A and C may have been on opposite sides of B, and could therefore have twice the distance T between them.[2]

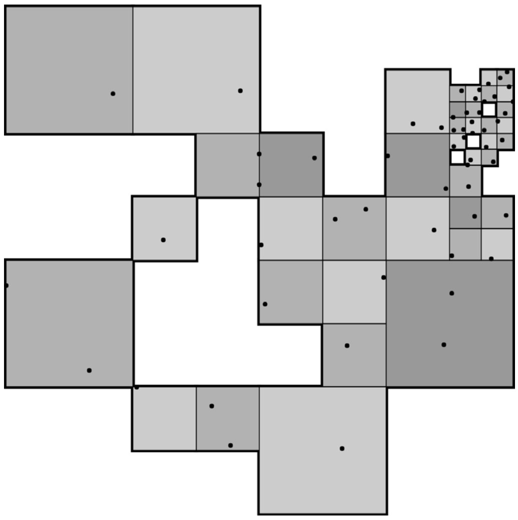
Another setting of the network is the maximum allowed number of network links L starting at a particle. By limiting the connectivity, an overcrowded network can be avoided. The longest connections are always removed first by this filter. L applies not to the total amount of connections converging at a particle, but on the amount of connections that originate at a particle. For convenience all bidirectional connections are also removed, so we are left with the smallest (shortest) network which still contains all the information.

In terms of spatial planning, a network represents the social interaction ratio of particles for a specific level of scale. By limiting the distance of connections in the network, social clusters emerge.

1. Networks are notoriously difficult to compute since they involve extra iterations; every particle must evaluate every other particle for proximity. This quickly results in enormous computing times which is fatal for the usability of software. Connectivity networks here, are computed using a standard Search grid nearest neighbour search. For every cell C in the grid, a local cluster of cells M is created. M contains C and the eight surrounding neighbours. (Cells along the edges of the grid are special cases). Within M all network connections for particles located in C are parsed. Upon completion all particles in C are evaluated for occupancy. If there exist particles in C that are not yet saturated, a new cluster M is created containing all cells that have a certain minimum and maximum distance to C. M is updated with an ever growing radius until the search distance exceeds the connection threshold or until all particles in C are fully occupied. This outward spiral approach drastically reduces overhead distance and sorting calculations and can be more than 3000 times faster than a brute force approach.
2. The implementation of Connectivity networks here is not mathematically solid. Since this is an often used derivative, optimization was an important factor in the algorithm. The resulting networks are based on quadratic distances rather than euclidian distances. The upshot is that networks here are not genuinely topological.

B.2.5.5 Quad Trees

In computational geometry, a kd-Tree is an iterative subdivision of search space which indexes contained particles. "Binary trees", "Quad Trees" and "Octrees" simply refer to the number of subdivisions every iteration generates (2, 4 and 8 respectively).



A typical Quad Tree. Empty branches are not drawn.

A Quad Tree is created by iteratively subdividing a region of space until all branches contain less than a preset threshold amount of indexed particles[1]. Since every branch in the Quad Tree contains more or less the same amount of particles (typically the subdivision threshold is a low value), the local dominant scale of a spread will emerge. But when the subdivision threshold is high, other properties can be read.

A Quad Tree is less suited for searches than a Search grid, since the structure of all subdivided regions is stored in a branching fashion. Thus adjacent coordinates in model-space are not necessarily located in adjacent quads. However, since Quad Trees adapt to local complexity they can yield better searching times when the original particle spread is severely heterogeneous.

Trees come in many different types, each optimized for a certain search behaviour. A fairly complete cross section of available tree algorithms[2] is provided by Frantisek Brabec [Brabec, Samet 1998]:

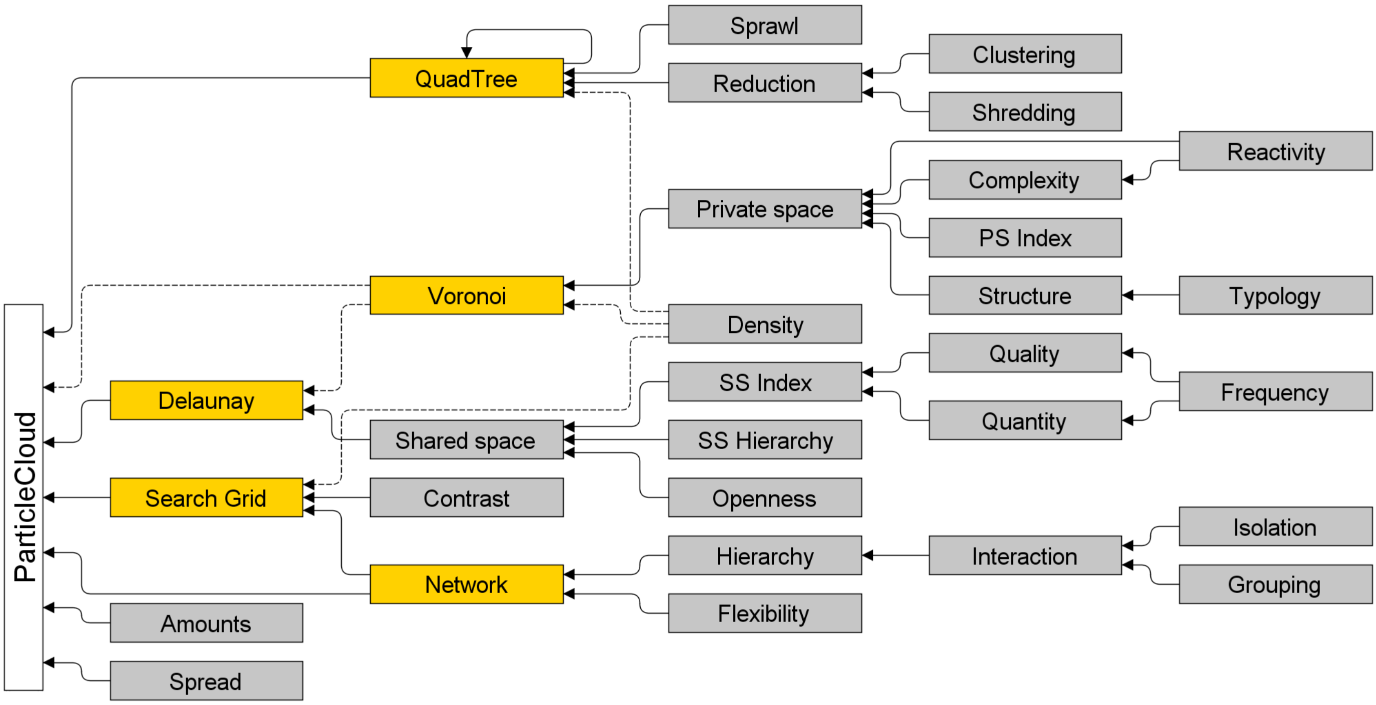
- K-d trees (**K** refers to the number of dimensions the tree resides in)
- MX trees (**M**atri**X**)
- PR trees (**P**oint-**R**egion)
- Bucket PR trees
- Bucket K-d trees
- 2D range trees
- Priority trees
- PK trees

Each of these structures can be implemented for any number of dimensions and any multiple of two subdivisions. 'Quad Trees' as discussed in this paper refer exclusively to '2D Bucket PR Quad Trees'.

1. A Quad Tree is created by iteratively subdividing a closed region of space into smaller -but geometrically identical- regions. The first tree-node is the smallest possible (world-axis aligned) square which contains all particles. If the total amount of particles contained within this square exceeds a preset limit T, the square is subdivided into four smaller squares, each of which inherits the particles which it contains. Every new square in turn performs the threshold test and subdivides accordingly. This process terminates when all squares contain T particles or less. Trees are a widely used method in computation for partitioning sets.
2. All stated algorithms have a hyperbolic counterpart as well.

B.2.6 Derived properties

The point of derivatives is to supply methods for measuring properties. Because it lacks shape, only very few properties can be extracted directly from a spread. This is of course the paradox of Proximity sketching, only by abandoning shapes can we scientifically evaluate a sketch, but we need to generate shapes to do so. Technically, the amount of information that can be read from a spread is infinite. An algorithm can be devised to evaluate any thinkable property. The platform discussed here has been equipped with 24 different properties. Some of these are computed directly from a derivative, some depend on other properties. The graph below lists the entire programmatic structure of the platform:



Derivatives are drawn in yellow, properties in grey. Dependencies are drawn from right to left, i.e. the property "Openness", requires the property "Shared space", which in turn requires the Delaunay derivative. Optional dependences are drawn with dotted lines, i.e. "Density" can be computed either via the Voronoi derivative, the Quad Tree derivative or the Search Grid derivative. Structure and dependencies as indicated above, only apply to the current implementation of Proximity sketching. It is not part of the Proximity paradigm.

When writing algorithms for properties, one must first determine what the definition of a certain property is. For instance; what is 'social hierarchy'? A possible answer (the answer used here) is that 'hierarchy' is the inequality of neighbours. When two of your neighbours differ, you are unlikely to treat them the same way, thus, there is a certain measure of hierarchy. Since we do not design the personalities and characters of people, we can only use the positional data of neighbouring dwellings. The Delaunay derivative provides ways to search for neighbouring particles, and it is thus the best starting point. Then, somehow we need to define 'inequality between neighbours'. Since every triangle in the Delaunay mesh connects three neighbours, the shape of any given triangle is an indication of equality. In this case I have used the ratio of the longest triangle edge divided by the shortest triangle edge. I.e. sharp triangles will be less equal than blunt triangles. This provides us with an index for inequality per triangle. Subsequently, for every particle all adjacent triangles must be found, and their respective values combined (averaged, floored or any other superimposition) to get the social pressure for a certain particle. If we wish to define hierarchy not between direct neighbours, but for all neighbours within -say- a 50m radius, we should use a Connectivity Network instead of a Delaunay Triangulation. Every derivative described in the previous paragraphs, focuses on a certain type of relation:

- Delaunay Triangulations map the **local social** relations between particles
- Voronoi Diagrams map the **local private** (non-social) properties of particles
- Connectivity Networks map the **global social** relationships between particles
- Search Grids map the **social partitioning** of particles
- Quad Trees map the **social density** of particles

So, in order to analyze the social weight of high-density particles, we need both the Voronoi derivative (in order to compute the density) and plot that value against the connectivity index as obtained through a Connectivity Network derivative. By performing this computation for every particle in the spread, we get a map which highlights the social/density gradient in a model.

A similar approach has been used for 24 other properties, all of them outlined in the upcoming paragraphs. They widely differ in complexity and application and the total set has no special meaning. It is simply a collection of possible properties which can be derived from particle clouds.

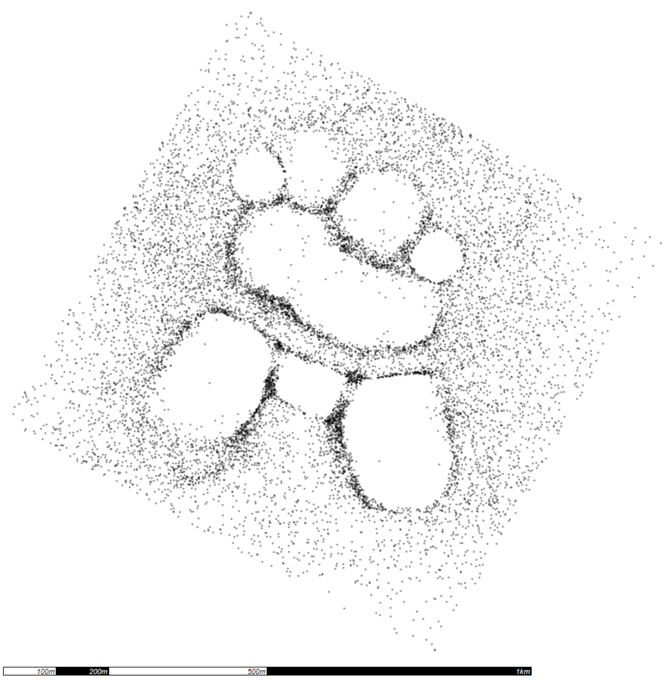
B.2.6.1 Amounts

This property is simply a number representing the total amount N of particles per legend unit. It is the most basic measure imaginable, but that does not mean it is redundant. If a design is required for 20.000 homes, the planner needs to know if he has catered for this demand. Counting actual dwellings is a fairly cumbersome pastime when your model-representation consists of pencil lines on a piece of paper.

B.2.6.2 Spread

This isn't a property as such, since nothing is being calculated. It is merely the collection of all particle clouds in the model. The output is a simple image where evert particle is represented by a dot. Optionally, different types of particles (if super-binary legends are used) will be manifested as different colours and/or different radii.

Although we can perform almost no direct measurements on a bare spread, the human mind is capable of creative assessment (see [2.2.3 Closure]) and thus -in a sense- the spread can be treated as a property.

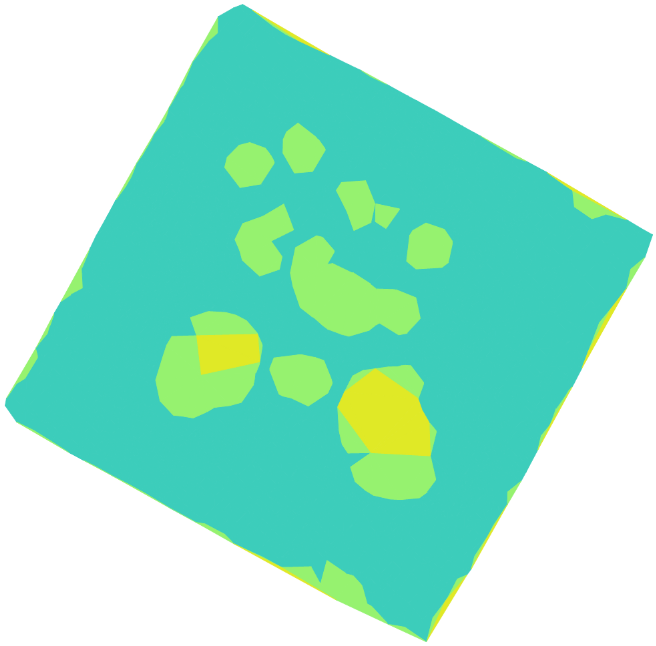


The original spread. All particles are drawn with similar properties.

B.2.6.3 Shared Space

Shared Space (SS) is an abstract property in that it is not evaluated directly. It is only used by other properties. Essentially, SS is nothing more than a localization of the Delaunay Triangulation. Every triangle in the Delaunay mesh represents the space shared by three neighbouring particles. Most particles have more than one of these shared spaces. Properties that elaborate on SS:

- Frequency
- SS Index
- Openness
- SS Hierarchy



Map with Shared Space posterization.

B.2.6.4 Contrast

Contrast -in this case- is expressed as difference in density on a specific level of scale. Contrast is computed via a Search grid[1]. The Search grid cell size corresponds with the required level of scale. Typically, these levels are inherited from the logarithmic scale series:

Lower limit	Domain base value	Upper limit
...	**...**	...
0.1	**0.3**	1.0
0.3	**1.0**	3.0
1.0	**3.0**	10.0
3.0	**10.0**	30.0
10.0	**30.0**	100.0
30.0	**100.0**	300.0
...	**...**	...

This series extends infinitely in both directions[2]. For our purposes we can limit the domain to regional sizes. Every row in the table above indicate one level of scale. Every level consists of a lower domain value, an upper domain value and a base value. Note that every level extends to the base values of its adjacent levels.

1. The number of empty cells in a Search Grid is an indication for Search Grid significance. If too small a cell size is used the false accuracy of the spread will be magnified. Ideally, the number of empty cells in a search grid is between 25% and 75%. These are rough estimates based on trial and error research.
2. The advantage of using a Search Grid derivative to compute contrast, is that we can limit the contrast to a certain level of scale. Typically, contrast can be computed through any derivative which uses shared edges (i.e. Delaunay and Voronoi).

Contrast then, is computed for all borders between Search Grid cells. The final Contrast value is a result of absolute and relative contrast. Absolute contrast is the amount of difference between adjacent cells and relative contrast is the ratio between cell content;

$$C_{abs} = |A - B|$$

$$C_{rel} = \frac{\min(A,B)}{\max(A,B)}$$

Absolute contrast can be any integer value in $\mathbb{N}^+$, relative contrast can be any floating point value between zero and one.

Feedback on this property is two-fold. First, a map is drawn where high contrast transitions are highlighted, this map gives an impression of where contrast is to be found in the model. I.e. where the designer needs to focus his repairs if the desired amount of contrast is not reached or exceeded. Secondly, a histogram showing the total length of several contrast groups. This makes it easier to compare different models for contrast occupancy.

High Contrast drawn in black, medium Contrast in yellow and low Contrast in white. Contrast has been computed for 30m radius.

B.2.6.5 Sprawl

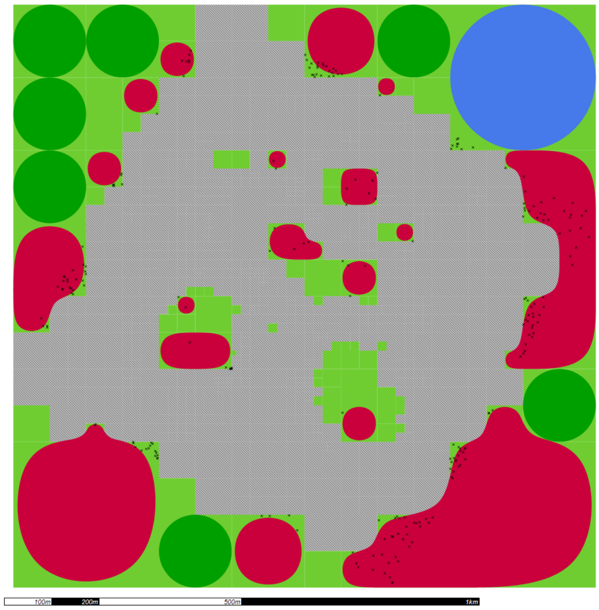
In modern spatial planning, de-urbanization has led to a phenomenon called sprawl. Sprawl has many definitions, but it is almost always used in a negative sense. During suburbanization the notion arose that low density housing would improve living conditions for typical family households. In areas where lots of space is available these new ideas were quickly put to wide-spread use. Lately, more and more research indicates that low-density dwellings cause more problems than they solve and sprawl is now falling rapidly out of favour. Unfortunately there is no solid definition of sprawl, but this platform attempts to provide a few footholds for detecting sprawl.

First, a Quad Tree derivative is created for $T = 30$. Subsequently, all adjacent branches whose density evaluates as being within a preset domain will be grouped together as sprawl-clusters[1]. Those areas that exceed this domain are treated as "proper" urban density and are drawn as a grey hatch. Areas that contain lower densities are drawn in green. Also, areas with sub-sprawl densities that are larger than 1 hectare will be filled with an additional dark green spot. Areas that do contain dwellings, but do not exceed the lower density limit, are drawn with a blue spot. Several sprawl-density-domains (in particles per hectare) have been preset:

1. Quad branch grouping is not a trivial task and occasionally the algorithm fails. First all edges of all branches are stored in an array of line segments. All of these line segments are then split using the endpoints of all other line segments. From this new array all segments that occur twice are removed (both instances are removed). Finally all remaining segments are rejoined into closed polylines. For this algorithm to work properly a certain tolerance has to be taken into account.

Lower limit	Upper limit	Typology
0.1 pph	1.0 pph	Extremely open space
0.2 pph	2.0 pph	Open space
0.5 pph	5.0 pph	Low density suburban
1.0 pph	10.0 pph	Medium density suburban
1.5 pph	15.0 pph	Medium density suburban
2.0 pph	20.0 pph	High density suburban

Every density domain results in a different map. There are some problems with the sprawl map which are a result of the precipitous nature of Quad Trees. If few particles are located in a large Quad Tree branch, the branch as a whole will have a low density, even if all particles are grouped together in a high-density cluster. Before Quad Tree branches are designated as sprawl, a test should be performed which tells us whether the particles are distributed evenly across the branch area, or whether they are grouped together. I have not added this test and in the example map on the right, about half of the sprawl regions should be discarded as a result of this.



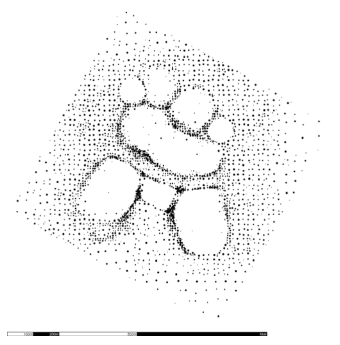
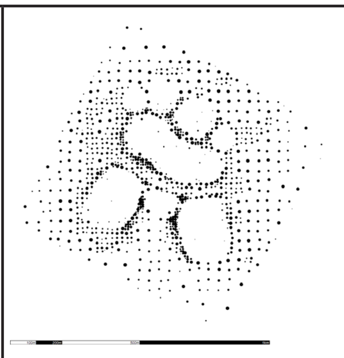
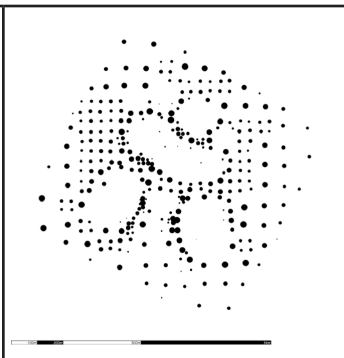
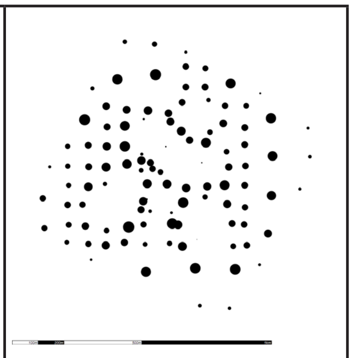
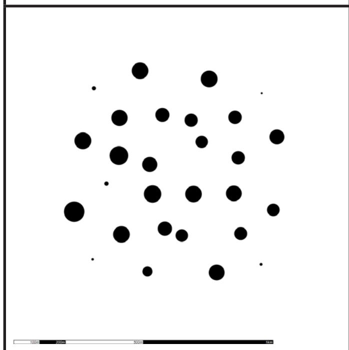
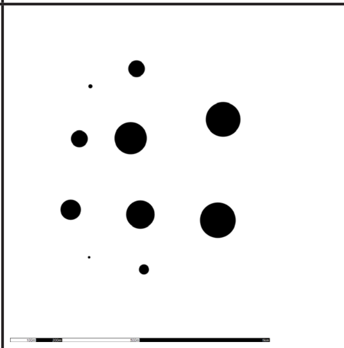
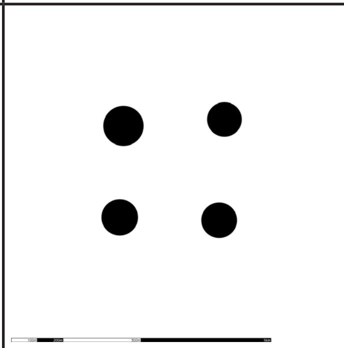
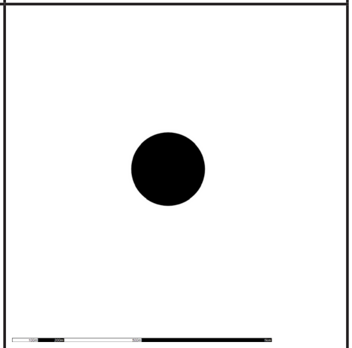
Sprawl for density domain $\{1.0 - 10.0\} d \cdot ha^{-1}$

A typical example of suburban sprawl in the US.

B.2.6.6 Reduction

The original particle cloud probably contains many particles. Apart from the significant false accuracy per particle[1], such a large amount is hard to grasp in a way that exceeds impression or feeling. By averaging particles in an iterative fashion the total number of particles is reduced and the false accuracy is partly removed since the process of averaging tends to annihilate anomalies.

In reduction diagrams the particles in the original cloud have been iteratively grouped and added together to create a representative reduction. Every reduction is associated with a threshold value *T* which controls the maximum allowed amount of particles per reduction group. Every group is drawn as a circle around the average coordinate of all the particles in that group, where the area of the circle is the sum of the area of the assimilated particles.

			
Reduction for *T*=10	Reduction for *T*=30	Reduction for *T*=100	Reduction for *T*=300
			
Reduction for *T*=1.000	Reduction for *T*=3.000	Reduction for *T*=10.000	Reduction for *T*=30.000

Reduction graphs provide information on difference and equality for clusters of dwellings (i.e. ensemble, neighbourhood, district, etc.). Reduction diagrams allow us to judge the impact a certain intervention entails and to what extent different designs actually differ.

B.2.6.7 Private Space

Just like Shared Space, Private Space (PS) is an abstract property. PS is a localization of the Voronoi Diagram. Every cell in the Voronoi Diagram represents[2] the space that belongs to its base particle. Properties that elaborate on PS:

- Complexity (see [2.4.16 Complexity])
- PS Index (see [2.4.17 Private Space Index])
- Structure (see [2.4.18 Structure])

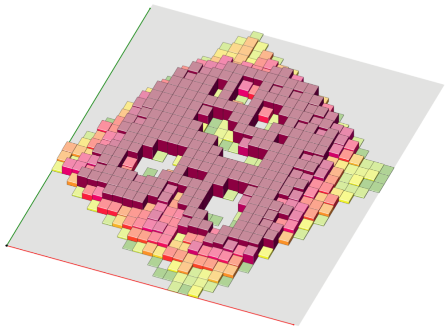
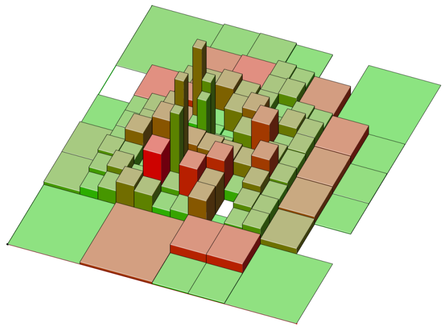
1. False accuracy is a result of the sketching techniques used in Proximity. When the first rough sketch of a model is made, the positioning of particles is only approximate. But the particle coordinates are digitally stored in double-precision floating point values. These coordinates imply a much higher level of accuracy than actually intended by the user. One solution would be to add a tolerance domain to every particle, thus only specifying where the particle could be, instead of where it is.
2. "Represents" does not mean "is identical to". A cell is an indication of private space, not a manifestation.

B.2.6.8 Density

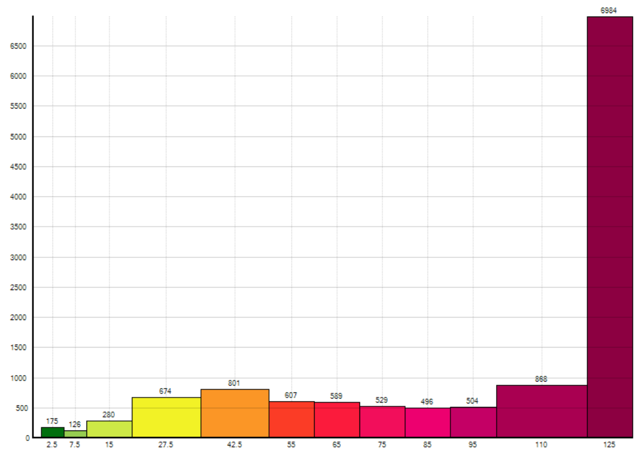
Density is always expressed as particles/hectare and it can be computed in many ways. Here, density is computed via three different derivatives[1]:

- Voronoi Diagrams
- Search Grids
- Quad Trees

For density calculations we need a number of particles N and the area A they occupy. In case of a Voronoi cell N is always 1 and A equals the surface area of the Voronoi cell. This is a very 'honest' method for density computation, but it does not work for cells that are on the boundary of the diagram, since their area is infinite. In case of Search Grids and Quad Trees, N equals the length of the list of referenced particles and A equals the surface area of the current grid cell or tree branch.

		
Voronoi density	*Search Grid density*	*Quad Tree density*

Density is a widely used property in Urbanism which allows us to quickly link a hypothetical area in a design to an existing city model. This is a common analysis method for sketch designs, existing city maps with a matching density are pasted over the sketch regions to give an impression of the urban weave.



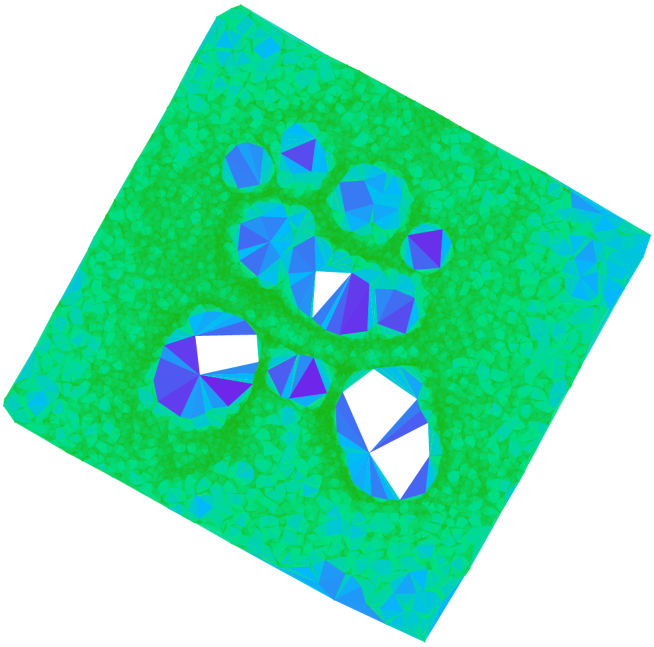
Density histogram derived from Voronoi density.

In Spacemate (M. Berghauser-Pont, P. Haupt, 2004), Permeta architects coin a better way to measure urban densities. Instead of dwellings·hectare^{-1} a combination of FSI, GSI and OSR[2] can be used to more accurately catalog typologies. However, this system suffers from the same dysfunctions as Search Grid and Quad Tree densities, and it requires a profound knowledge of the details of a build area. It is less suited for Proximity analysis, but it could be used to add an extra layer of properties to particle subsets.

1. Alternatively, the Density of any coordinate C in the Model-space can be computed by counting the number of particles within sphere S around C. This method results in a blurred representation of Density.
2. Floor Space Index, Ground Space Index and Open Space Ratio. See http://www.spacemate.nl for further details.

B.2.6.9 Shared Space Index

The Shared Space Index (SSI) is simply a wrapper property for the Shared Space abstract property. SSI is linked with the area of the local Shared Space. Several special cases are dealt with separately. Since the local Shared Space is always shared among three neighbours, the area of the space is an indication for social pressure. Nearby neighbours result in high social pressure. Social pressure per particle is identical to the highest pressure in all adjacent triangles.

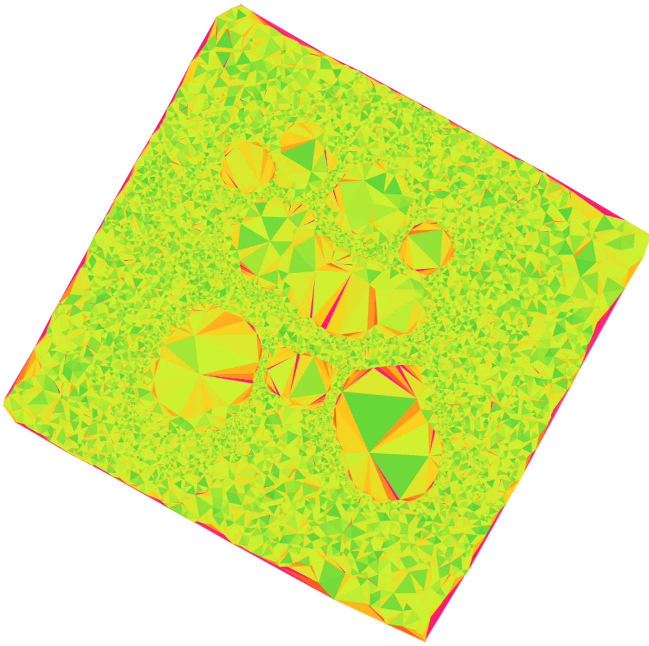


Shared Space Index map. Green areas indicate a high social pressure. Faces exceeding 3.000m² *are ignored.*

B.2.6.10 Shared Space Hierarchy

The Delaunay face ratio distribution is a measure for the social inequalities of a spread. For neighbours A and B to be equal, they need to have the same distance to the particle of reference P. Shared Space Hierarchy (SSH) is a result of Shared Space ratio.

A Delaunay triangulation minimizes the internal angles of all triangles, which means it always connects the most equal neigh-bours in a spread, but this is only a result of the Delaunay ruleset, not a prerequisite for it. Thus, if the particle cloud is inherently unequal, the Delaunay triangulation will reflect this. A problem with the ratio-analysis is that it only takes small translations per particle to change the ratio of most faces. False accuracy problems are therefore very likely to occur.



Shared Space Hierarchy map. Red areas indicate high social inequality.

B.2.6.11 Openness

Every Delaunay face connects three particles in such a way that the circumcircle of the face contains only those three particles. Delaunay triangles can thus be used to locate gaps in the social fabric of the particle spread. "Gaps" are defined as circles with a certain radius ***R*** that contain less than a certain amount ***T*** of particles. A few combinations make sense:

Gap description	**Radius** *{R}*	**Threshold** *{T}*
Garden	3m	0
Playground	10m	1
Neighbourhood park	100m	10
Small city park	300m	100
Large city park	1.000m	1.000
City landscape	3.000m	10.000
Landscape park	10.000m	100.000
Landscape	30.000m	1.000.000

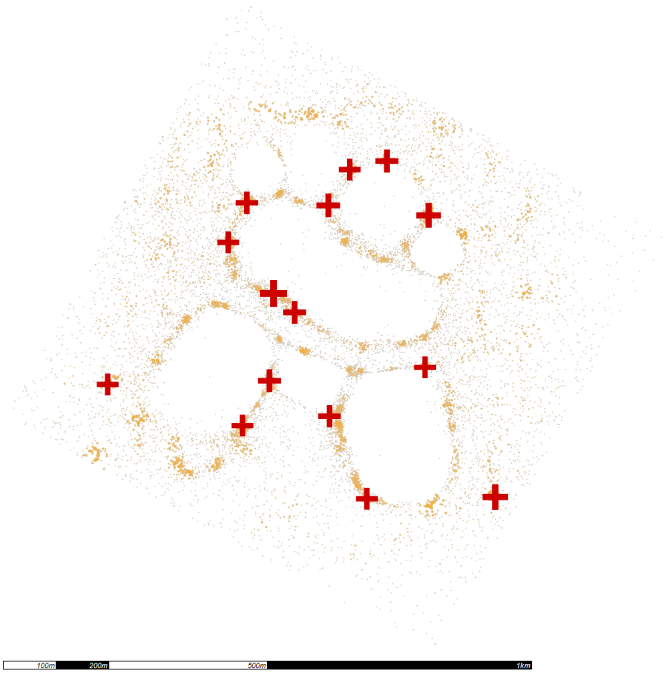
Since the area of a triangle circumcircle is known, gaps can be found with relative ease. The complex of gaps is called "Openness". To find gaps that contain less than ***T*** particles, a separate Delaunay derivative must be constructed, where the input is no longer the original particle spread, but a Quad Tree reduction where the Quad Tree group size equals the gap threshold.

Note that this property is only an indication of potential. It does not find all gaps in the model, only the locations where gaps might occur.

Openness map. Pink dots indicate playgrounds, orange dots small city parks and green dots large city parks.

B.2.6.12 Hierarchy

A connectivity network plots relationships between particles. In the entire network, some particles are better connected than others, this is both a result of local connectivity occupancy and global connectivity ratio. For a specific level of scale the Connectivity network contains a number of key-nodes that occupy frequent positions within the network. "Frequency" in this case is a result of superfluity-frequency and exchange-frequency; a node which occupies a unique road towards other nodes has a high superfluity-frequency and a node which provides access to many other nodes has a high exchange-frequency. Hierarchy is a measure for these frequencies. Hierarchy is a completely relative measure and it should not be used to compare different models.

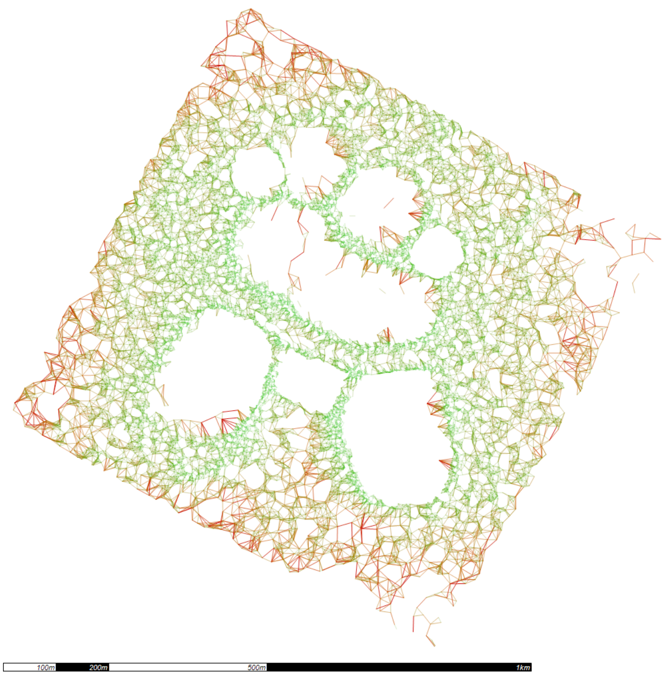


Hierarchy diagram. Key nodes manifested as red crosses.

[Hierarchy is algorithmically defined as transit-probability. I.e. travel between two random nodes A and B in the network and all nodes along this path will have an increased hierarchy value. Key-nodes are defined as nodes that exceed 80% of the maximum hierarchy in the network. Near coincident key-nodes are removed from the network. The solution provided here is not an exact one and it might differ between documents. It should therefore be treated with caution.]

B.2.6.13 Flexibility

Flexibility is a measure for network strain. The length limitation T of a network connection, is an abrupt threshold. From a social point of view, there shouldn't be a large difference between a neighbour who lives 10m away and neighbours who live 11m away. Network strain is an indication for the reach of a specific connection. If a connection is well within T the link is treated as 'relaxed'. Once the length of the connection approaches T it becomes 'strained'. Strained connections are drawn in red and they provide feedback on the kind of connectivity for the current level of scale, rather than just the amount of connectivity.



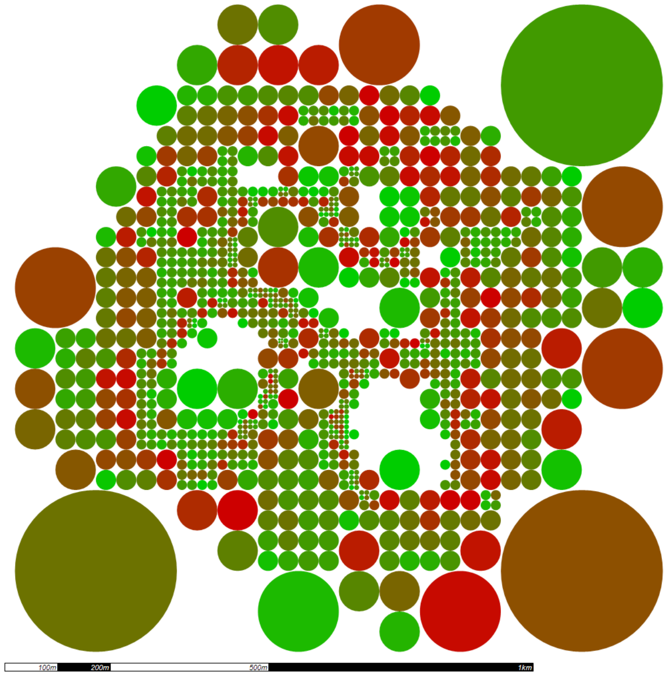
Flexibility diagram. Straining connections are drawn in red.

B.2.6.14 Clustering

Clustering is a sub-property of reduction. Clustering occurs when phenomenologic proximity (see [2.2.1 Proximity]) occurs in a reduction graph. Overlap of reduction clusters is an exceptionally strong case of clustering.

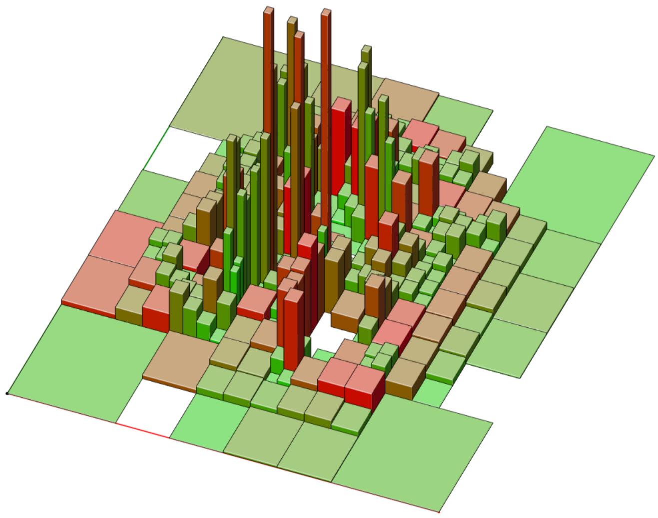
B.2.6.15 Shredding

The amount of shredding of a spread, is the amount of local density difference. The Quad Tree derivative is a good way to visualize this. Large branches next to small branches indicate high shredding values. Human perception of shredding is diversity. The difference between Contrast and Shredding is the integration of scale. Contrast is computed for a specific, preset level of scale while Shredding is computed for a preset accuracy. As a result, Shredding also shows the local scale magnitude.



Shredding map. Quad Tree branches are drawn as circles where the fill colour is derived from branch occupancy.

By colouring the branches in a Shredding graph, additional information can be stored. Green can be used to indicate low-pressure branches that are not in danger of subdivision. By adding another dimension the density of branches can be displayed. By displaying two properties at once, competitive perception groupings will emerge (see [2.2.2 Similarity]).

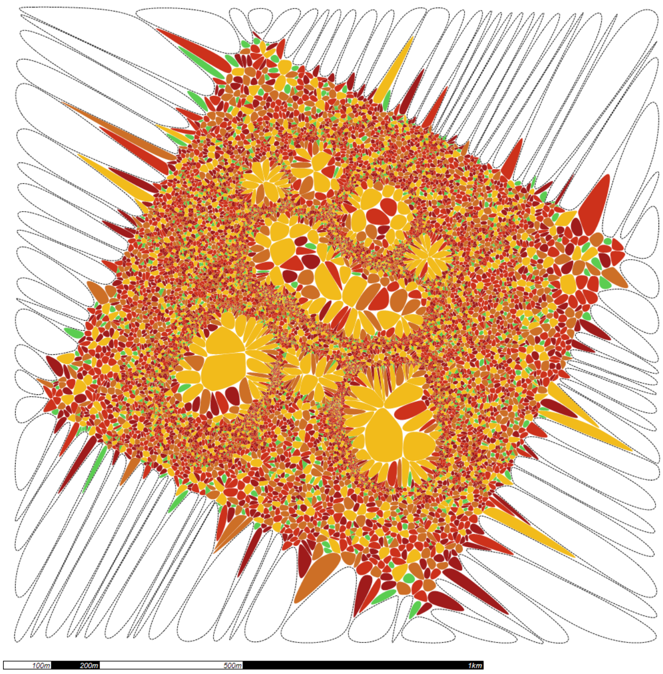


Isometric Shredding map. Quad Tree branches are drawn as blocks where the fill colour is derived from branch occupancy and block height is derived from local density.

B.2.6.16 Complexity

Complexity is a per particle measure for neighbour occupancy. The occupancy equals the sum of all adjacent particles (as indicated by the Delaunay derivative) multiplied by their distance factor. (The distance factor equals the $\log(D)$ where D is the euclidian distance.)

Particles with many neighbours will be more complex than particles with few neighbours. Particles with nearby neighbours will be more complex than particles with distant neighbours. Complexity is an indication of social flux, which usually expresses itself in active, busy neighbourhoods.

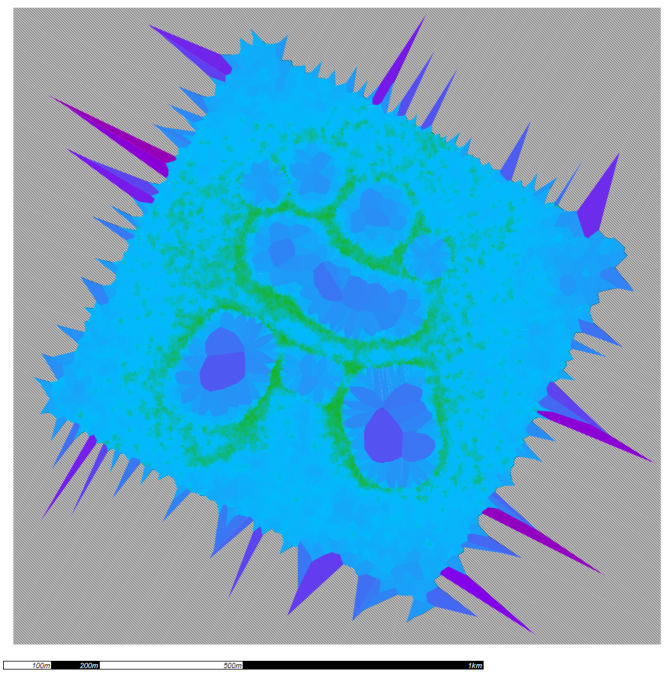


Complexity map. Cells along the boundary of the model-space are drawn in white.

B.2.6.17 Private Space Index

The Private Space Index (PSI) is simply a wrapper property for the Private Space abstract property. PSI is linked with the area of the local Private Space. Several special cases are dealt with separately. PSI is nearly synonymous with Density, with the small difference that Density is mostly a numeric property while PSI is a graphical property.

Very large cells (large areas or long circumference) and cells along the boundary of the Voronoi space are not a proper representations of Private Space. They are ignored during PSI analysis.

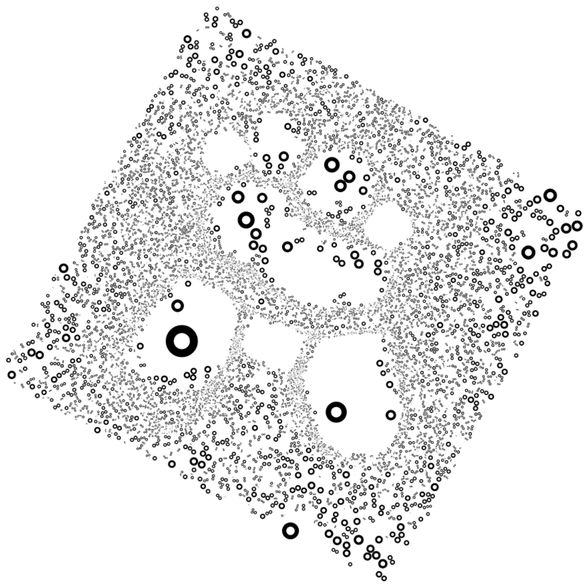


Private Space Index map. Cells along the boundary of the model-space are ignored.

B.2.6.18 Structure

Voronoi cells are poor visualizations of Private Space since -among them- they cover all surface area in the model-space. Properties such as Density and Complexity are based on topological and numeric values. The local shape (the geometric spin-off of topological Private Space) is also an important factor in the character of a spread. Structure is the sum-total of all local shapes.

The local shape depends on a particles relation to its Voronoi cell. Particles that are relatively close to the edge of the corresponding Voronoi cell are assumed to have a smaller PSI than indicated by the cell area. By plotting the directions of all particles (the direction vector D starts at particle P and points towards the projection of P onto the closed boundary of the corresponding cell) as circles, the local shape emerges.



Structure map.

B.2.6.19 Quality

Quality is a measure of uniqueness and can be computed on a per particle basis by measuring the properties of the set of converging faces C in the Delaunay triangulation at that particle location[1]. The final Quality value of a particle is the sum-total of all Delaunay faces in C multiplied by the square-root of its area. Large faces are typically more rare than small faces. Quality is a purely relative property since we cannot imagine a null-value nor a highest value. In the current platform, Quality is an abstract property. It is only used by the Frequency property.

B.2.6.20 Quantity

Quantity is a measure of relative size and can be computed on a per particle basis by measuring the properties of the set of converging faces C in the Delaunay triangulation at that particle location. The final Quantity value of a particle is the highest ratio value of all Delaunay faces in C. Quantity is an absolute value. In the current platform, Quantity is an abstract property. It is only used by the Frequency property.

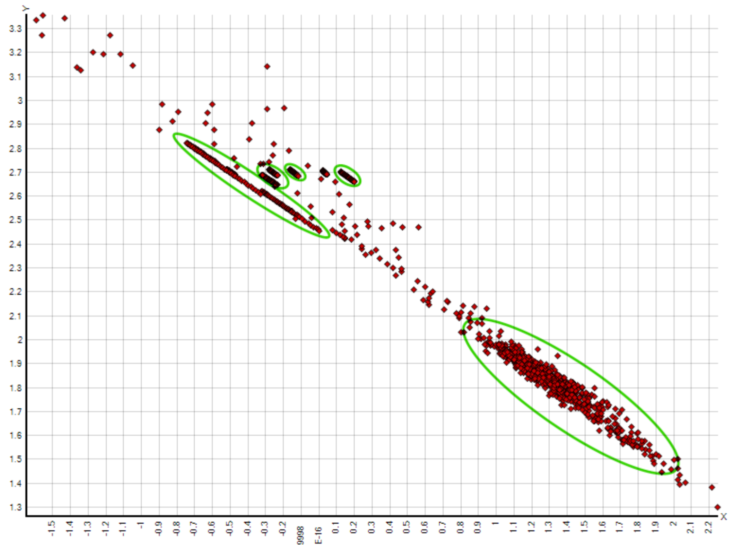
B.2.6.21 Interaction

Network interaction is a measure for social group growth. The number of nodes that can be reached by a maximum path length increases quadratically in a homogenous network. In varying networks this growth can differ drastically. Especially if the network is disjoint. Interaction cannot be linked to a common property when particles represent dwellings, so it has been omitted here.

1. A Delaunay mesh always approaches the same face/vertex ratio. As the size of the vertex array grows, the number of faces divided by the number of vertices quickly approaches to 2.00. For small amounts this ratio can be as low as 0.5, but 500 vertices is already enough to nearly reach the ratio limit. This means that the average amount of faces converging at a vertex is usually about 6. Note that these rules only apply to two-dimensional delaunay triangulations.

B.2.6.22 Reactivity

Reactivity is a dual property that depends on local cell area and local cell circumference[1]. Smaller cells have a higher reactivity and cells with more neighbours have a higher reactivity. When all reactivity values are plotted in a dual-axis graph, clusters emerge of particles with similar reactivity values. These clusters typically appear on iso-reactivity lines ranging from long circumference to large area. However, clusters can also appear in other groupings. Reactivity is a measure for social interaction diversity. Many different reactivity clusters indicate a diverse social weave. This tool is better suited for comparison than evaluation.

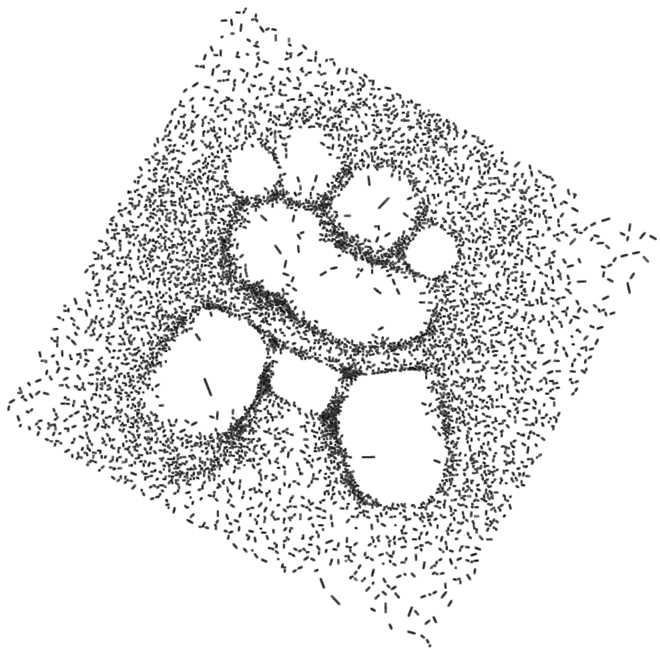


*Reactivity graph. The Voronoi cell area is plotted along the x-axis {**log(A)**}, Voronoi cell circumference along the y-axis {**log(C)**}. The current particle spread contains several reactivity typology cluster (marked in green).*

B.2.6.23 Typology

Typology is a wrapper property for Structure. The code is identical, but the representation of local shape features direction instead of size. In Typology lines are drawn from the particle coordinates to the projection of the particle onto the Voronoi cell boundary. This manifestation of local shape provides visual information on local orientation.

Optionally, orientation vectors can be drawn to intersect the entire Voronoi cell.

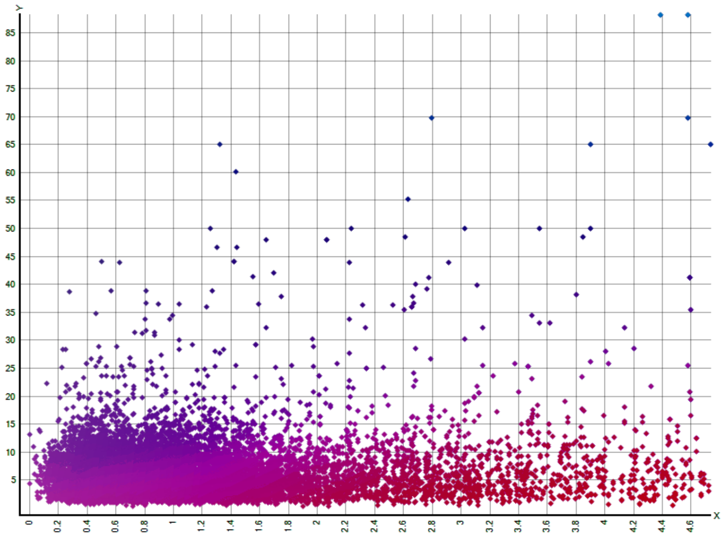


Typology map.

1. Algorithmic definition of reactivity: $R_i = log(A_i) \cdot log(C_i)$
Where R_i is the local reactivity, A_i is the local cell area and C_i is the local cell circumference.

B.2.6.24 Value

The Value[1] of a particle ***P*** is a dual property derived from Quality and Quantity. Quality indicates that ***P*** has access to relatively open space in one or more directions. Quantity indicates the diffusion of Quality between ***P*** and ***P***'s neighbours. Value can be plotted on a per particle basis or as a property of the spread as a whole.

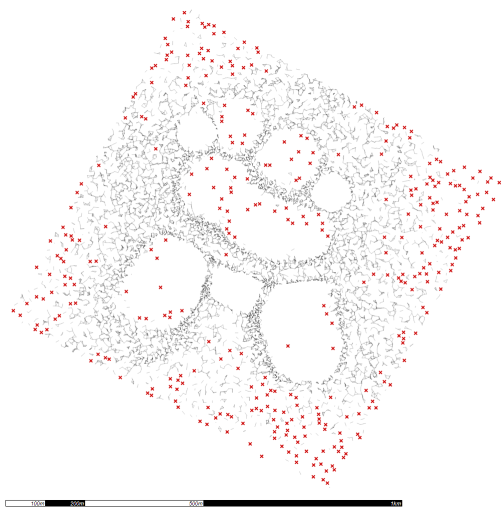


In this dual graph the combined properties of Quality and Quantity have been plotted. Along the x-axis the relative quality and along the y-axis the square-root of the absolute quantity. Since this graph contains relative data, it is not suited for comparison. Note that quality and quantity only apply to the small scale, social properties of particles. No statements are made concerning other types of quality.

B.2.6.25 Isolation

A particle is isolated, when it features no connections in a Connectivity Network. This network is always bound to a certain level of scale. Isolation is a boolean property and its meaning depends on the current level of scale:

Radius	Meaning of a particle
3m	an apartment home
10m	a complete dwelling in a larger volume
30m	a dwelling which is part of a group
100m	a dwelling which is part of a neighbourhood
300m	a dwelling which is part of a village
1000m	a dwelling which is part of a district
3000m	a dwelling which is part of a region



*Isolation for **R=10m***

The isolation value for an entire particle spread is a float point value indicating the amount of isolated particles as a percentage of the total amount of particles.

1. It should be noted that "Value" refers to social value, not economic value.

B.2.6.26 Grouping

Grouping is the dual property of isolation. A particle which is part of a group cannot be isolated. A group index can be represented as a float rather than a boolean, since 'being a member of' is a gradient property. Number of connections, local group size and connection length all contribute to the group index of a particle.

For the particle spread as a whole, the grouping can be indicated as the number of connections (multiplied by their respective weight) as a percentage of the total amount of particles.

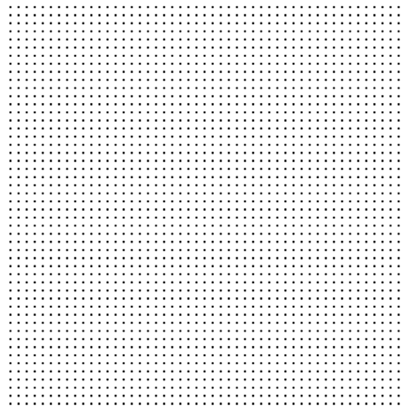
B.2.7 Property case studies

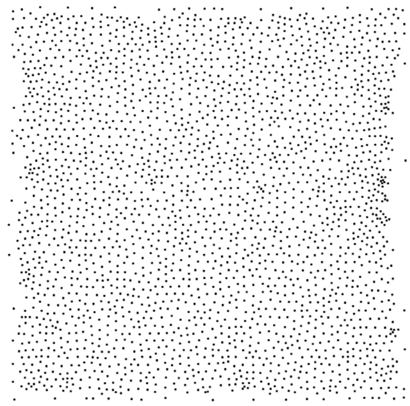
Some of the before-mentioned properties are absolute, others are relative. Relative properties can be used for comparison purposes, absolute properties can also be used for evaluation of single models. To see how (partly) absolute properties behave[1], I have created 8 different models:

1. Standard point grid, 50 by 50 points
2. Same as *{1}*, but with randomized and equalized coordinates
3. Same as *{1}*, but with manually created bulges
4. Same as *{2}*, but with manually created bulges
5. Random filling of a curved region
6. Same as *{5}*, but with a bias towards the edge
7. A set of small clusters
8. A severely manipulated model

All models contain the same amount of particles (2500) and roughly occupy the same amount of space (100 ha). Properties that cannot be used for comparison have not been included. When meaningful, a short assessment of the property value is added.

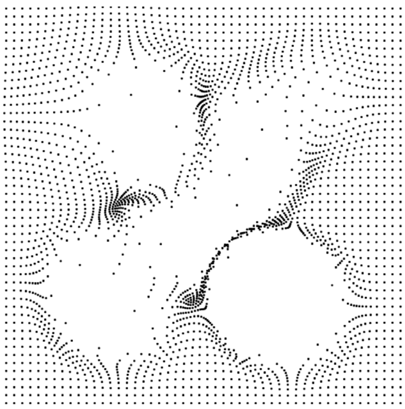
1. To know how a property works, does not necessarily mean the behaviour is predictable. Even I am sometimes surprised by the results of a spread analysis. In a sense, this is a desirable effect; predictable software isn't going to teach us anything we did not already know.

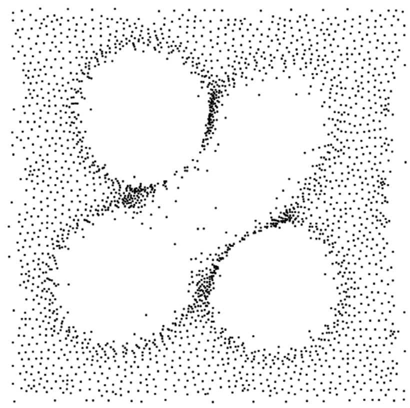




Model 1

Model 2



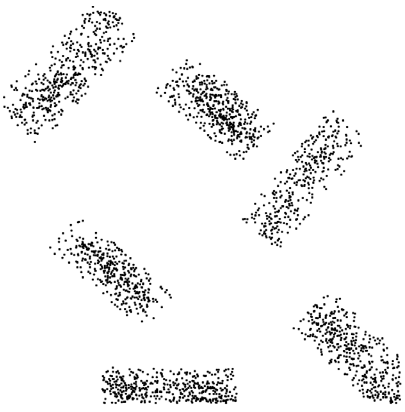


Model 3

Model 4

Model 5

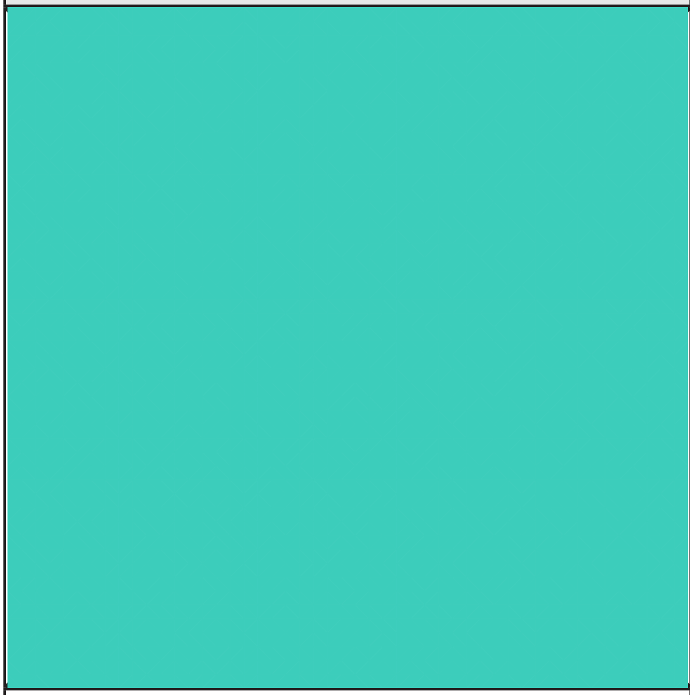
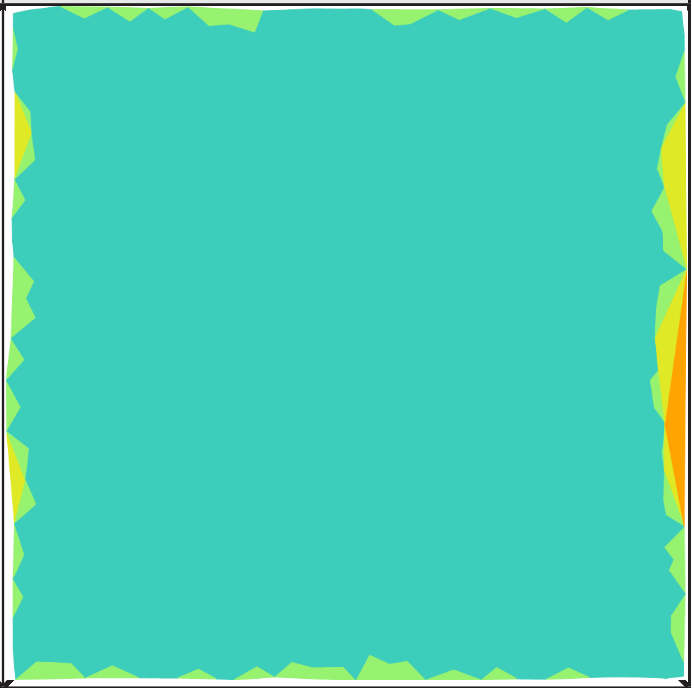
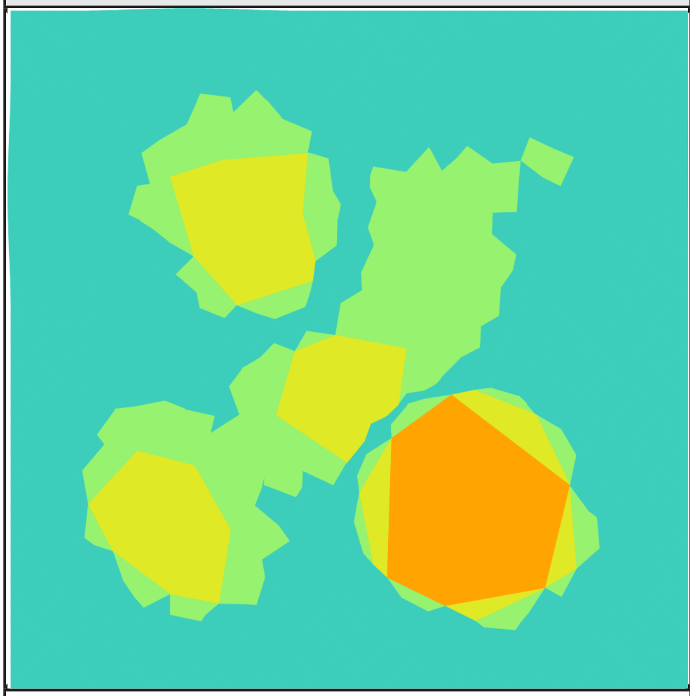
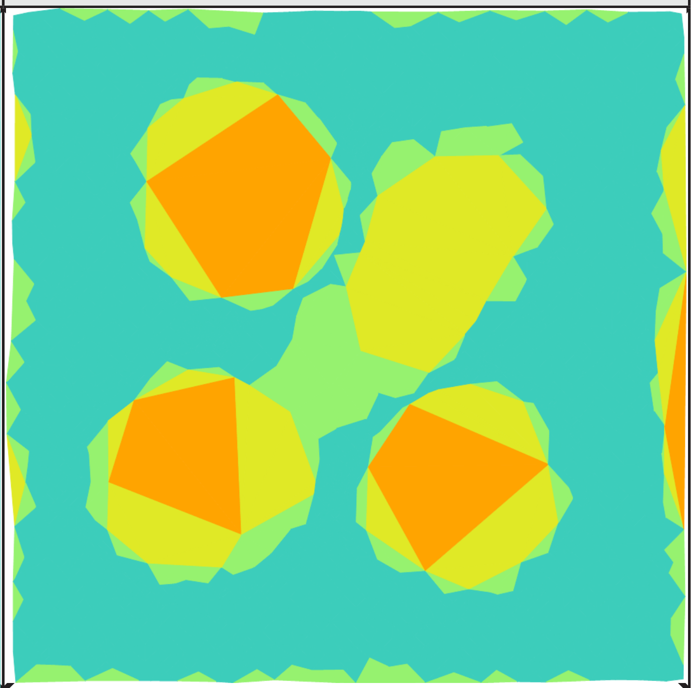
Model 6



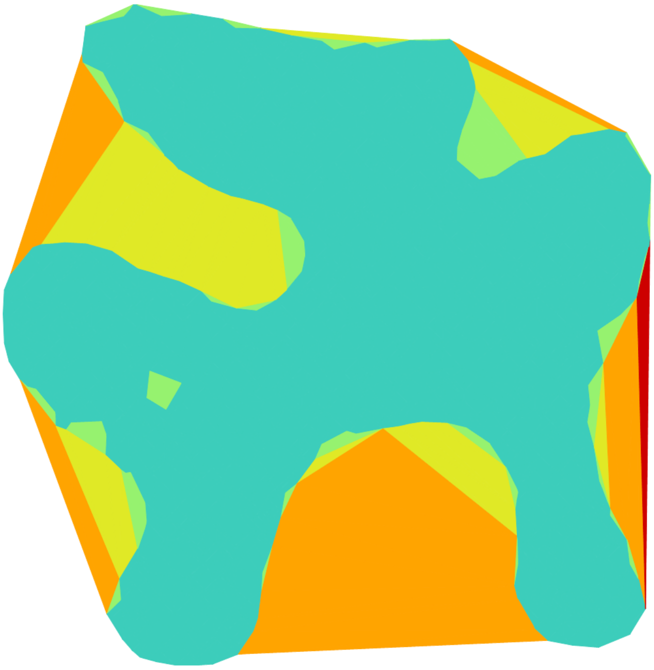
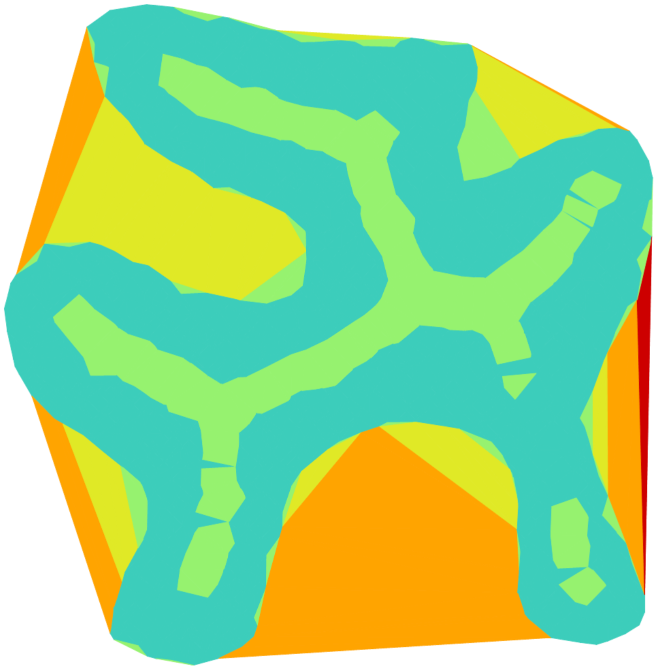
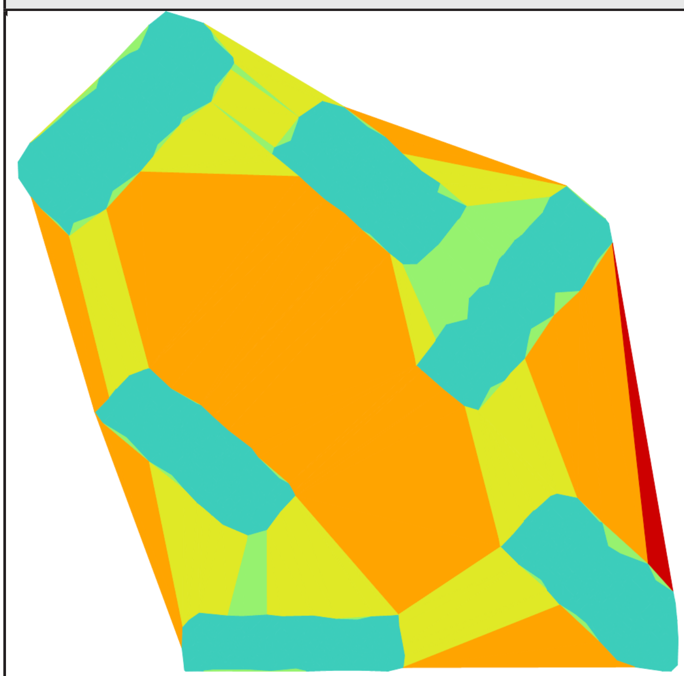
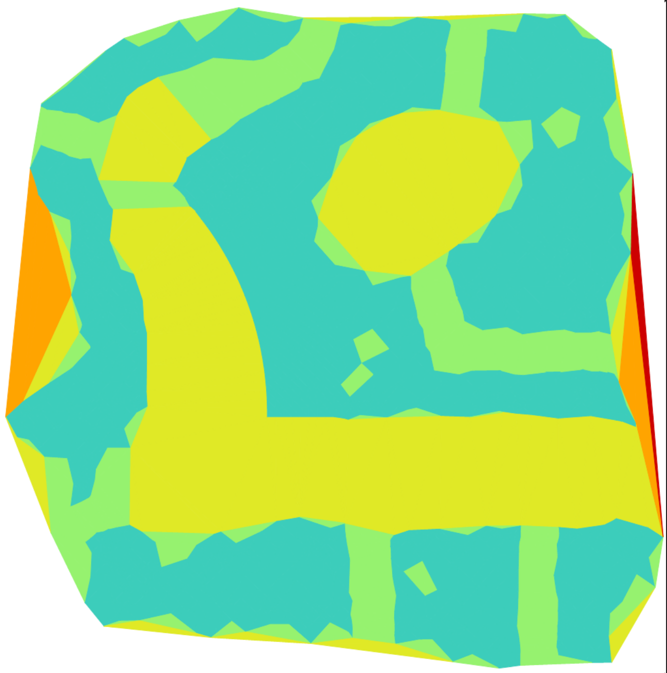
Model 7

Model 8

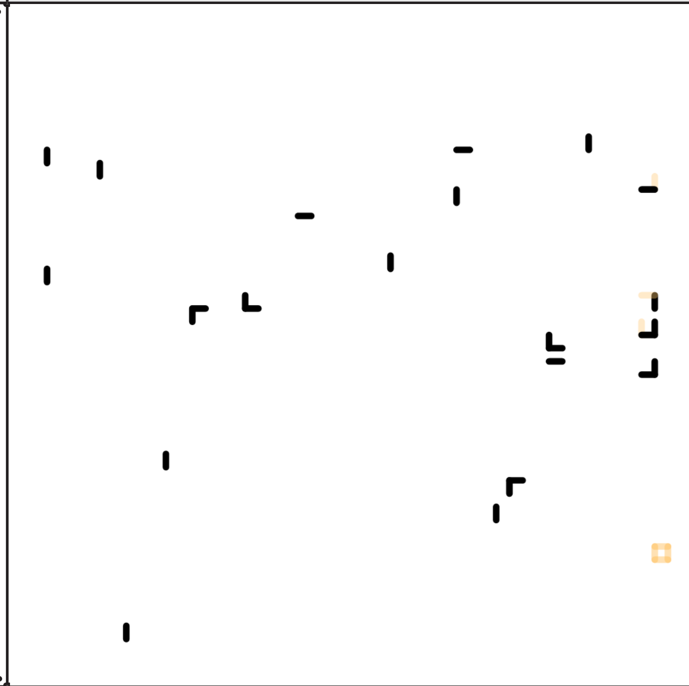
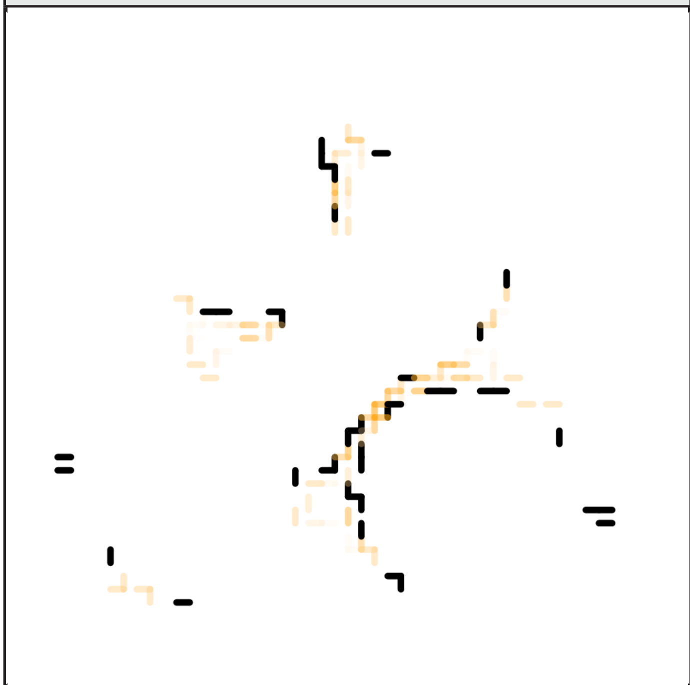
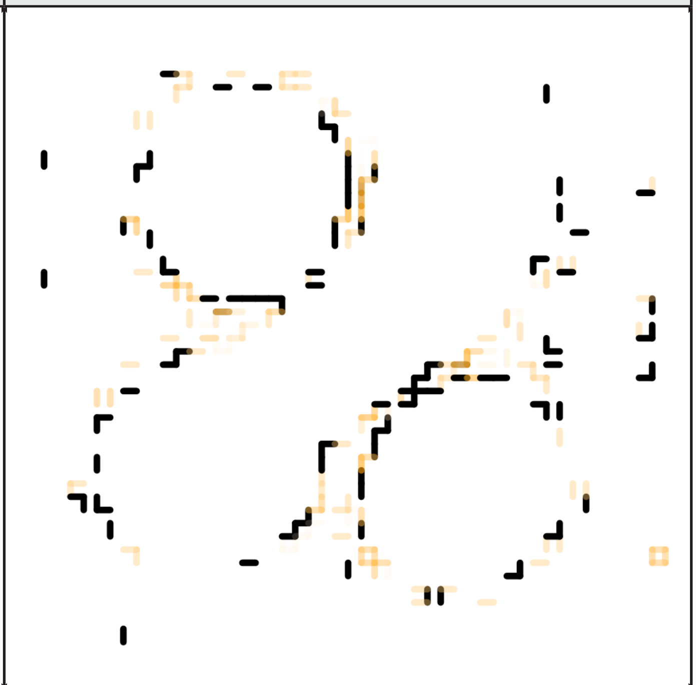
Shared Space

Model 1	*Model 2*
	
All triangles are identical, hence there is no difference in Shared Space.	Triangles differ, but *M2* has an equal distribution of particles and thus the areas of all particles are more or less the same. The edges are coloured since these particles were ignored during the equalization iteration.
Model 3	*Model 4*
	

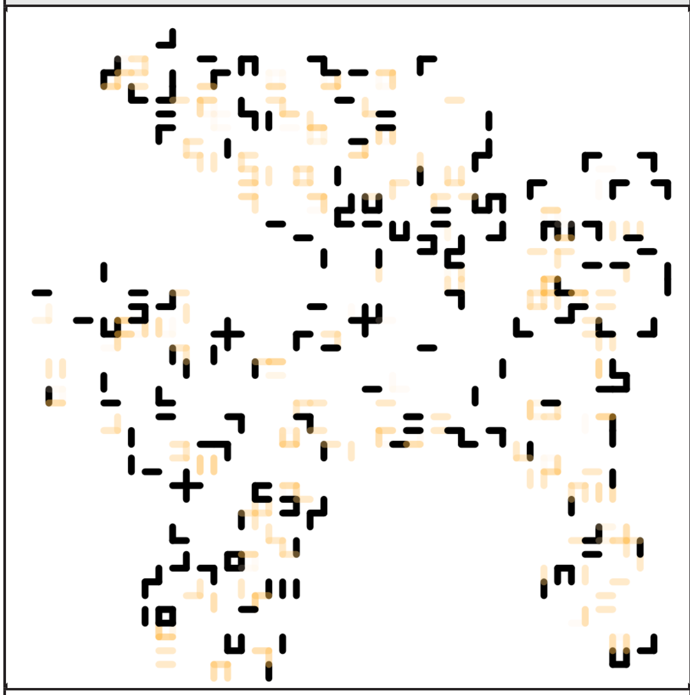
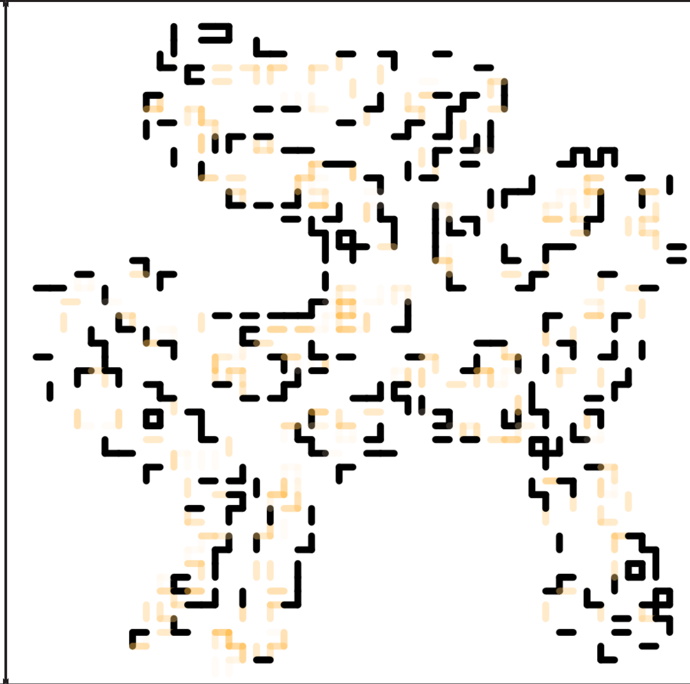
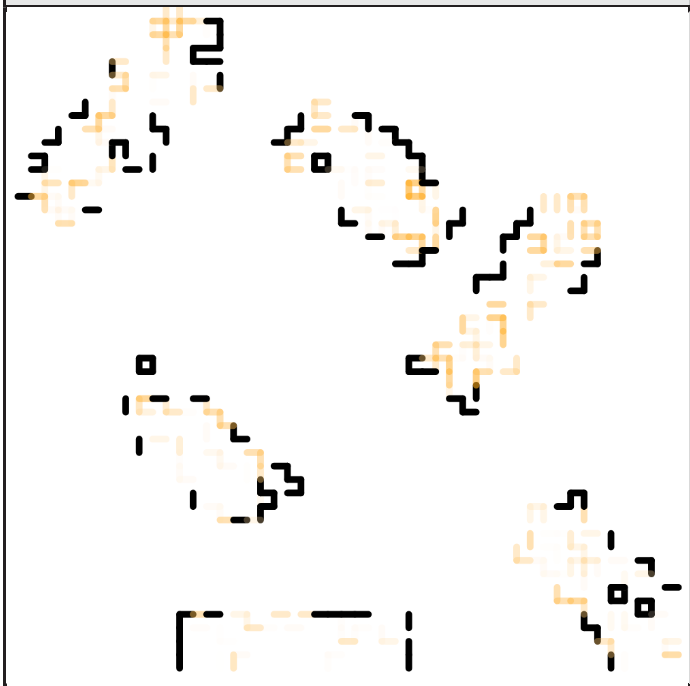
Shared Space

Model 5	*Model 6*
	
Red triangles exceed the preset meaning threshold. The occurrence here is a freak event.	Note how well the posterized areas mark the intended 'shape' of the spread. This is an indication our spread was drawn at the correct scale. Abnormal low or high densities tend to disrupt the contours.
Model 7	*Model 8*
	
Note the classification of open space into three different types. These roughly coincide with the human experience of open space. This is not an accident, the gradients have been calibrated with social distance [E. Stolk, 2002]	Even the difference between 'open space' and 'open corridors' is highlighted.

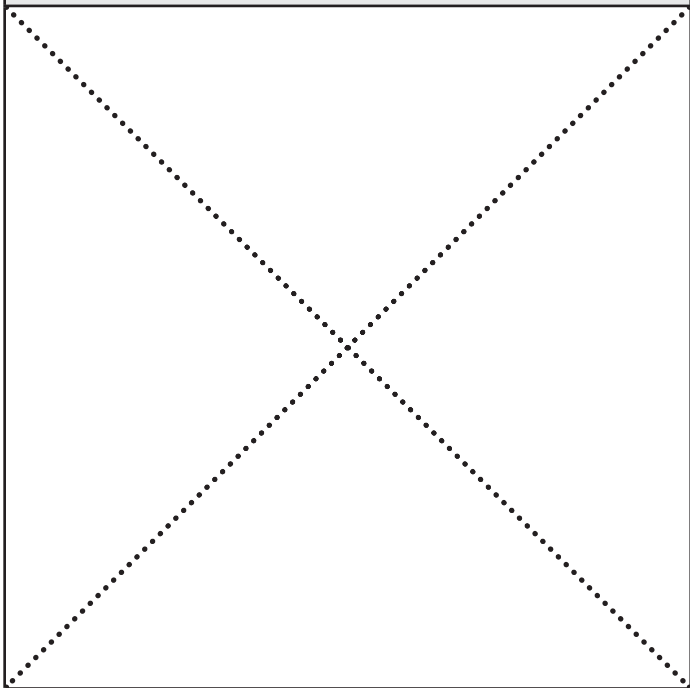
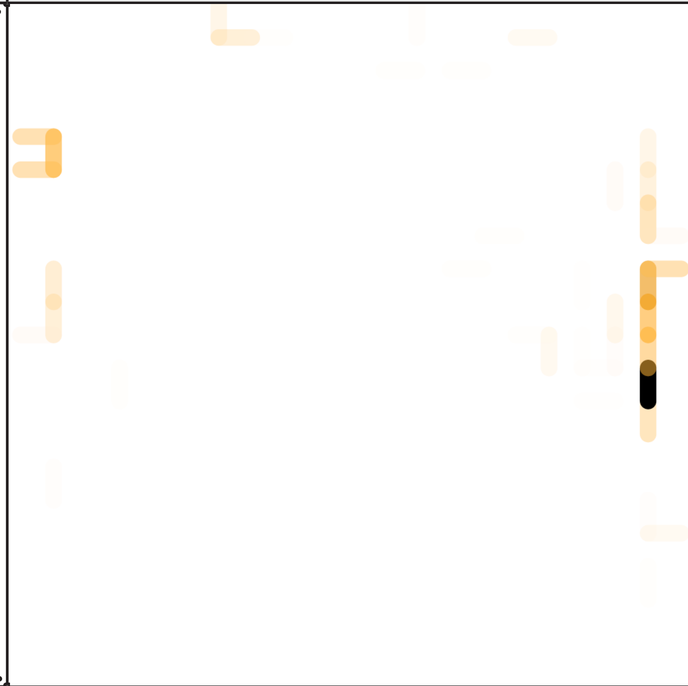
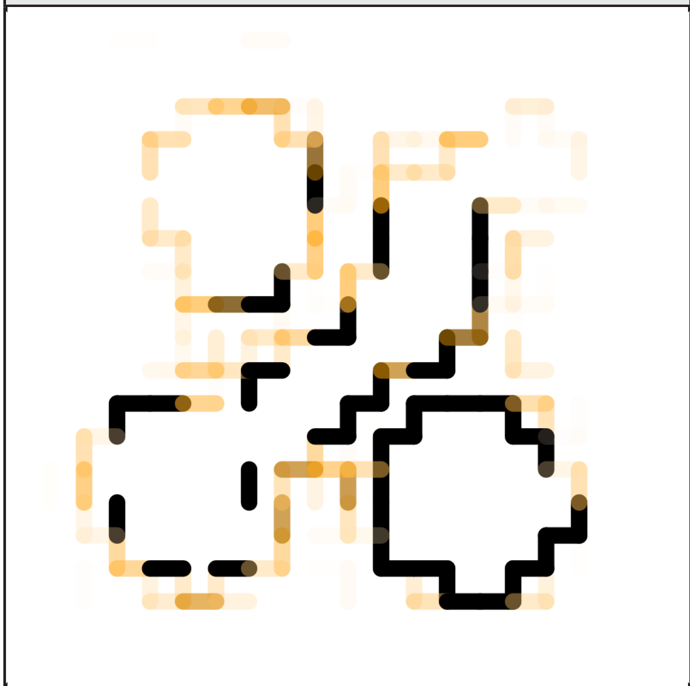
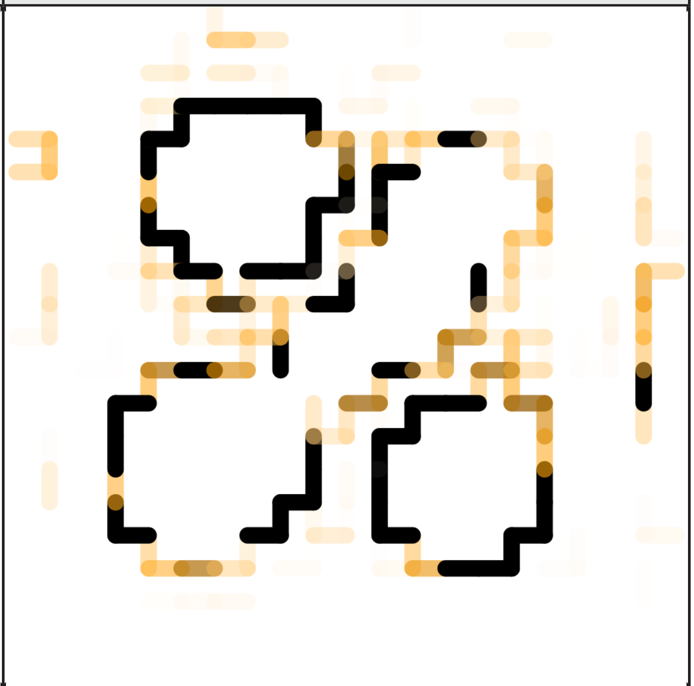
Contrast *{R=20m}*

Model 1	*Model 2*
	
M1 contains no contrast.	Contrast typically only makes sense when multiple lines are joined into contrast boundaries. In this case, the computed contrast is a result of noise and will probably not be experienced as strong contrast. Still, the definition of contrast is mathematically rigid.
Model 3	*Model 4*
	
Here -unlike *M2*- the contrasting transitions of the Search Grid line up to form connected contrast borders.	Note the difference in contrast between *M3* and *M4*. This was not terribly apparent from the spread property.

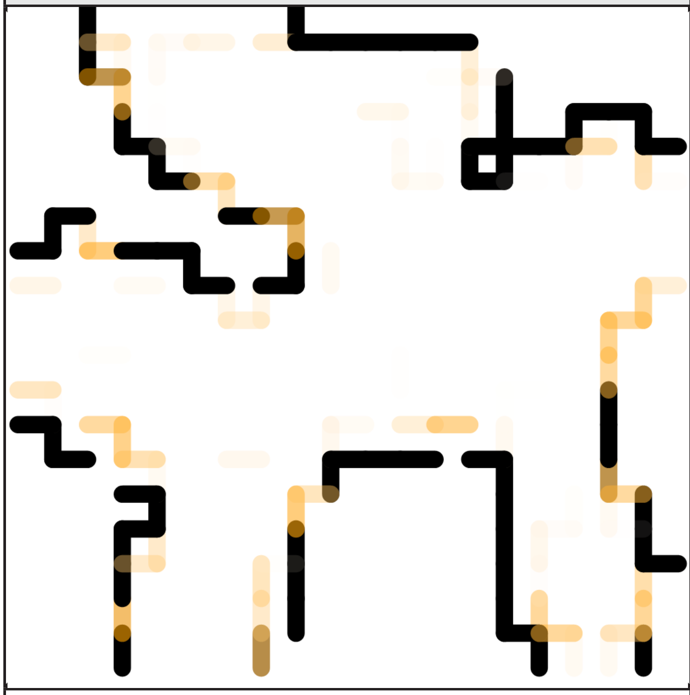
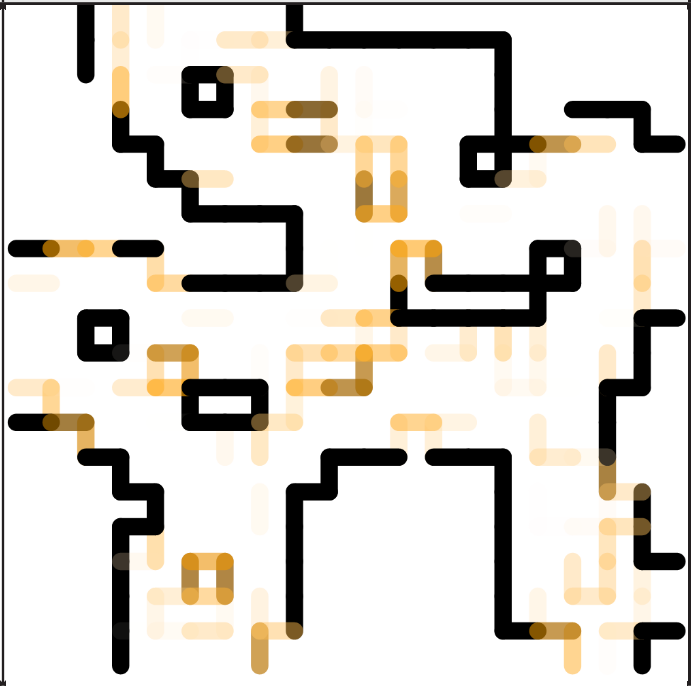
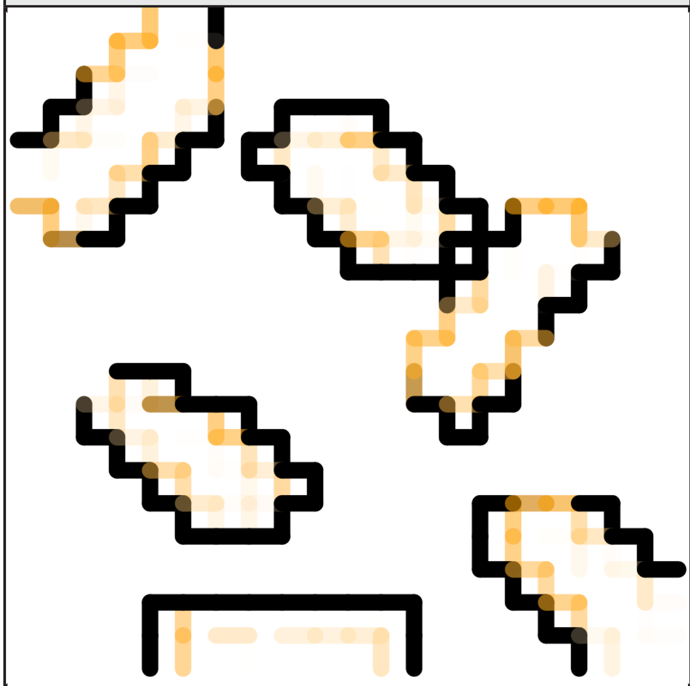
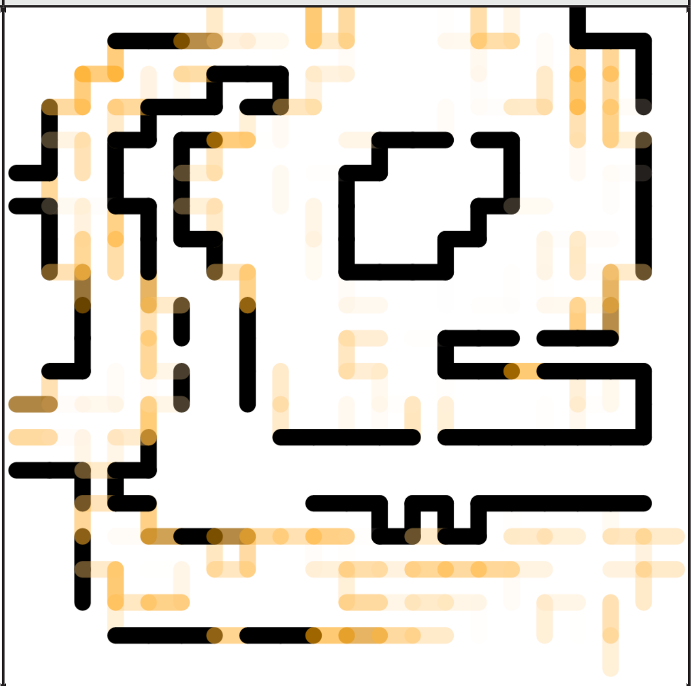
Contrast *{R=20m}*

Model 5	Model 6
	
With *M5* and *M6* the contrasting transitions form areas instead of curves. Since it is natural to view contrast as a boundary (transition), it is fairly hard to imagine a region with ‘constant contrast’. The mathematical definition of contrast does not exclude this.	
Model 7	Model 8
	
At this level of scale the outlines of the separate clusters are still quite permeable. I.e. pedestrians and cyclists should not be hindered while entering or leaving the clusters.	

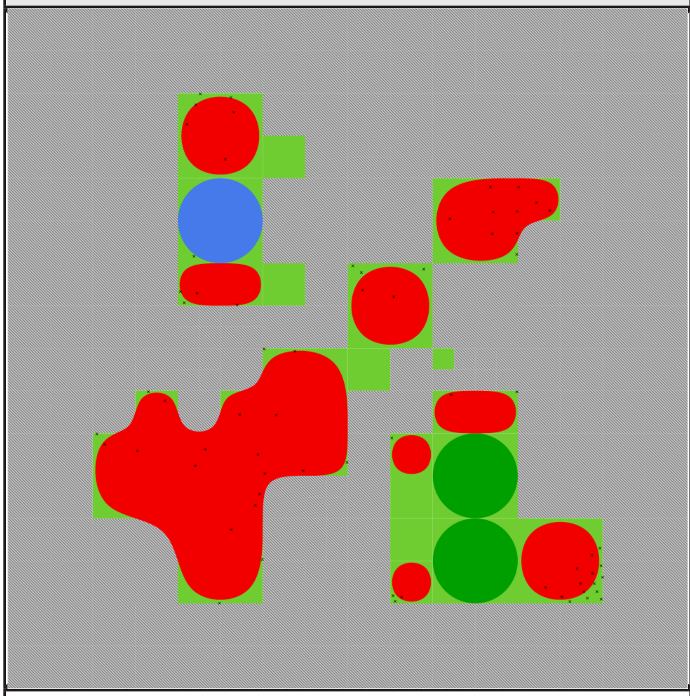
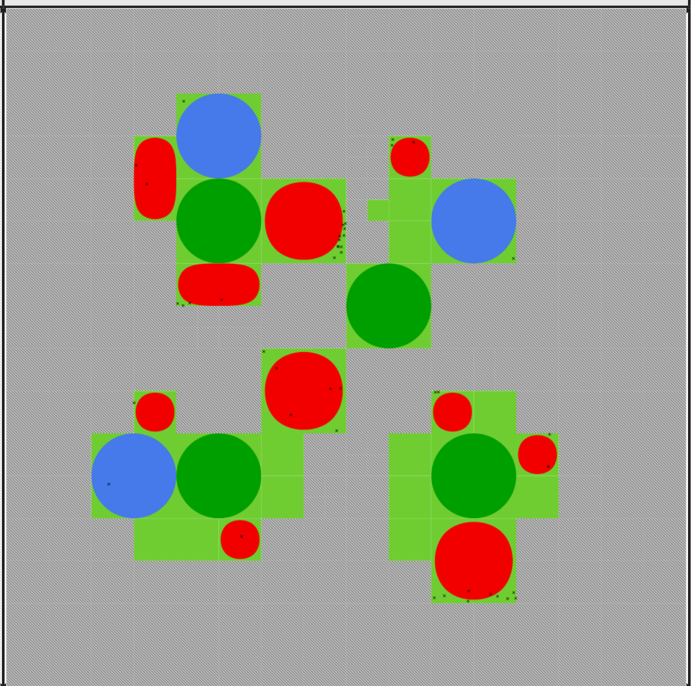
Contrast *{R=50m}*

Model 1	*Model 2*
	
M1 has no contrast.	Contrast has disappeared from this spread on this new level of scale. The remaining contrast is the result of the curved outline of the spread; i.e. the visible contrast here is between the particle cloud and the empty space beyond.
Model 3	*Model 4*
	
	The difference between *M3* and *M4* is far less clear when viewed at $R = 50$.

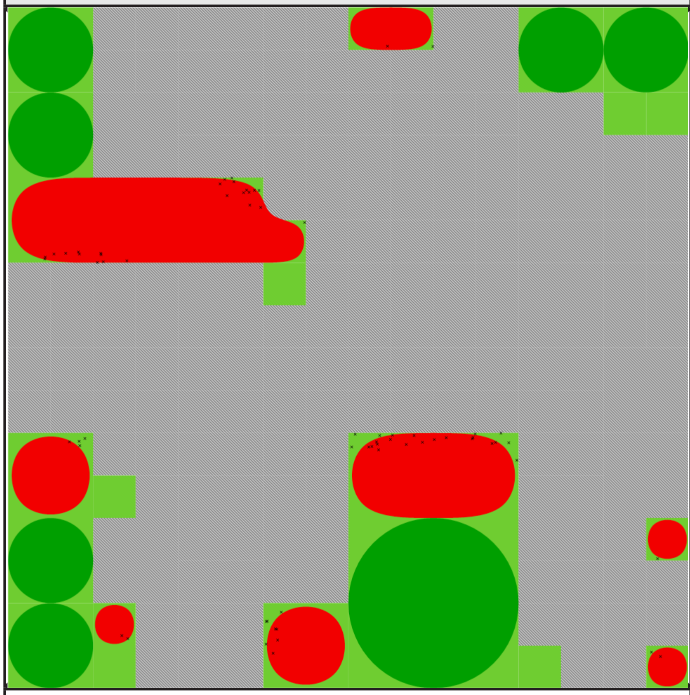
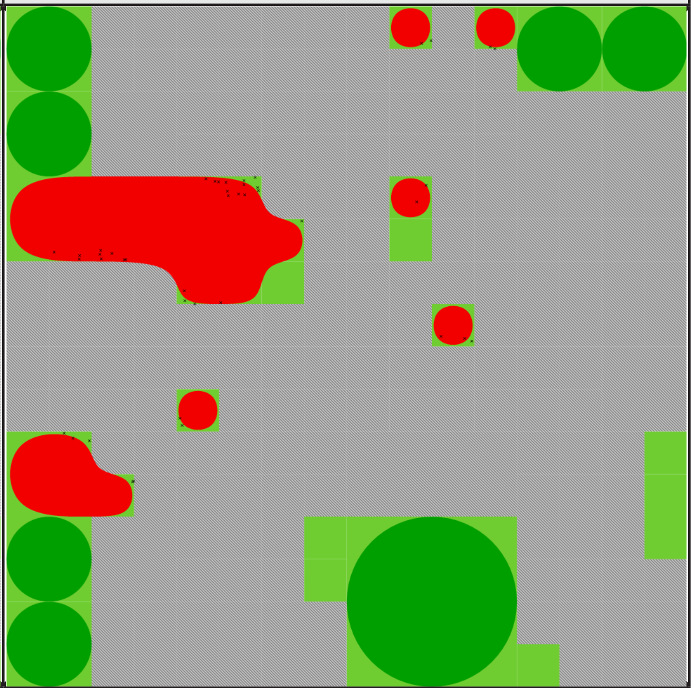
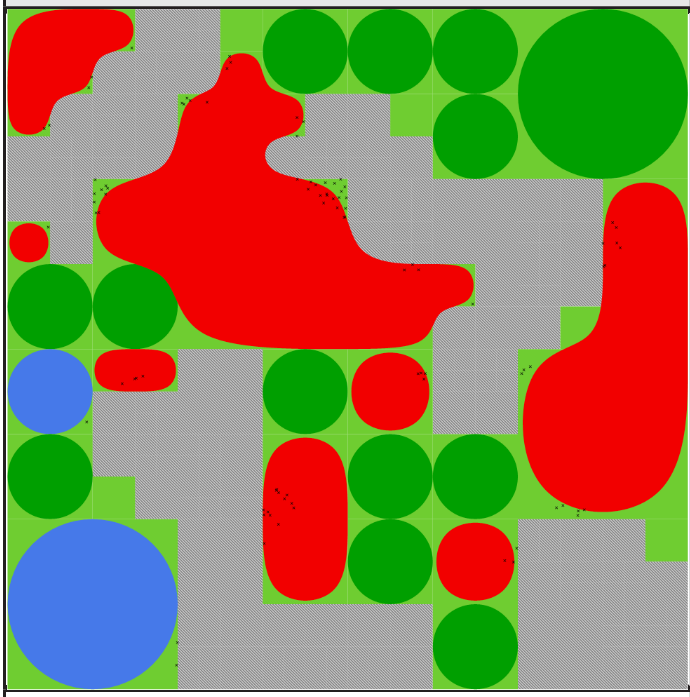
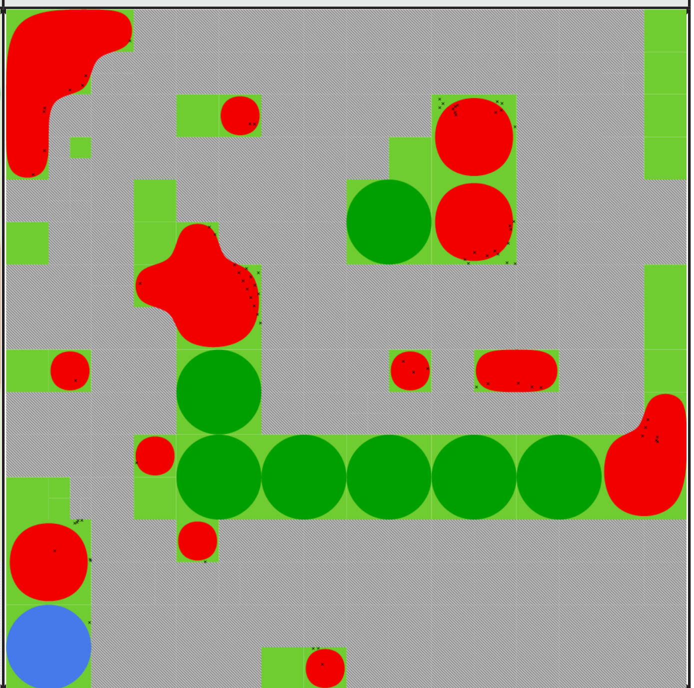
Contrast *{R=50m}*

Model 5	*Model 6*
	
Unlike the comparison between ***M3*** and ***M4***, ***M5*** and ***M6*** become less alike on this higher level of scale. A typical example of scale-paradox [T.M. de Jong, 1992].	
Model 7	*Model 8*
	
On this level of scale the clusters are very pronounced. I.e. heavy traffic will experience these clusters as obstacles.	***M8*** has -just like ***M7***- a much stronger contrast on $R = 50$. Apparently the open spaces in this model are meant for slow traffic.

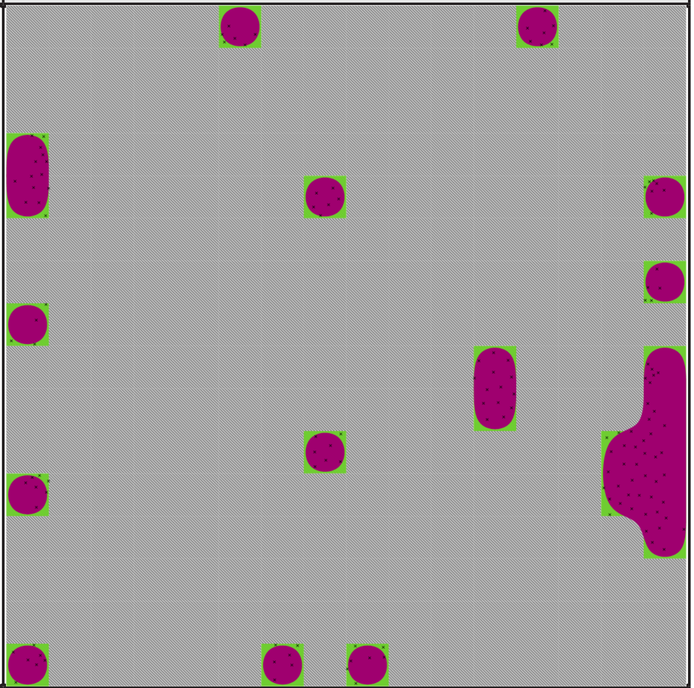
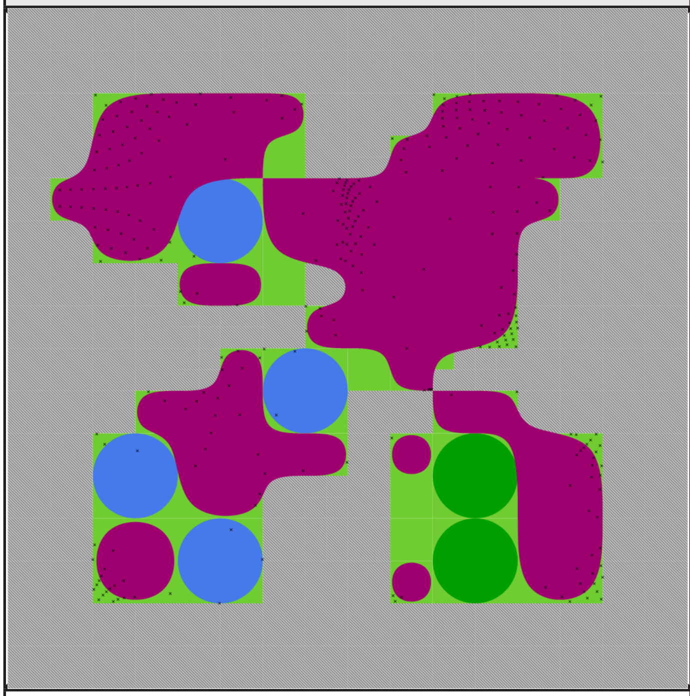
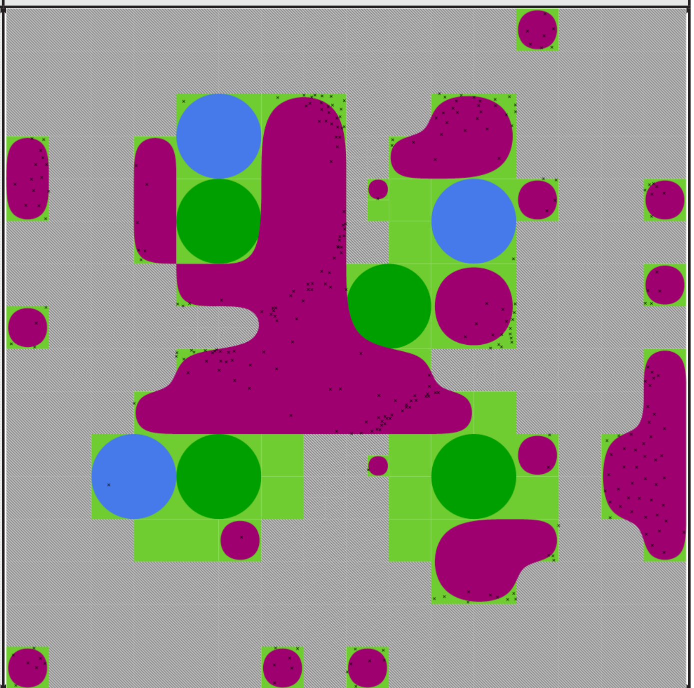
Sprawl *{min = 1.0; max = 10.0}*

Model 1	*Model 2*
M1 contains no sprawl.	***M2*** contains no sprawl.
Model 3	*Model 4*
	
	M4 contains less sprawl than ***M3*** for the same reason why ***M4*** contains more contrast than ***M3***. This difference is predictable when knowledge about contrast is available.

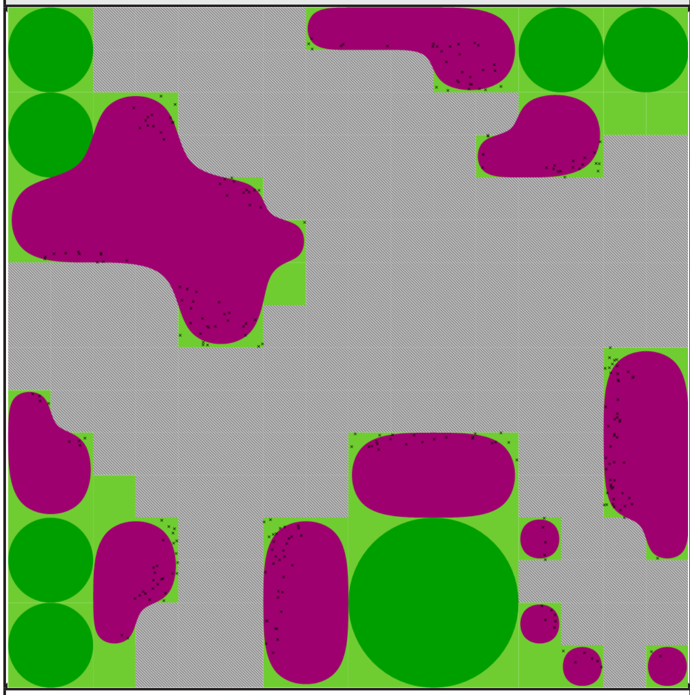
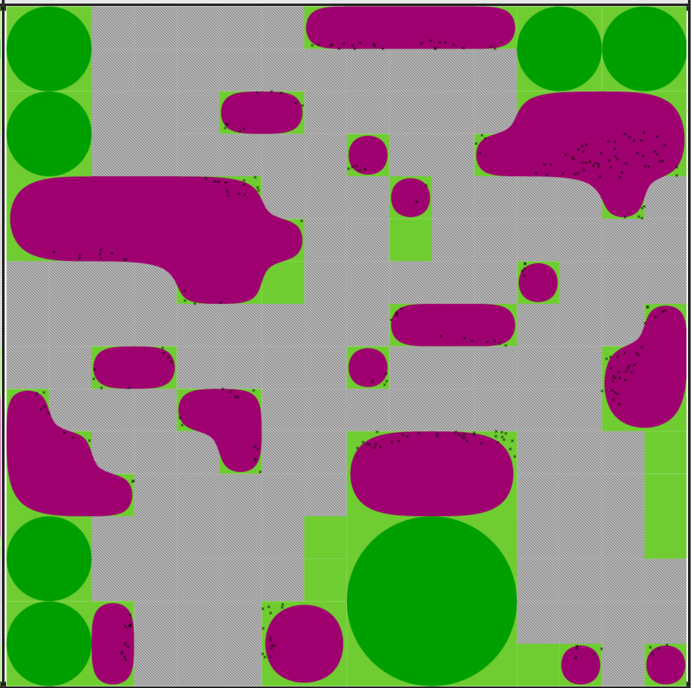
Sprawl *{min = 1.0; max = 10.0}*

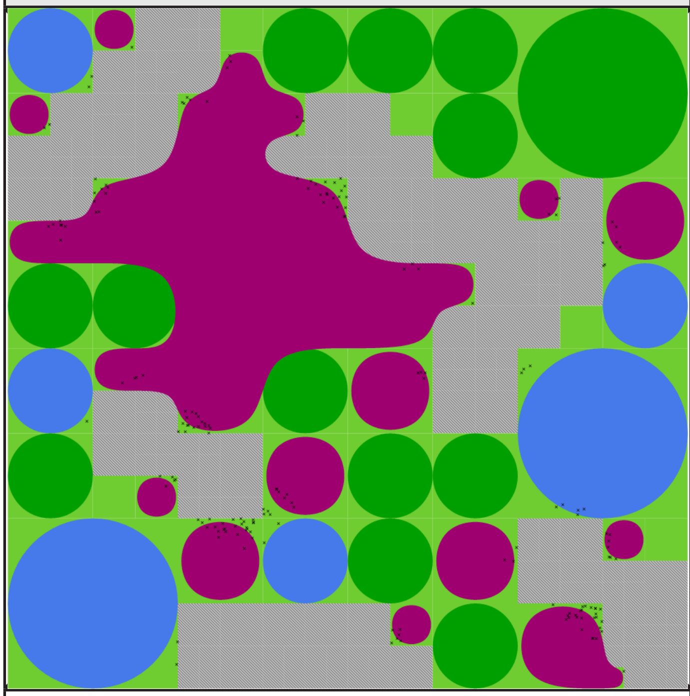
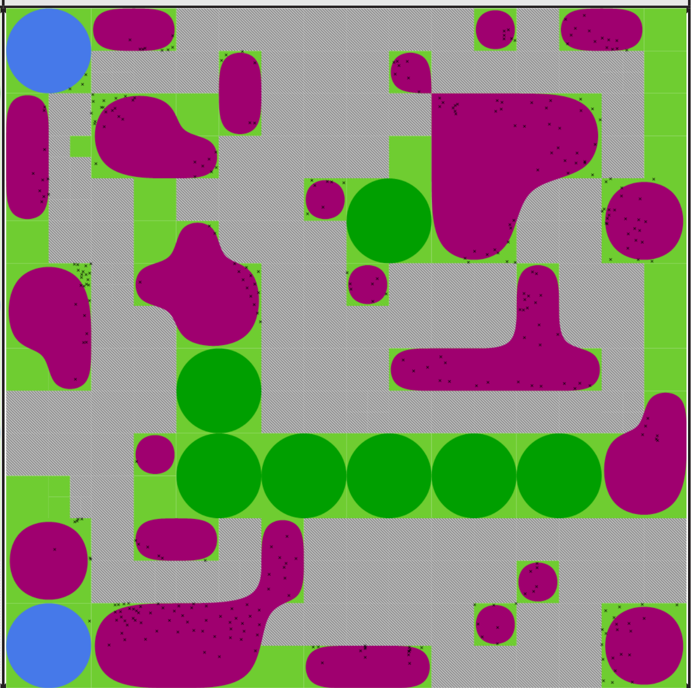
Model 5	*Model 6*
	
	Sprawl is fairly identical for ***M5*** and ***M6***. Interestingly, the central open space of ***M6*** is barely manifested in the sprawl diagram.
Model 7	*Model 8*
	
Note that the sprawl algorithm does not handle this case well. The large red blob in the middle is mostly empty, yet is has been designated as sprawl.	Here -unlike with ***M6***- the central open spaces are reflected by the sprawl map.

Sprawl *{min = 2.0; max = 20.0}*

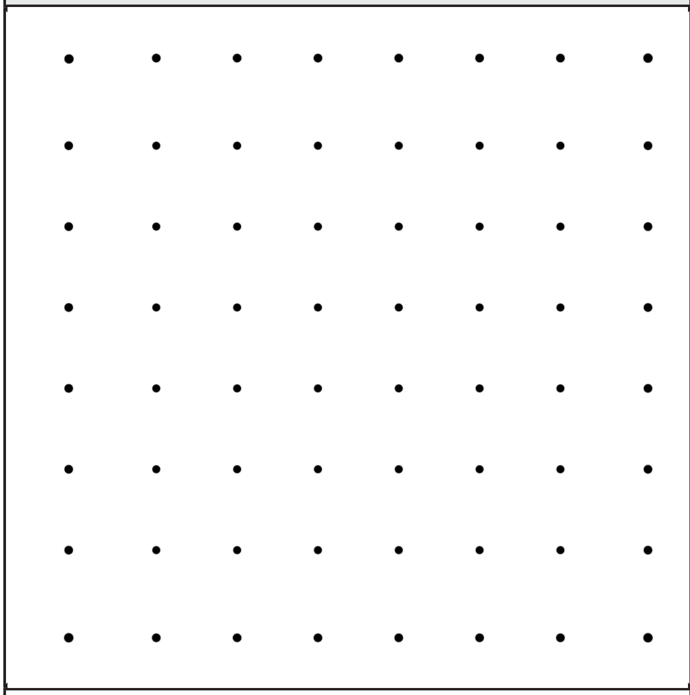
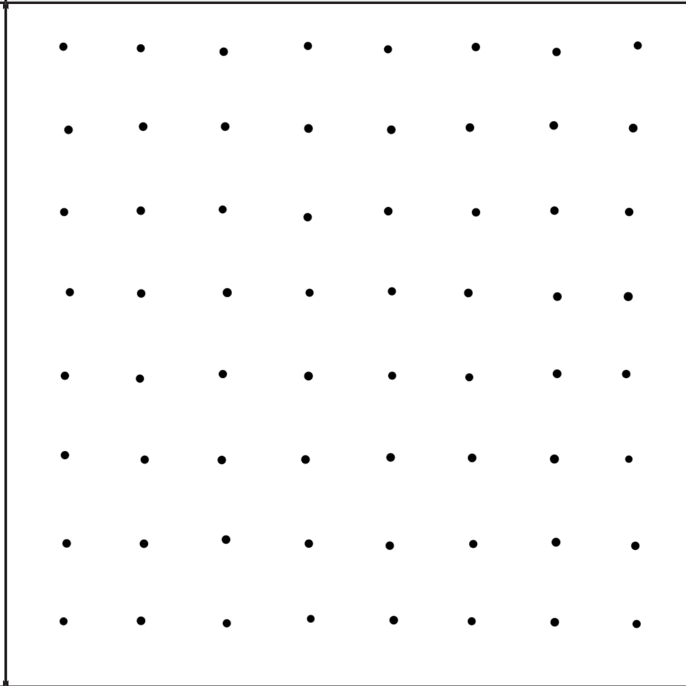
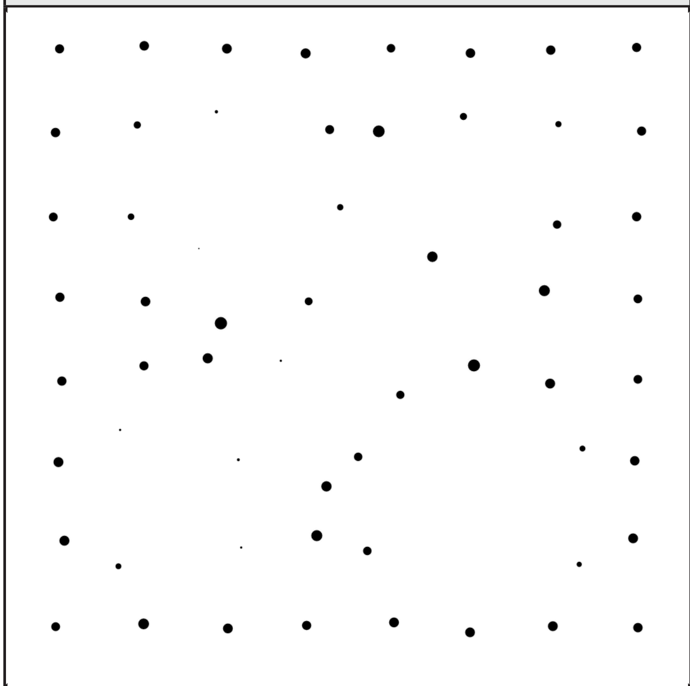
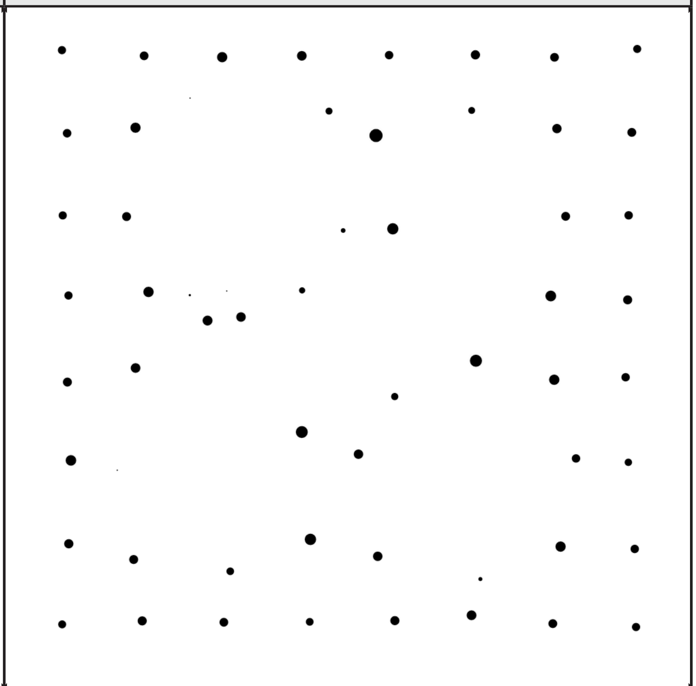
Model 1	*Model 2*
	
M1 contains no sprawl.	Small isolated sprawl groups should not be treated as a negative characteristic of a spread. The problem with sprawl is that it occupies **large** amounts of surface area.
Model 3	*Model 4*
	
Sprawl is much more serious here. Approximately 50% of the build surface area has been designated as sprawl.	Sprawl is slightly less severe in ***M4*** than ***M3***. Also note that the current sprawl domain extends into rather dense urban weave. 20 dwellings per hectare is not necessarily something to be ashamed of.

Sprawl *{min = 2.0; max = 20.0}*

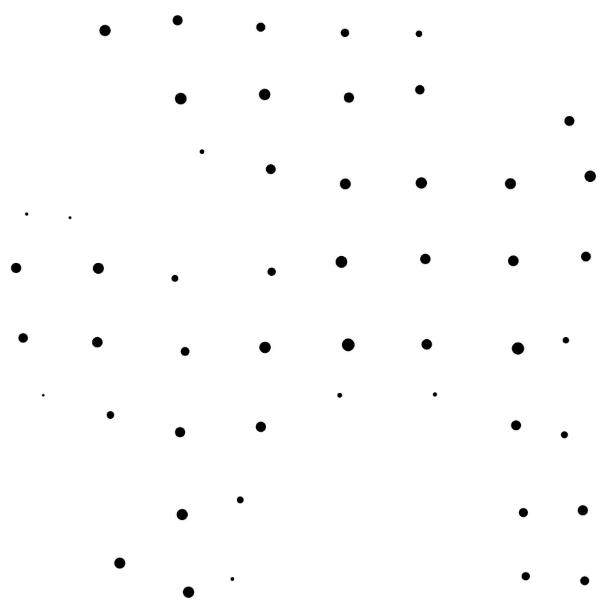
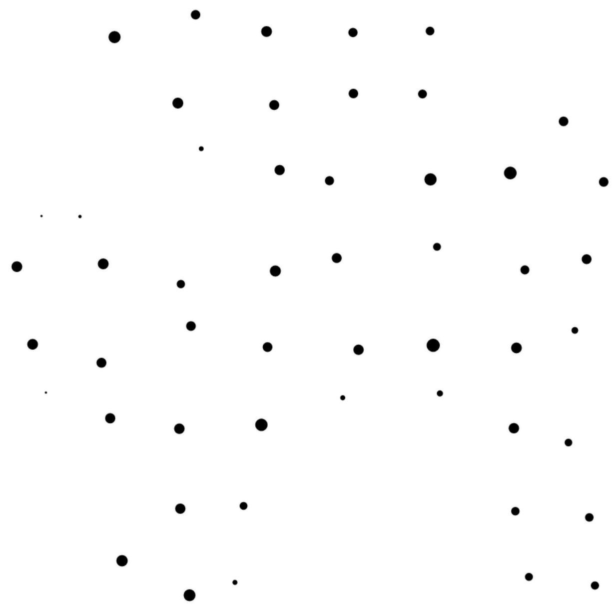
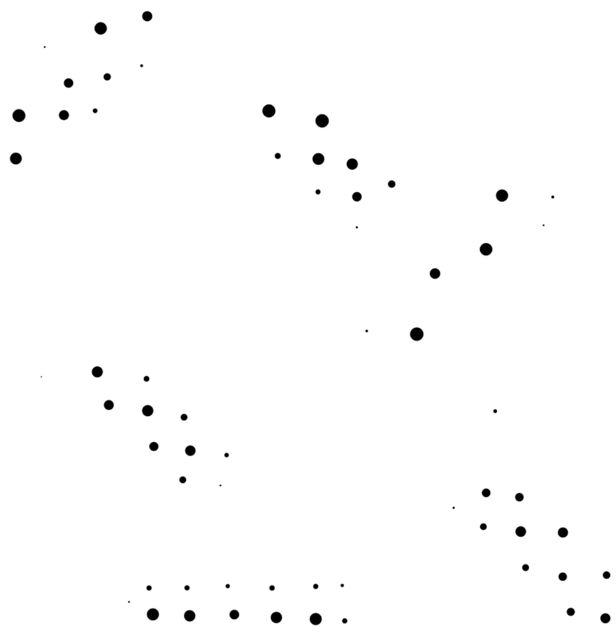
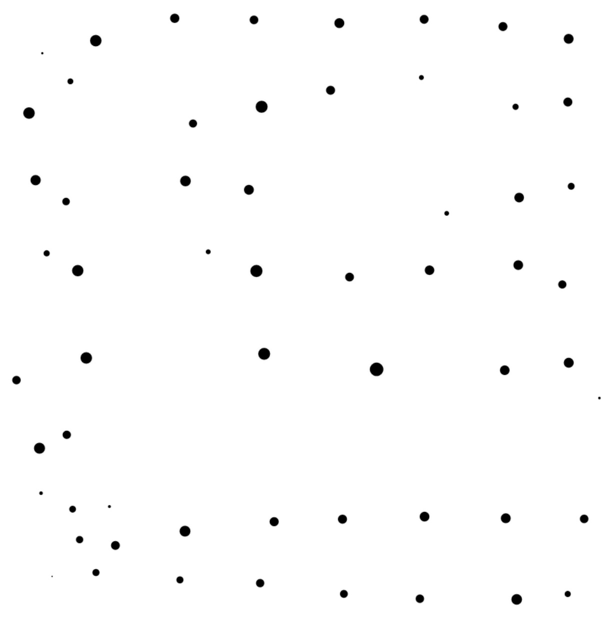
Model 5	Model 6
	
	Again, only a minimal difference exists between ***M5*** and ***M6***.

Model 7	Model 8
	
The same problems apply to this sprawl map as to the previous one. One possible solution for this erroneous behaviour of Quad Trees would be to add an additional subdivision threshold which is linked by the surface area of the Quad branch.	

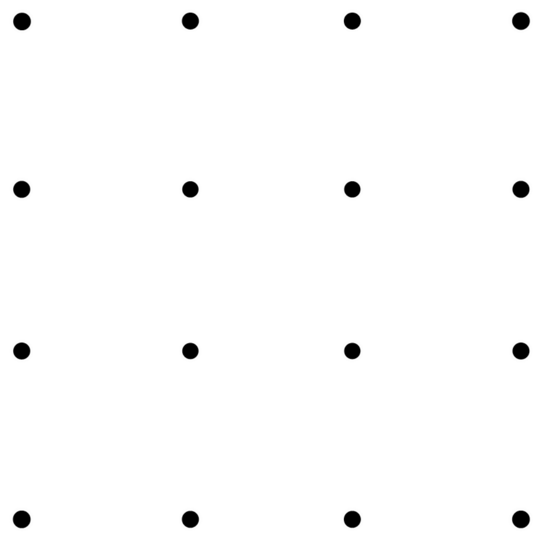
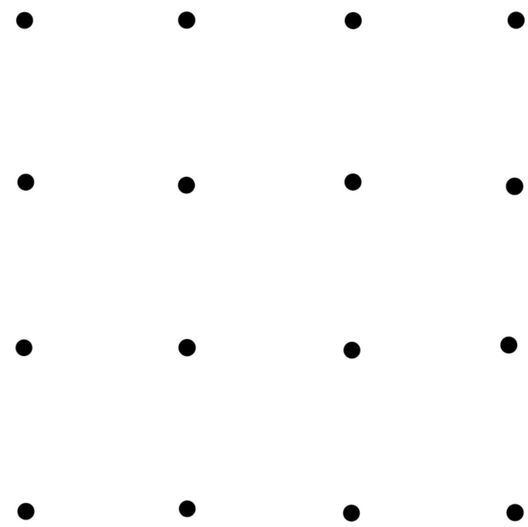
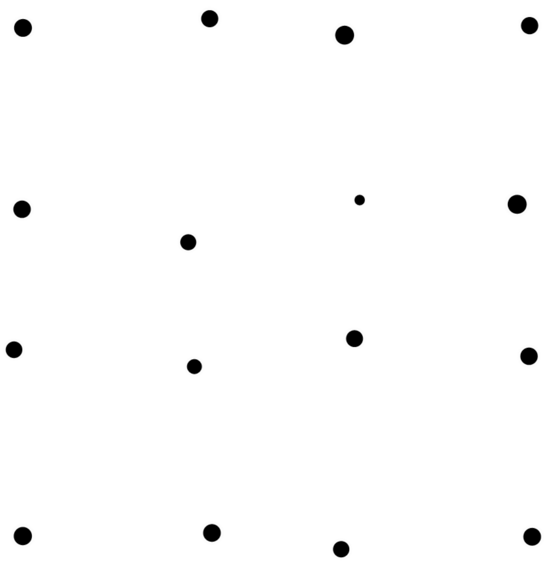
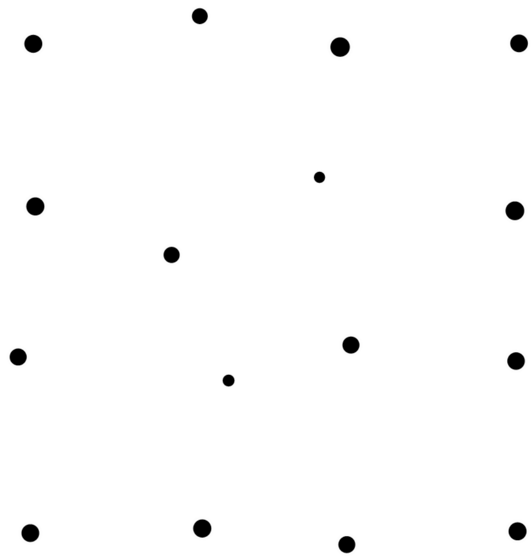
Reduction *{T = 100}*

Model 1	*Model 2*
	
M1 contains 64 reduction groups.	***M2*** contains 64 reduction groups. ***M1*** and ***M2*** do not differ at this reduction level.
Model 3	*Model 4*
	
M3 contains 59 reduction groups.	***M4*** contains 59 reduction groups. ***M3*** and ***M4*** barely differ at this reduction level.

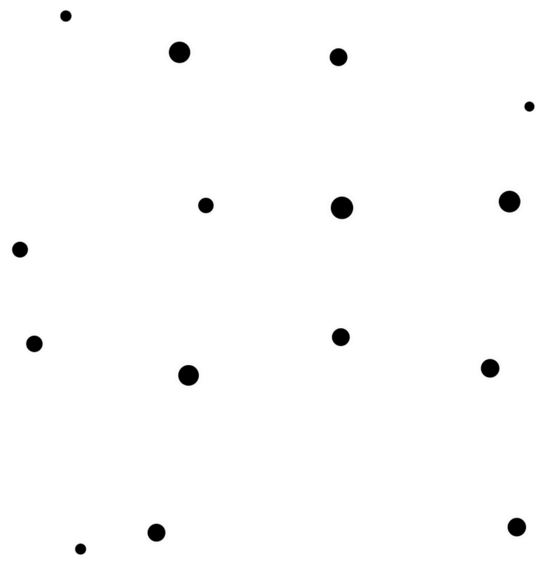
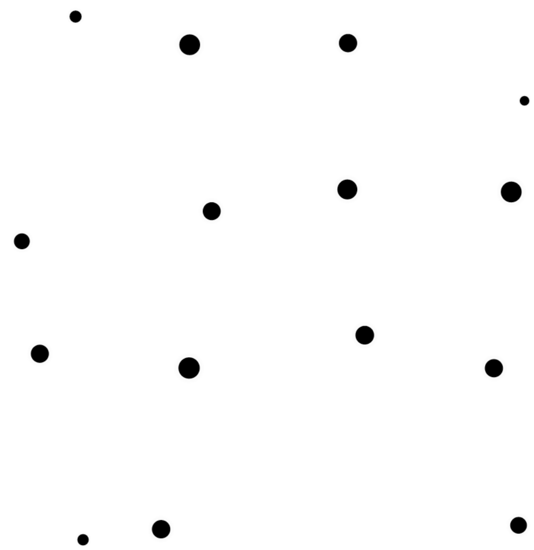
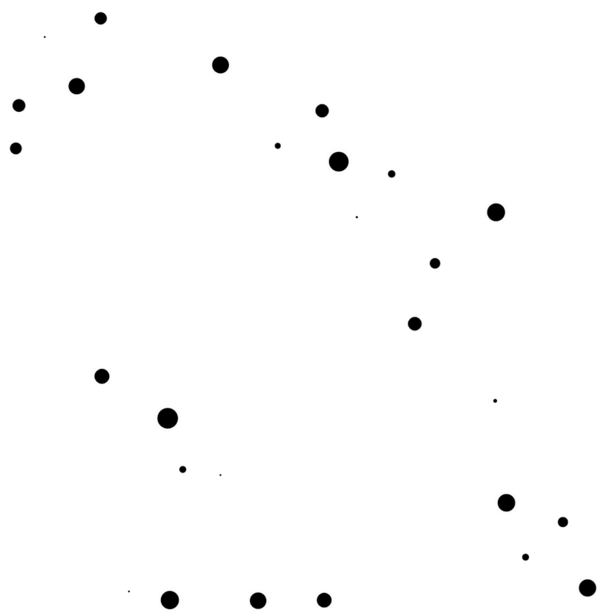
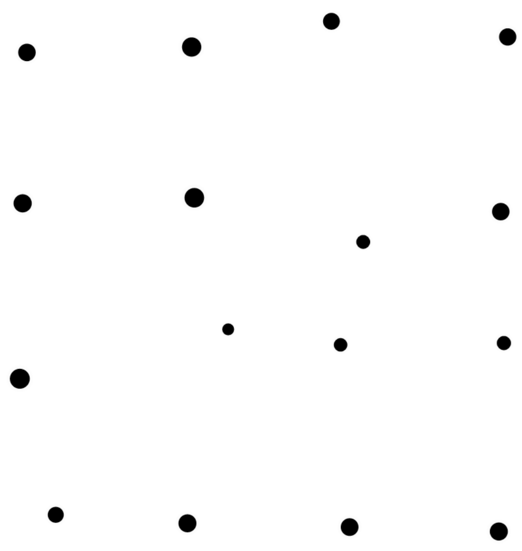
Reduction *{T = 100}*

Model 5	*Model 6*
	
M5 contains 51 reduction groups.	***M6*** contains 51 reduction groups. ***M5*** and ***M6*** barely differ at this reduction level.
Model 7	*Model 8*
	
M7 contains 62 reduction groups, but many of them are relatively small.	***M8*** contains 59 reduction groups.

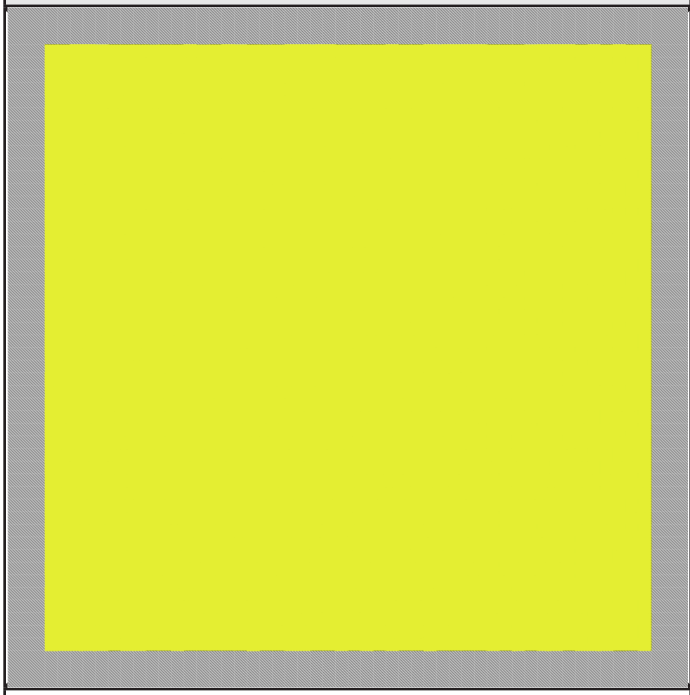
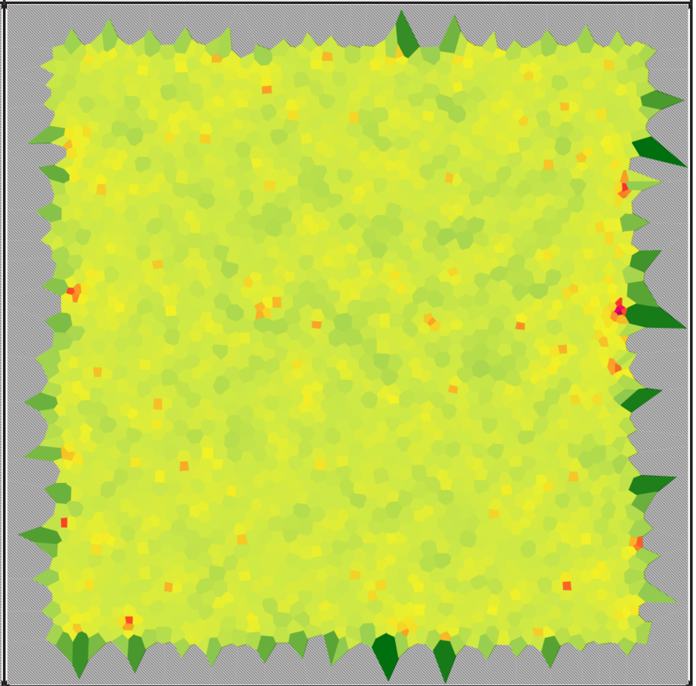
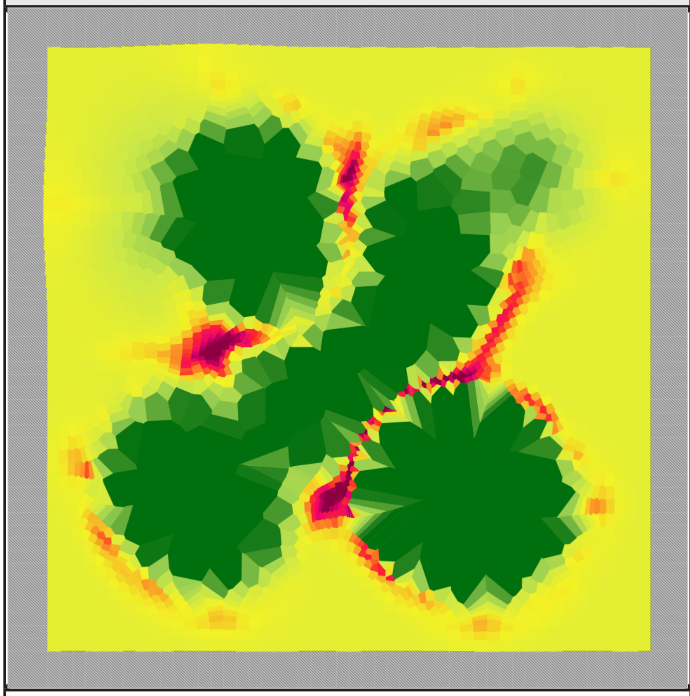
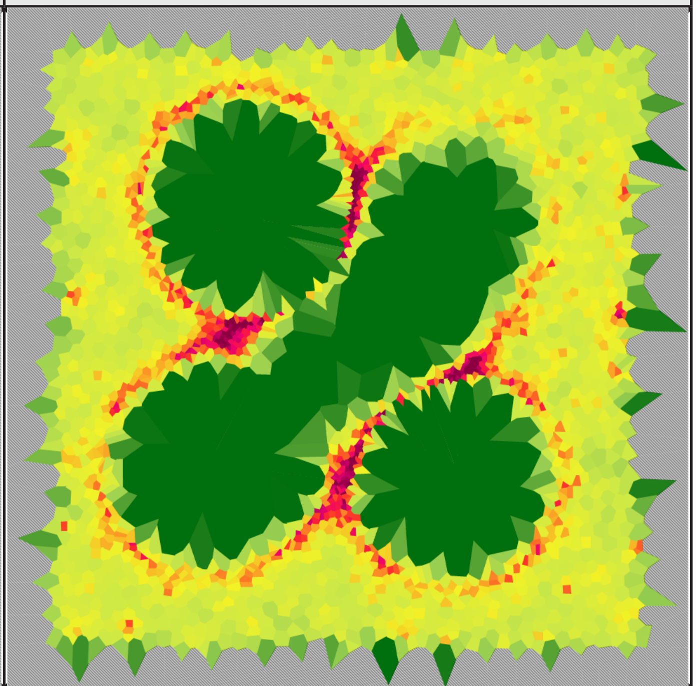
Reduction *{T = 300}*

Model 1	*Model 2*
	
M1 contains 16 reduction groups.	***M2*** contains 16 reduction groups. Differences only decrease when ***T*** increases.
Model 3	*Model 4*
	
M3 contains 16 reduction groups.	***M4*** contains 16 reduction groups.

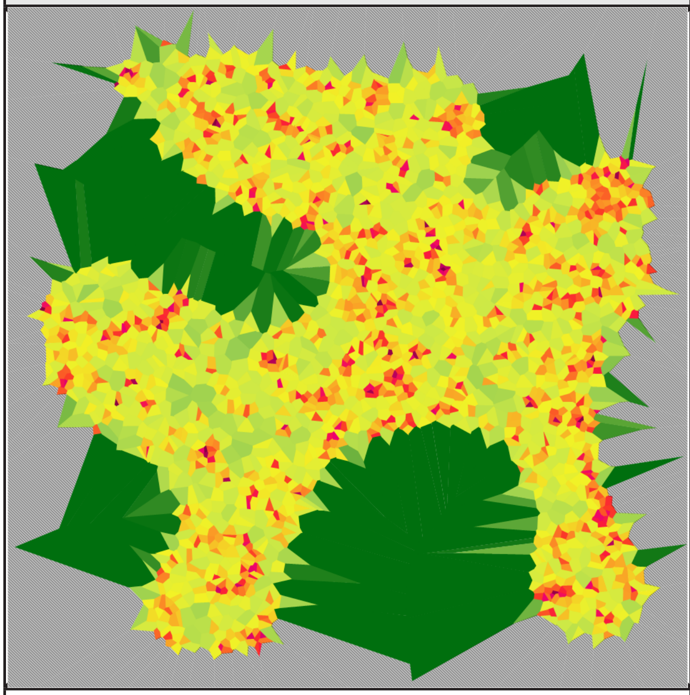
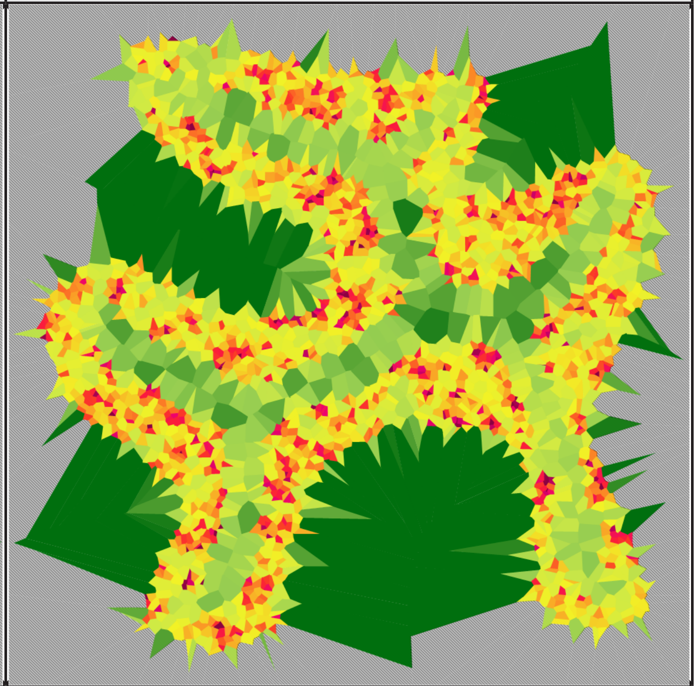
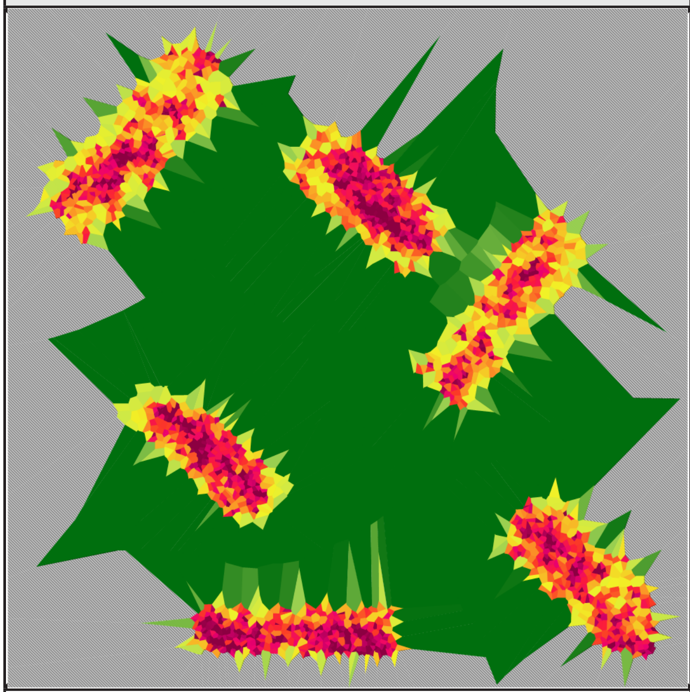
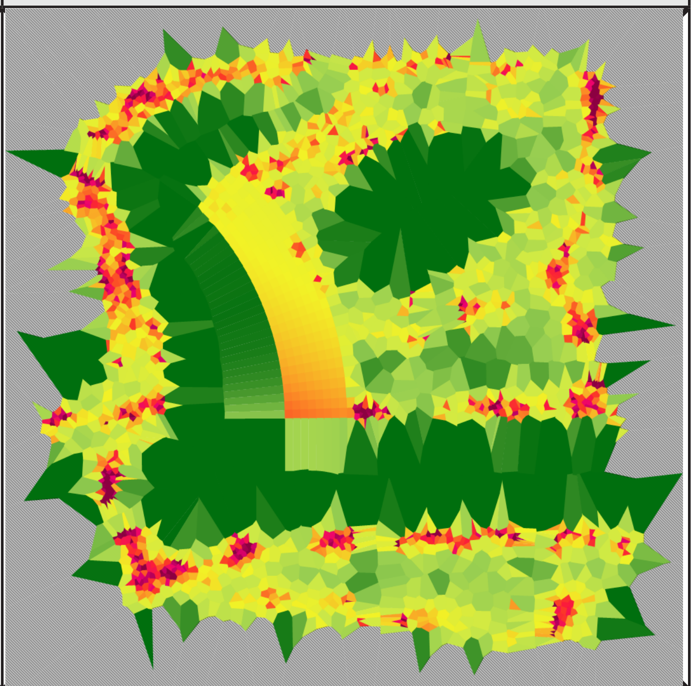
Reduction *{T = 300}*

Model 5	Model 6
	
M5 contains 15 reduction groups.	***M6*** contains 15 reduction groups.
Model 7	Model 8
	
M7 contains 27 reduction groups.	***M8*** contains 16 reduction groups. Model 8 is starting to look like Model 1~4.

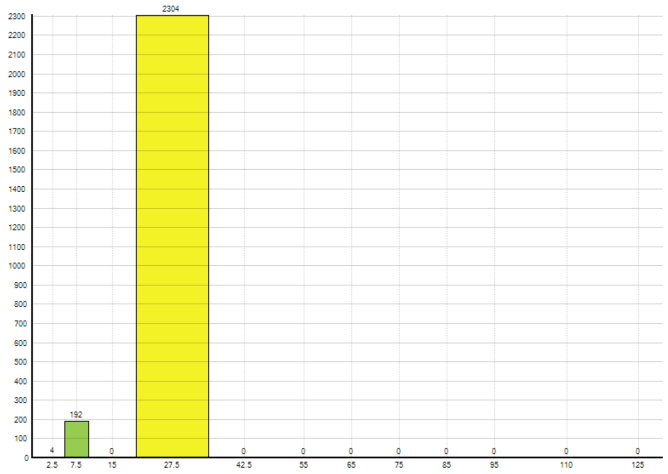
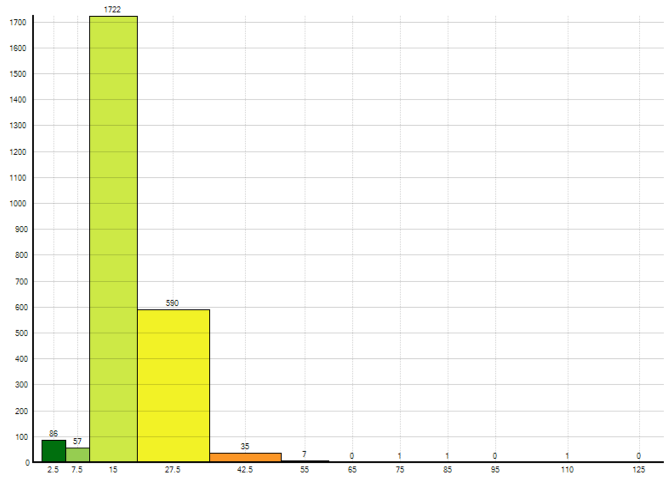
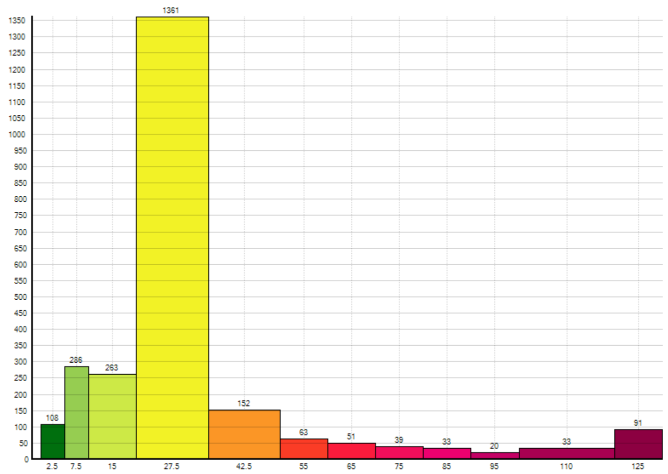
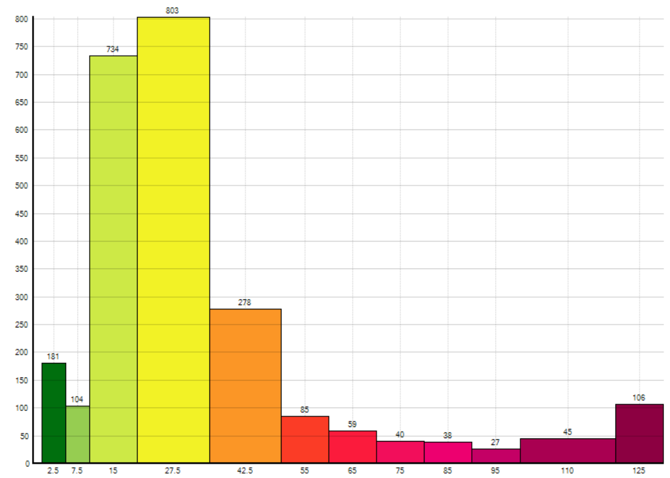
Voronoi density

Model 1	*Model 2*
	
All voronoi cells are identical, hence the density is identical. Cells along the edges are ignored.	Global density is identical to ***M1***, but local density differs. This visual difference may not be as large as it appears (see [Voronoi Density Histogram]).
Model 3	*Model 4*
	
	Clearly a randomized spread allows for better distortion than a gridded spread. Unlike ***M3***, the density contours are coincident with the bulges.

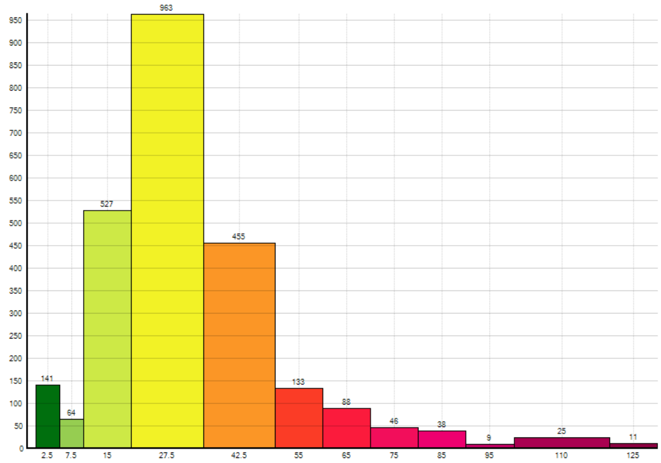
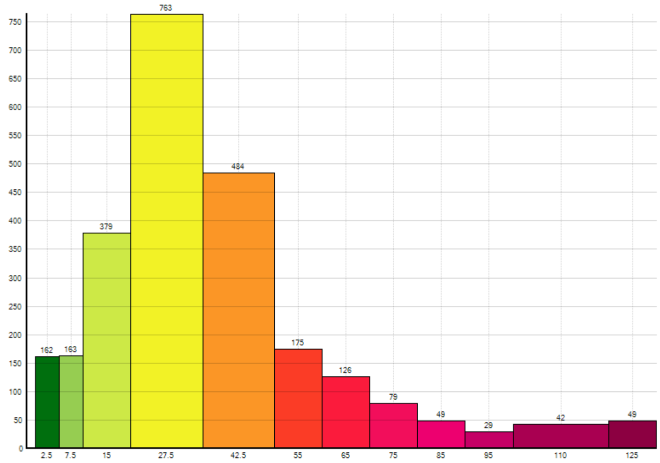
Voronoi density

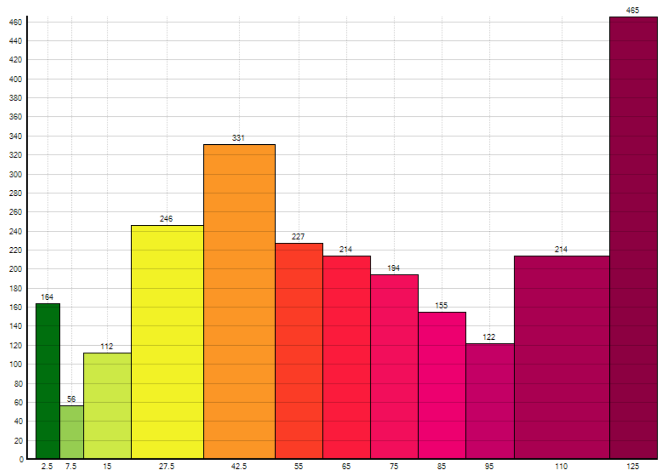
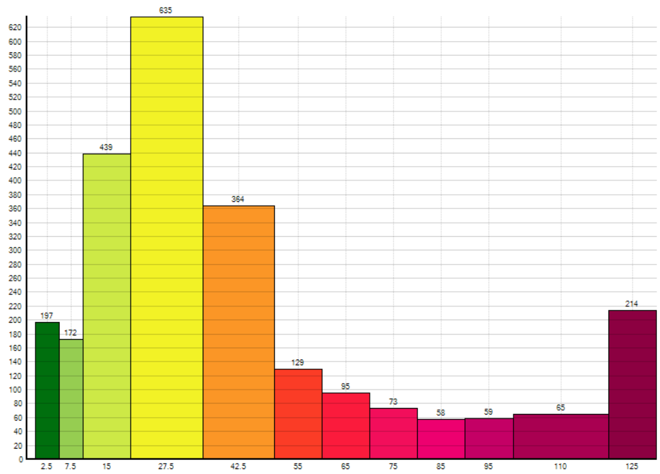
Model 5	*Model 6*
	
M5 shows the typical noise which occurs when a completely random model is analysed.	
Model 7	*Model 8*
	
	The protrusions of yellowish cells into the green open spaces of ***M8*** is not a realistic representation of private space. Search Grid and Quad Tree density algorithms perform better here.

Voronoi density histogram

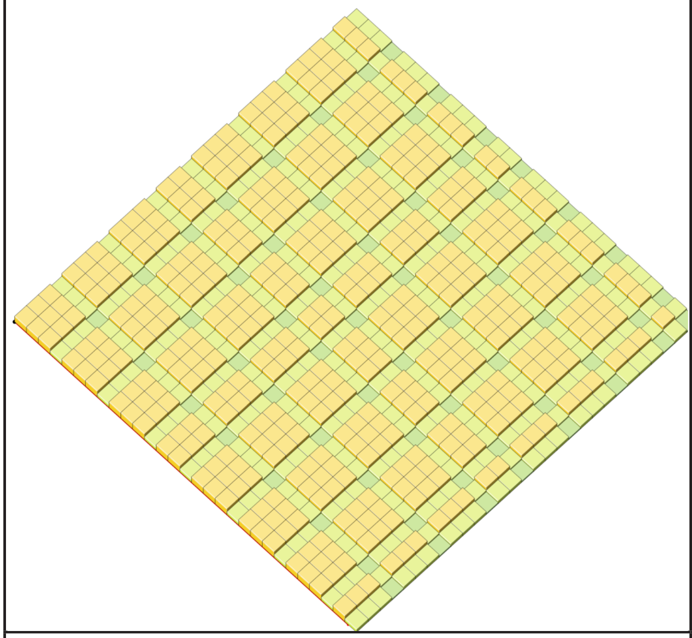
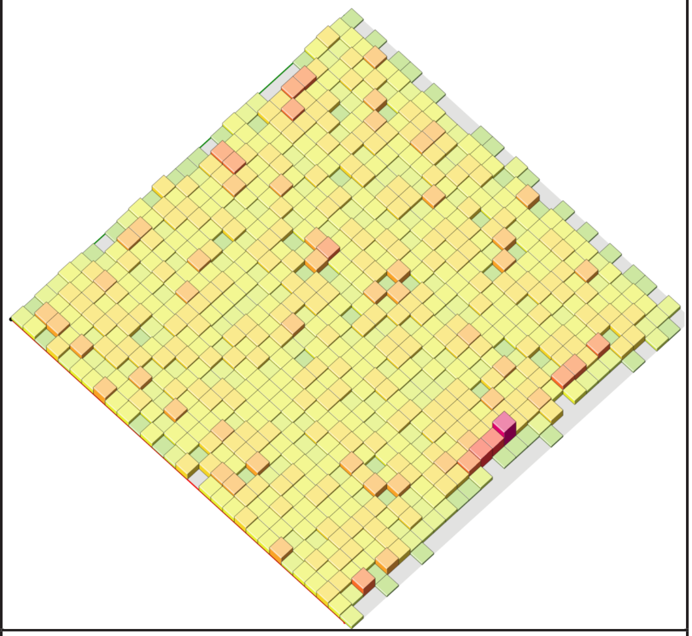
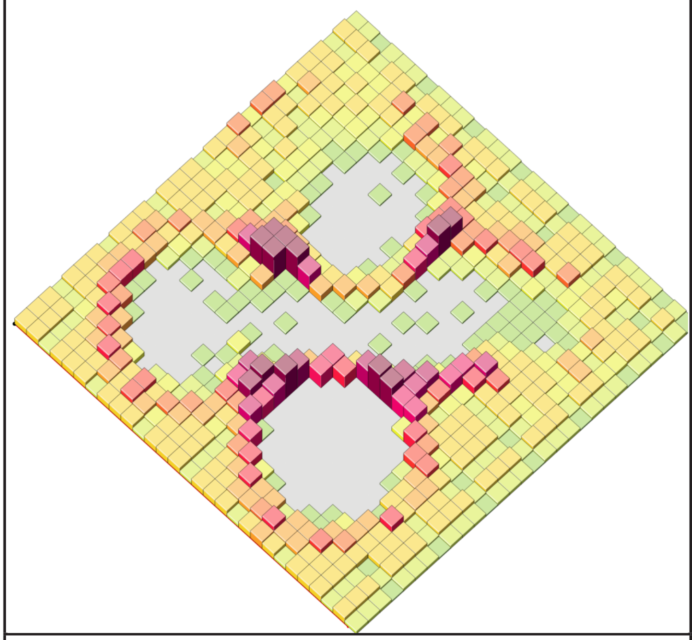
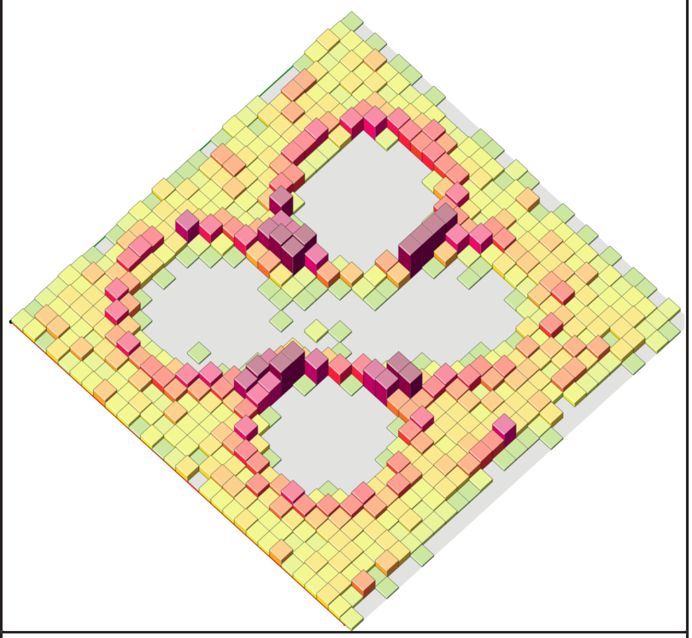
<table>
<tr><th>Model 1</th><th>Model 2</th></tr>
<tr><td></td><td></td></tr>
<tr><td>The small green bar is a result of the boundary edges. They have not been ignored in this histogram.</td><td>The difference between M1 and M2 is indeed significant.</td></tr>
<tr><th>Model 3</th><th>Model 4</th></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td>The properties Density and Sprawl have a large overlap. In effect, the amount of Sprawl can be more accurately judged by Density histograms than by the Sprawl maps. Sprawl maps provide information about where Sprawl occurs.</td></tr>
</table>

Voronoi density histogram

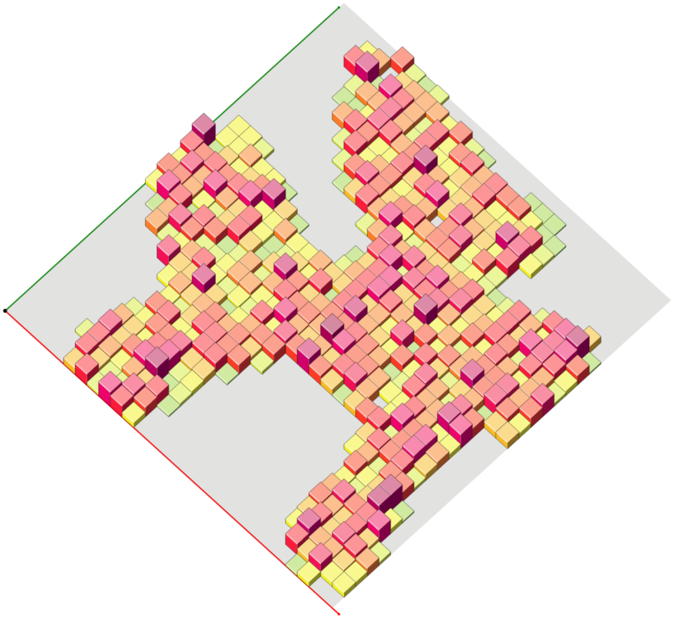
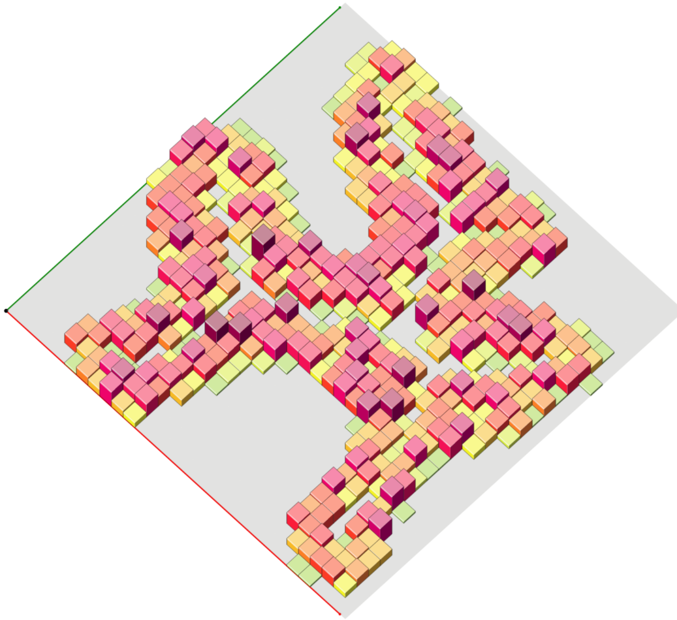
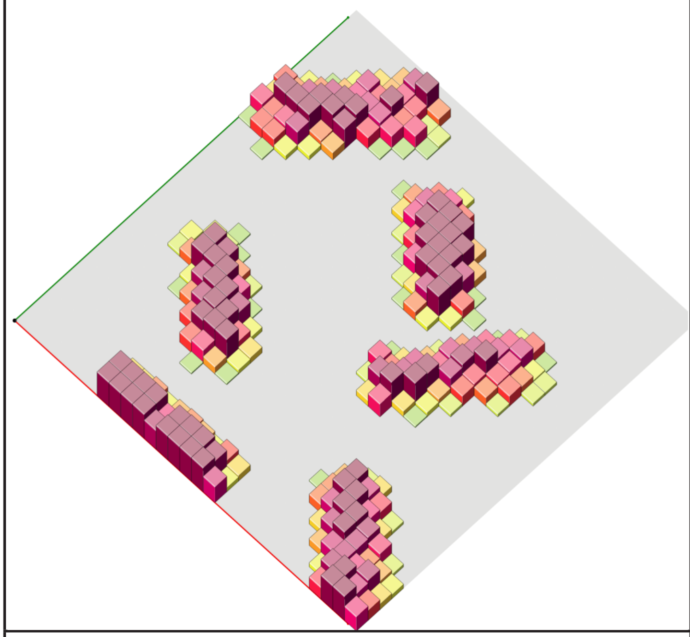
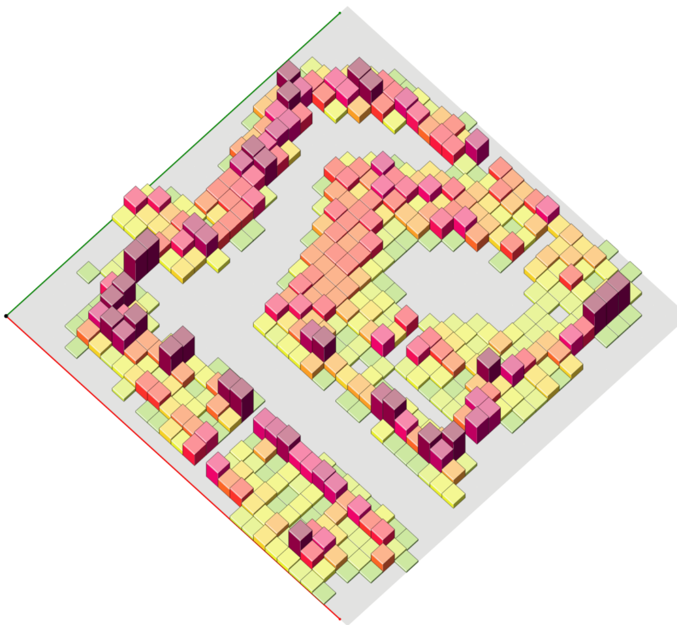
<table>
<tr><th>Model 5</th><th>Model 6</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td></td><td>Note that the yellow bars in M5 and M6 are of equal height but not of equal value;
(M5 = 963; M6 = 763)

The histogram is scaled to fit the display rectangle.</td></tr>
<tr><th>Model 7</th><th>Model 8</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>This histogram range is preset. It is unclear what the density distribution is >125 dwellings per hectare. The Search Grid Density diagrams do not clip the bars and it thus shows the distribution of high-density areas as well.</td><td>Note that the density value of Voronoi cells along the edges of the clusters is very low. This is not a proper assessment. Particles along the border of high density areas are likely to also have a high density. This is why there are more density properties.</td></tr>
</table>

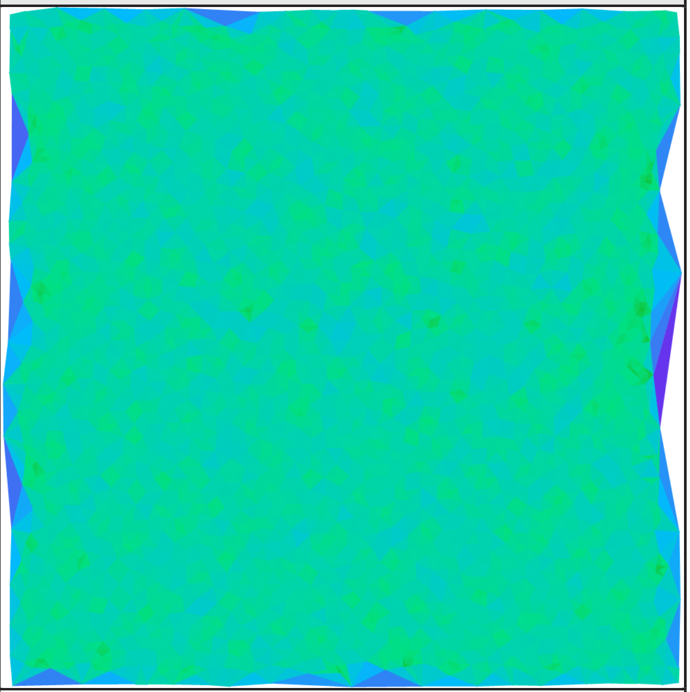
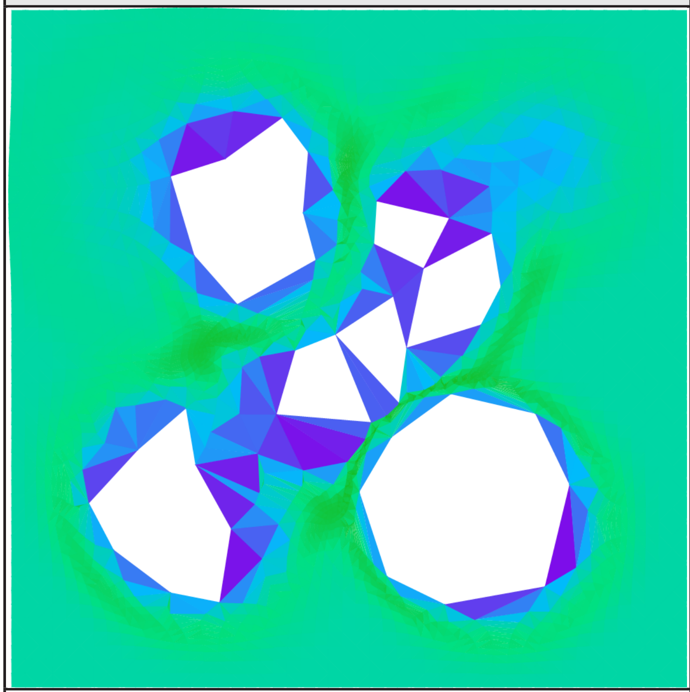
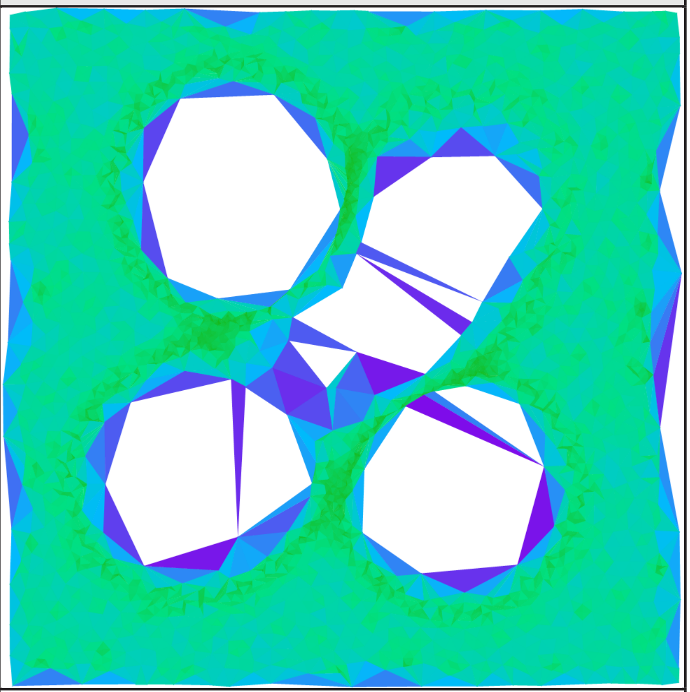
Search Grid density

Model 1	*Model 2*
	
Although the density of every particle is identical, rounding errors result in a Moiré pattern with Search Grids. It is very difficult to measure ordered input using ordered derivatives. This typically results in interference noise.	With unordered input this interference is not a problem.
Model 3	*Model 4*
	
	The height of every block is synonymous with the density. It becomes clear that the open spaces are much more vacant in ***M4*** than they are in ***M3***.

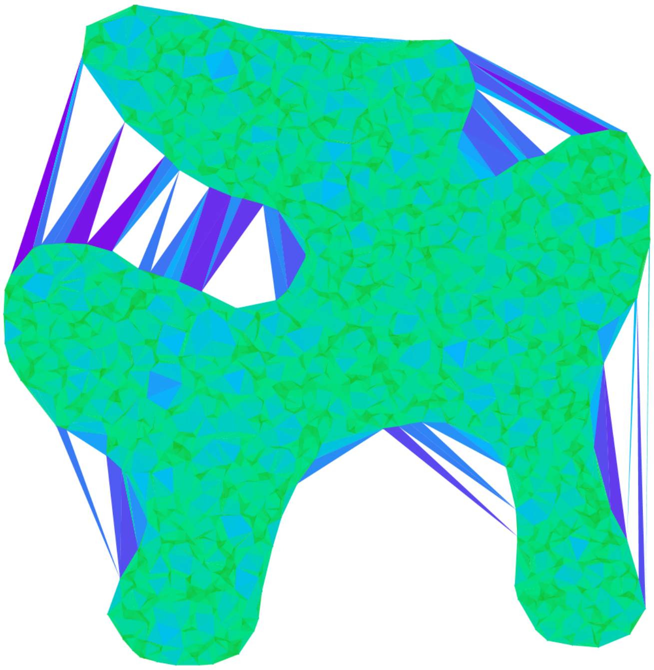
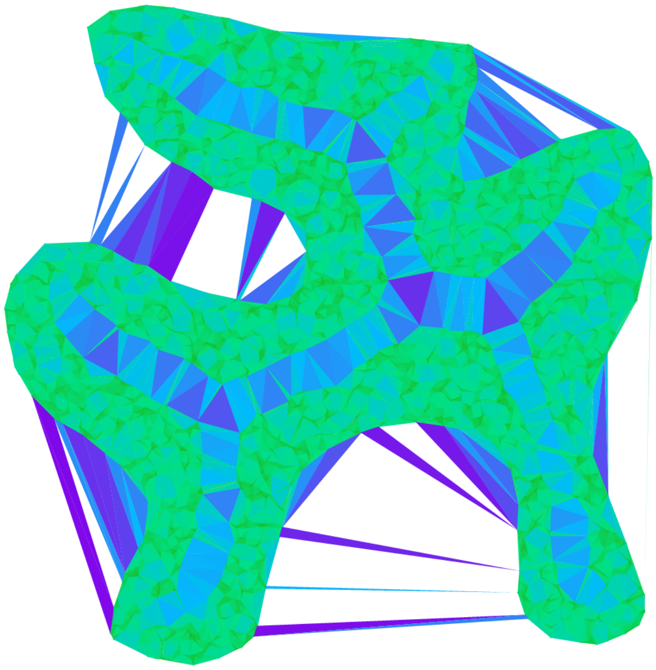
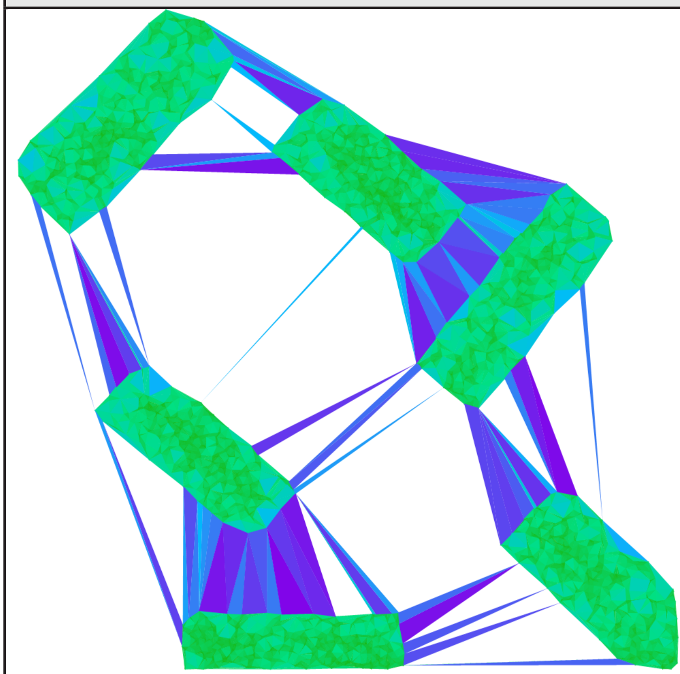
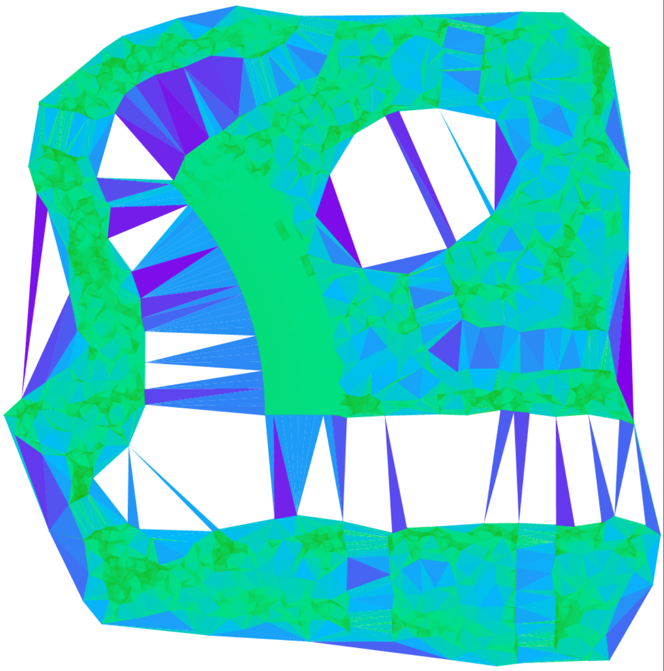
Search Grid density

Model 5	*Model 6*
	
Even though the density is uniformly distributed over the free-form region, the blocks along the edges have a lower density. This affect is called anti-aliasing in the graphics industry. There are several algorithms available for reducing anti-aliasing artifacts.	
Model 7	*Model 8*
	
Anti-aliasing artifacts also cause rotated borders to be blurred. This problem manifests itself with all Search Grid constructs.	In a sense, density and building height are overlapping properties. But one should be careful not to treat them synonymously. With Proximity sketching, the planner creates densities. But as the Shredding property shows, Density is not the whole picture.

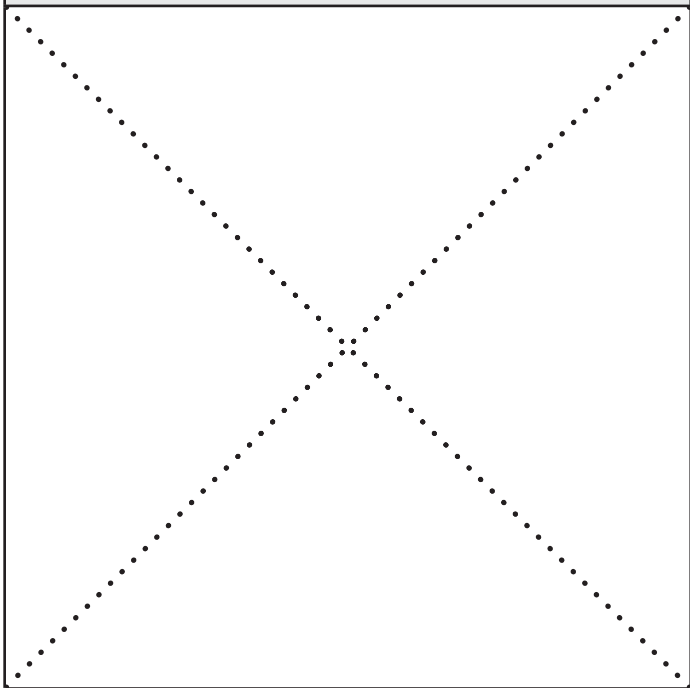
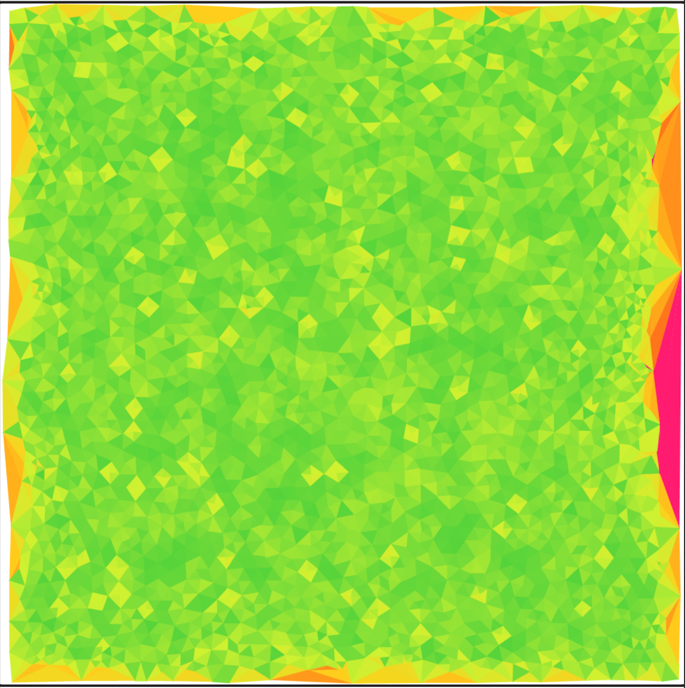
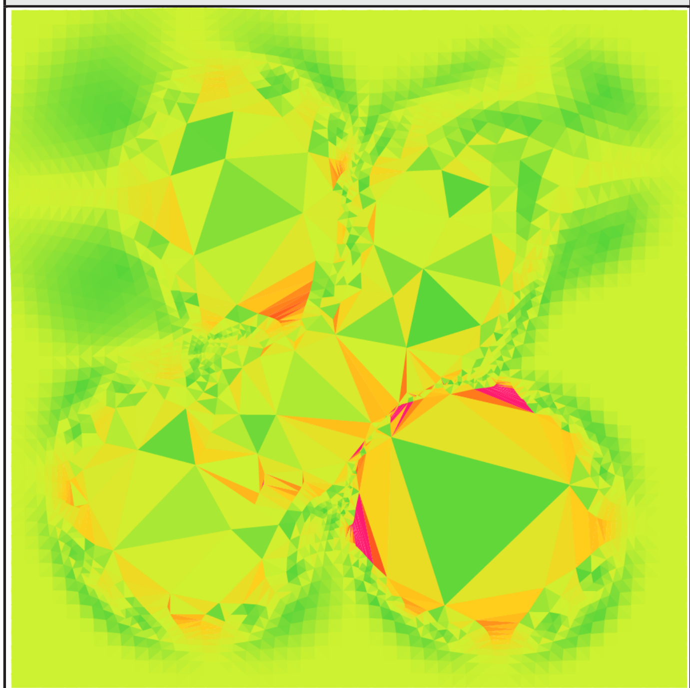
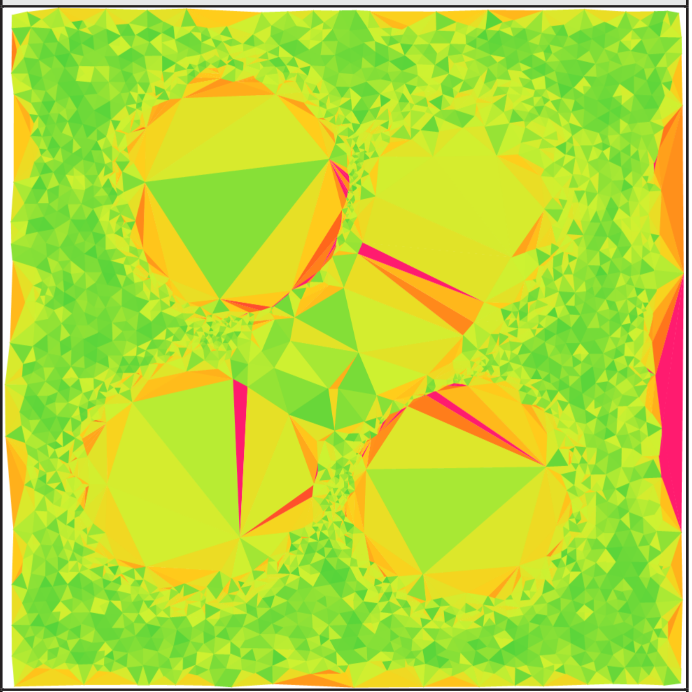
Shared Space Index

Model 1	*Model 2*
	
M1 contains no variance in social pressure.	***M2*** contains minor variance in social pressure. Since all triangles have been equalized the differences have been minimized without breaking topology.
Model 3	*Model 4*
	
It is tempting to view SSI as a dual of PSI since a Voronoi diagram is the dual of a Delaunay mesh. SSI gives us a better representation of public space. Where PSI often generates cells that penetrate the public area, triangles in the SSI can be culled so as to avoid this.	Culling thresholds apply both to triangle area and maximum edge length. In these cases only triangles that exceed 3.000m^2 have been removed. The public (shared among >3 particles) space thus emerges.

Shared Space Index

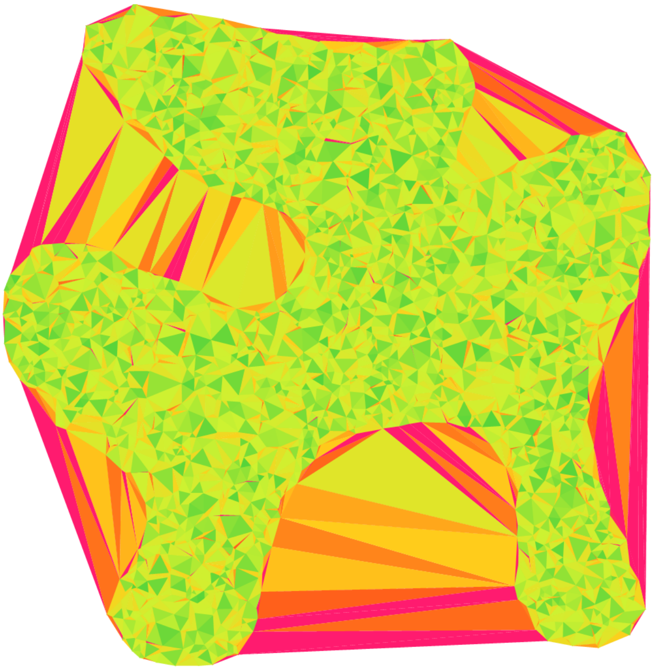
Model 5	*Model 6*
	
Culling by maximum triangle edge length would also remove many of the triangles in the meshes of ***M5~M8***.	
Model 7	*Model 8*
	
Note that removing triangles only by area is a naive and insufficient method. Maximum edge length and exposure properties should be used to achieve a more realistic social culling.	

Shared Space Hierarchy

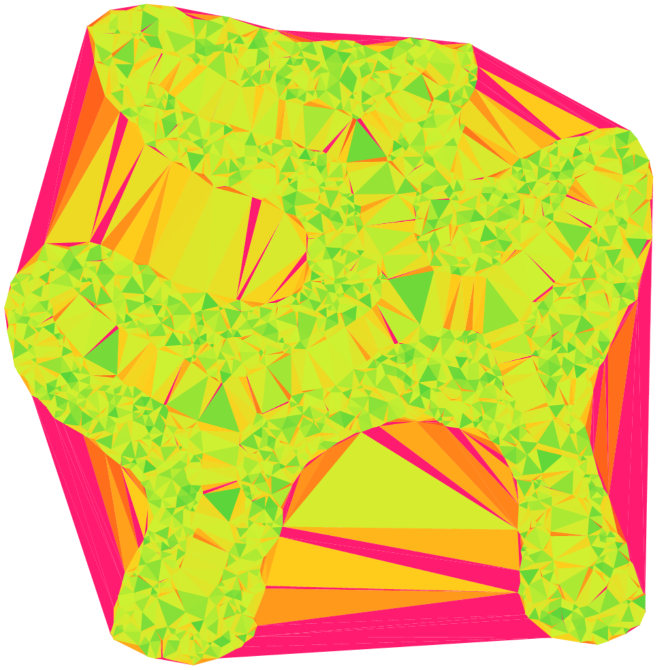
Model 1	*Model 2*
	
All triangles have the same ratio, there is no hierarchy.	Because the spread is equalized, hierarchy has been reduced. Truly random spreads (***M5*** and ***M6***) have a completely different Shared Space Hierarchy signature.
Model 3	*Model 4*
	
Interesting effect in ***M3*** is the clustering of SSH typologies. Typically this only occurs when ordered input is distorted.	The distortion of unordered input results is a very noisy SSH distribution. Though, as the SSH histograms indicate, the amounts of all SSH types is roughly identical in ***M3*** and ***M4***.

Shared Space Hierarchy

Model 5

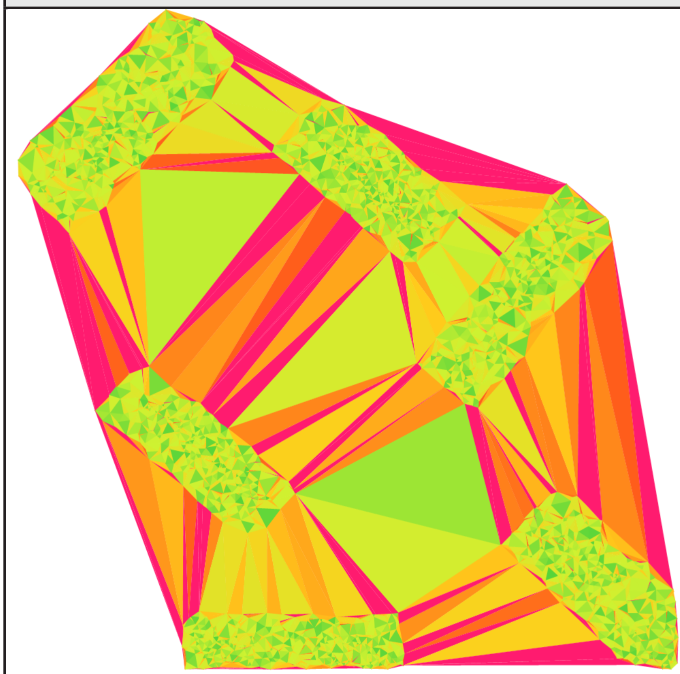


Model 6

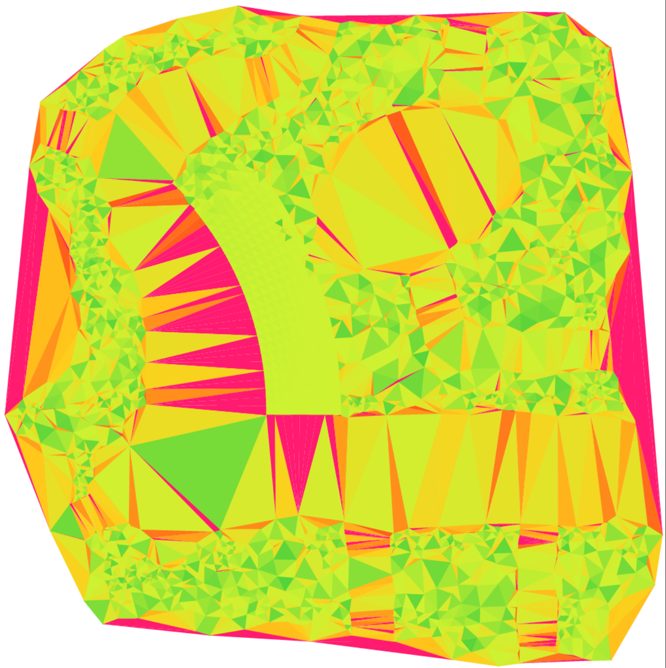


Here, a problem with the SSH algorithm emerges. If particles have very close neighbours on one side, they are unlikely to link with particles that are very far away in opposite directions. A possible solution for this would be to add an additional if clause to the algorithm which is called when the neighbour distances for a particle vary wildly. Then, a small check is performed for nearby, but unlinked particles. This way a more complete picture of social hierarchy emerges.

Model 7



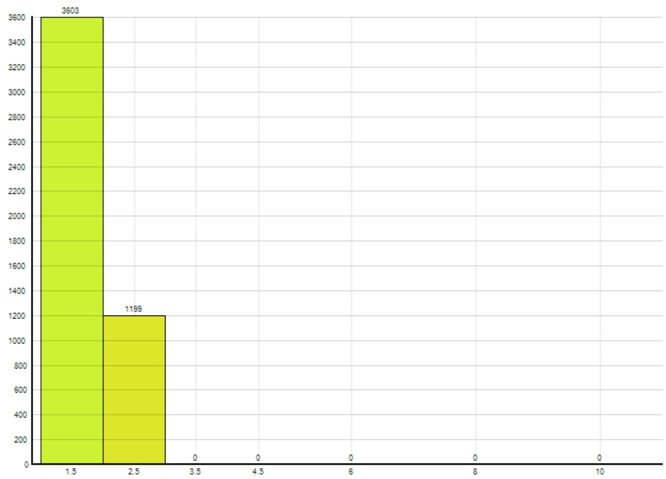
Model 8



M7 and ***M8*** suffer from the same algorithm dysfunction as ***M5*** and ***M6***.

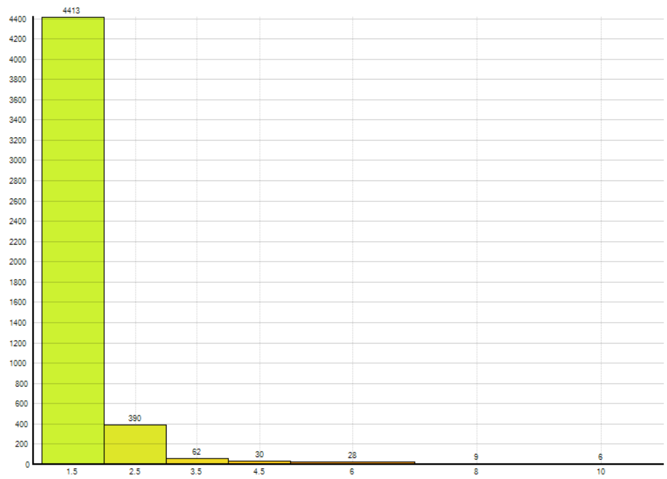
Shared Space Hierarchy Histogram

Model 1



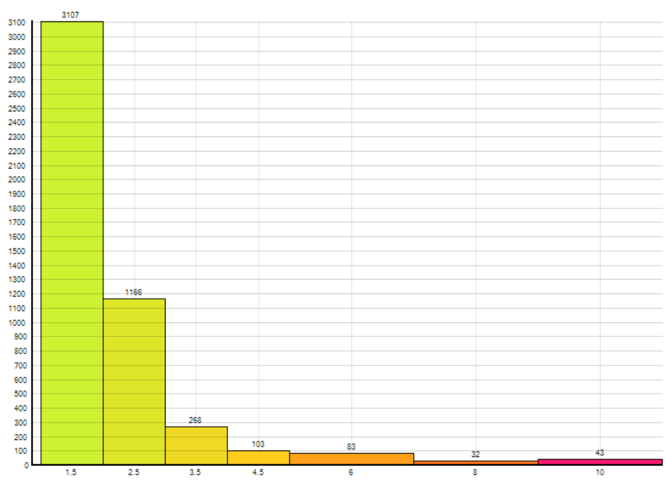
The ratio of all triangles in a square grid equals 2.0.; $\mathit{Sqrt}(2) / \mathit{Sqrt}(2 * 0.5^2)$. This ratio is exactly on the threshold between the first and second bar in the histogram. 32-bit mathematics will therefore sometimes compute a value slightly above 2.00. This explains the second bar.

Model 2

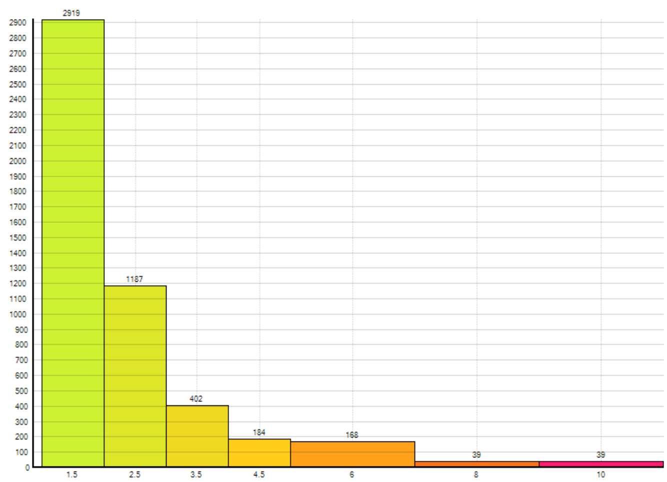


M1 contained only orthogonal triangles, while the equalization pass in ***M2*** creates many equilateral triangles. This is why the first bar in the histogram is actually larger than in ***M1***.

Model 3

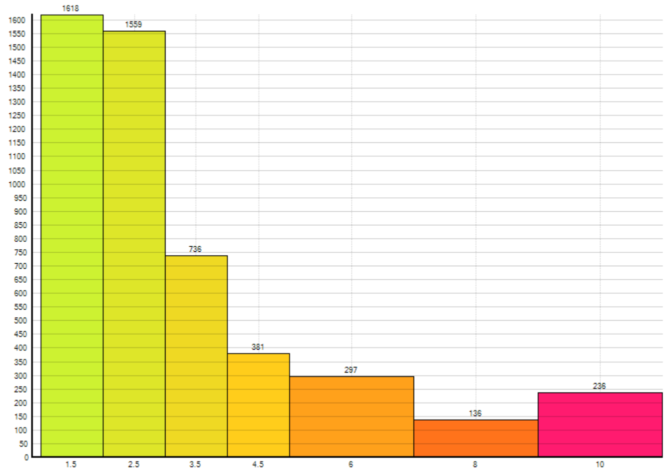
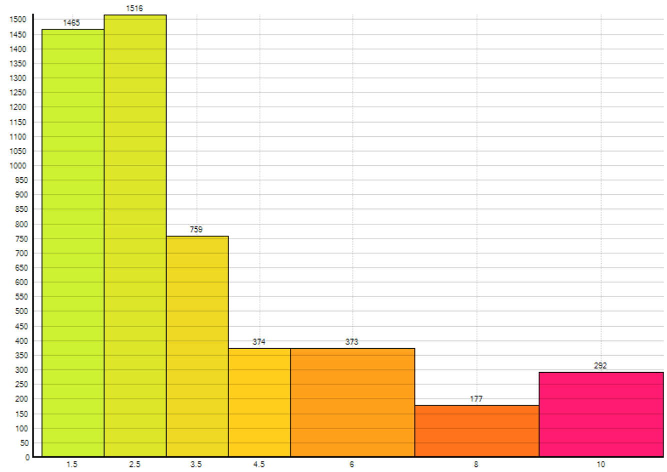
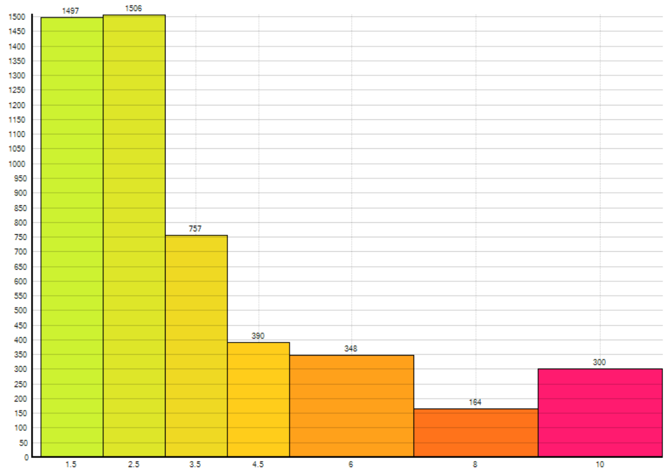
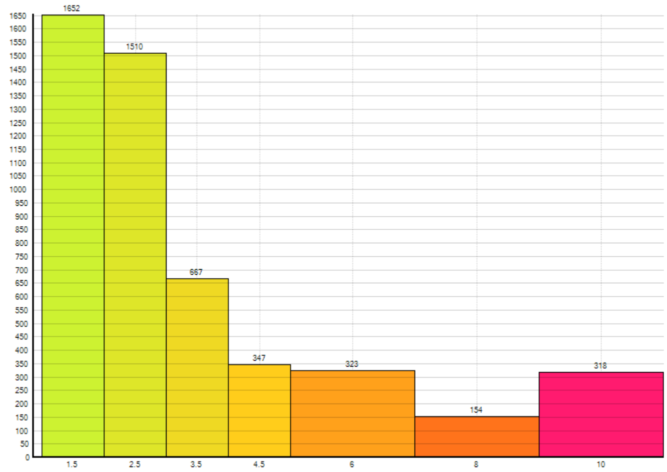


Model 4

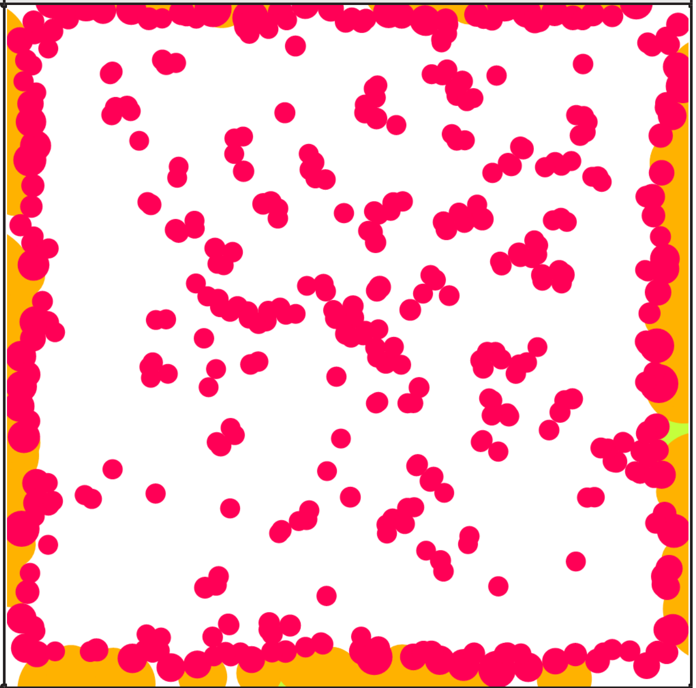
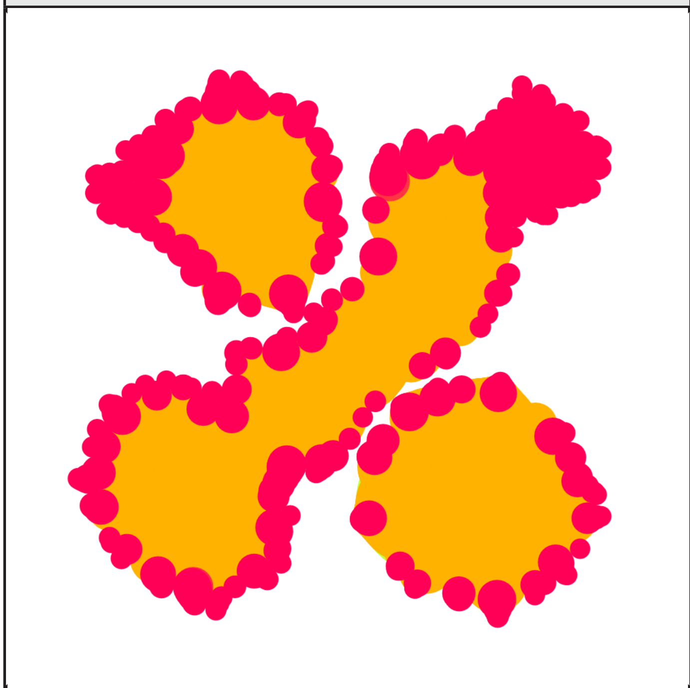
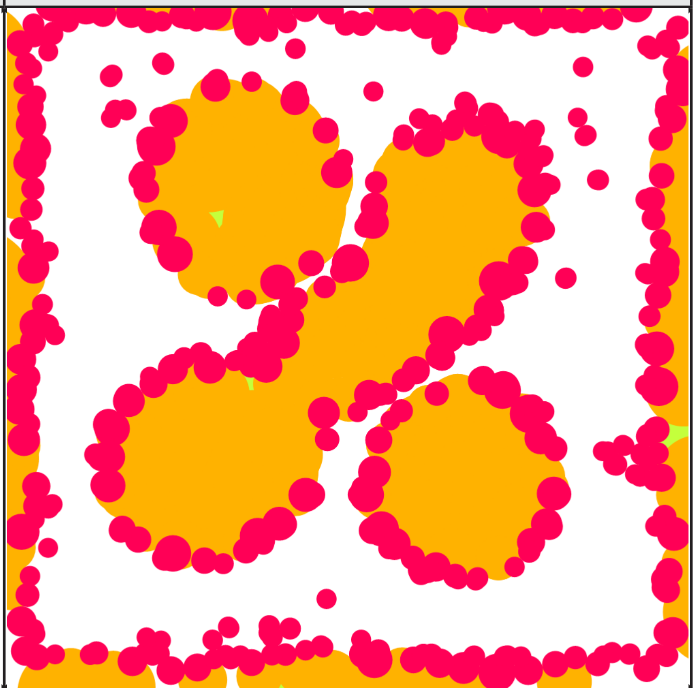


Though the hierarchy-maps are wildly different, the histograms show us that ***M3*** and ***M4*** are very similar with respect to SSH.

Shared Space Hierarchy Histogram

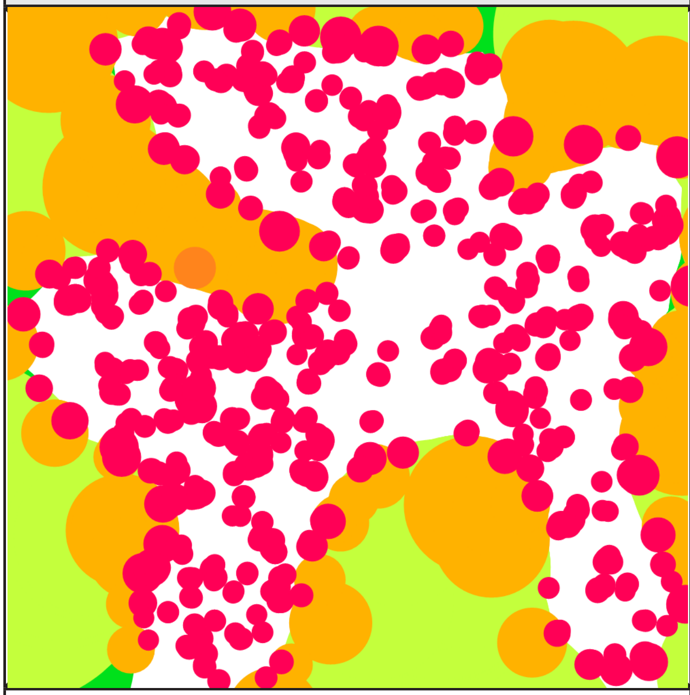
<table>
<tr><th>Model 5</th><th>Model 6</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>The difference between M5 and M6 is also very small. This makes sense since they are both random spreads.</td><td></td></tr>
<tr><th>Model 7</th><th>Model 8</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>M7 is also a random spread and thus very similar to M5 and M6. SSH can thus be used as an indicator for order and spatial sorting.</td><td></td></tr>
</table>

Openness

Model 1	Model 2
	
M1 contains no gaps.	Interestingly, by shuffling a spread by a small amount a complete public space structure emerges. Where *M1* was not very suited for families with young children, *M2* is.
Model 3	Model 4
	
	The main difference between *M4* and *M3* with regard to Openness, is the thickness of the pink layer around the bulges. This thickness indicates a higher variety of open space typologies.

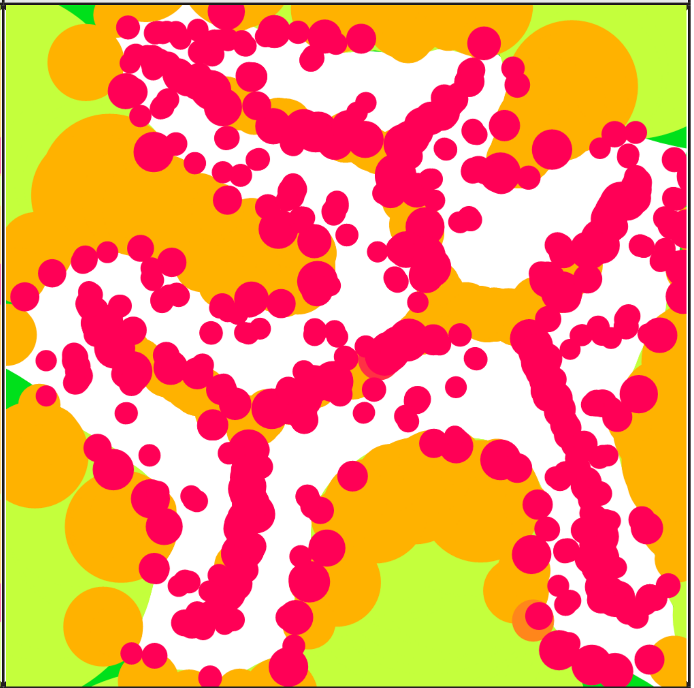
Openness

Model 5



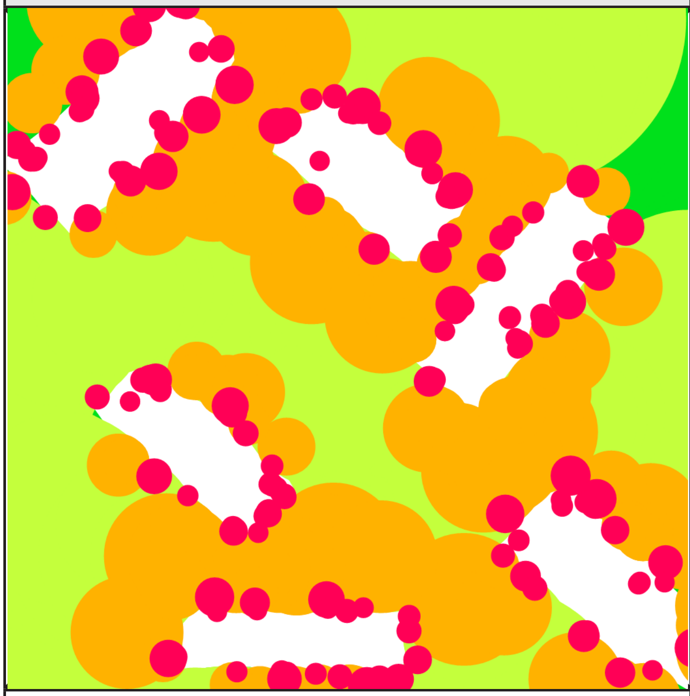
Even though the density of particles is much higher here than in ***M1*** and ***M2***, there is plenty of room for playground sized gaps. Still, larger open spaces are expelled towards the fringes of the spread.

Model 6

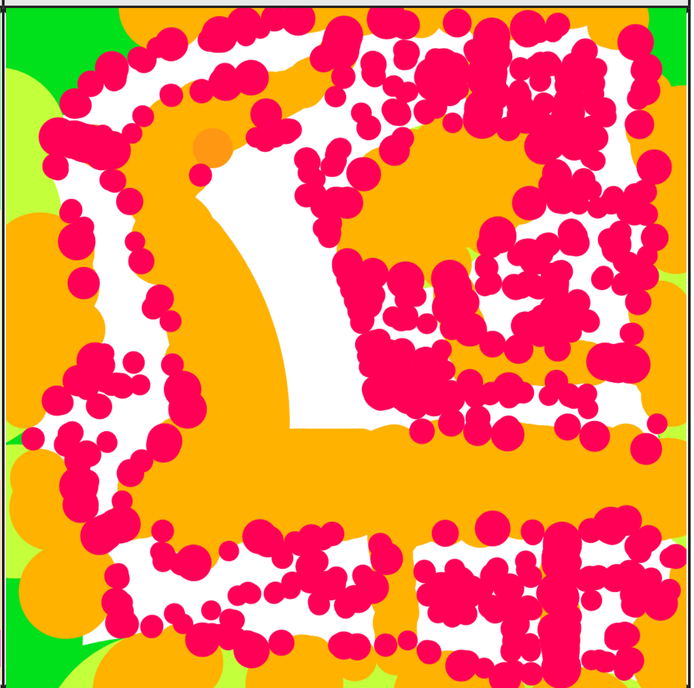


The open core of ***M6*** allows for the placement of park sized gaps within the spread. Playgrounds are noticeably less numerous in the high density ribbons than in ***M5***.

Model 7

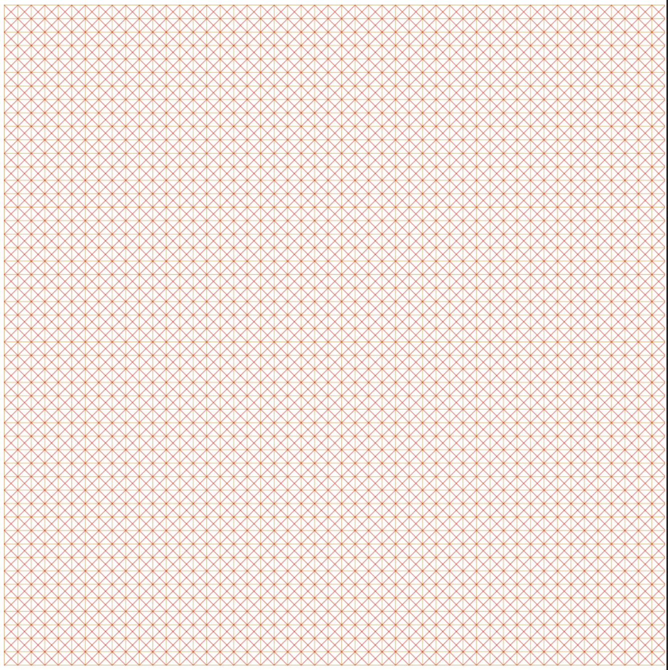
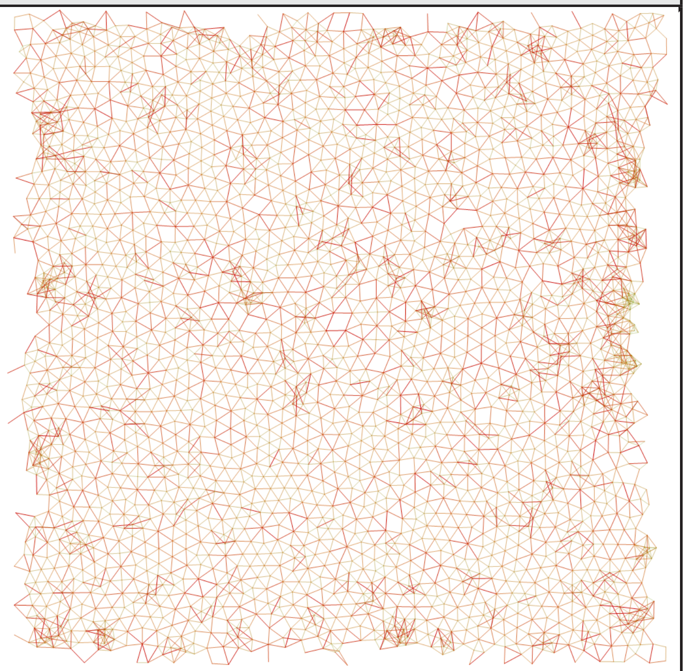
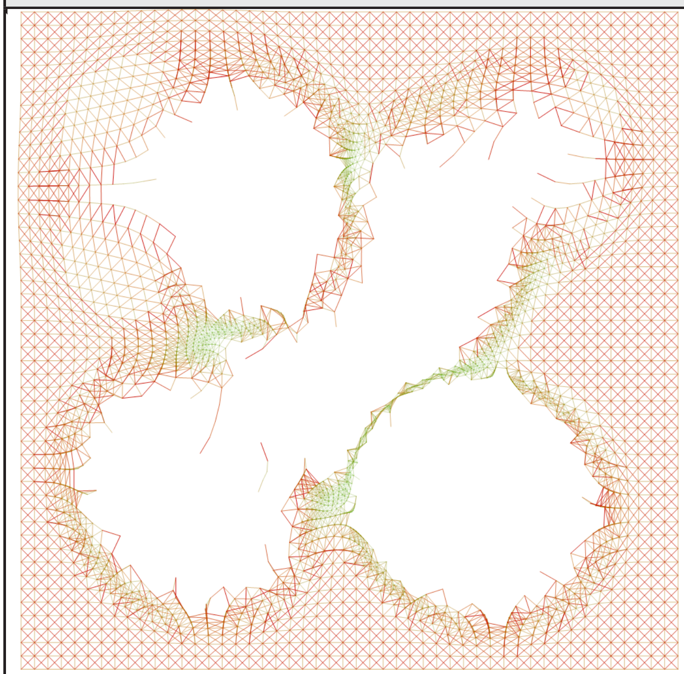
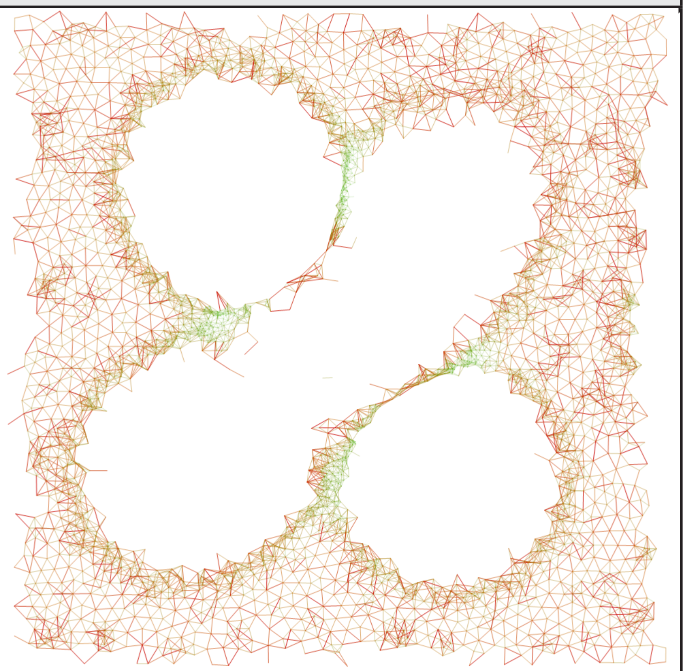


Model 8

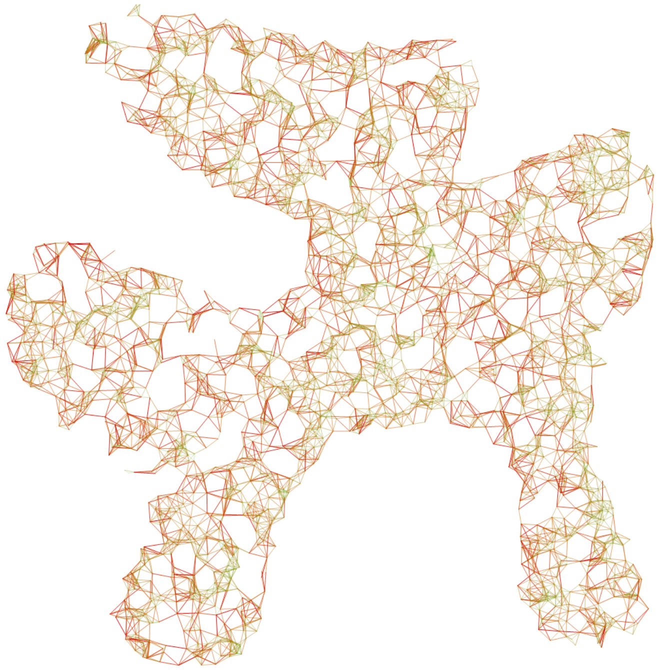
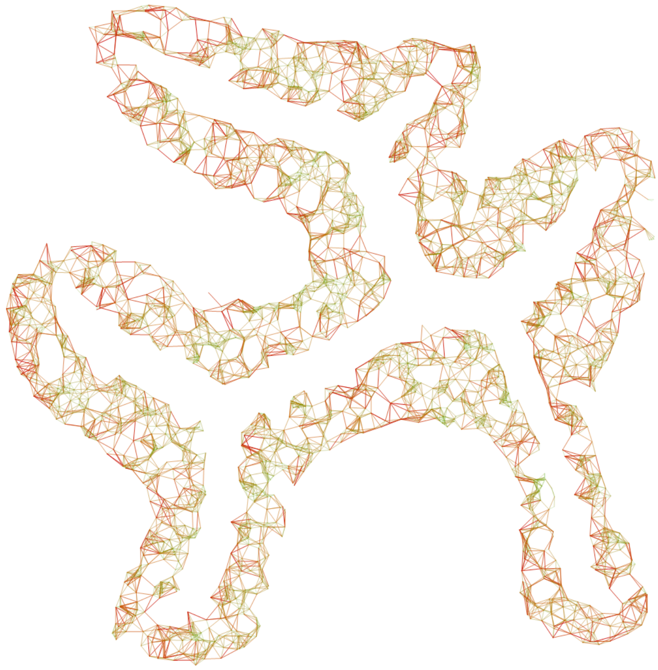
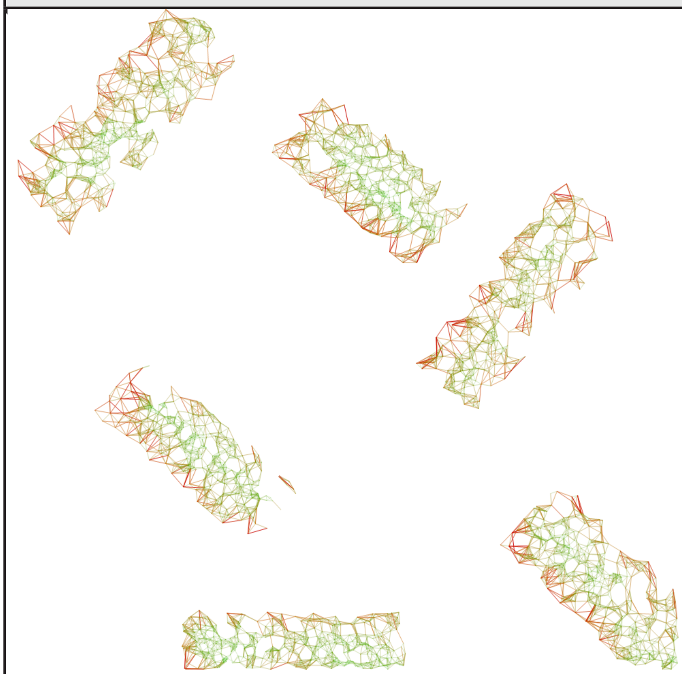
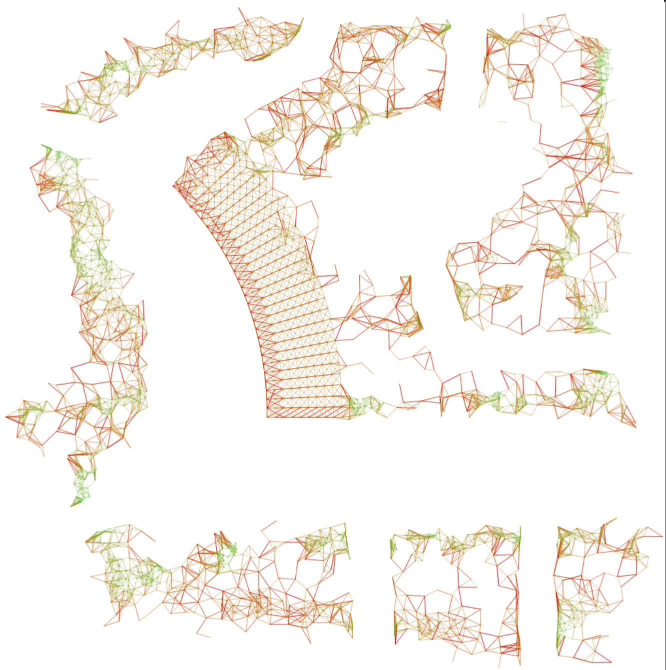


(white)	Non-open
(red)	Playground
(orange)	Small city park
(light green)	Large city park
(green)	Landscape

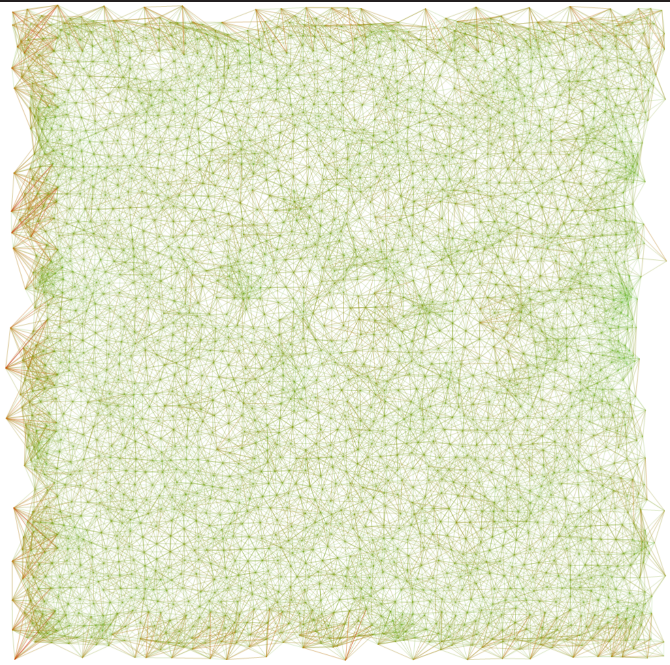
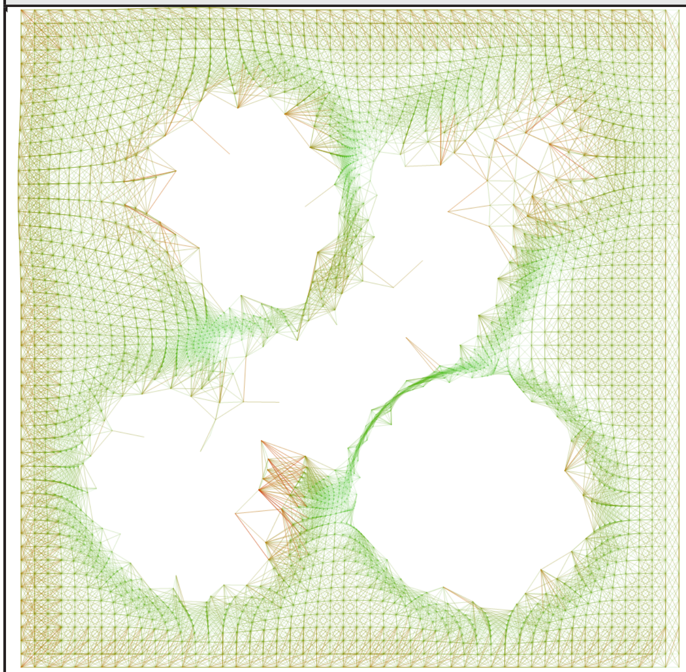
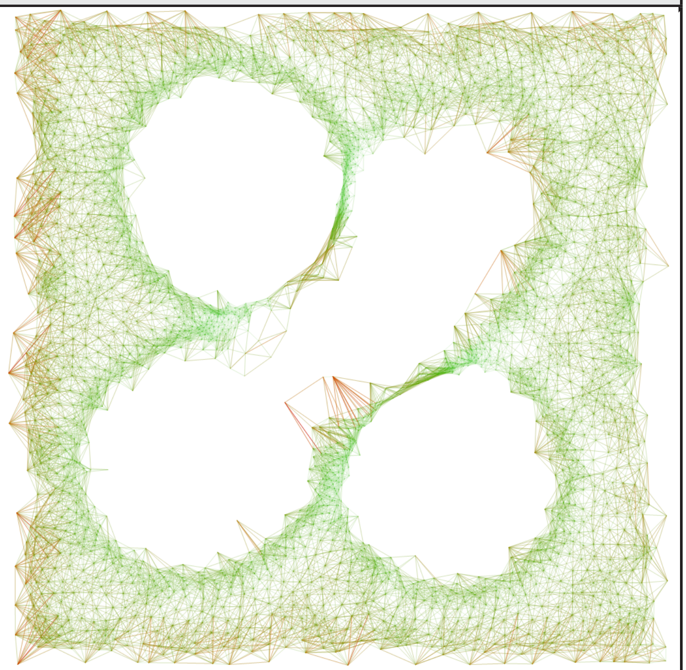
Flexibility $\{R = 30\}$

Model 1	*Model 2*
	
Almost the entire network is strained at this level of scale. The low density of the particles excludes close neighbours.	Local clustering results in slightly less strained islands, but the whole is nearly identical to ***M1***.
Model 3	*Model 4*
	
The high density ribbons are less strained. This is an obvious result.	

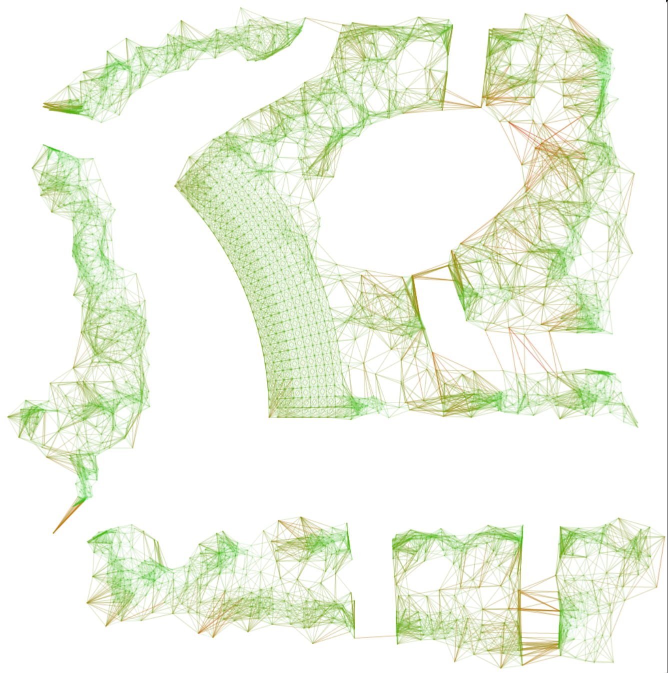
Flexibility *{R = 30}*

Model 5	*Model 6*
	
Random placement of particles results in large flexibility differences throughout the network.	
Model 7	*Model 8*
	

Flexibility *{R = 100}*

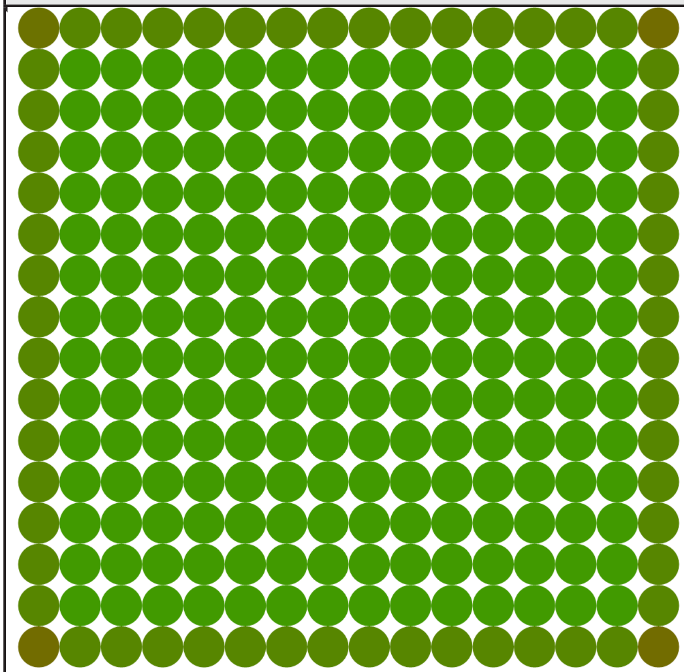
Model 1	Model 2
	
At this level of scale the network is completely relaxed.	Some strain still remains around the edge particles.
Model 3	**Model 4**
	
There is a large amount of straining around the bulge edges, indicating that -from a human point of view- the bulges are not clearly delineated.	Unlike *M3*, *M4* does have clear bulge contours.

Flexibility $\{R = 100\}$

Model 5	*Model 6*
At this level of scale the network is completely relaxed.	At this level of scale the network is completely relaxed.
Model 7	*Model 8*
	
Isolated particles are easily detected, but isolated groups are more complex entities. Connectivity Networks are ideal for spotting them.	Isolated groups are even better visible in ***M8***. The combination of neighbour distance and neighbour count allows us to detect social borders even in complicated spreads.

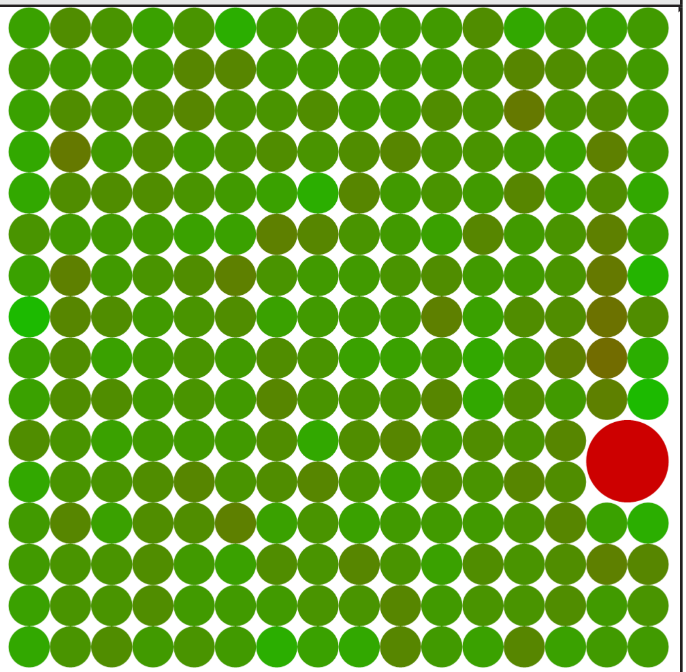
Shredding $\{T = 30\}$

Model 1



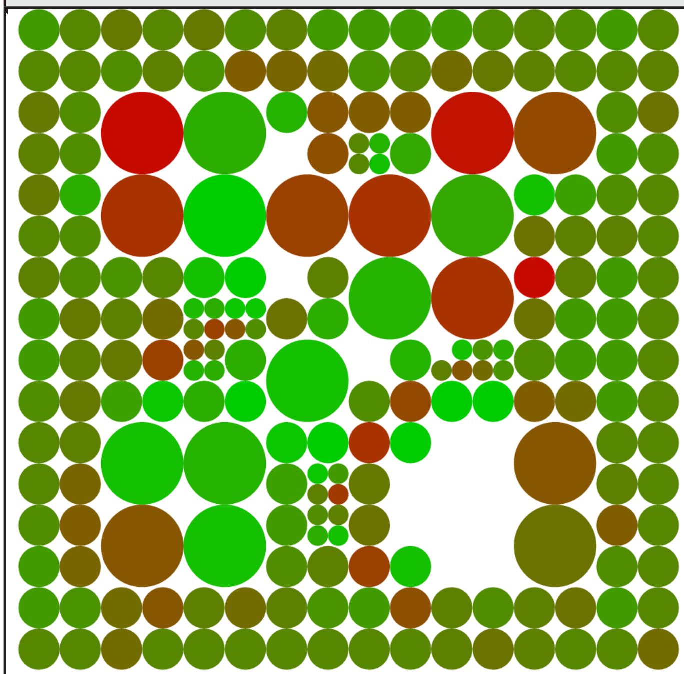
The discolouration around the spread boundary is the result of a 32-bit computing rounding error.

Model 2

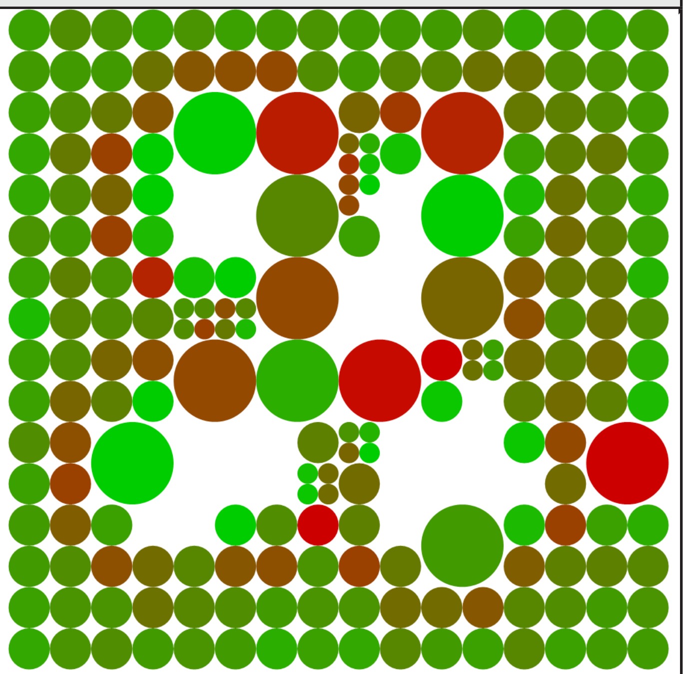


A typical example of biased derivatives. The red circle disappears when affine transformations are applied to the spread.

Model 3

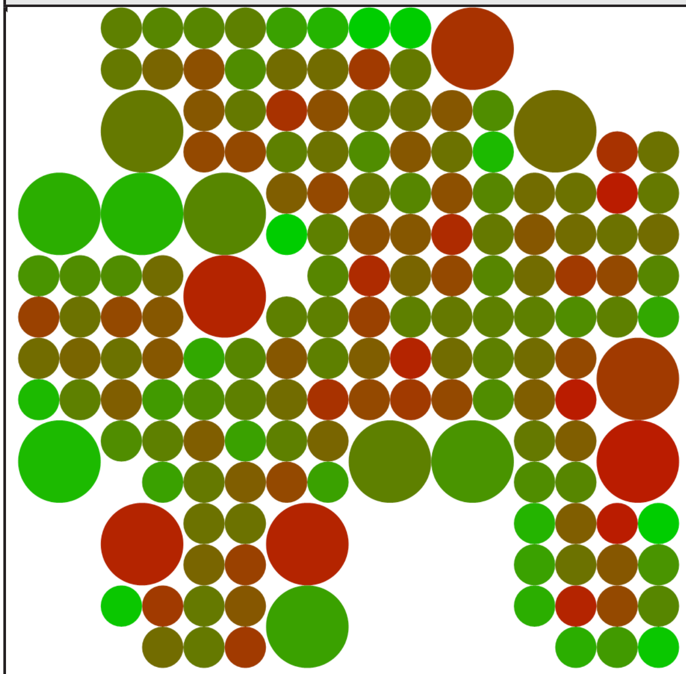
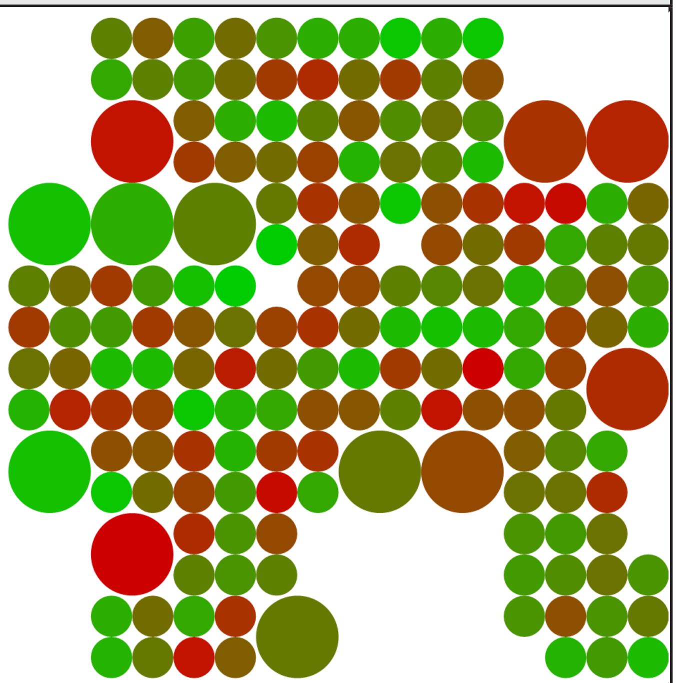
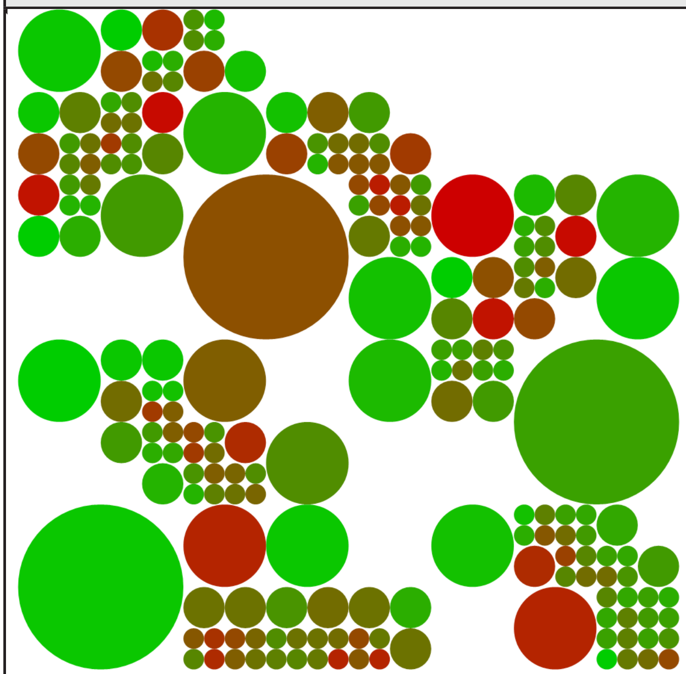
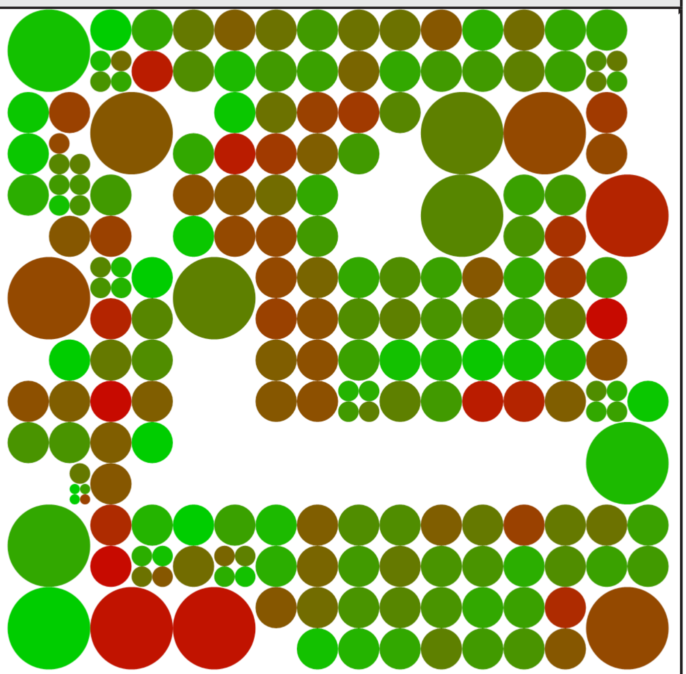


Model 4

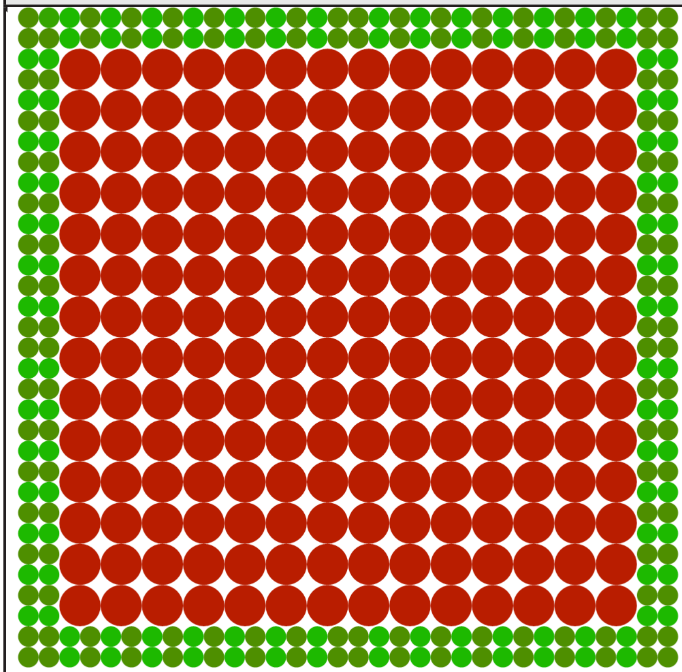
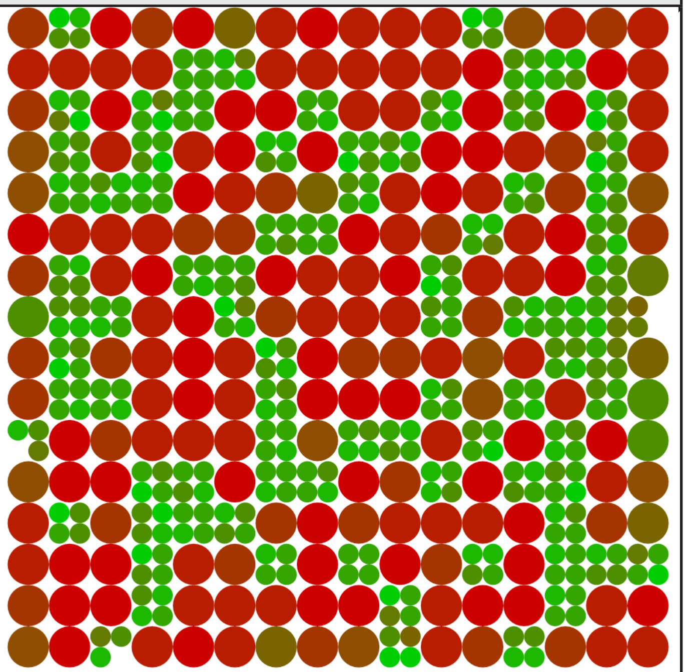
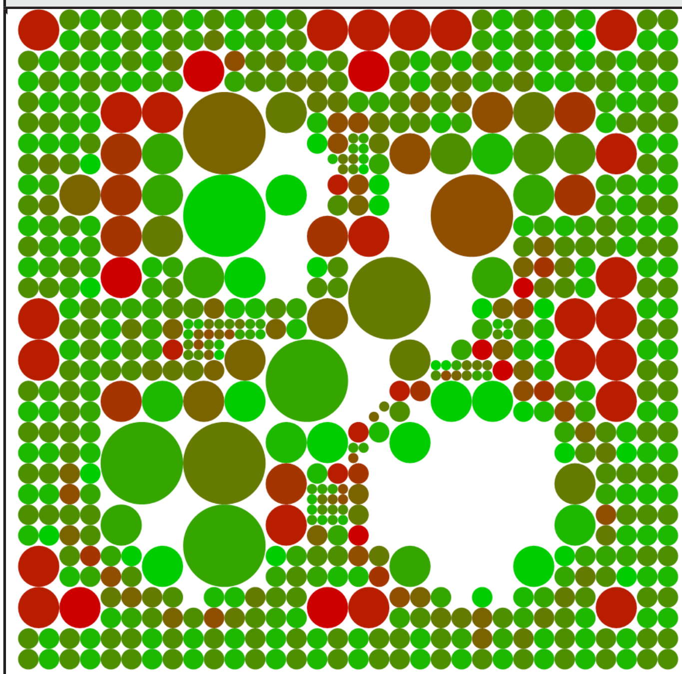
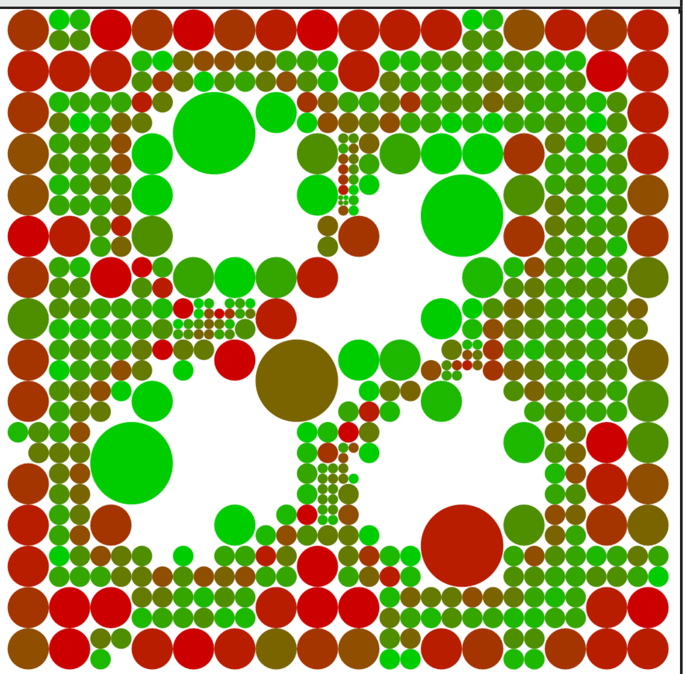


The added area of all empty Quad branches (empty branches are not drawn) is much higher in *M4* than in *M3*. This corresponds with the properties of Density, Flexibility and Openness.

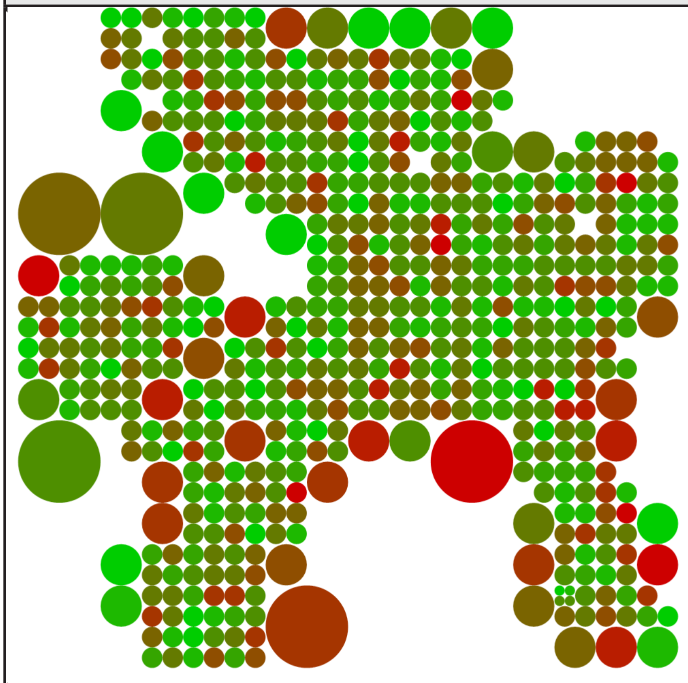
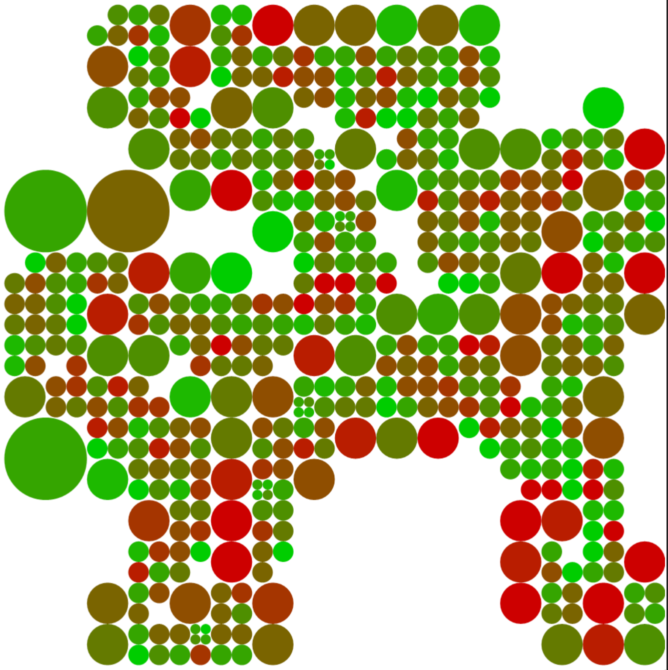
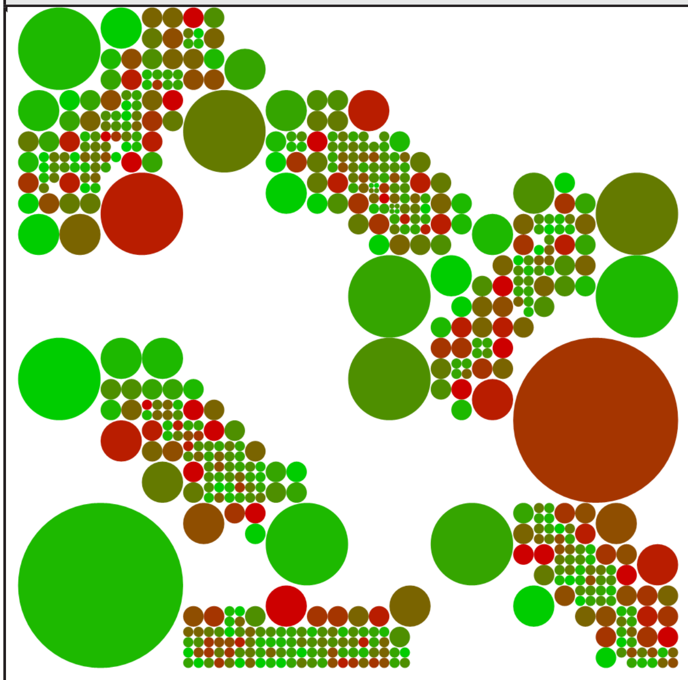
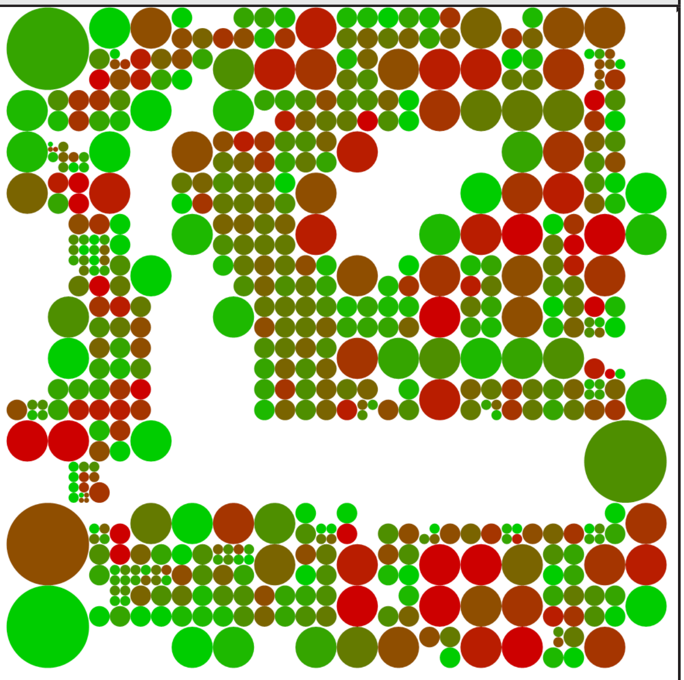
Shredding *{T = 30}*

Model 5	*Model 6*
	
On this level of scale ***M3*** and ***M4*** obviously differ far more than ***M5*** and ***M6***. Group size threshold T needs to be decreased further before differences between ***M5*** and ***M6*** will become evident.	
Model 7	*Model 8*
	
Green circles indicate relatively empty Quad branches, so in that sense the openness between the separate clusters is represented correctly. Visually however, this openness has disappeared. The solution as proposed in the topic [Sprawl ***{min = 2.0; max = 20.0}***] would	also provide solace in these cases.

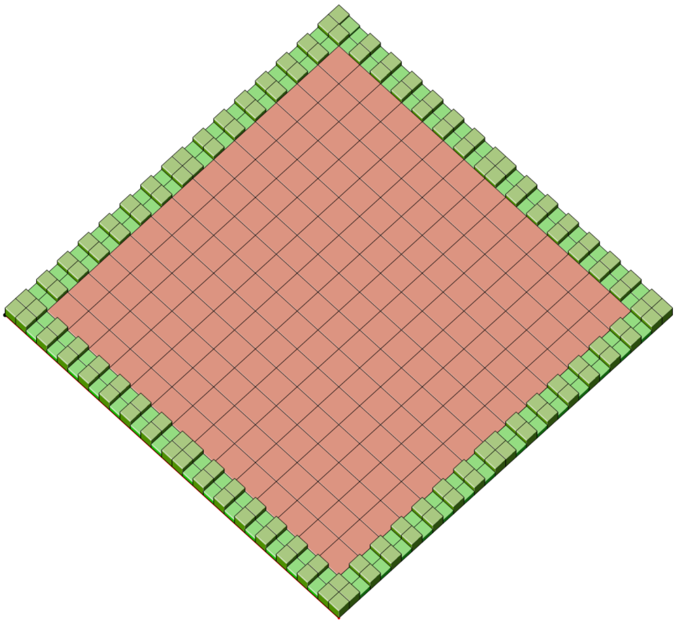
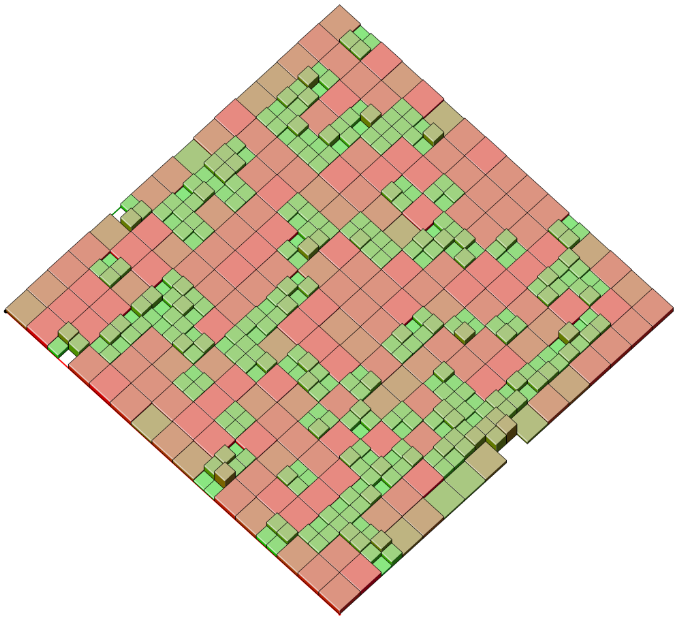
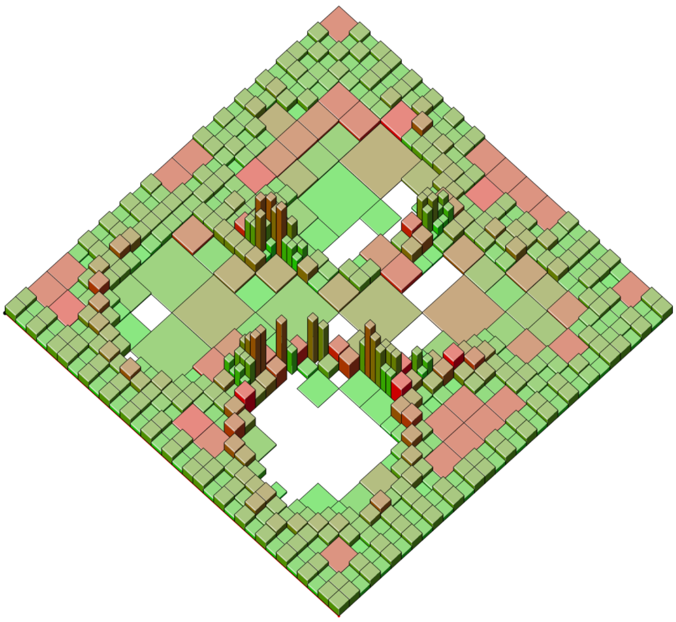
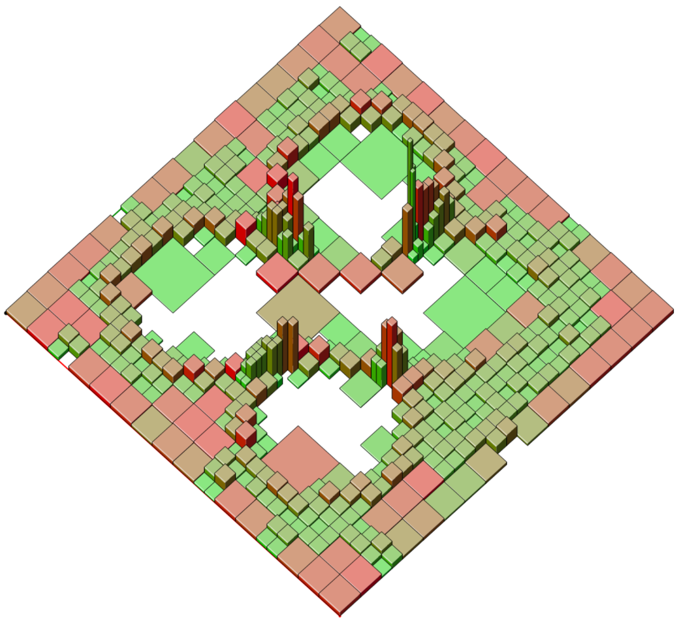
Shredding $\{T = 10\}$

Model 1	*Model 2*
	
The manifestation of the 32-bit computing problem is far more severe on this reduction level.	Since the occurrence of large, red circles is not erratic, this is not a problem of bias. This Quad Tree manifests the underlying clustering properly.
Model 3	*Model 4*
	
The 32-bit issue does not appear with $M3$ because the bulges have pushed a few particles beyond the bounding box of $M1$. Hence, far fewer particles are coincident with the Quad Tree box.	

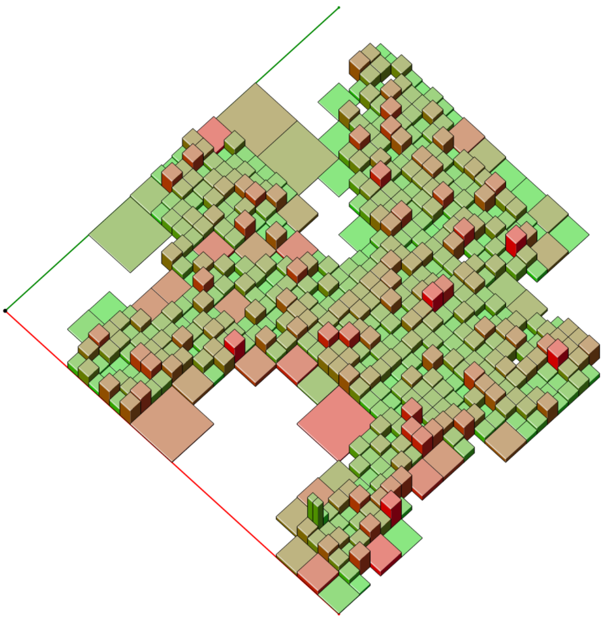
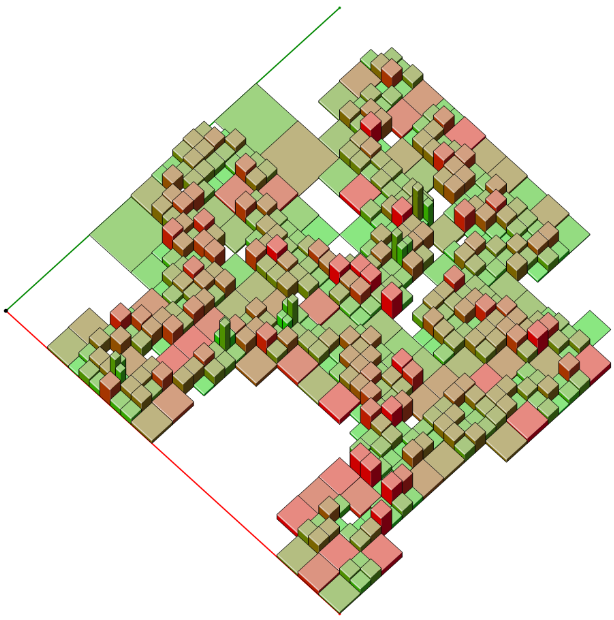
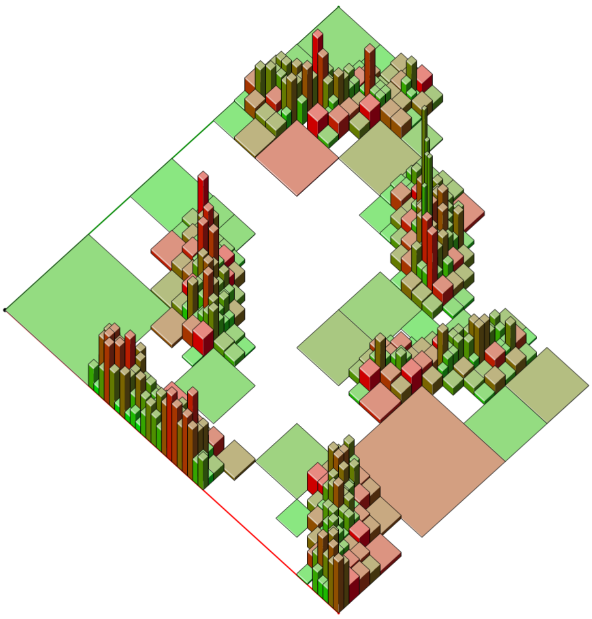
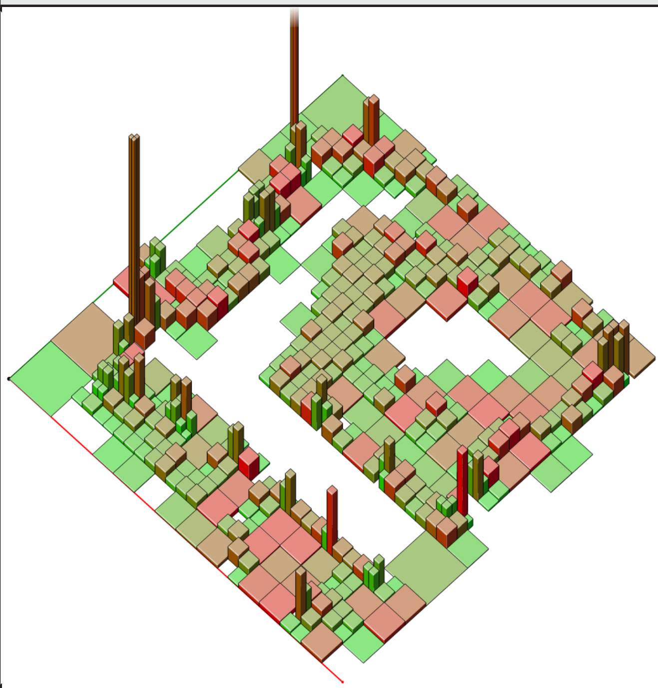
Shredding *{T = 10}*

<table>
<tr><th>Model 5</th><th>Model 6</th></tr>
<tr><td></td><td></td></tr>
<tr><td>Quad Tree groupsize has been reduced far enough for differences between M5 and M6 to emerge.</td><td></td></tr>
<tr><th>Model 7</th><th>Model 8</th></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>

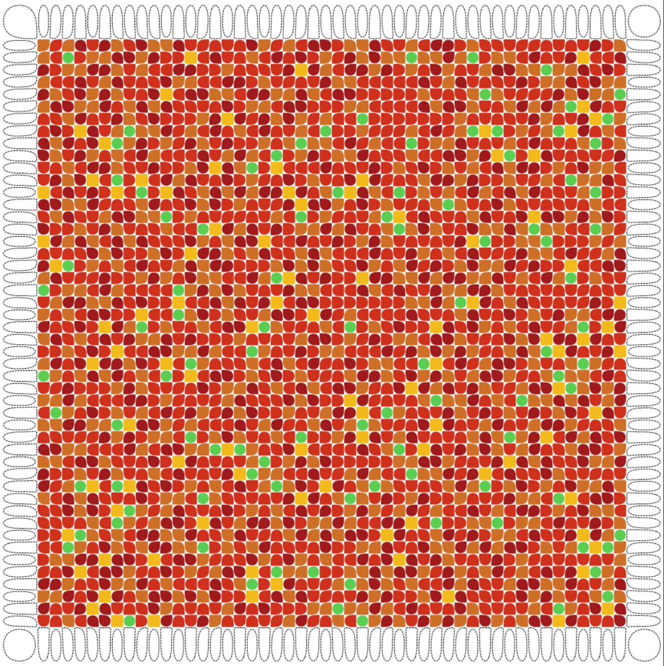
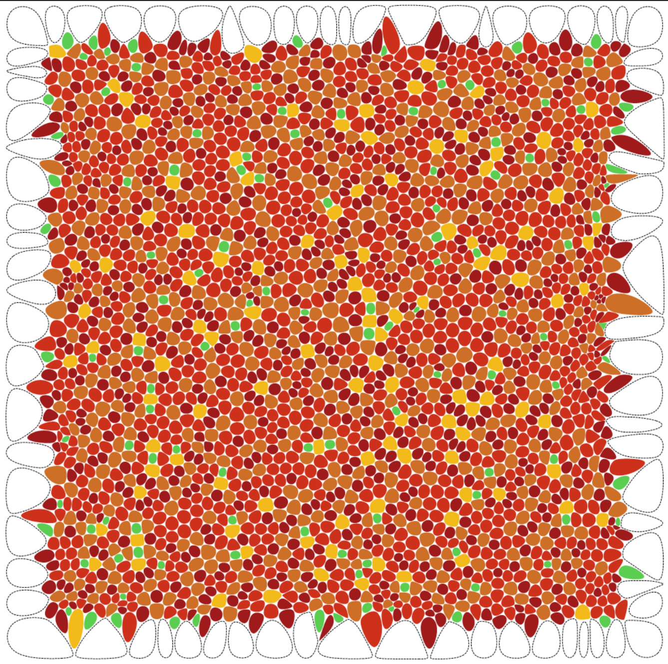
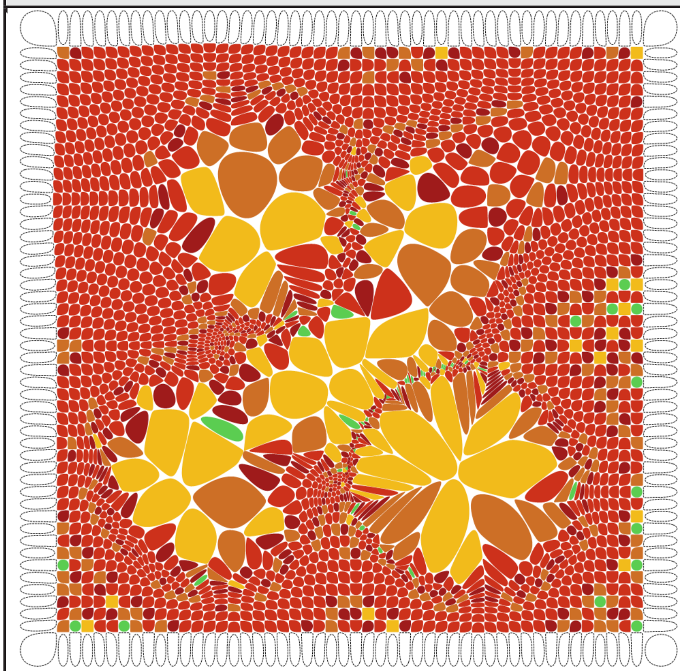
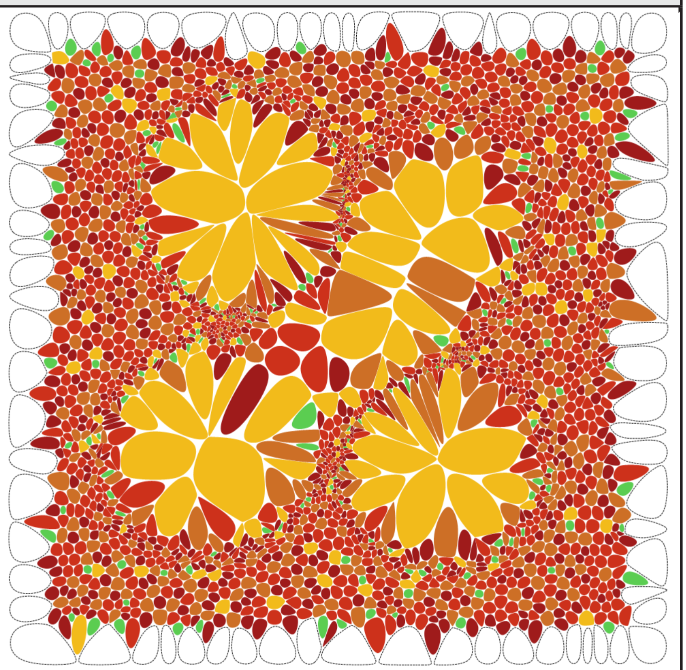
Shredding vs. Density *{T = 10}*

Model 1	*Model 2*
	
Density computations do not suffer from 32-bit issues. They do however suffer from Moiré interference. The alternating pattern along the edges of the Quad Tree is a testimony to this.	
Model 3	*Model 4*
	

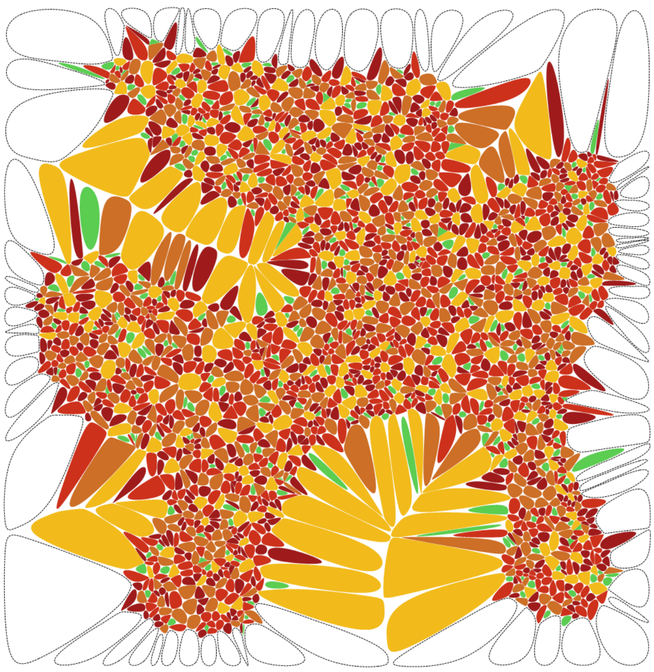
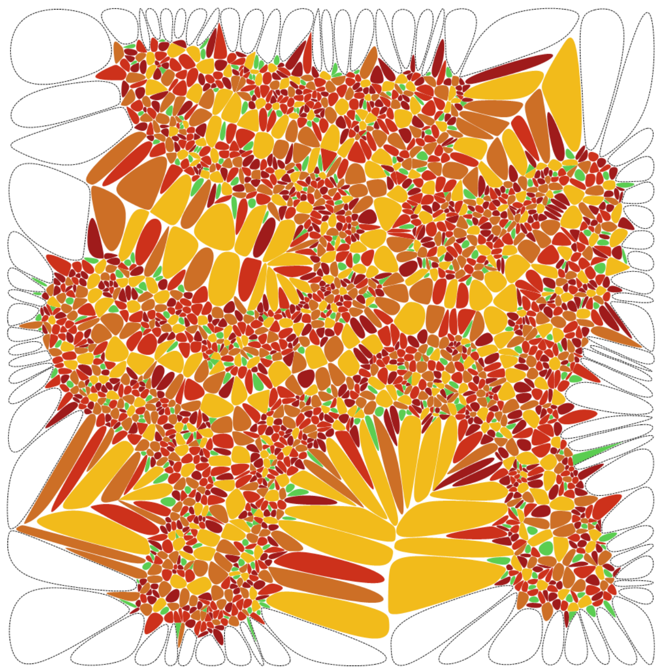
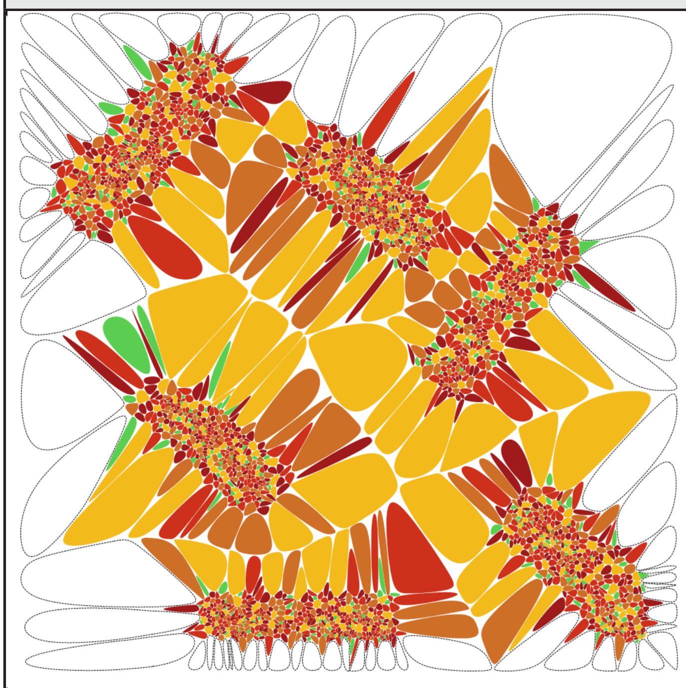
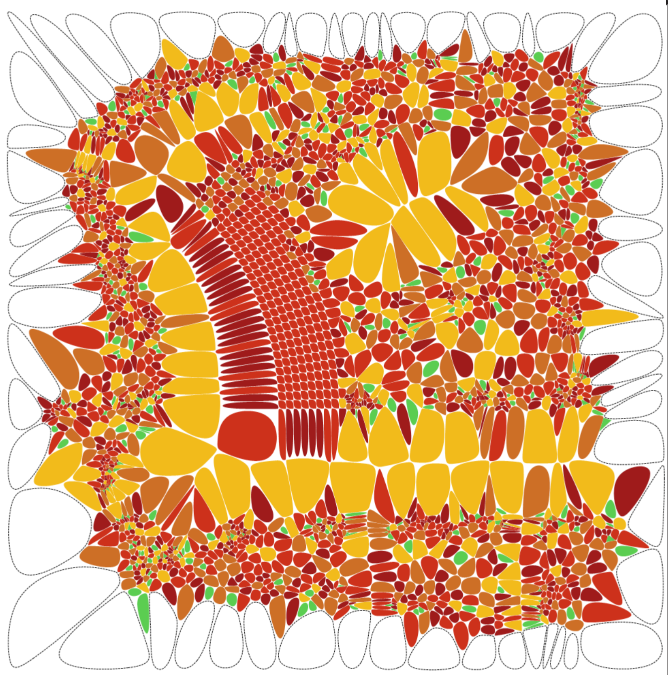
Shredding vs. Density *{T = 10}*

<table>
<tr><th>Model 5</th><th>Model 6</th></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><th>Model 7</th><th>Model 8</th></tr>
<tr><td></td><td></td></tr>
<tr><td>The Quad Tree Density map shows a much more varying manifestation than the Search Grid Density map. This is because Quad Trees are more affine than Search Grids; there is less averaging.</td><td>Density peaks in M8 do not indicate per se a much higher density than you would find in M1~M7. Simply, the shredding groups are smaller here and smaller measurement areas always result in higher densities.</td></tr>
</table>

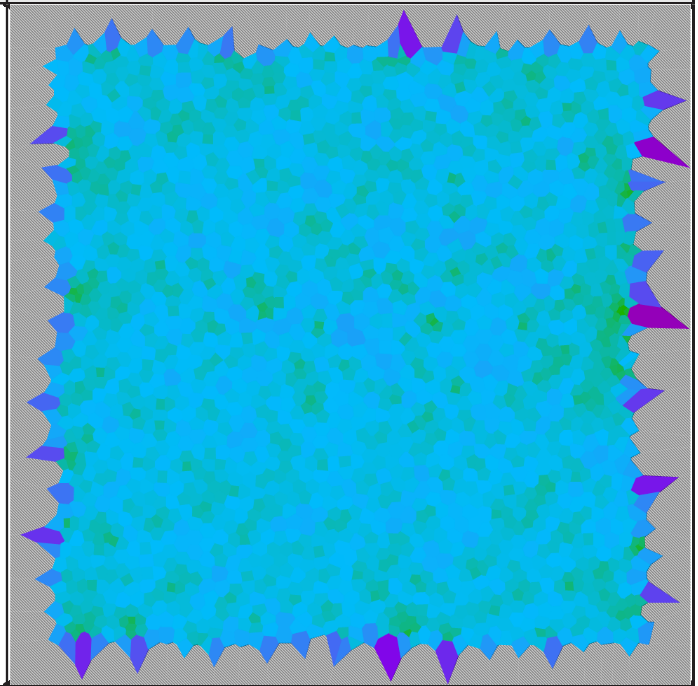
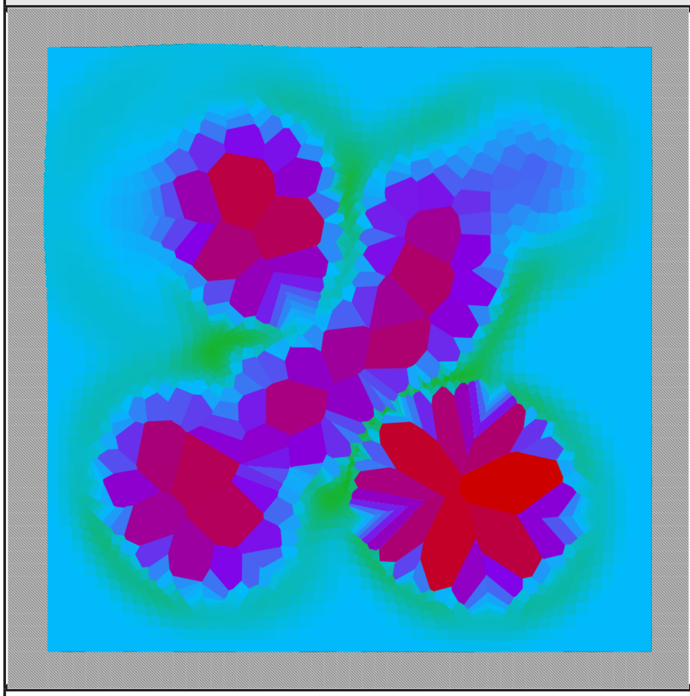
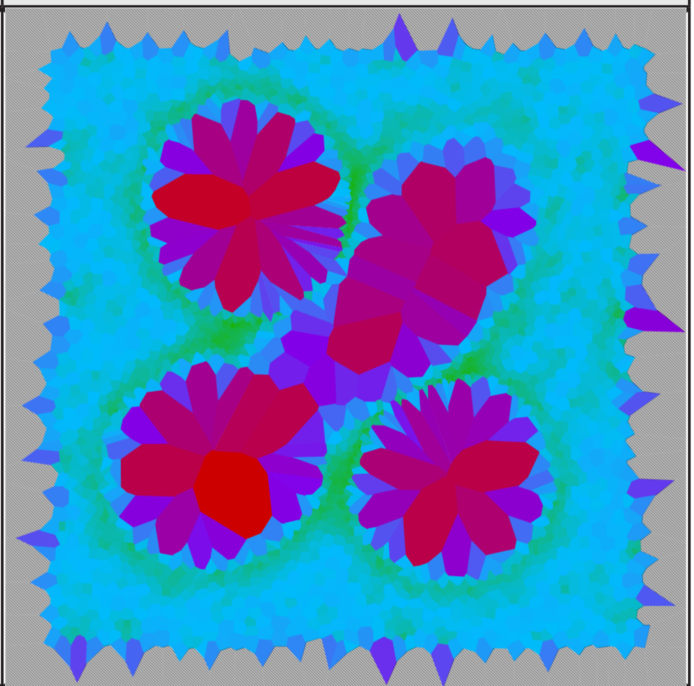
Complexity

Model 1	*Model 2*
	
The dithered effect of the Complexity map is a result of a pre-process algorithm which shuffles the entire model slightly. This is done to avoid computational issues in some algorithms. The effect could be avoided by adding a threshold value to cell-segment length.	On unordered input the problem as seen in ***M1*** does not occur. The random translations of particles do not exceed 1mm in either direction.
Model 3	*Model 4*
	
Only those areas that have been completely unaffected by the bulges still have the dithered noise as seen in ***M1***.	

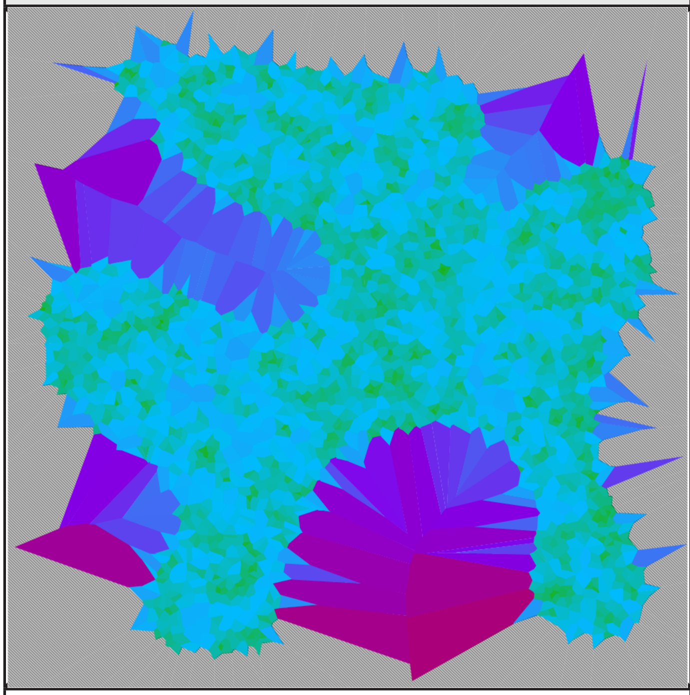
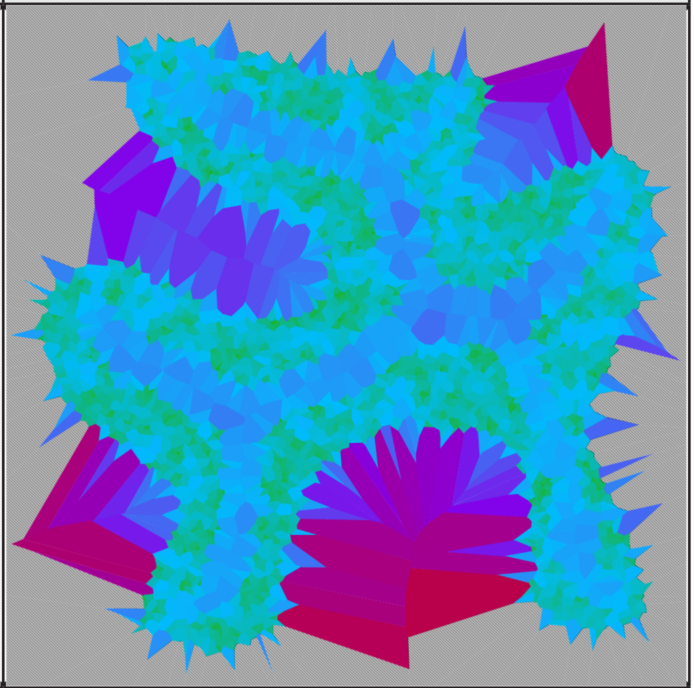
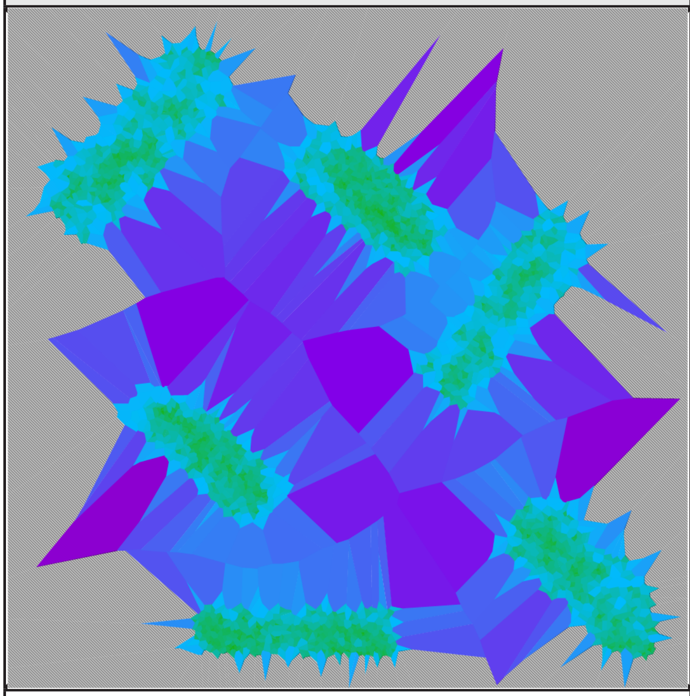
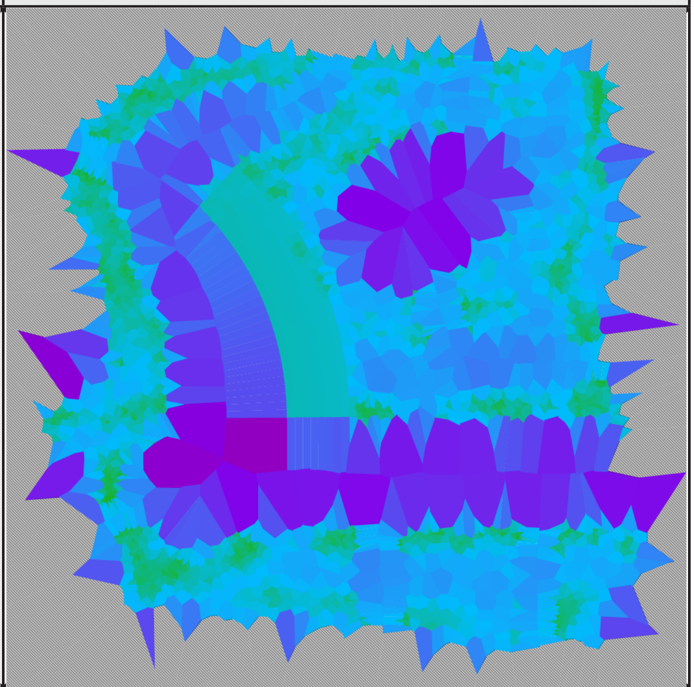
Complexity

Model 5	*Model 6*
	
Model 7	*Model 8*
	
	The size and colouration of Complexity cells highlight the distribution of certain land-use typologies in the spread.

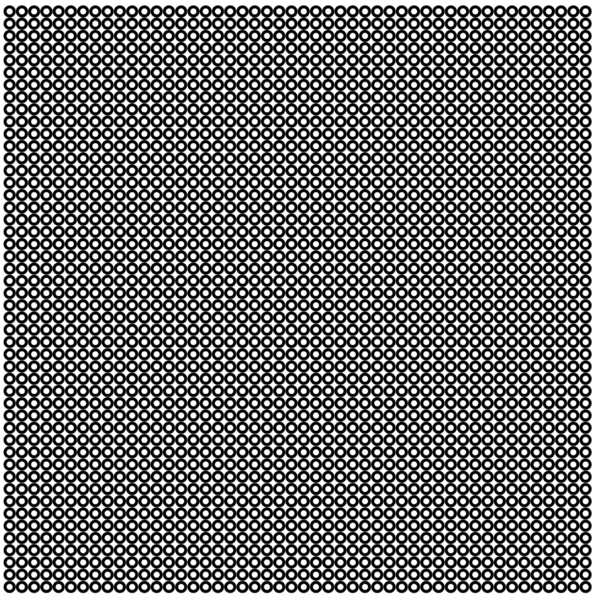
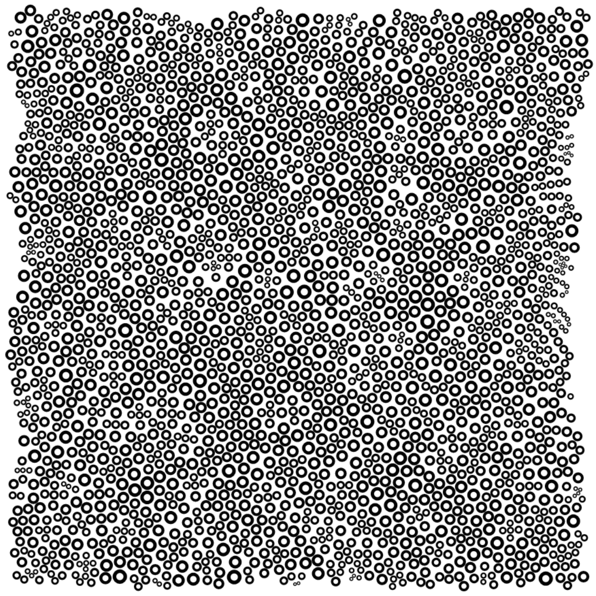
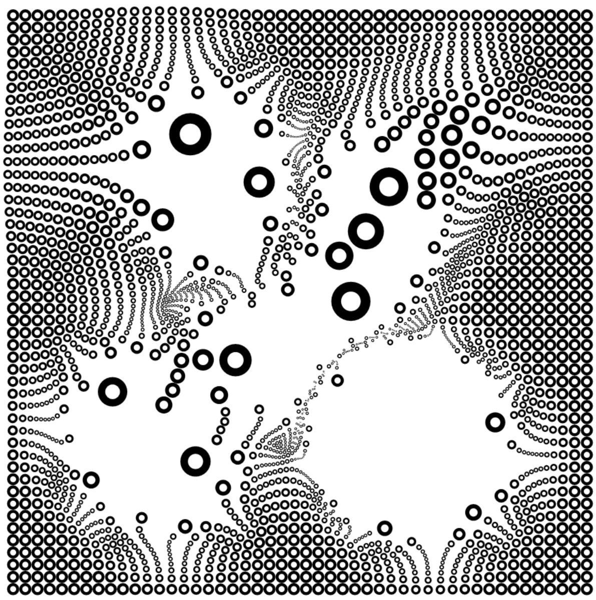
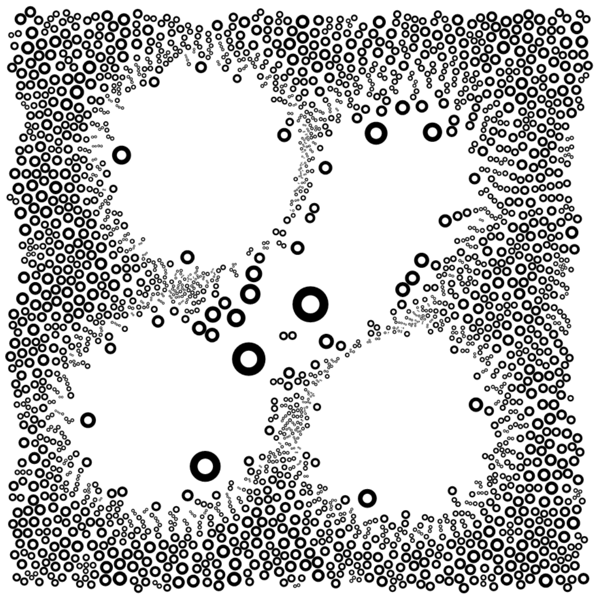
Private Space Index

Model 1	*Model 2*
	
The circumference of all Voronoi cells is identical. All cells are squares. (Actually, due to the shuffling pass described in the Complexity topic most cells are pentagons or hexagons. But these extra edges are extremely short)	In an equalized spread most Voronoi cells are hexagons with more or less equal length edges.
Model 3	*Model 4*
	

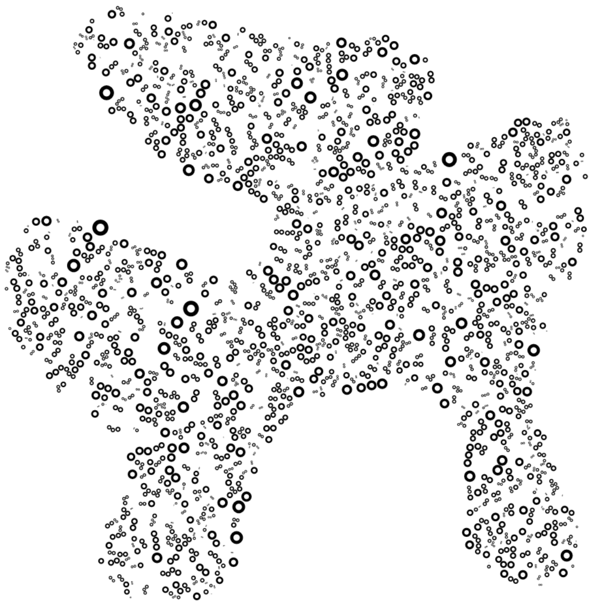
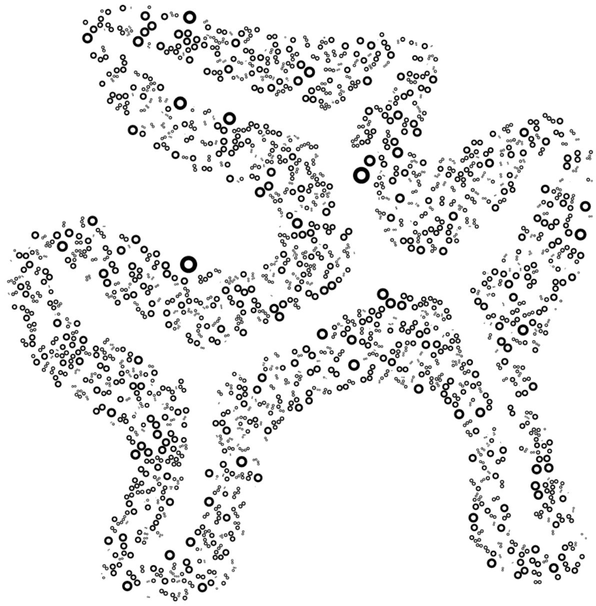
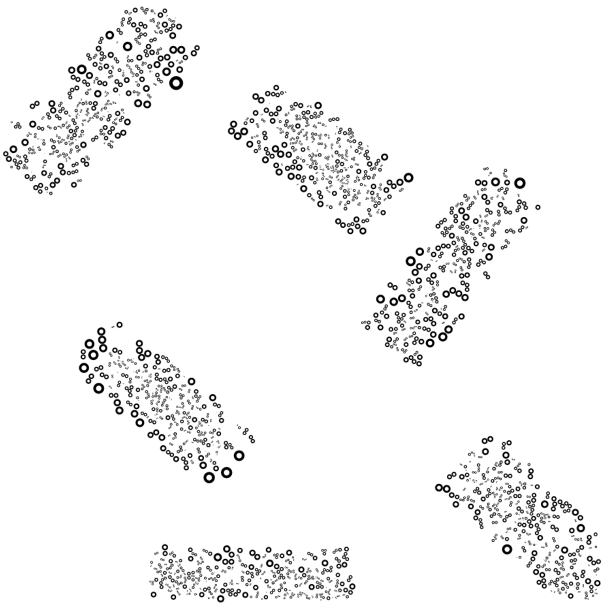
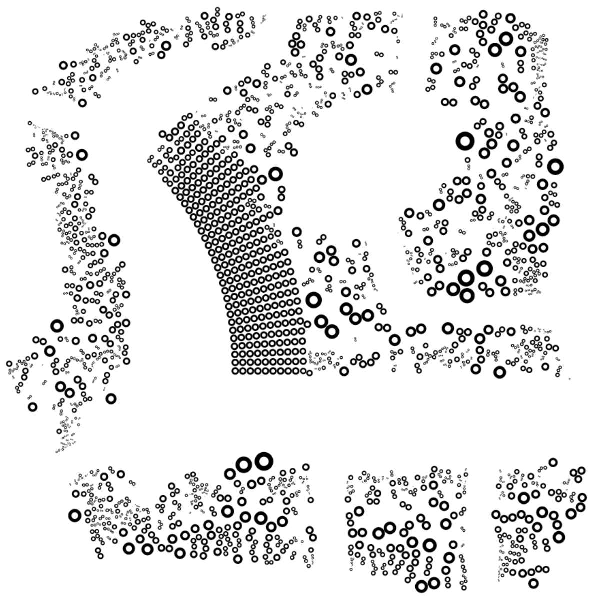
Private Space Index

Model 5	*Model 6*
	
Unequalized Voronoi cells do not resemble equilateral hexagons as well as equalized cells. Hence, they contain more difference in segment length. That is why there is more green in ***M5~M7*** than in ***M2***.	
Model 7	*Model 8*
	

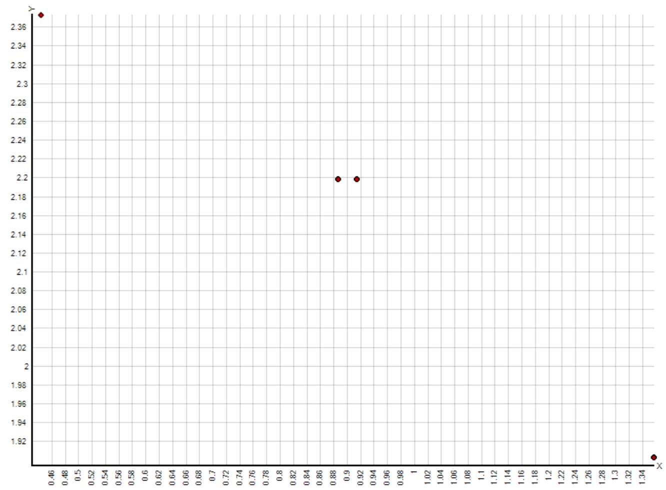
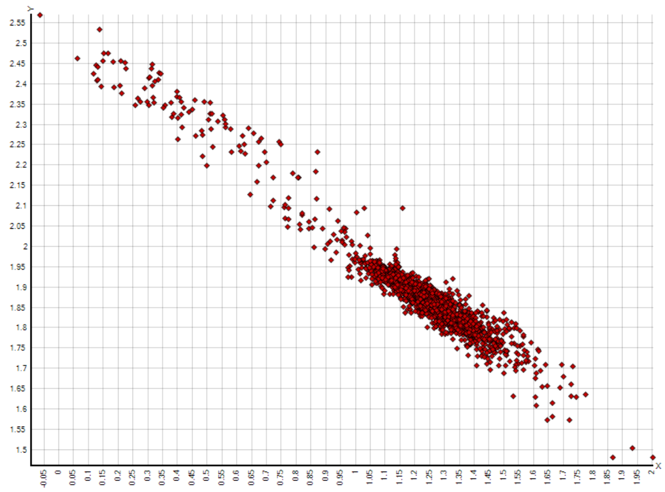
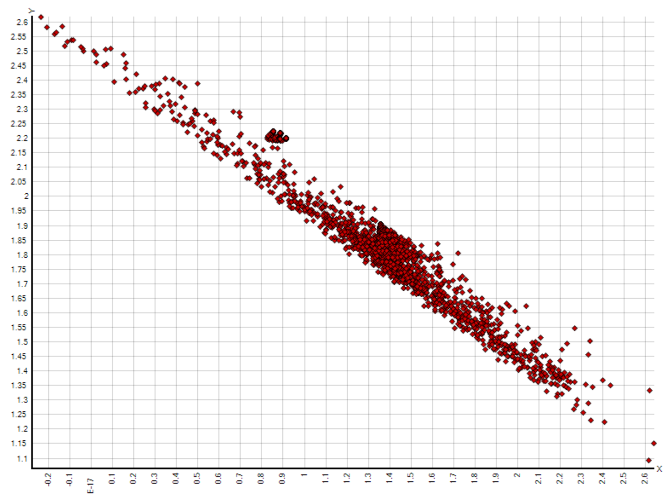
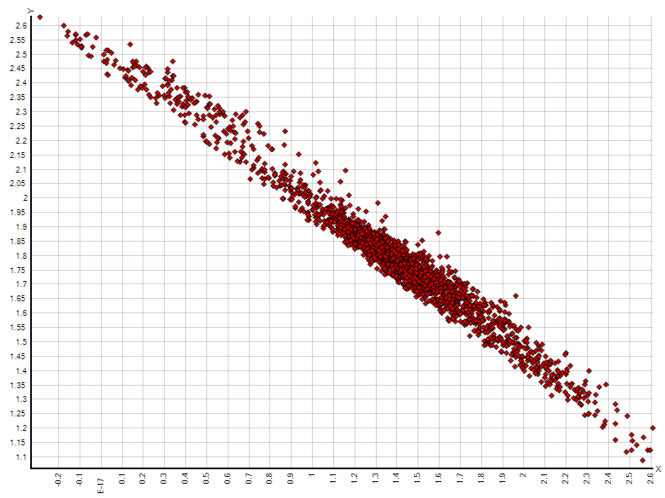
Structure

Model 1	*Model 2*
	
A square grid contains no orientation and every particle is located at the exact centre of the Voronoi cell.	Structure size differs, but the equalization makes sure the gaps between Structure dots are kept small.
Model 3	*Model 4*
	
Once neighbour distances start to differ, particles tend to string together to form sequences. Structure maps provide a different way of looking at public (open) space.	Large dots indicate isolated dwellings. The advantage of the Structure map is that Voronoi cell size is not an important property. Hence, the Structure map displays both private and public area.

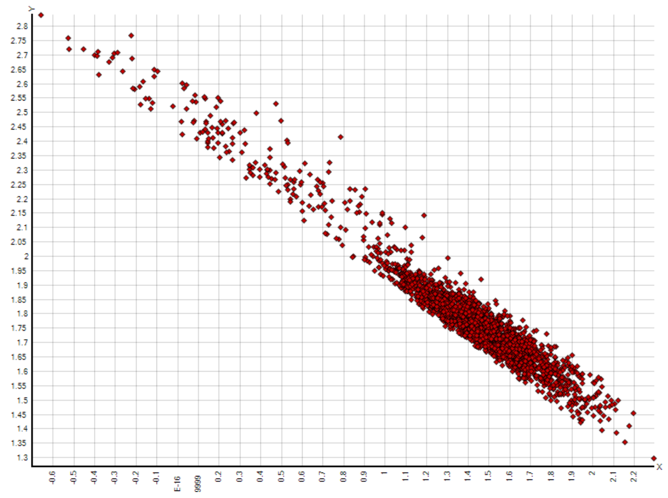
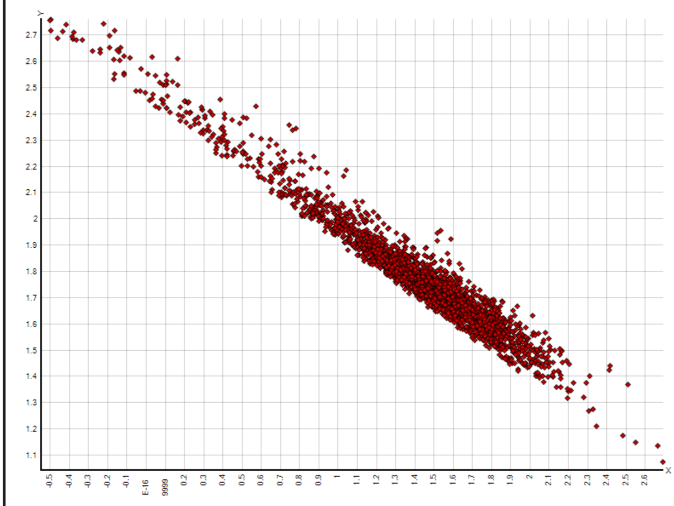
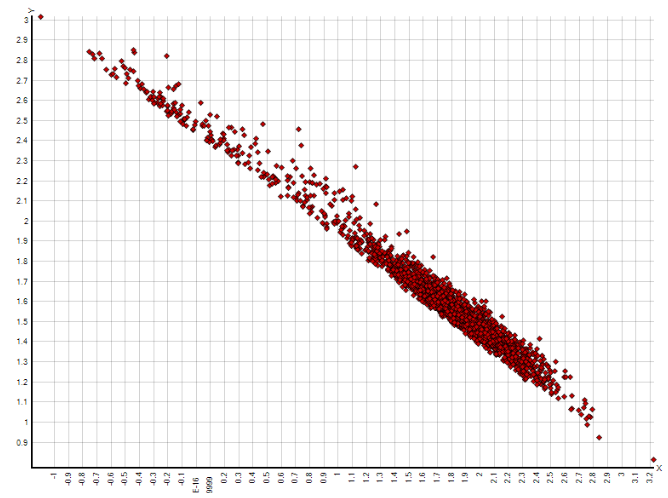
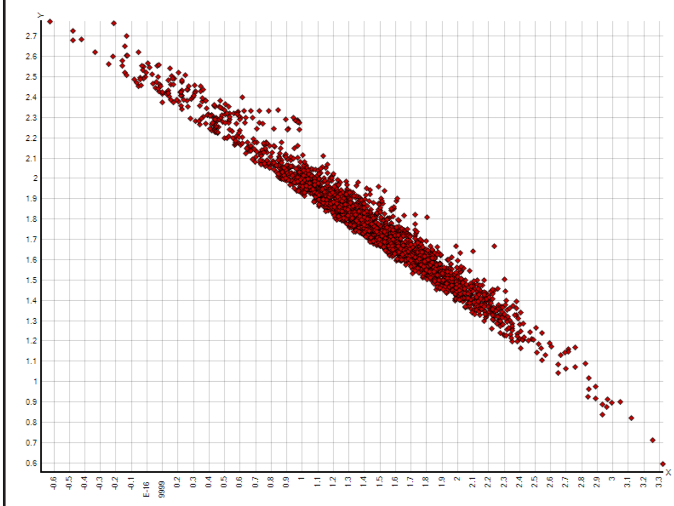
Structure

Model 5	*Model 6*
	
In the case of unordered, unequalized spreads, Structure strings tend to be short. Still, strings can be easily spotted since they contain same size Structure dots (see [2.2.1 Proximity] and [2.2.2 Similarity]).	The Structure map is derived from the Voronoi Diagram, yet the open space is clearly visible. Even the transition from public to (semi)private spaces is represented. Structure is an example of a complex representation of a spread.
Model 7	*Model 8*
	
Here, the Structure map even seems to create stronger silhouettes than the Spread map.	

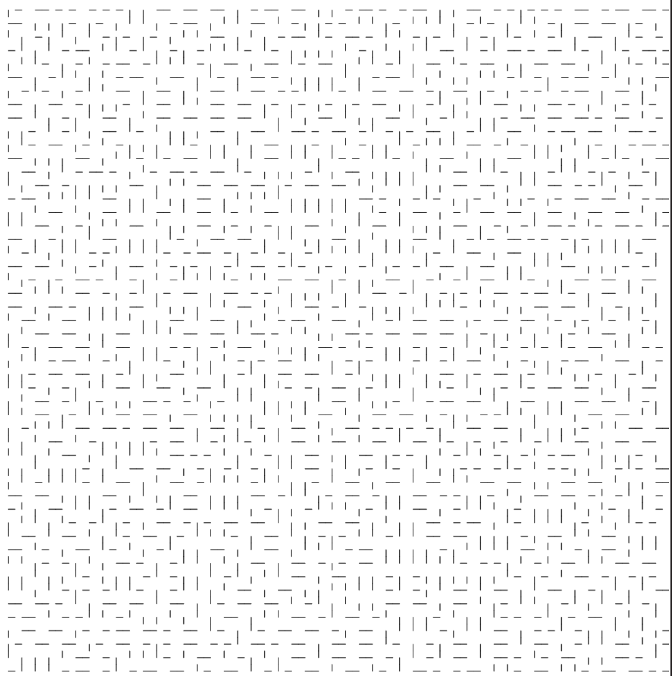
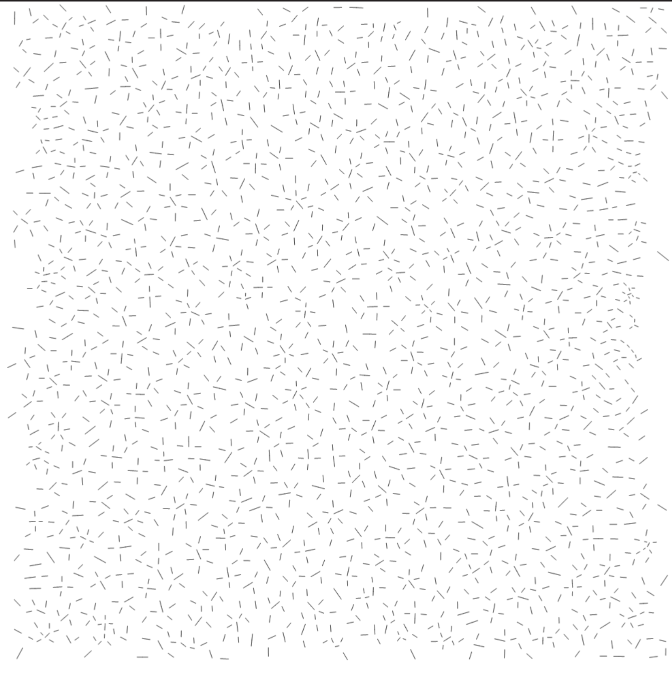
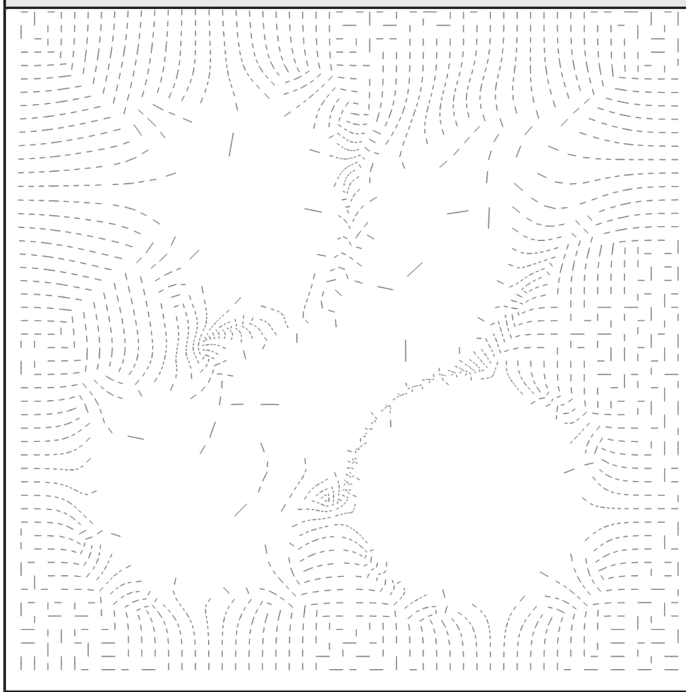
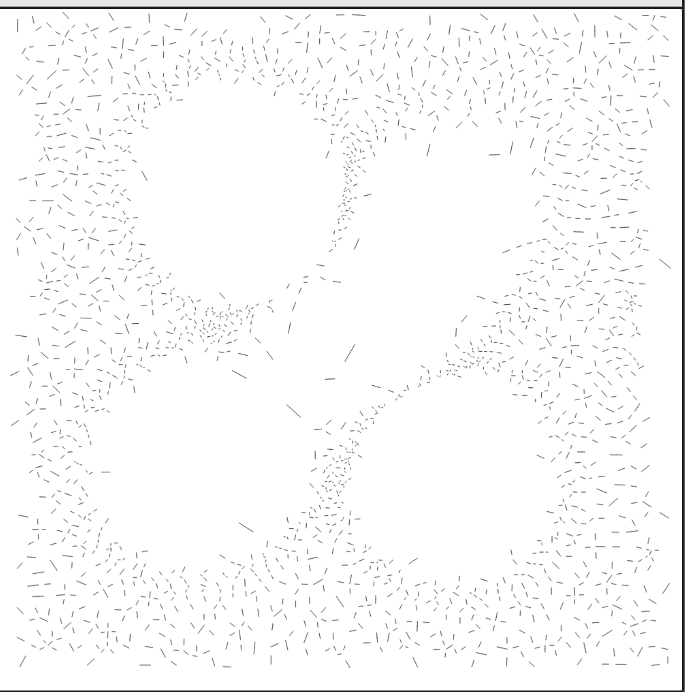
Reactivity

Model 1	*Model 2*
	
Particles are nearly identical and thus they all flock on the same Quantity/Quality coordinates. Note that the axes are scaled to fit the graph-rectangle. The difference between the left-most and right-most values is far smaller here than in ***M2~M8***.	The equalized spread is slightly less diverse than simple randomized spreads. Most particles flock on a specific domain of a Reactivity typology.
Model 3	*Model 4*
	
In this dual-graph a small group of Reactivity typology occurs near ***{0.85, 2.2}***. The occurrence of clusters away from the standard Reactivity line is a rare event but it usually happens with distorted organized input.	

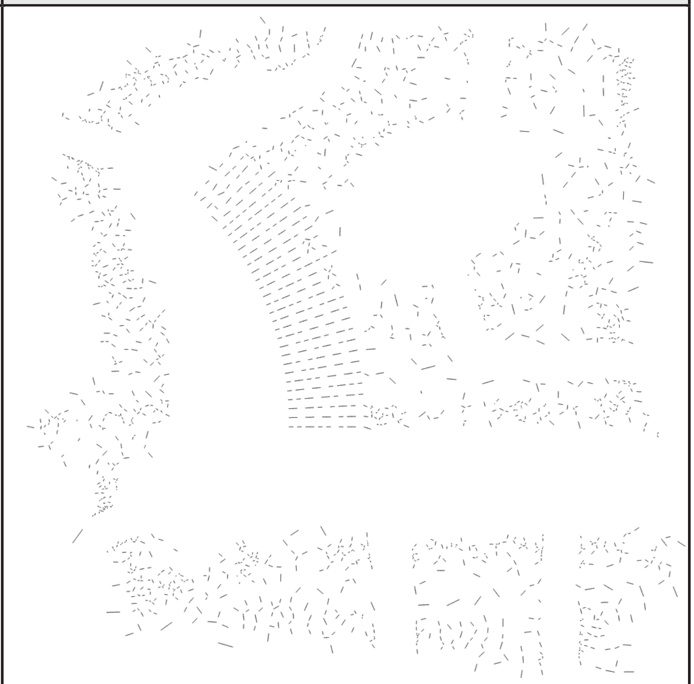
Reactivity

Model 5	*Model 6*
	
	The distribution of particles over the iso-Reactivity line is less focused in ***M6*** than in ***M5***. This indicates a higher diversity of land-use.
Model 7	*Model 8*
	
	In most cases, the Reactivity graph does not tell us a great deal of interesting stuff...

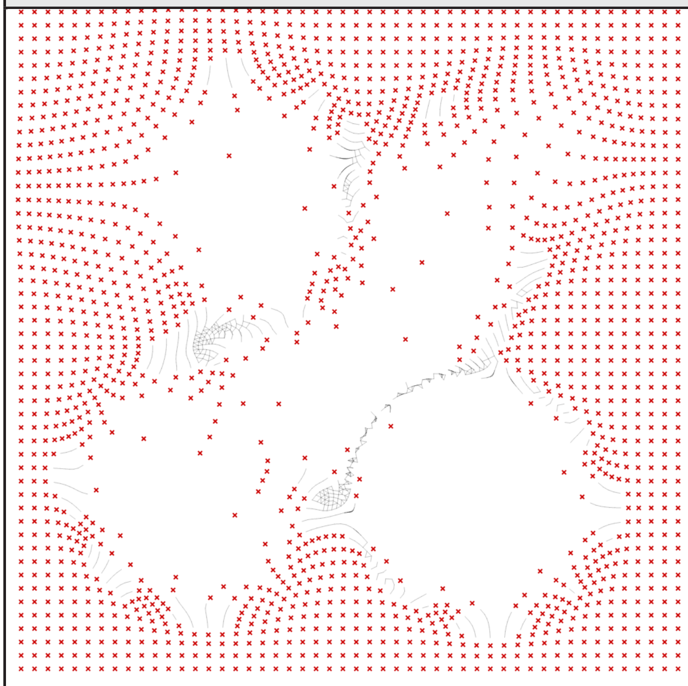
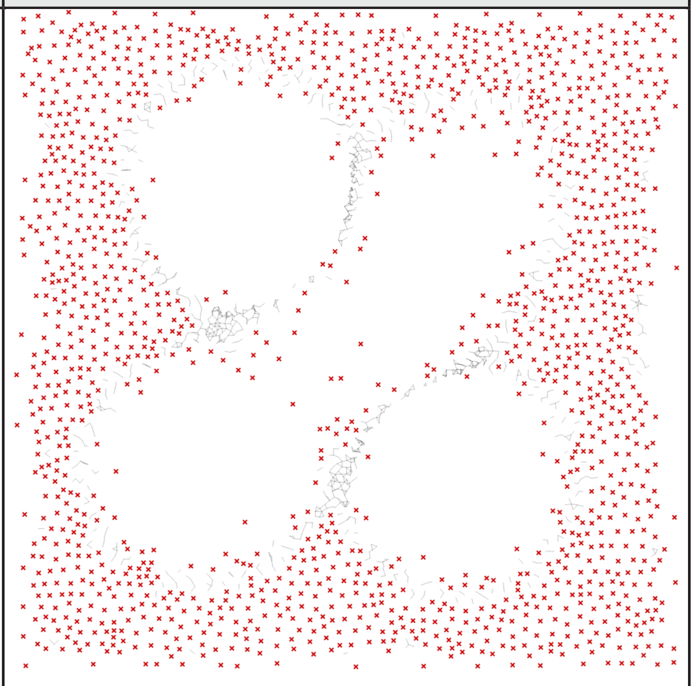
Typology

Model 1	*Model 2*
	
Orientation does not exist in *M1*, but the shuffling pass will create extremely minor orientation values. An additional check for orientation strength might be in order. The image as a whole clearly suggests a null-environment.	Note the occurrence of small groups of inward facing particles. The orientation is not random, but very pronounced on this small level of scale.
Model 3	*Model 4*
	
In this case, the allotment typology map displays nearly the same information as the Structure map, be it without the local size component. Hence, weak orientations such as near the edges of the spread are more visible.	

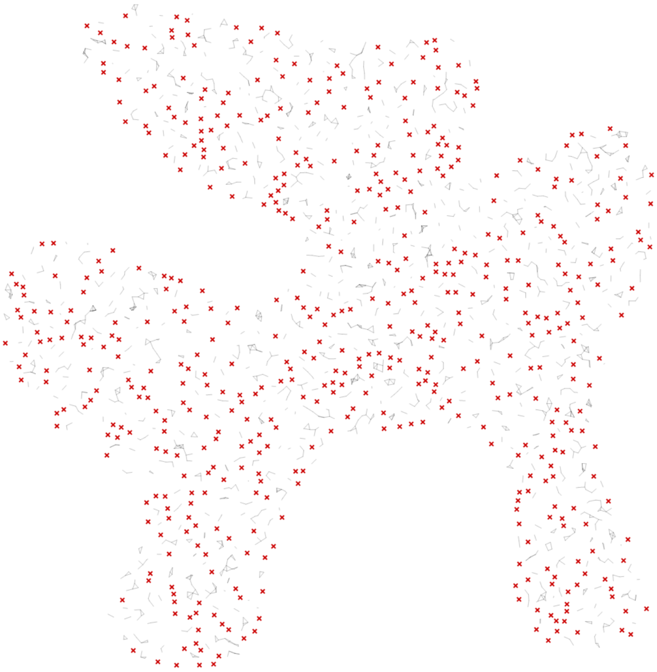
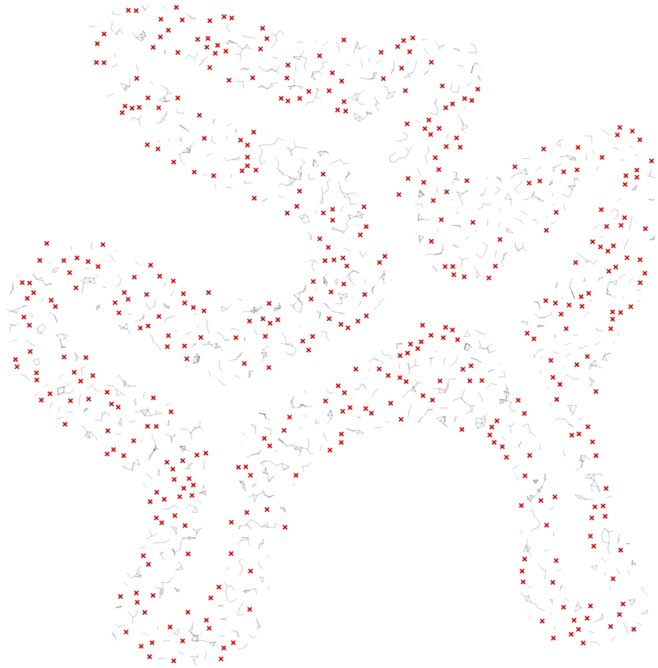
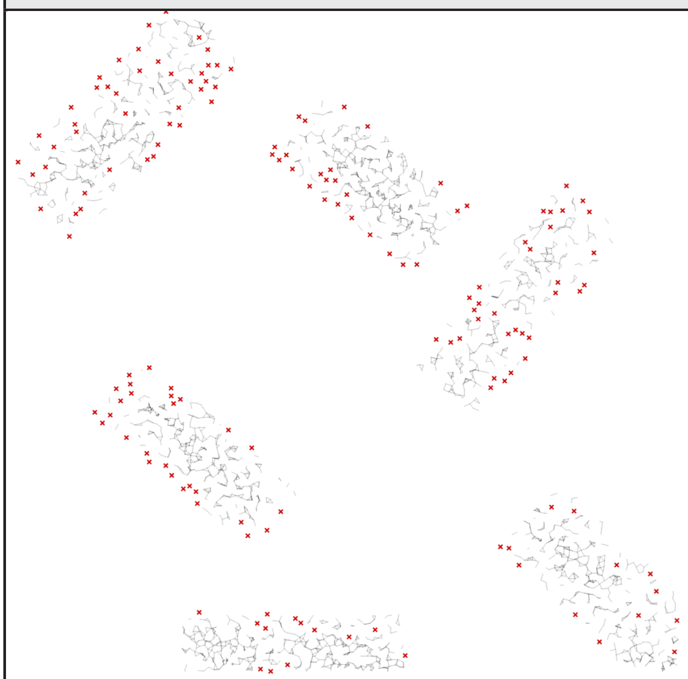
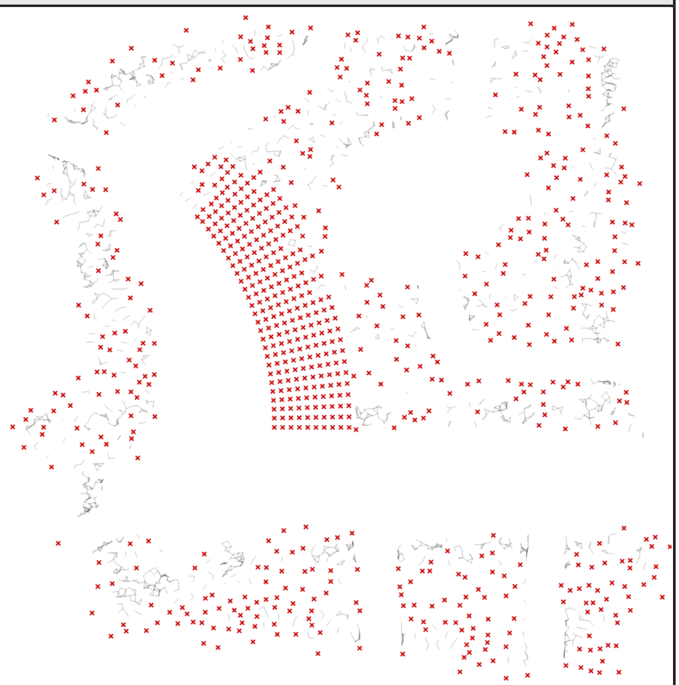
Typology

Model 5	*Model 6*
	
Unequalized spreads result in a random Typology map. The only remaining information is the length of the Typology line segments.	
Model 7	*Model 8*
	

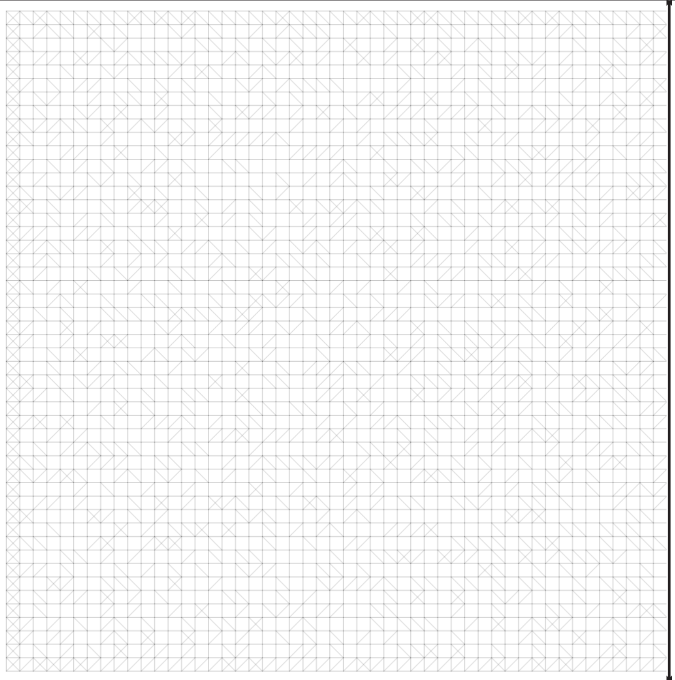
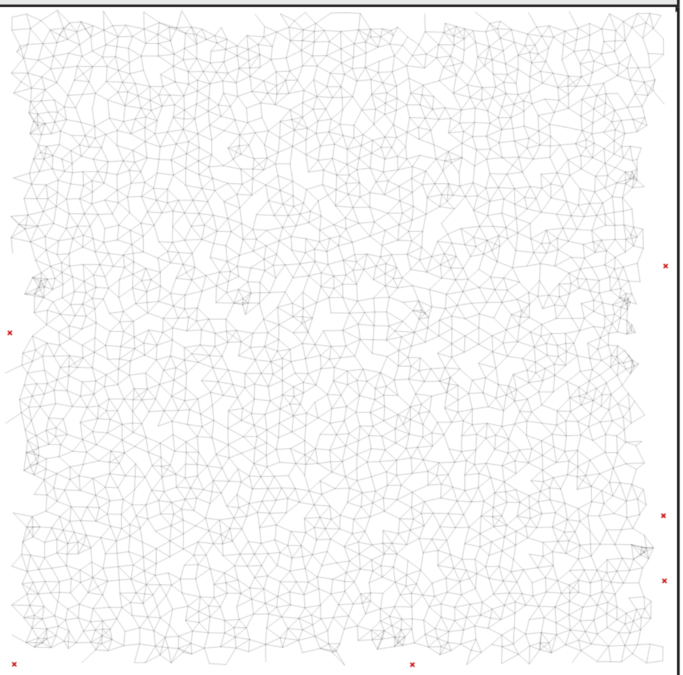
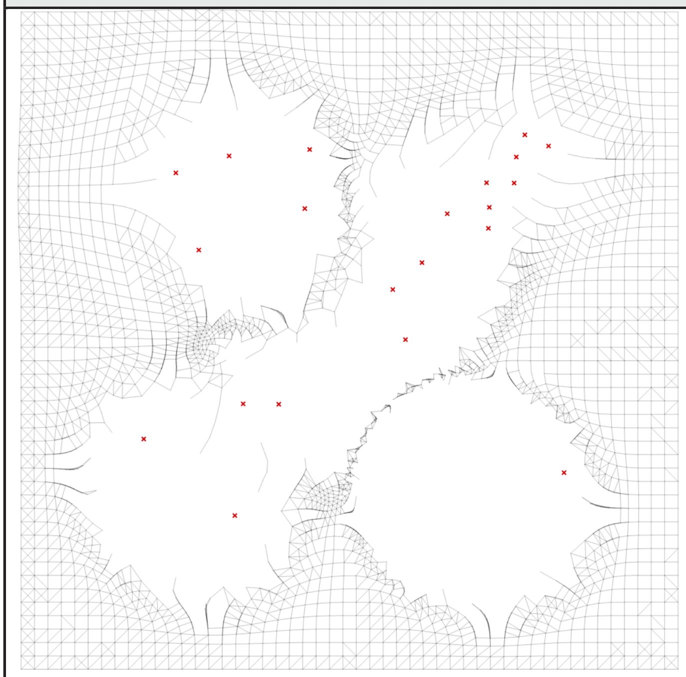
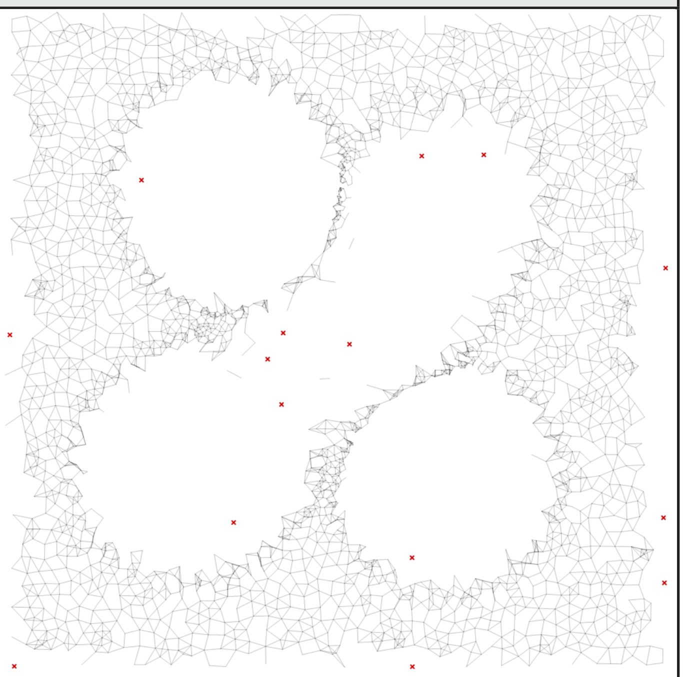
Isolation *{R = 10}*

Model 1	*Model 2*
All particles are isolated on this level of scale.	All particles are isolated on this level of scale.
Model 3	*Model 4*
	
71% of particles are isolated. This information tells us something about the amount of different buildings out spread implies.	59% of particles are isolated.

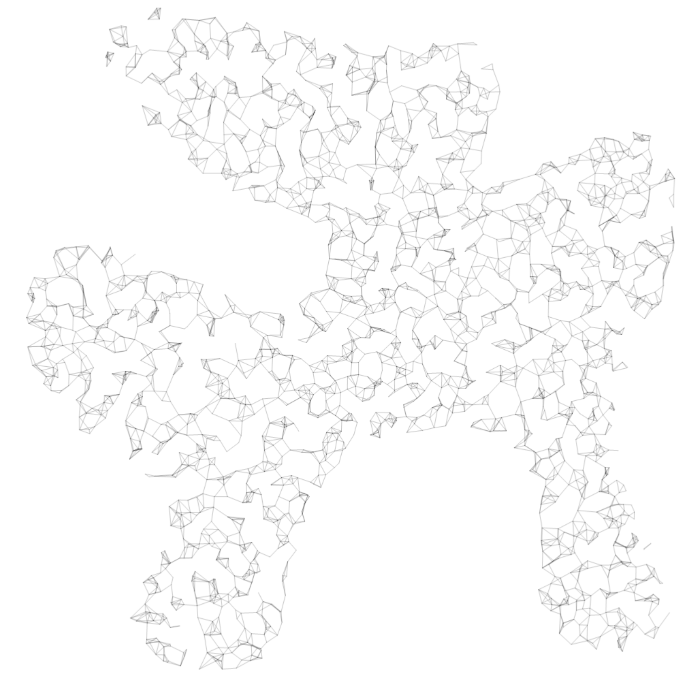
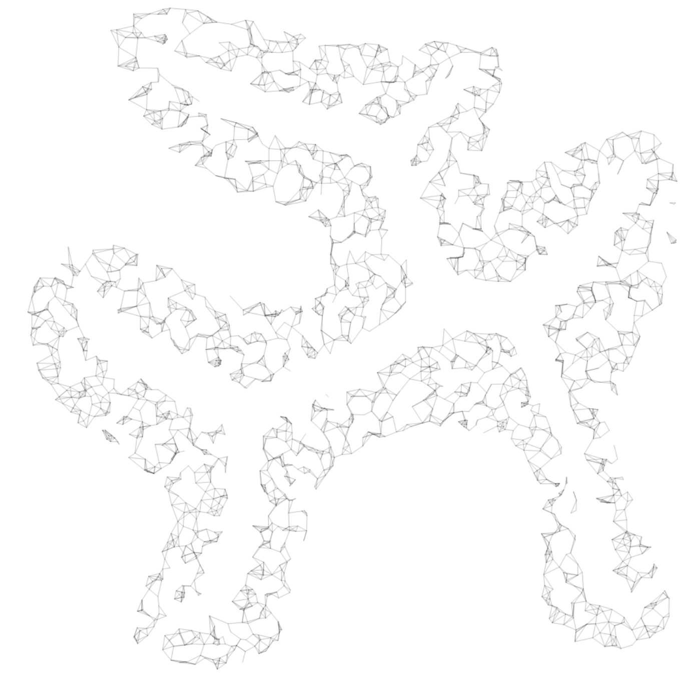
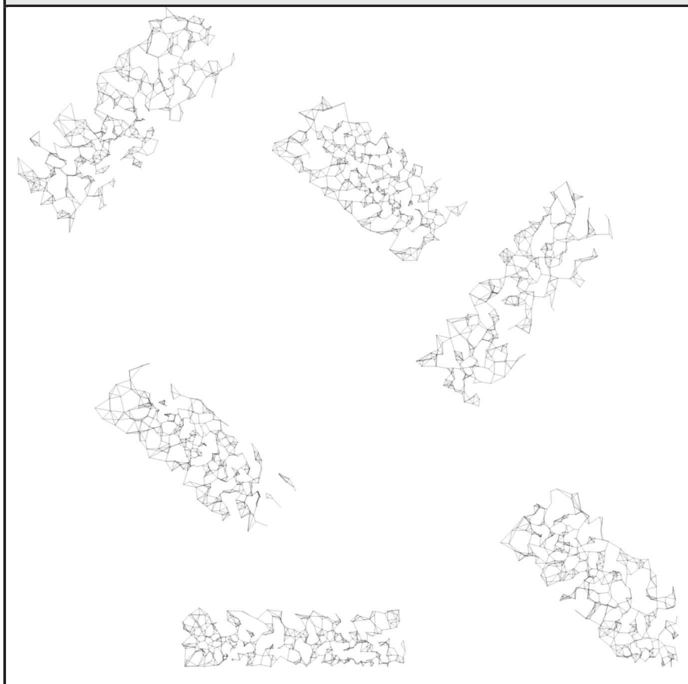
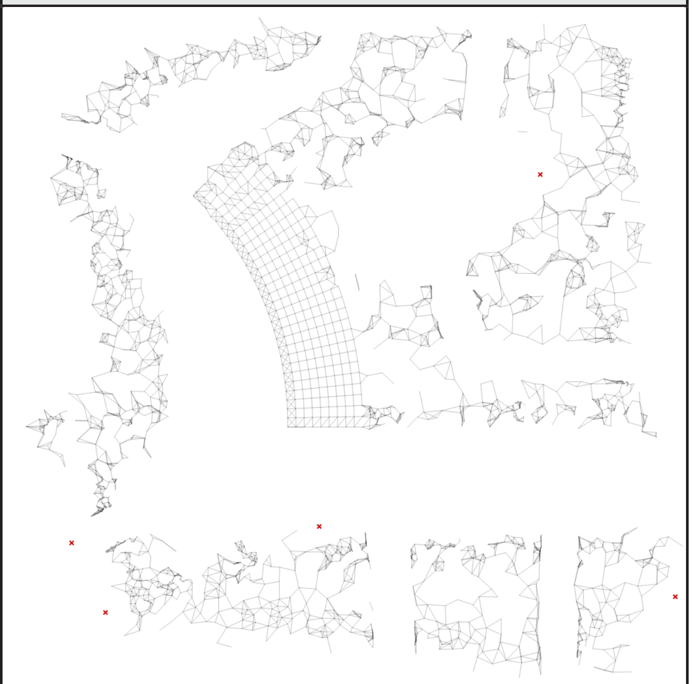
Isolation $\{R = 10\}$

Model 5	*Model 6*
	
21% of particles are isolated.	16% of particles are isolated.
Model 7	*Model 8*
	
6% of particles are isolated.	31% of particles are isolated.

Isolation $\{R = 30\}$

<table>
<tr><th>Model 1</th><th>Model 2</th></tr>
<tr><td></td><td></td></tr>
<tr><td>M1 contains no isolated particles on this level of scale.</td><td>M2 contains 6 isolated particles on this level of scale.</td></tr>
<tr><th>Model 3</th><th>Model 4</th></tr>
<tr><td></td><td></td></tr>
<tr><td>M3 contains 21 isolated particles on this level of scale.</td><td>M4 contains 15 isolated particles on this level of scale.</td></tr>
</table>

Isolation $\{R = 30\}$

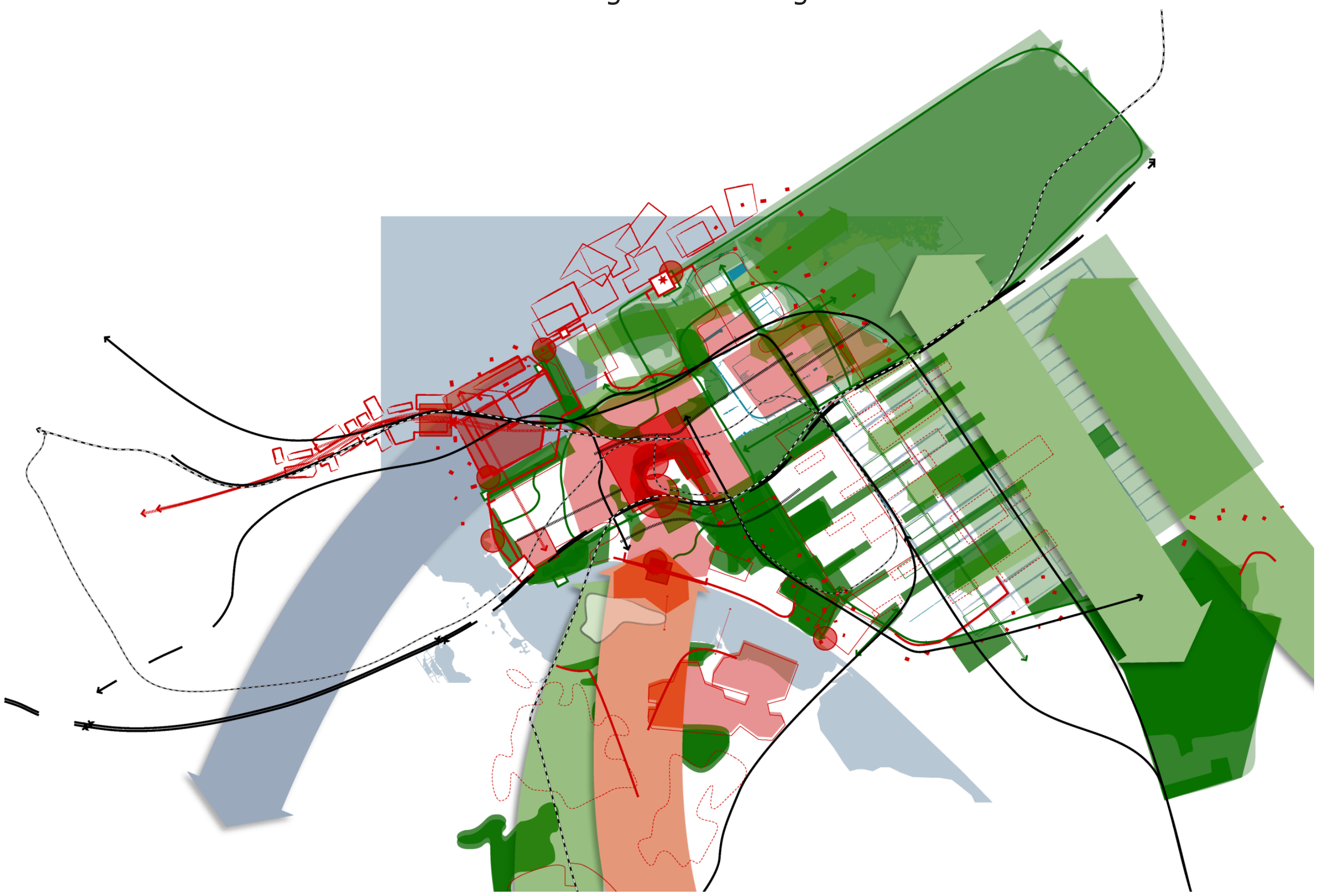
Model 5	*Model 6*
	
M5 contains no isolated particles on this level of scale.	***M6*** contains no isolated particles on this level of scale.
Model 7	*Model 8*
	
M7 contains no isolated particles on this level of scale.	***M8*** contains 5 isolated particles on this level of scale.

B.2.8 Non-hypothetical case study.

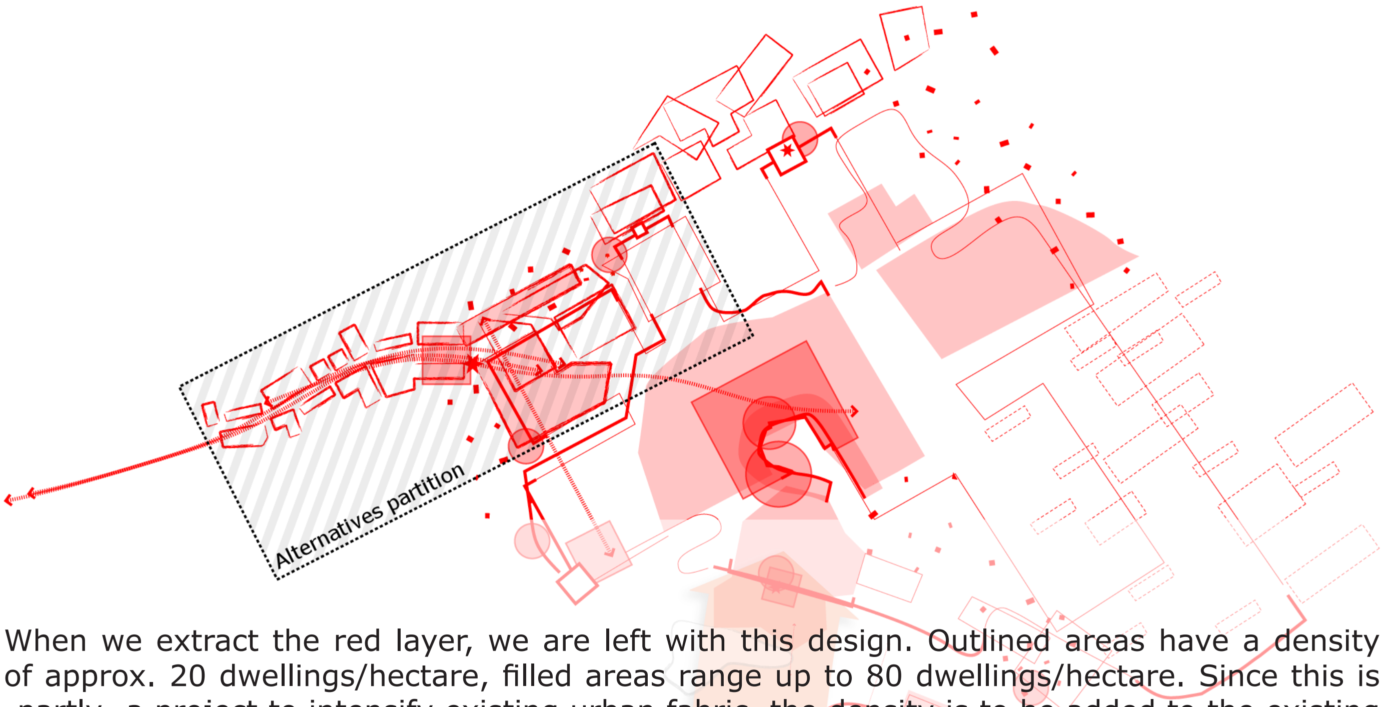
The above eight cases (see [B.2.7 Property case studies]) are hypothetical and were designed specifically to illustrate the inner workings of the software. In this paragraph an existing design will be put to the test. The design was made in a three day workshop (1st, 2nd and 3rd of March 2005) called RGBG Almere[1]. It has been copyrighted by P.G. de Bois and K.A. Buurmans from Atelier Almere and is used here with their permission.

(Almere is a new city in Holland. The land it sits on (Zuidelijk Flevoland) has been reclaimed from the water in 1968 and the first inhabitants appeared in 1976. Currently the city has 175.000 inhabitants and its growth plans caters for as much as 400.000. Almere functions as a pressure valve for Amsterdam. The idea is that people who work in Amsterdam would prefer a calm surrounding to live in, enter Almere. Several high-capacity connections between the two cities have been build to lubricate the transit of commuters. Almere has been having great difficulty to get rid of the 'commuter-town' syndrome. Almere lacks a proper city centre and most of its quarters have been build in low-density typologies. The individual city-parts lack character which makes the whole city boring and 'never-ending'.)

The Four Faces (as the design has been christened) is an interesting case since it features many different typologies. The map below is a result of the super-imposition of the four layers. I shall not make an assessment of the design at this stage.



1. RGBG stands for Red-Green-Blue-Grey and Almere is a city in The Netherlands. RGBG is a design method developed by Atelier Almere, which works through a divide and conquer strategy. Four teams make a cognitive map of a specific layer (R=build area, G=unbuild area, B=water, G=infrastructure) and these maps are then superimposed on each other to form a complete design.



When we extract the red layer, we are left with this design. Outlined areas have a density of approx. 20 dwellings/hectare, filled areas range up to 80 dwellings/hectare. Since this is -partly- a project to intensify existing urban fabric, the density is to be added to the existing density to get the final value. My interest here lies with the string of city-blocks along the top of the design. These low density areas are constructed on the shore and some even in the water. They are grouped along an infrastuctural axis which features both road and rail.

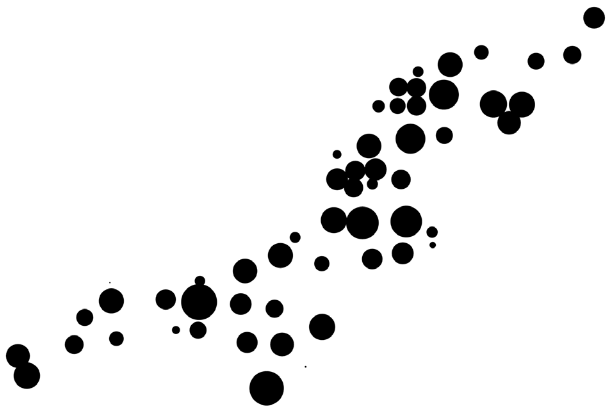
We can derive a set of rules (demands) for evaluation from the history of the site (unfortunately there is no record of this from the design group):

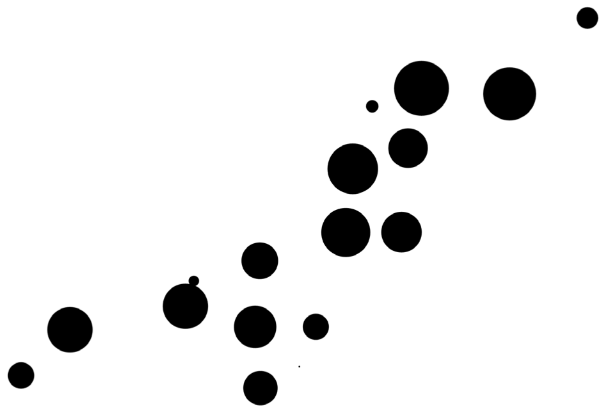
1. The new areas should be city-like. (Getting rid of the commuter-town syndrome)
2. Dwellings in these new areas should be in or near calm spaces. (Adhere to the original philosophy of the city)
3. Areas should be different from each other. (Tackling the boredom)
4. There has to be plenty of space for children to play away from traffic backbones.
5. Sprawl should be minimized. (Prevent further waste of surrounding land)
6. Social interaction should be optimized. (During the day the number of inhabitants will be drastically reduced. The remaining people should not become isolated)

This condensed list contains many soft demands such as "city-like", "different", "social interaction". Most of us will be able to assess the design on these items, but this assessment is tacit knowledge. We cannot convey it to those who do not see it. The following Proximity analysis will provide some solid arguments. I have omitted those images that do not contribute to the analysis.
The plan did not feature further information so the orientation and placement of individual particles is according to the most basic scheme; orientation is always along one of the primary axes of the local shape and all particles have been equalized. Also, three partial parallel designs were created for purposes of comparison, each of which has been modified in a single manner:

- **Alternative A**. Particles have been pushed towards the edges of the local outline, or pulled towards the center. Particle numbers and high-scale distribution are left intact. Local boundaries are not breached. Please note that this is an *elaboration* of the original design. Alternative A is a subset of the designs which are implied by "The Four Faces".
- **Alternative B.** Particles have been duplicated. This is synonymous with using a higher density distribution among the regions. Alternative B differs from the original design, but only in legend, not in drawing.
- **Alternative C**. The entire particle cloud has been scaled down by a factor of two. Alternative C differs from the original in both legend and drawing.

Reductions:

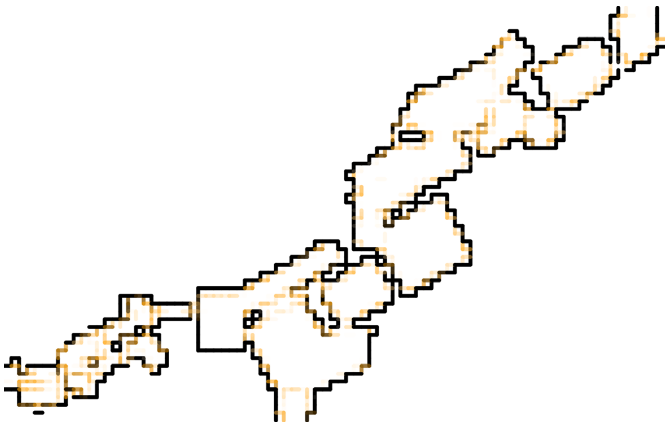

The Four Faces

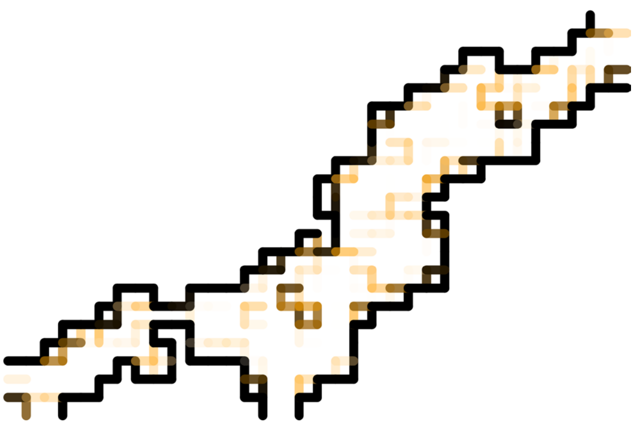

The Four Faces

Reduction diagram for T=3,000 on the left. A groupsize of 3,000 indicates a neighbourhood, even though many groups are clearly smaller than 3,000 dwellings. The emergent picture is variant, which indicates the design is compliant with demand #3. There are many different types of neighbourhood and many different relations between them. There are even some overlaps.

The diagram on the right is a reduction map for T=10,000. The celebrated 'difference' as implied by the T=3,000 map failed to propagate to this higher level of scale. In this map, all super-neighbourhoods are about the same size with about the same distance between them. Still, it is not easy for a design to be notably different on this level of scale. Almere is not an exceptionally large city. None of the alternatives managed to get a better ('better' as measured against demand #3) result for T=10,000.

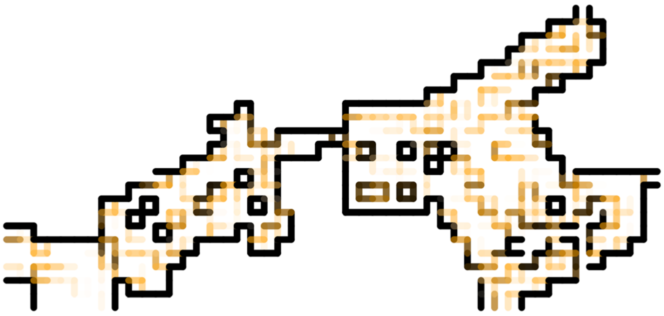
Contrast:


The Four Faces


The Four Faces

Contrast map for R=200m on the left. On this level of scale the design acts as a single entity. It has a complex border which means the ratio Circumference/Area is large. This in turn indicates that demands #2 and #4 are being satisfied. One should not forget that all the empty space to the north of the design is in fact water and not accessible space. Also, many of the outline crevasses are due to infrastructural incisions and thus not very suitable for playing children. There is little contrast to be found within the build areas. This is very predictable since they have been created with a more or less constant density.

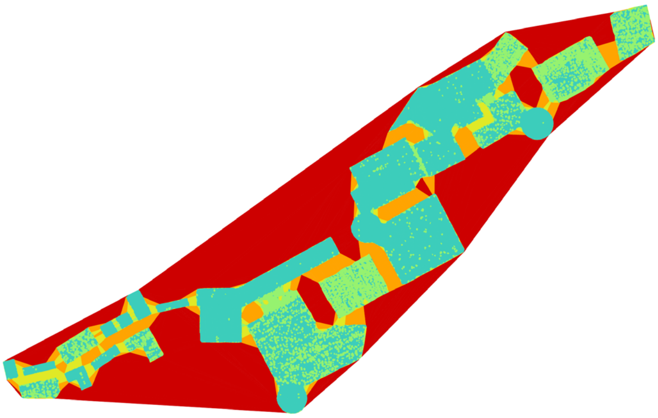
The Contrast map for R=500m on the right tells a slightly different story. On this higher level of scale the contour meanders disappear. There is still little contrast on the inside, which is perhaps a bit surprising. On this scale demand #3 seems to be lacking, while demand #1 is satisfied.



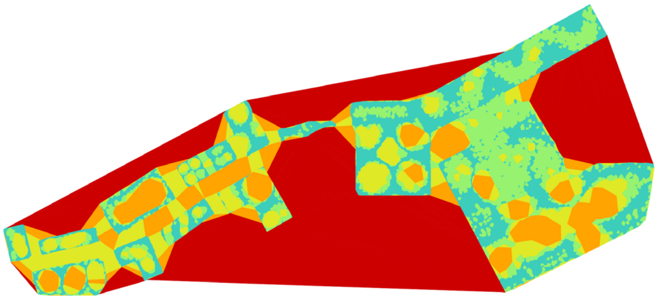
Alternative A

Alternative A was created with an increase of contrast in mind. The adjacent contrast diagram for R=200m of Alternative A already shows a much higher diversity within the city boundary. The newly introduced contrast is not spread evenly across the contents, indicating that diversity on this level of scale is less likely to be interpreted as equality on a higher level of scale.

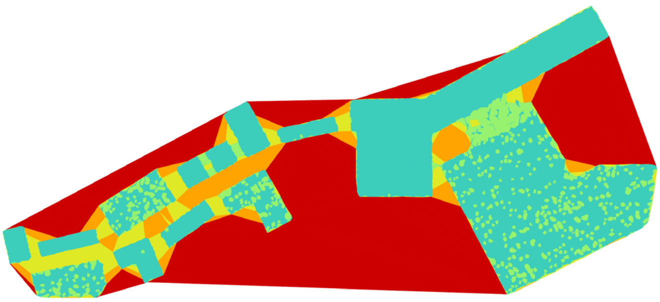
Distribution validity:



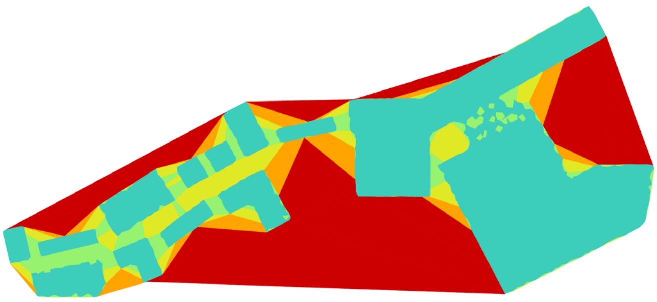
The Four Faces



Alternative A



Alternative B



Alternative C

Distribution validity maps for all four designs. The emotional outlines of the original design (top-left) seem to be compliant with the set of non-red areas. This is an indication that no major errors-of-scale have been made. We also get a clear separation of 'dense' and 'open' space. Orange indicates a strong relation between neighbourhoods. This map states there are no serious issues with this facet of the design.

Since Alternative A (top-right) did not alter the the outline of the local regions, nor the overall density of those regions, the validity map remains largely unchanged on the outside of the city boundary. On the inside however, many new open and semi-open spaces have been found.

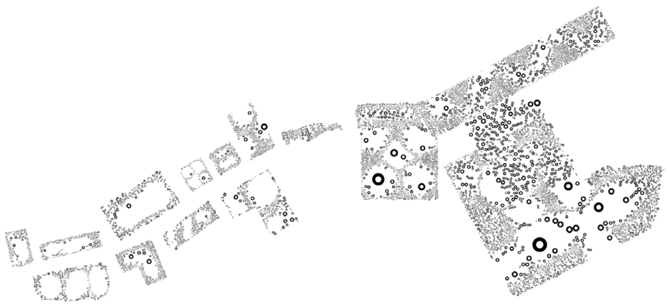
Alternative B (bottom-left) shows no changes for similar reasons.

Alternative C (bottom-right) on the other hand seems to suffer greatly from the different size of the particle set. The outline as indicated by the red areas has become convoluted and large sections of mainly open space have suddenly become part of the 'city-bubble'. Another strong indication that the original design was made on the right scale.

Private Space Structure:

The Four Faces

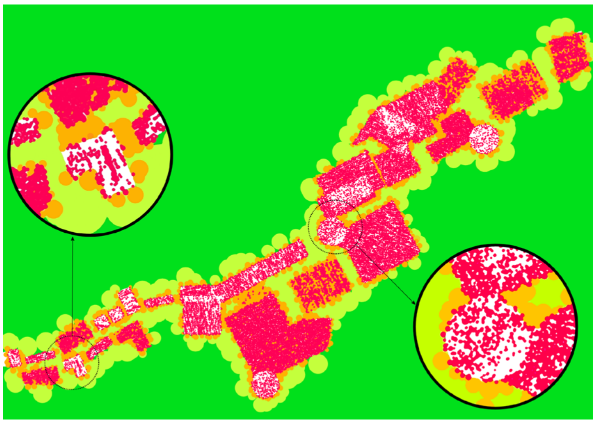
Private space structure map. The orientation and equalization of the spread has removed local differences in typology. The resulting map still gives us useful information though. The radius of the individual dots, the amount of open space between them and their progressive change among neighbours, provides feedback on three different character properties. Here, they all tell the same story. Particles do not differ amongst themselves, only when we look at particle groups do we see a clear division into different typologies. The structure map claims that demand #3 has been met, while demand #1 and #6 have been violated.



Alternative A

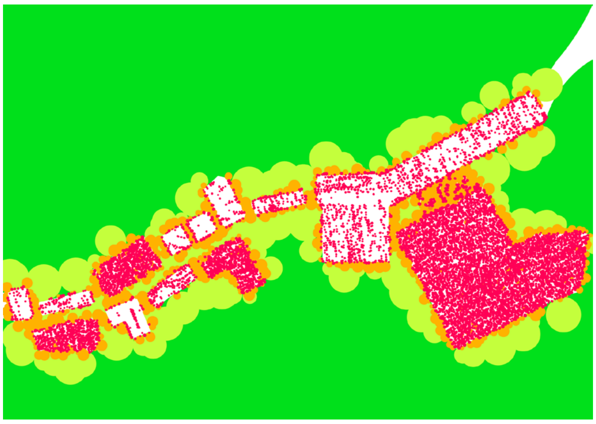
The private space structure map of Alternative A shows a wholly different picture. The manual shifting of particles has ruined the equalization of the initial cloud, resulting in large local differences in private space. In this respect, Alternative A performs much better at **all** demands.

Openness:

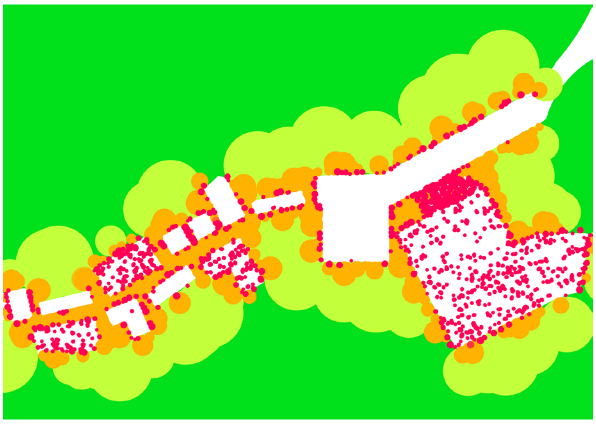


The Four Faces

Alternative A



Alternative B



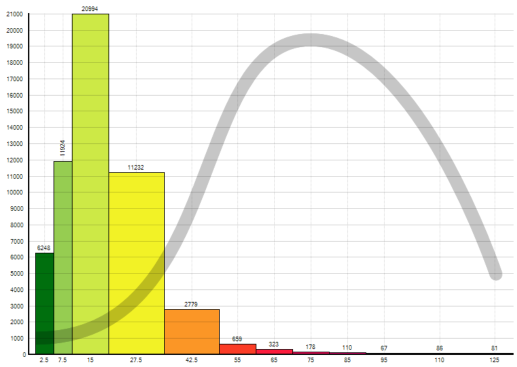
Alternative C

Openness map. The almost total absence of white in the original design (top-left) indicates that this design is very open. Only in the most dense clusters does some white become visible. This is good news for demand #4, but bad news for nearly all others.

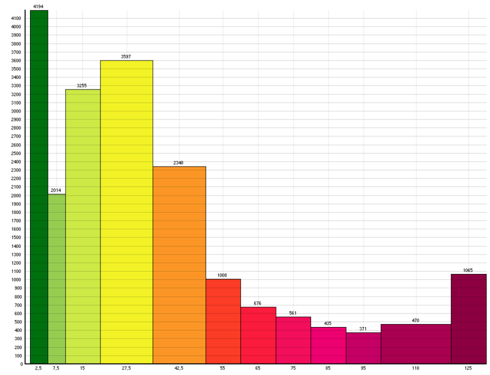
Not surprisingly, all three alternatives perform better since they all feature areas with far more densely packed particles. Alternative A (top-right) is still fairly modest, but it contains approximately twice as much white space as the original design. In additional, the available white space has been grouped together to form clusters and ribbons. Demands #1, #3 and #6 are more suited better with such a dense structure.

Alternative B has the double density of the original design (all particles have been duplicated). Hence, there is more white space available, but it is distributed evenly across the cloud. Alternative C has four times the density value of the original (all particles are packed into a quarter of the surface area) and is therefore almost completely white. This is not a good thing, since demands #2 and #4 are opposed to such a vast, dense spread.

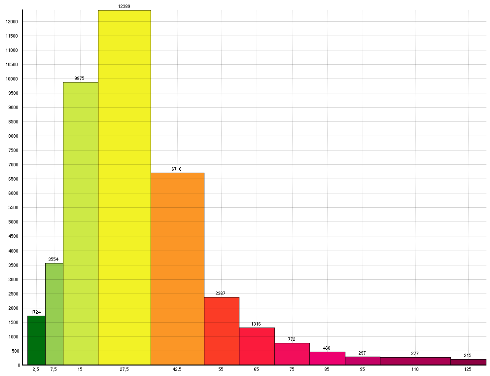
Density:



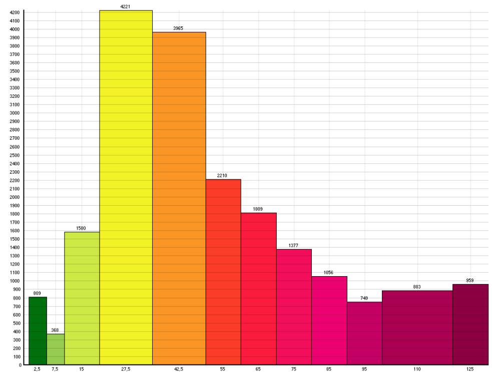
The Four Faces



Alternative A

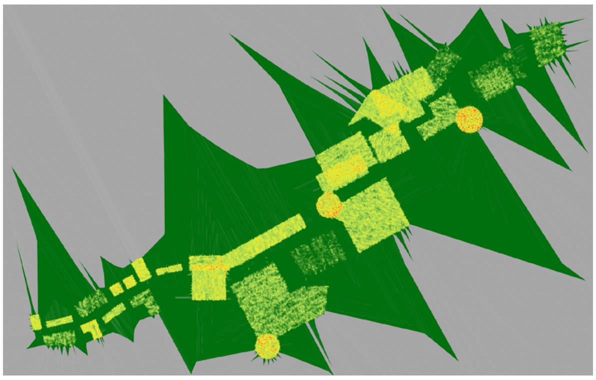


Alternative B

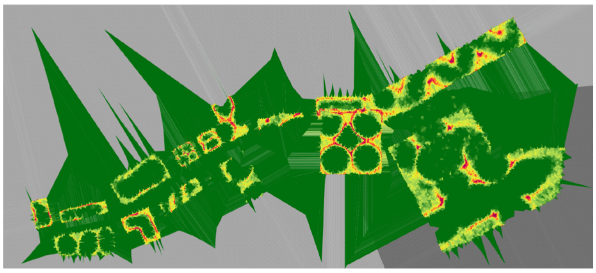


Alternative C

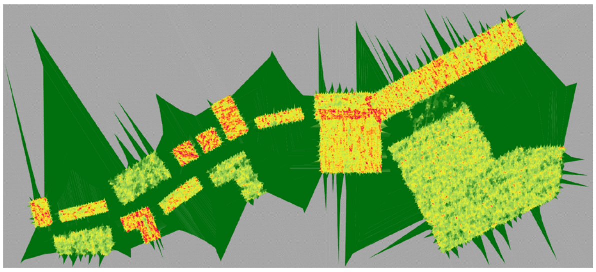
The density graphs of the four design strains interestingly show that none of them are truly urban in character. Though we expect the sum total of all low-density surface areas of any urban region to be larger than the total of all high-density surface areas, the *number* of high-density particles should be far more proportionally for the simple reason that more of them occur in the same area; that is of course what high-density means. A very common distribution for down-town urban systems in Holland is drawn as a grey curve in the top-left graph. Even Alternative C, which appeared to be so dense in the Openness analysis is lacking. In fact, there is very little difference between alternatives A and C, which is a conclusion one could not have guessed based on the previous sections. Even the Voronoi Density graphs (see below) do not convey this result...



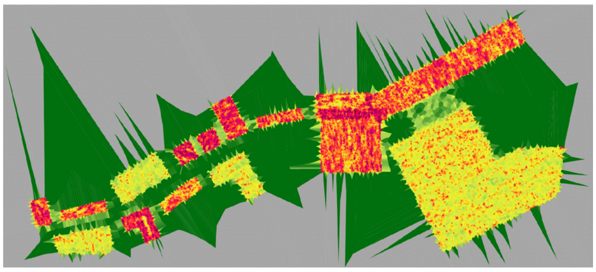
The Four Faces



Alternative A

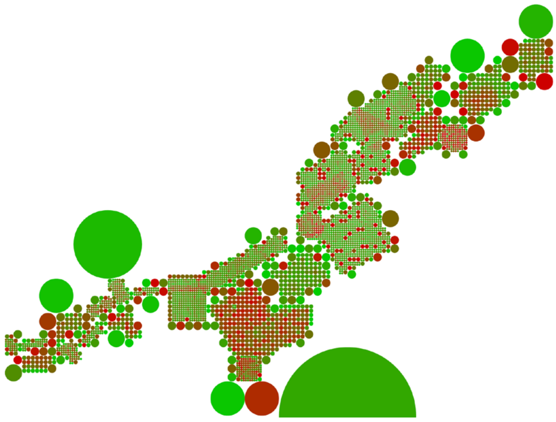


Alternative B

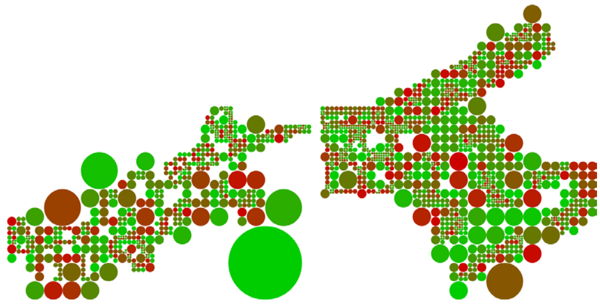


Alternative C

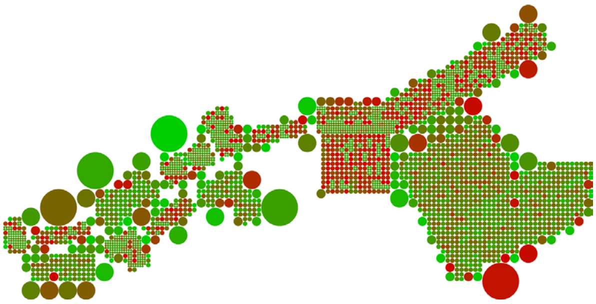
Shredding:



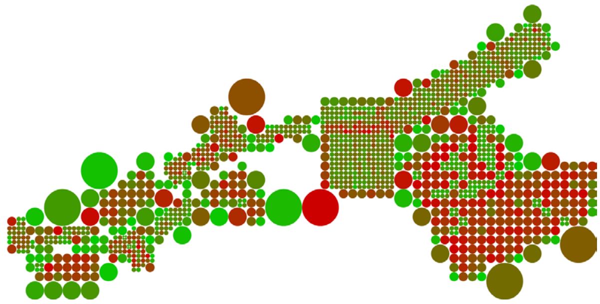
The Four Faces



Alternative A



Alternative B



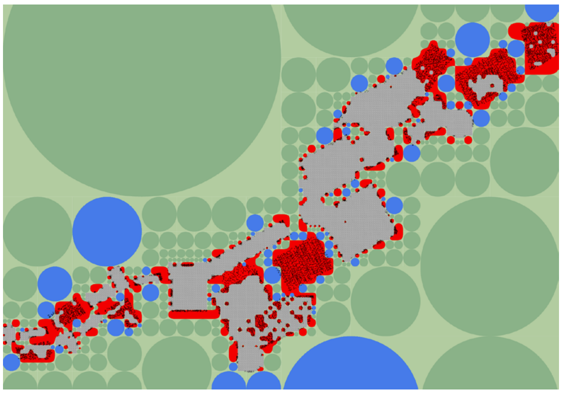
Alternative C

The shredding map emphasizes the unity of the design as a whole, but also the lack of difference within it. Very few areas exhibit internal shredding. Urban areas are typically more divers than this. Demands #1 and #3 are violated. Since Alternatives B and C do nothing to counter the distribution of particles at large, we expect them to resemble the original design in this respect. Fortunately, our expectations are met.

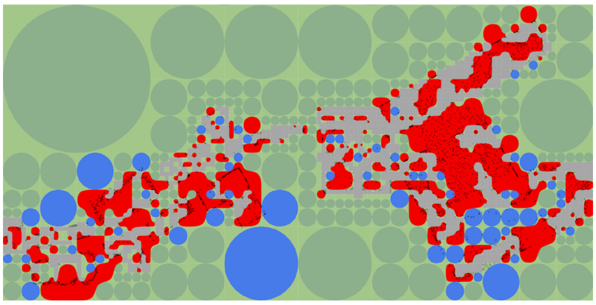
(I must admit it seems a bit unscientific to rejoice the analytical results when they are in sync with predictions, as well as when they turn out to be counter-intuitive. The experimental nature of this project justifies such appearently arbitrary behaviour. On the one hand it is pleasing to know that -at least some of the results- seem to be correct; it is an indication of trustworthyness for the whole paradigm. Yet this whole exercise would be pointless if the analysis only tells us what we already know. Whenever results do not turn out to be conform expectation, I attempt to offer an explanation of why this is so. However, the experimental nature of this project could *also* mean some results are simply wrong.)

Apart from meeting expectations concerning the local diversity of the respective designs, the shredding map does offer an additional advantage; it can be used as a measure of how much more diversity Alternative A actually offers. Not just "more", but "2.6 times more". I have made this point before, but it is an important one and I feel it must be emphasized.

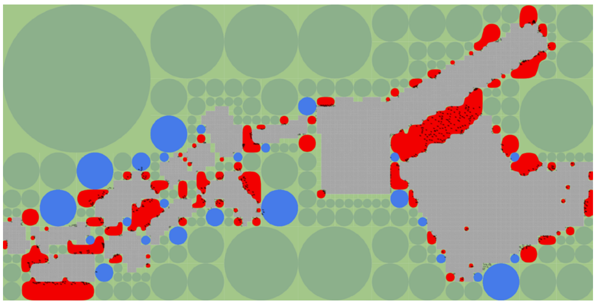
Sprawl:



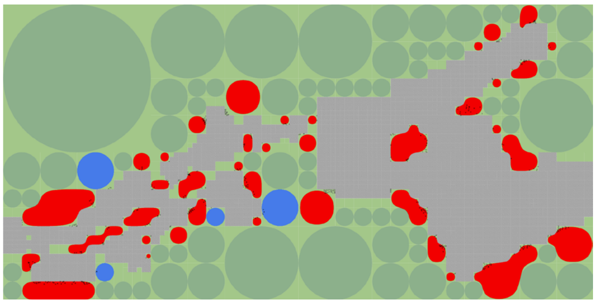
The Four Faces



Alternative A



Alternative B



Alternative C

The only demand not yet dealt with is #5; "Sprawl should be minimized". We have plenty of data on the distribution of density so far, but none yet on the lack of it. Like many other properties, sprawl is not a clearly defined one. Whether an area is identified as sprawl may depend on cultural, architectural and even personal factors. In this case I have chosen the {min=1.0; max=10.0} map, but the other presets yielded a fairly similar outcome.

The sprawl map tells us two things:
1. How much area is 'wasted' (sprawl is almost always used as a negative characteristic)
2. How many particles are contained within sprawl boundaries

The point that sprawl maps are severely non-affine has already been raised in paragraph [B.2.6.5 Sprawl] and in the maps above we can clearly see where it has gone wrong. We should treat this part of the analysis with scepsis.

The upper left map shows that the original design scores rather poorly on both sprawl characteristics. Very large, connected areas have been flagged and many particles reside within the red boundaries. Interestingly though, Alternative A, which has scored better on all other properties, receives a severe blow. There is a slight reassurance since many of the red blobs contain dense clusters of particles which indicates a non-affine problem. But even without these suspect blobs, Alternative A still performs the worst.

Since sprawl is fairly scale-independant, Alternatives B and C yield nearly similar results. Indeed, the difference between them should be contributed once more to non-affine issues.

Conclusion:

The results of the analysis is that demands #2, #4 and #6 are fully satisfied. Demand #3 suffers from some contradictory evaluation, but does not appear to be a serious problem, in the end it all comes down to what is meant by “difference”. Demands #1 and #5 are severely frustrated, but this is partly a problem of the bland translation from the classical design medium to the proximity medium. Local intensification (Alternative A) have solved many issues regarding sprawl, but the overall density of the plan as a whole is too low for an urban region.

On the whole it appears to be a good design. Not in the sense of “a good finished design”, but rather in the sense of “a design strain worthy of further development”[1]. The original particle cloud and Alternative A are both subsets of the implied design and between them they resolve nearly all issues.

We’ve also learned that simply packing in more dwellings isn’t a fool-proof solution and in fact only makes it worse in some cases. There is more to cities than high density.

1. See the design process diagram in paragraph [*A.4.3 Emergent Planning*]

B.2.9 Assessment of the project

I first started thinking about sketching software for urbanism in my fourth year of study. Though no solid solution was clear to me yet, it was obvious that there were many lapses in the design process as we know it. It wasn't until I approached T.M. de Jong in the summer of 2004 that things started to move. I have always been attracted to the teachings of methodology as offered by T.M. de Jong in his many publications and he seemed -and was- the logical choice. The one thing he could not evaluate was the actual software I was writing. This proved to be a big problem since so much[1] had to be written before tangible results appeared.

Over the course of the project, the direction in which we were heading changed more than once. The biggest mutation occurred about a year into the project (the halfway point) when it was decided to abandon the shape-approach and to focus completely on spread as a design tool. Even though it rendered much of what had been finished useless, problems suddenly started to solve themselves all over the place.

One year after this key-event, as I'm writing these lines, the project has become something far beyond what I imagined, let alone planned for. I am the first to admit there are many problems still unsolved, both peripheral and elemental. The software is not running as fast as I would like (it still takes approximately ten minutes to compose an analysis document for a spread with 50,000 particles) and some drawing tools are still far from fluent. These issues are mostly due to my inexperience with programming and optimization but since this is an experimental project, they are not terribly important. Problems at the core of the system do not exist, for the simple reason that the core is so incredibly simple. It allows for an almost infinite elaboration of principles and theorems, both for evaluation and simulation.

The implementation I have written touches briefly upon the subject of evaluation and comparison and does so with varying degrees of success. Some properties fail to deliver in certain cases and others do not really tell us anything we did not already know. This is a subject for future research and elaboration. The biggest problem with the platform as it exists today is the non-affinity of many derivatives, and thus many properties. This issue is profound and threatens the scientific nature of the Proximity paradigm, or at least the scientific nature of this particular implementation. I have composed an indicatory list of property problems. More dots indicate a more severe impact on the property.

- Non-affine (Small linear transformations result in abrupt changes)
- Precipitous (Small deformations result in abrupt changes[2])
- Scale dependency (Linked to a certain level of scale[3])
- False-accuracy sensitivity (Exact particle coordinates play an important role)
- Low relevance (Value is of little use in evaluation or comparison)
- Random dependency (Value contains a random component)
- Relative (Not suited for comparison[4])
- Preset limited (Value contains a preset component[5])
- Non-characteristic (Value does not resemble the model[6])
- Small set sensitivity (Value becomes useless on small spreads)

1. The software as it stands today measures nearly 22,000 lines of source code. This is less than half the amount of code that has been abandoned over the course of the project.
2. Linear transformations affect the entire spread, they are homeomorphic. Deformations affect a subset of particles in a potentially non-linear fashion.
3. This is not necessarily a problem. Truths are often scale-bound so we need mechanisms to evaluate the same properties on different scales. The problem is, that a certain scale must be chosen. If a slightly different scale results in a wholly different value, the usability of a property is compromised.
4. Again, this is not a problem per se, it just means the field of application for this property is reduced.
5. If the user has access to presets, this limitation can in fact become a plus. It requires a fair amount of knowledge on the part of the user though.
6. Non-characteristic properties inform us about problems, but they do not offer information on where these problems are located. They are therefore less suited for evaluation. They are useful for comparison.

Property \ Problem	Non-affine	Precipitous	Scale dependency	False-accuracy sensitivity	Low relevance	Random dependency	Relative	Preset limited	Non-characteristic	Small set sensitivity
Amounts									•••	
Spread				•	•					
Shared Space (index)				••		•				
Contrast	•••	••	•••					•••		
Sprawl	••	••		•				•••		••
Reduction	••	••		•					•	••
Density	•	•		•		•		•		
Shared Space Hierarchy				••	•	•			•	
Openness				••				•		
Hierarchy	•	•	•••	•	•		•••	•••		•
Flexibility	•	•	•••	•	•		•••	•••		
Clustering	•	•	•••	•	•					
Shredding	••	••		•			•			••
Complexity		••		••	••	•	•••	•	•	
Private Space (Index)										
Structure							•••			
Quality				••	••		•••		•	•••
Quantity				••	••		•••		•	•••
Interaction	•	•	•••		•		•••	•	•••	
Reactivity		••		•	••		•••		••	
Typology		•		•			•			
Value				••	•••		•••		••	•••
Isolation	•	•	•••	•						•
Grouping	•	•	•••	•						••

Fortunately, the software is modular and these problems can be tackled piecewise. Further derivatives should be developed and along with them, further and more intricate properties. A lot of work on this field has already been done[1] and all that is needed, is a platform which combines the powers of computational design with the foundation of human culture. Existing platforms such as Space-syntax, GIS, SpaceMate and others unmentioned, are (by their very digital nature) compatible with Proximity sketching. Proximity is not a model or a data-base or a file-format. It is a representation of reality in a scientifically valid reduction. Every professional field which deals with spatial planning is therefore at some level compatible with Proximity. Legislation, economics, social studies, mass-psychology, politics and many more.

1. Burry, Wolfram, Stolk, Salingaros, Franz, Hillier, Turner and Figueiredo de Medeiros to name but a few.

Even without these additions, this implementation of Proximity is a useful tool for the spatial planner. Limitations aside, it enables the user to generate large amounts of analytical data within very short spans of time. It is helpful to individuals since it allows them to spot certain problems before they become progressive, it is helpful when used in a team-environment since it allows for scientific comparison of different designs and it is definitely advantageous when used by an individual in team context since it provides the user with argumentation and factual data to promote his or her design.

The aforementioned limitations at this point include:

- Over simplified legends[1]
- Particle inactivity[2]
- Lack of actors[3]
- Lack of independent networks[4]

This is not a complete list, only the most pressing items have been included. Still, they differ in severity. Some only require additional code to be written, others demands changes at the very kernel of the current implementation. It is my opinion that further research and elaboration should not be based on the available code. A new platform must be written which is not limited by preset particle characteristics. This new platform should optimally utilize the possibilities as provided by the object-oriented paradigm (OOP):

- Encapsulation[5]
- Inheritance[6]
- Polymorphism[7]

When these qualities are respected and applied consistently from the start, the platform as a whole will be flexible concerning the end-user, flexible concerning updates and flexible concerning third party additions. Of course OOP is used to a certain extent in the current implementation, but many core-classes are fixed and they do not support additions.

The ideal run-time environment for such a project is a non-platform, non-language bound syntax such as dotNET. DotNET allows third party developers to write additions in the programming language of their choice, it is in complete compliance with OOP and it supports web-based execution which means the program need not be installed but can be run via a web-page. Additionally dotNET is based on the GDI+ system which provides many advanced drawing tools that are vital to the graphical front-end and output of Proximity sketching.

Creating such a platform is not a trivial task and it will require a great deal more than the efforts of a good-willing amateur.

---

1. Currently, the written implementation only supports a single type of particle; a dwelling. Ideally, any type of particle should be possible. This requires a very OOP approach of the Proximity paradigm which was deemed too intricate for this study. As it stands, all properties are optimized for the distribution of dwellings. To fix this limitation a rewrite of the source code kernel will be required.
2. It would be useful if particles could be preprogrammed with a certain behaviour. This would augment Proximity from an analysis-tool to a design-tool. To fix this limitation new routines and classes should be written, but it does not require changes to the kernel.
3. Controlling entities do not exist. There is in effect little difference between actors and 'thinking' particles (see [B.2.1 Agents and Particles] and [B.2.3 Actors]). Effectively, actors tend to be smaller in number and can project more advanced behaviour.
4. Roads may be just a spread of asphalt grit, but it serves no purpose to treat them this way in a spatial planning sort of perspective. Networks are in fact an extremely important component of spatial planning and they require a non-particle approach. Proximity is not inherently incompatible with non-particle systems, it just requires a lot of additional coding.
5. MSDN Library: "Encapsulation means that a group of related properties, methods, and other members are treated as a single unit or object. Objects can control how properties are changed and methods are executed." Encapsulation lessens the responsibility of the user without reducing his/her options. This is vital when users are given access to the inner workings of classes.
6. MSDN Library: "Inheritance describes the ability to create new classes based on an existing class. The new class inherits all the properties and methods and events of the base class, and can be customized with additional properties and methods." This is vital for the structure of the platform. Very few restrictions should be placed upon particle types for example, but an abstract base class for particles must exist.
7. MSDN Library: "Polymorphism means that you can have multiple classes that can be used interchangeably, even though each class implements the same properties or methods in different ways." This is vital for the flexibility of the platform. New classes must be allowed to behave as they please, without breaking laws imposed on them by base classes.

Bibliography

This is not a research project. Although many sources have contributed to the process, it is very hard to estimate the extent of individual chunks. The following list is woefully incomplete (especially the digital section) for the plain and simple reason that I do not remember the respective origins of all the things that I know. The following list contains sources without which the project would not have existed, sources that have had a significant impact on the way the project progressed and sources which are completely superfluous.

- COLOMBO, F. (1994) Stedebouwkundige verkenning van verstedelijkingsmogelijkheden tussen Delft en Zoetermeer na 2005
- JONG, T.M. DE (1992) Kleine methodologie voor ontwerpend onderzoek
- JONG, T.M. DE; VOORDT, D.J.M. VAN DER; et. al. (2002) Ways to study and research; Urban, architectural and technical design
- JONG, T.M. DE; MOENS, R.; et. al. (2002) Grondgebied, Ecologie en Techniek; Schaalgelede stadsecologie voor ontwerpers
- STELLINGWERFF, M.C. (2005) Virtual Context; Investigating the characteristics and opportunities of digital visualisation media for situated approaches to architectural design in an urban environment
- STUURGROEP ZUIDPLAS (2005) Ontwerp Intergemeentelijk Structuurplan Zuidplas
- MEYER, H; BURG, L. VAN DEN; et. al. (2005) In Dienst van de Stad; 25 jaar werk van de stedenbouwkundige diensten van Amsterdam, Den Haag, Rotterdam
- MINSKY, M (1974) A Framework for Representing Knowledge
- SALINGAROS, N.A. (1998) Theory of the Urban Web
- SALINGAROS, N.A. (1999) Urban space and its information field
- SALINGAROS, N.A.; WEST, B.J. (1999) A universal rule for the distribution of sizes
- SALINGAROS, N.A. (2000) Complexity and Urban Coherence
- SALINGAROS, N.A. (2000) The Structure of Pattern Languages
- SALINGAROS, N.A. (2001) Remarks on a city's composition
- SALINGAROS, N.A. (2003) Connecting the Fractal City
- SALINGAROS, N.A. (2004) Design methods, emergence and collective intelligence
- SALINGAROS, N.A. (2004) The Information Architecture of Cities
- BOIS, P.G. DE; BUURMANS, K.A.; et. al. (due 2006) RGBG; Strategic Model; a Scenario Analysis & Design Method
- KOSTOF, S. (1991) The City Shaped; Urban Patterns and Meanings Through History
- SCHUMACHER, P.; FONTANA-GIUSTI, G.; et. al. (2004) Zaha Hadid; Major and recent works
- BERGHAUSER PONT, M.; HAUPT, P. (2004) Spacemate; the spatial logic of urban density
- DAWKINS, R. (1982) The Extended Phenotype; the long reach of the gene
- ALEXANDER, C.; ISHIKAWA, S.; SILVERSTEIN, M.; et. al. (1977) A Pattern Language; Towns, buildings, construction
- OKABE, A.; et. al. (2000) Spatial tessellations; concepts and applications of voronoi diagrams
- EDELSBRUNNER, H. (2001) Geometry and Topology for mesh generation
- CHAZELLE, B. (2000) The discrepancy method; randomness and complexity
- SAMET, H. (1998) Multidimensional Point Data
- LEIGHTON R.; HUTCHINGS E. (1985) Surely you're joking, Mr. Feynman!
- VON NEUMANN, J. (1958) The Computer and the Brain
- MILLER, G. (1956) The Magical Number Seven, Plus or Minus Two

The following entries represent digital sources

- TARBELL, J. (2005) Complexification; Gallery of Computation
- TARBELL, J. (2005) Levitated; Anti-Gravity
- SHANNON, R. (2006) HTMLSource
- REYNOLDS, C.W. (1987) Flocks, Herds, and Schools: A Distributed Behavioral Model
- HEUER, R.J. (1999) Memory: How Do We Remember What We Know?
- MACKENZIE, B. (1997) Developing Imagery skills
- KEEBLE, D.; MCILHAGGA, W. (2003) Physiology of Vision and Perception
- RUTTER, D. (2002) Your computer is not alive
- CAMPBELL, K.; et. al. (2000) Psychological Schools of Thought; Gestalt Psychology
- UNDERWOOD, M. (2000?) Redundancy and entropy
- FIELDING, A. (2003) Cluster Analysis
- CESARE, A.; et. al. (2001?) A Tutorial on Clustering Algorithms; K-Means Clustering
- SOFTSURFER.COM (2004) Bounding Containers for Polygons, Polyhedra, and Point Sets
- HAUSNER, A. (1996) Convex hull; Graham's scan algorithm
- BATTY, M.; BESSUSI, E.; CHIN, N. (2003) Traffic, Urban Growth and Suburban Sprawl
- SEDGEWICK, R.; WAYNE, K. (2004) Creating Data Types
- WEISSTEIN, E.W. (1999-2006) Mathworld
- ROWLEY, T. (2000?) Minimal Enclosing Circle
- COETZEE, D.; et. al. (2004) Euclidian minimum spanning tree
- FISHER, R.; PERKINS, S.; et. al. (2003) Spatial Filters: Gaussian Smoothing
- KÖNIG, R. (2004?) Laboratorium für räumliche Prozesse
- SUNDAY, D. (2004) Polyline Simplification; The Douglas-Peucker Algorithm
- OHYAMA, T. (1999) Voronoi Diagram page
- BOURKE, P. (1993) Efficient Triangulation Algorithm Suitable for Terrain Modelling; An Algorithm for Interpolating Irregularly-Spaced Data

Image reference

All images made by D. Rutten except:

Page	Source
1	Mondriaan, P. (1913) Compositie #2, lijn en kleur*
9	National Braille Association (1971) Stone axe
11	Pieter Bruegel de Oudere (1563) De toren van Babel
18	Schiaparelli, G. (1888) Mars canals*
19	Hernandez, G. (2001) Romanesco broccoli
20	Weber, B. (1999) Grid City*
20	USGS (????) Birds nest
24	Le Corbusier (1952) Ronchamp sketch
24	Anonymous (2004) Ronchamp sketch*
25	Hadid, Z. (1990) Rheinauhaven design
28	Colombo, F. (1994) Regional concepts
50	Smart Growth America (????) Suburban sprawl
116	Buurmans, K. (2006) RedGreenBlueGrey*
117	Buurmans, K. (2006) Red*

*Original has been pre- or post-processed by D. Rutten.

Proximity; how to represent reality in sketches is a 'study by design' project which explores the (im)possibilities of a new drawing theorem in urban and regional design. By stepping away from traditional media and methods, new doors are opened up which allow for a scientific look on even the most rudimentary sketch designs.

This paper discusses the theoretical paradigms involved, a first step approximation of the digital implementation and the results which flowed forth.

David Rutten was born in 1980 in Nijmegen and attended highschool in Amersfoort. At the age of fourteen he was awarded shared first price for an urban redevelopment project for the city of Amersfoort in a contest among highschool students. Based on this feeble success he entered the Delft University of Technology in 1999, where he started a study of architecture and urbanism. In august 2003 he was invited to give a lecture on scripting at a CAD user meeting in London organised by Will Alsop architects and subsequently to teach a four-day workshop in Vienna (part of the Greg Lynn curriculum) for students of the University of Applied Arts. In 2004 he spend four months as an intern with a Toulouse-based tensile architecture firm where he helped to develop in-house software for surface relaxation and form-finding.
As a result of the publication of the web domain associated with this graduation project (http://www.UrbanProximity.net), he was invited to discuss the project at a workshop for computational design organized by EZCT architects at the University of Malaquais, Paris in early 2006.

TU Delft