Uniwersytet Jagielloński w Krakowie
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Jakub Wida

Nr albumu: 1113470

Losowe upakowania układów złożonych z dysków z wykorzystaniem kart graficznych

Praca magisterska
na kierunku Informatyka Stosowana

Praca wykonana pod kierunkiem
dr. hab. Michała Cieśli
z Zakładu Fizyki Statystycznej

Kraków 2019

Oświadczenie autora pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

....

Kraków, dnia Podpis autora pracy

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

....

Kraków, dnia Podpis kierującego pracą

Chapter 1

Introduction

TODO make this at the end, when all else is done

Contents

Chapter 2

Problem Overwiev

2.1 Random Sequential Adsorbtion

Random Sequential Adsorbtion is a stochastic process, that can be described as sequential insertion of given shapes onto an empty, limited euclidean space. The shapes are inserted with random position, and in such a way, that if the shape is colliding with an already inserted shape, it is then rejected in the process. The shapes and the space may be defined as having one or more dimensions, with any kind of inserted shape type. The shapes may be generated with random coordinates within given dimensions, as well as random rotation - although this may be optional.[1][2]

2.1.1 Shape Types

The inserted shape that were previously studied include rectangles, squares and other polygons, cubes and hypercubes. Also, the most basic shape usually investigated is a sphere - up to eight dimensions. Other studies involved spheroids, hyperdisks and, what is going to be further investigated in this thesis, disk polymers, or simply two dimensional groups of disks. In case of non spherical shapes, the shape angle can be randomly generated.[3]

2.1.2 Applications

The Random Sequential Adsorbtion can be applied to multiple problems. The process as well as the generated shapes can be used to model a variety of phenomena. These include the ion implantation in semiconductors, structure of the cement paste, particles in cell membranes, protein adsorption and settlement of animal territories.[4] In general, it can be used to model

[1] G Zhang. “Precise algorithm to generate random sequential adsorption of hard polygons at saturation”. In: *Physical Review E* 97 (Mar. 2018). DOI: `10.1103/PhysRevE.97.043311`, p. 1.

[2] Jens Feder. “Random sequential adsorption”. In: *Journal of Theoretical Biology* 87.2 (1980), pp. 237 –254. ISSN: 0022-5193. DOI: `https://doi.org/10.1016/0022-5193(80)90358-6`. URL: `http://www.sciencedirect.com/science/article/pii/0022519380903586`, p. 1.

[3] Zhang, “Precise algorithm to generate random sequential adsorption of hard polygons at saturation”, op. cit., p. 1.

[4] Ibid., p. 1.

tightly - but still randomly - packed particles.

A significant example is the Poisson Disk Sampling, which is a process of selection of points in subdomain, in such a way that within a given distance from the selected point, no other are taken. This distribution is used in computer graphics for rendering, texture generation, and more. In ray tracing, it is used to create soft shadow, motion blur and the depth of field. In physics, it can be used for mesh generation, interpolation and process modeling. In some cases, the generated meshes have improved quality and are generated more robustly.[5]

2.1.3 Challenges

While the principle behind RSA algorithms is simple, the execution of it's naive implementations will commonly lead to problems. As the figures are randomly generated and inserted to the area, the probability of rejection of said shape will grow with the saturation of the system. In simpler words, the more figures are already inserted, the more likely it is that a new one will overlap a pre-existing one, which leads to long execution times and uncertain end conditions. The goal of many approaches is to propose an algorithm that will be able to quickly generate shapes within given space, and have clear end conditions signifying a fully saturated state.

2.2 Thesis Goals

This thesis focuses on one particular subset of RSA, namely, the generation of polydisks in two dimensional space. It's goal was to create an algorithm that would take advantage of great degree of parallelization available through using the GPU. A pre-existing algorithm developed by dr. hab. Michał Cieśla enables quick sequential or thread-level parallel generation of polydisks in a 2D area. Such algorithm can be used to simulate the packings of particles on a surface; Which itself is usefull in physics and material research.[6]

[5] Mohamed S. Ebeida et al. “A Simple Algorithm for Maximal Poisson-Disk Sampling in High Dimensions”. In: *Computer Graphics Forum* 31.2pt4 (), pp. 785–794. DOI: 10.1111/j.1467-8659.2012.03059.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2012.03059.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2012.03059.x, p. 1.

[6] **ciesla**.

Chapter 3

Proposed Algorithm

3.1 Voxel-Based Algorithm

The basic algorithm used in multiple RSA applications involves the use of voxels. The voxels are defined as subspaces of the saturated area, which cover it entirely. After one or more shapes are inserted, the voxels may be removed if within them no more figures can be inserted. Developing an algorithm for "voxel rejection" is the main concern of those undertakings. Only within the non-removed voxels the figures can be generated. After inserting more figures, if the voxels are not rejected or the figures are rejected too commonly - the voxels are split into smaller voxels to be able to further reject them. If all voxels are removed, the system is saturated. An example of such algorithm, using the two dimensional circles as shapes is demonstrated in the figures 3.1, 3.2.

3.2 Sequential Algorithm

The algorithm developed by dr. hab. Cieśla is an expansion on the previusly described basic model. However, some changes have been made to accomodate the algorithm to a different shape, more efficient execution and a modification in the behavior of space.

3.2.1 Periodical Edge Conditions

The algorithm was supposed to operate using a space that, unlike previously shown model, would *loop* at the edges. This means, that if a shape is placed nearby the edge of given space, the shapes placed at the opposite edge of the space collide with it. They behave as if the opposite edge of the space would be a continuation of this edge. The periodical edge conditions serve to improve packing efficiency, and remove edge-related artifacts. In essence, the periodical edge conditions 'simulate' an infinite area, and thus may be helpful to model surfaces of far greater area than the provided space.[1]

[1] Michał Cieśla and Robert M Ziff. “Boundary conditions in random sequential adsorption”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2018.4 (2018), p. 043302. DOI: `10.1088/1742-5468/aab685`. URL: `https://doi.org/10.1088%2F1742-5468%2Faab685`, p. 1.

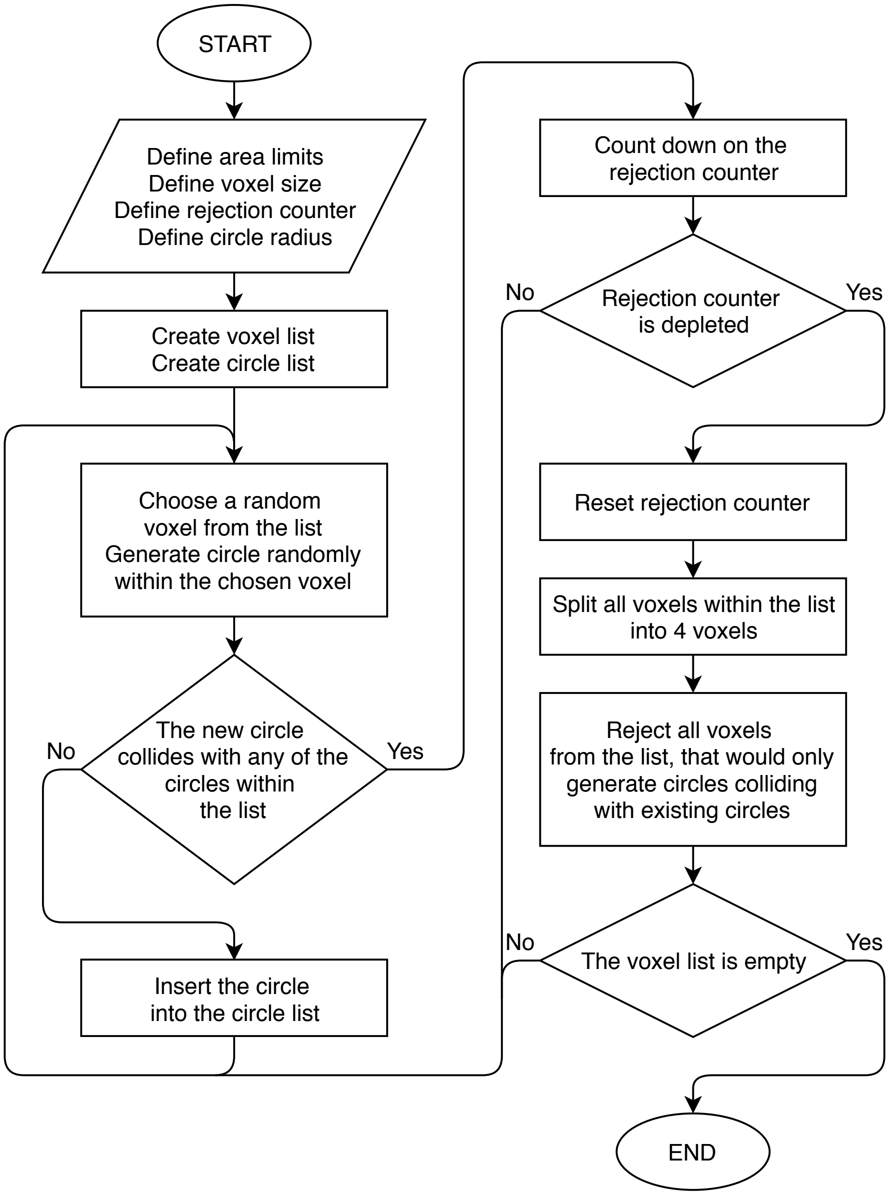
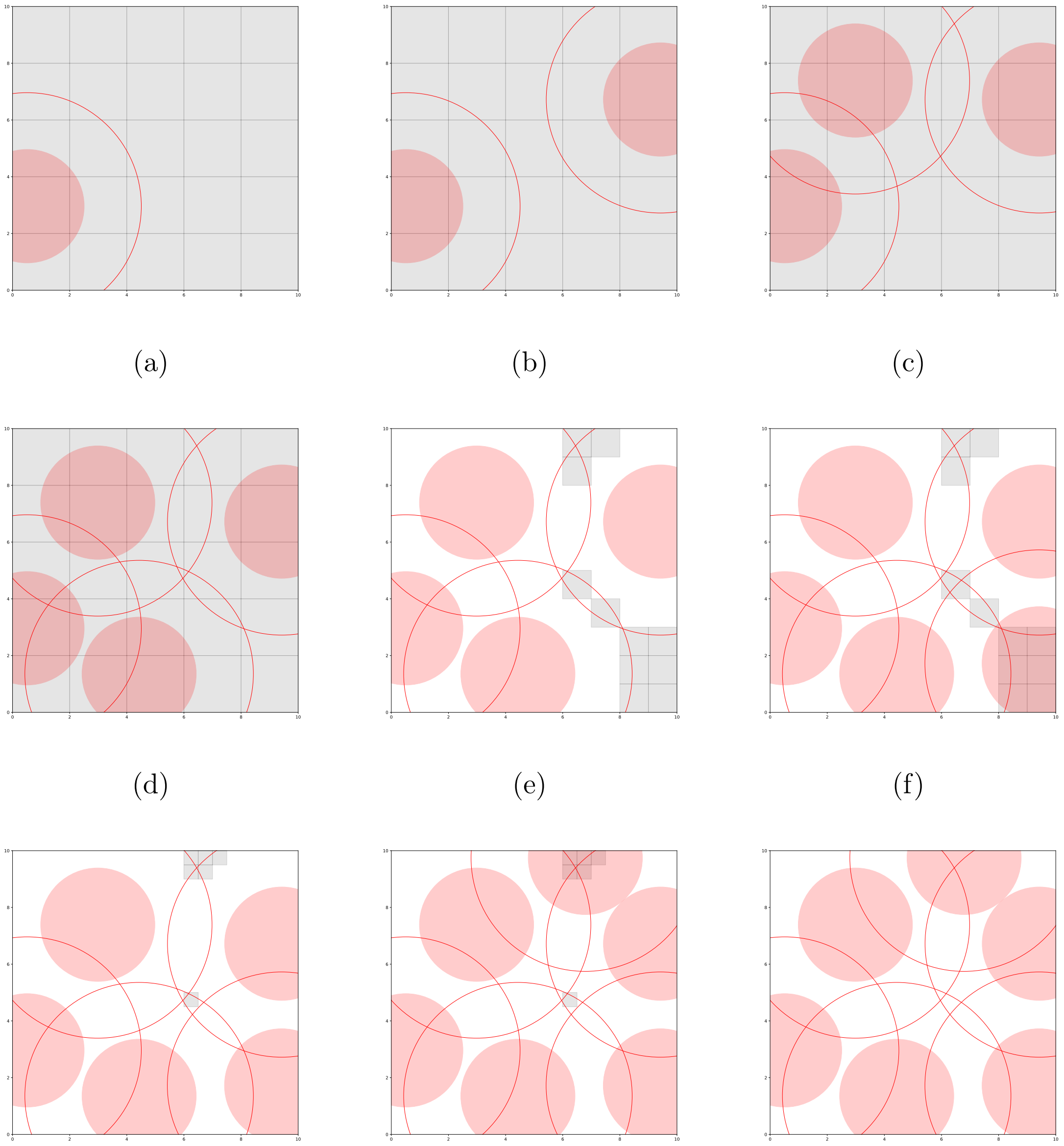

START
Define area limits
Define voxel size
Define rejection counter
Define circle radius
Create voxel list
Create circle list
Choose a random voxel from the list
Generate circle randomly within the chosen voxel
The new circle collides with any of the circles within the list
No
Yes
Insert the circle into the circle list
Count down on the rejection counter
Rejection counter is depleted
No
Yes
Reset rejection counter
Split all voxels within the list into 4 voxels
Reject all voxels from the list, that would only generate circles colliding with existing circles
The voxel list is empty
No
Yes
END

Figure 3.1: The flowchart of simple 2D RSA algorithm, using circles.
This algorithm can be expanded to remove voxels after every successful figure insertion, or to split voxels at a different condition.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.2: This figures illustrate a simple run of the described simple 2D RSA algorithm. World size is 10.0 by 10.0, the inserted circles have the radius of 2.0. The initial voxel side has length of 2.0. After 10 failed attempts at inserting the circle, the voxels are subdivied and rejected.

The figures *a,b,c,d* show the insertion of the first four circles. The figure *d* illustrates the effect of the voxel subdivision and rejection. Following figures illustrate next iterations of circle insertions, voxel subdivisions and removals, and finally, at figure *i*, completely saturated system with no remaining voxels.

The voxels are rejected if they lie entirely within the double radius of any circle - in such case any circle generated within the voxel would collide with the evaluated circle. This is determined by examining if distance to all vertices of a voxel is less than the circle's doubled radius. This doubled radius is marked here as a solid red outline.

3.2.2 Adjacency Matrix

During the execution of the algorithm, there are two places where a function must be performed at two objects, many different times. Firs, when the shape is inserted, it is determined if it collides with any other existing shape. The other time, it is when the voxel is rejected - it must be checked against all shapes, until one that would cause it's rejection is found. It is possible to limit the executions of these function to a far lesser number; By dividing the world into a number of cells, with sides as long as the radius of the shape's minimum binding circle, it is only nescessary to check the collisions of the shapes that belong to the same, or neighboring cells (similarly with voxels). This limits the number of collision checks that need to be performed to a theoretical maximum, based on how many shapes can fit into a three-by-three cell neighborhood.

This solution can also be helpful with the periodical edge conditions problem. Having a quick access to the cell given it's coordinates enables the program to easily reference shapes from the opposite edge, by tapping into the target cell and translating the shapes stored by it.

3.2.3 Polydisk Collision

Since the algorithm uses a shape consisting of a list of disks with given radiuses and a relative position to an arbitrary center, it is relatively easy to check if two such shapes collide. It is enough to loop over both lists of shapes' circles absolute positions, checking if any two collide. The much more complex problem is the voxel rejection. The first problem is that since the shapes can have a different angle, this needs to be implemented in the voxels structure - the solution was to add a third dimension, representing the available angles in which the shapes may appear.

UNFINSHED - ASK HOW THIS WORKS, HOW TO DESCRIBE IT

3.3 Parallel Application

Having the available algorithm for sequential execution, the main challenge consisted of modifying it to enable a highly parallel execution using the GPU. The modified algorithm executes some parts of the problem sequentially, as it was not possible to parallelize it entirely.

3.3.1 Block Graph

3.4 Implementation

3.4.1 PyCuda

3.4.2 Visualisation

3.4.3 Summary

TODO unfinished, do in near future

Chapter 4

Results Examination

4.1 Performance Evaluation

4.1.1 Parameter Influence over Performance

4.1.2 Shape Influence over Performance

TODO unfinished, do later

Bibliography

Cieśla, Michał and Robert M Ziff. "Boundary conditions in random sequential adsorption". In: *Journal of Statistical Mechanics: Theory and Experiment* 2018.4 (2018), p. 043302. DOI: 10.1088/1742-5468/aab685. URL: https://doi.org/10.1088%2F1742-5468%2Faab685.

Ebeida, Mohamed S. et al. "A Simple Algorithm for Maximal Poisson-Disk Sampling in High Dimensions". In: *Computer Graphics Forum* 31.2pt4 (), pp. 785–794. DOI: 10.1111/j.1467-8659.2012.03059.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2012.03059.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2012.03059.x.

Feder, Jens. "Random sequential adsorption". In: *Journal of Theoretical Biology* 87.2 (1980), pp. 237 –254. ISSN: 0022-5193. DOI: https://doi.org/10.1016/0022-5193(80)90358-6. URL: http://www.sciencedirect.com/science/article/pii/0022519380903586.

Zhang, G. "Precise algorithm to generate random sequential adsorption of hard polygons at saturation". In: *Physical Review E* 97 (Mar. 2018). DOI: 10.1103/PhysRevE.97.043311.