AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Inżynierii Metali i Informatyki Przemysłowej

# MULTISCALE MODELING

# „Report I”

Name and surname of the student: **Michał Pożarlik**

Signature of the student:

Signature of the reviewer:

**Kraków 2018**

# Table of Contents

# 1. Technology

Project was creating using Windows Forms technology. Basic language was C#, application was writing using IDE[1]: Microsoft Visual Studio 2015. UI[2] have been designed using standard editor integrated with VS.

This technology is still popular for Windows desktop application. The most important think about this decision was chose simple and fast technology. Microsoft deliver a lot of libraries integrated with Windows Forms, so it was easy to create user friendly application for simulation of grain growth.

This project is open-source[3], so for repository is using GitHub[i] website with public settings for repository. Visual Studio include tools for using GitGub, so all commits have been created using IDE.

The most important for decision about language was preference for developer. Knowledge is important for choose technology, especially, when project have quick deadline.

1 Integrated Development Environment – software application for software developers.
2 User Interface - interface with which a person may interact
3 More details: https://opensource.org/ (active for day: 14.11.2018)

# 2. User Interface

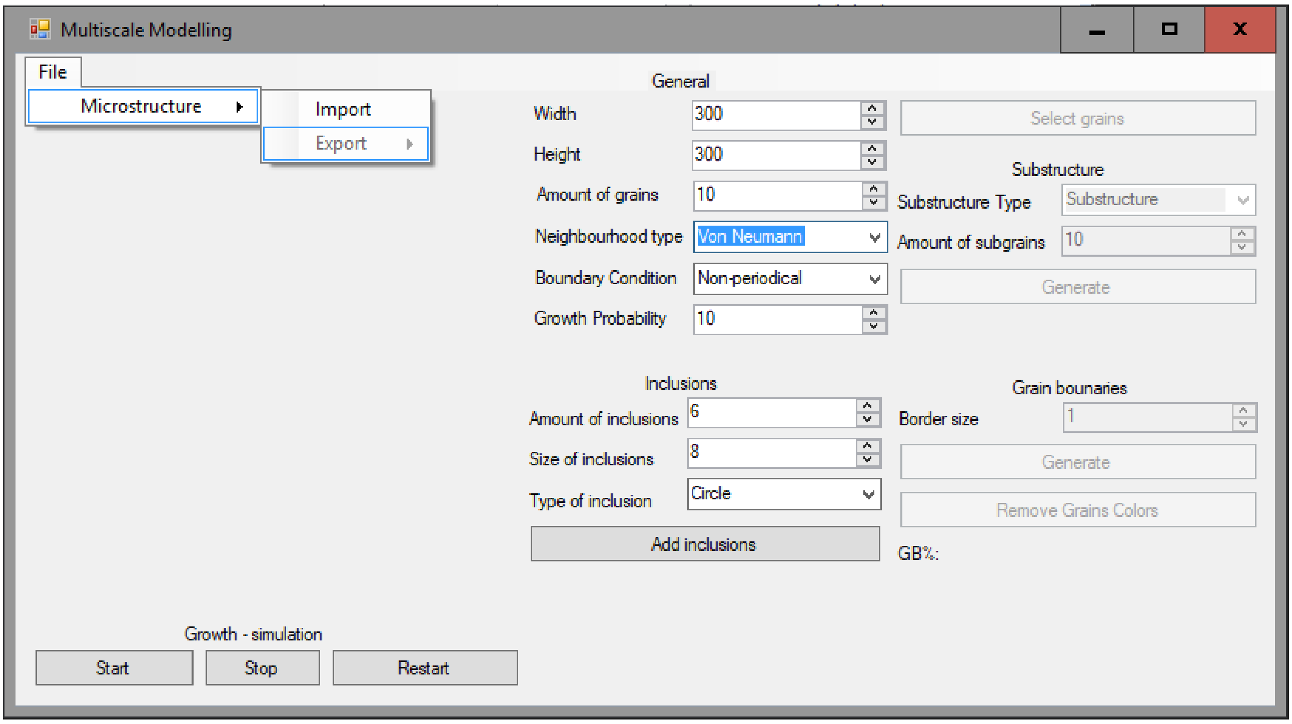



*Fig. 1: Application UI*

Advanced application for growth often needed much complicated interface. It is complex process, so user need to have access to change a lot of process parameters. Application should be easy to using and interface should be user friendly. Just to create interface more clear, user have active only needed options to current step of simulation.

Application [Fig. 1] contain subsections for control process of grains growth. In left side you can find output bitmap and basic simulation buttons. Right side contain fields for simulation control.

## 1. Simulation output

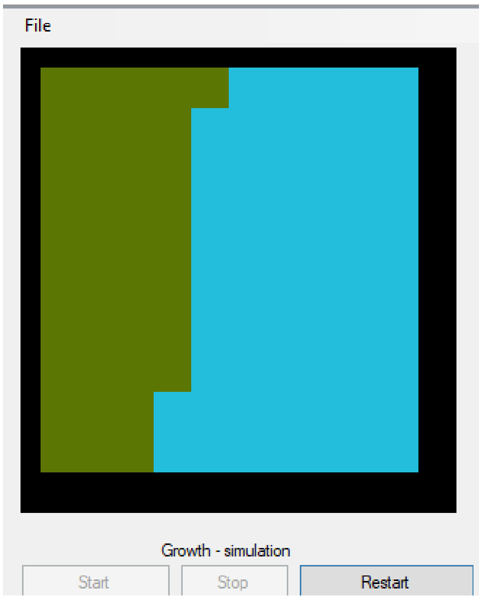



*Fig. 2: Output window - size 10x10*

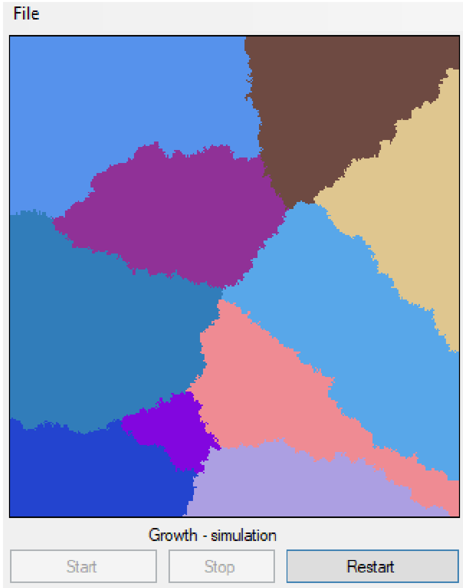



*Fig. 3: Output window - size 300x300*

Main of them is simulation output on left side of application. Nucleons are represented by colorful pixels, user can change simulation size, but always will see bitmap in one size. It is possible, because application rescale generated map to bitmap in application. You can see example outputs for two different size of maps [Fig. 2-3].

## 2. Simulation buttons

Below of output bitmap [Fig. 3], user can find simulation buttons, for control process:

- Start – button for start simulation, you need to click it if you want to generate grains, or substructure.
- Stop – when process is simulating, user can click stop button to stop simulation,
- Restart – after simulation user have possibility to restart simulation.

## 3. General growth settings

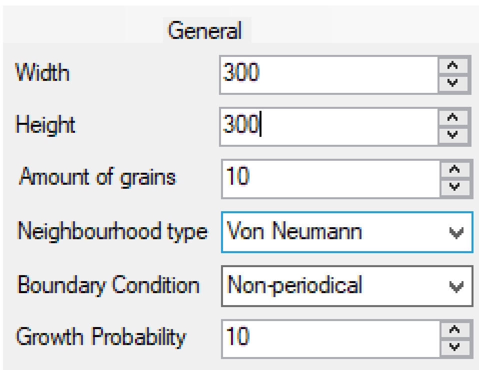



*Fig. 4: Growth settings*

Next section [Fig. 4] is about change mainly settings for growth simulation simulation.

More about each fields:

- Width – width of simulation map (output bitmap),
- Height – width of simulation map (output bitmap),
- Amount of grains – number of grains in map,
- Neighborhood type – type of grains neighborhood:
  - Von Neumann
  - Moore
  - Moore 2 - extension of Moore neighborhood
- Boundary Condition - currently unused – only implemented one option: Non-periodical BC
- Growth Probability – parameter for Moore extension neighborhood type

## 4. Inclusions settings

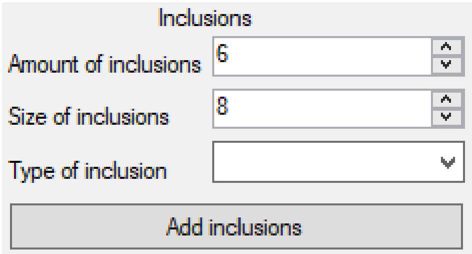



*Fig. 5: Inclusion settings*

Add new inclusions or change parameters for them you can find in next section [Fig. 5].

Section details:

- Amount of inclusions – number of inclusions to add
- Size of inclusions – size of inclusions to add (diameter or side of the square)
- Type of inclusion:
  - square
  - circle

## 5. Substructure section

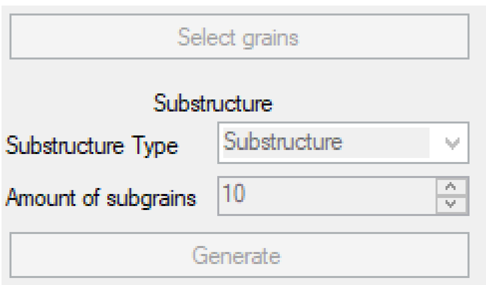



*Fig. 6: Substructure section*

It is possible to generate substructure or dual-phase structure in application. User need to use section Fig. 6 for regenerate structure.

Section details:

- Select Grains button – user need to use this button if want to select grains to processes, like: generate substructure, or generate boundaries for selected grains
- Substructure Type:
    - Substructure – clear unselected grains and generate substructure for empty fields
    - Dual phase – generate one grains from selected grains and generate new structure for other grains
- Amount of sub grains – number of grains to generate
- Generate button – initialize process

## 6. Grain Boundaries

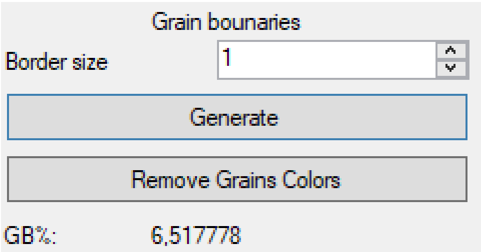



*Fig. 7: Boundaries section*

The last one section [Fig. 7] is about generate boundaries for grains.

Section details:

- Border size – size of border for each grains
- Generate button – add borders for: selected grains (if selected), or for each grains,
- Remove Grains Colors – remove colors for grains
- GB% - label with information about relation between boundaries and nucleons.

## 7. File control

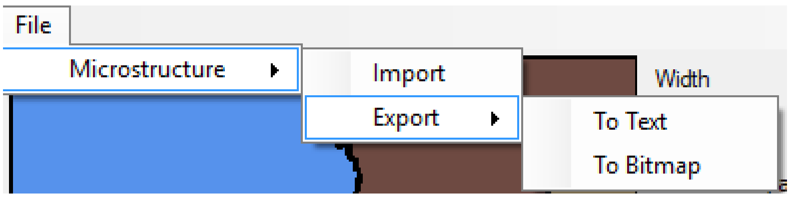



*Fig. 8: File control section*

Application have tool strip menu [Fig. 8] for file options.

- import – user can load bmp or xt files
- export – user can generate file with extensions like: txt or bmp

# 3. Application outputs

The application allows generate a lot of complex grains structure. This chapter cotains basic capabilities with descriptions, how to use them.

## 3.1. Basic growth

For change settings like: amount of grains, neighborhood or size you need to edit fields from basic section [Fig. 4]. The results for different configurations are presenting on output bitmaps [Fig. 9-16].

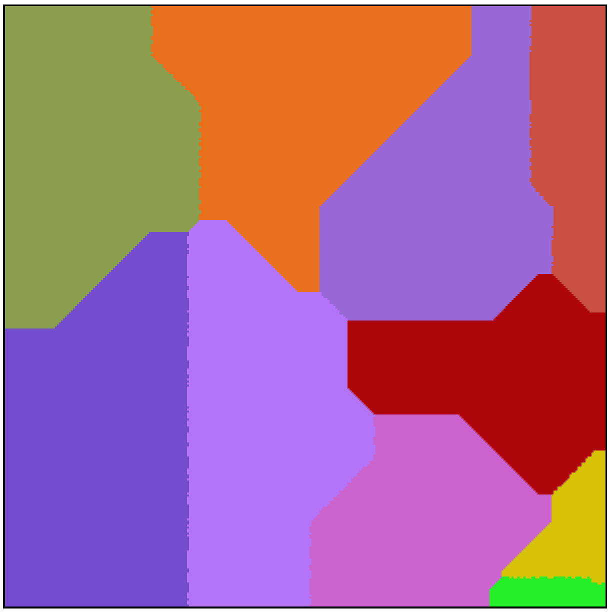
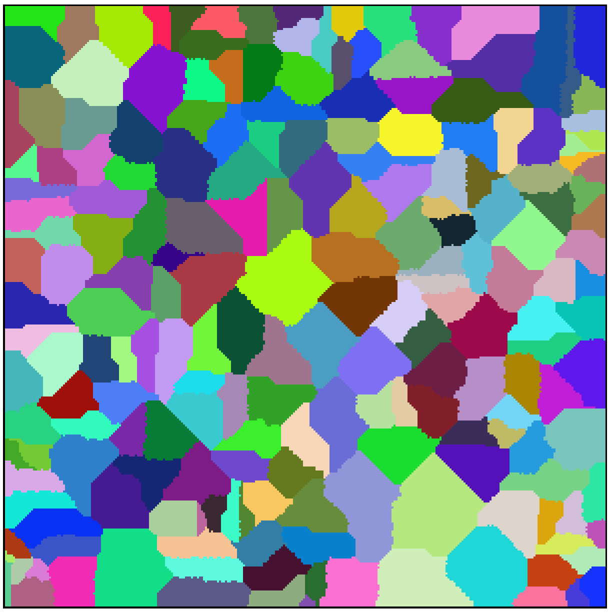
| Number of grains: 10<br>Neighborhood: Von Neumann | Number of grains: 200<br>Neighborhood: Von Neumann |
|---|---|
| <br>*Fig. 9: Application output - Von Neuman - small* | <br>*Fig. 10: Application output - Von Neuman - huge* |
| Number of grains: 10<br>Neighborhood: Moore | Number of grains: 200<br>Neighborhood: Moore |

*Fig. 11: Application output - Moore - small*

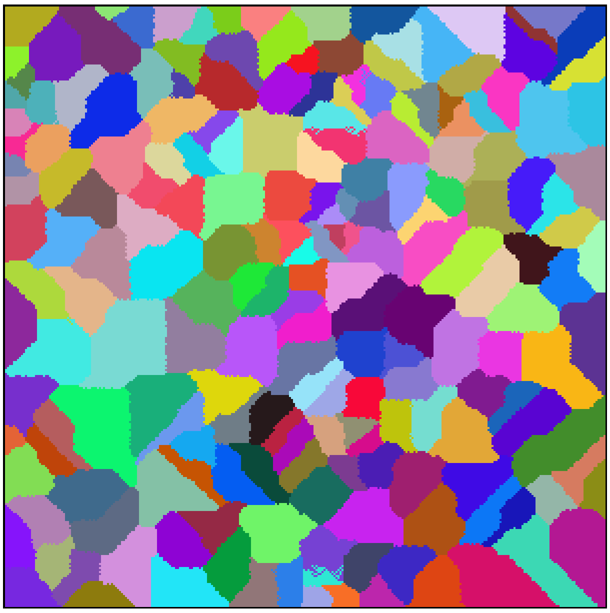


*Fig. 12: Application output - Moore - huge*

| | |
|---|---|
| Number of grains: 10<br>Neighborhood: Moore 2<br>Probability: 10% | Number of grains: 200<br>Neighborhood: Moore 2<br>Probability: 10% |

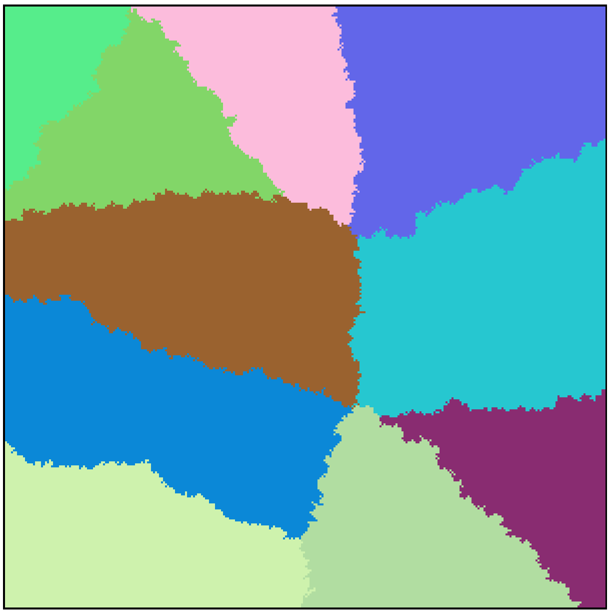


*Fig. 13: Application output - Extended Moore – small (probability 10%)*

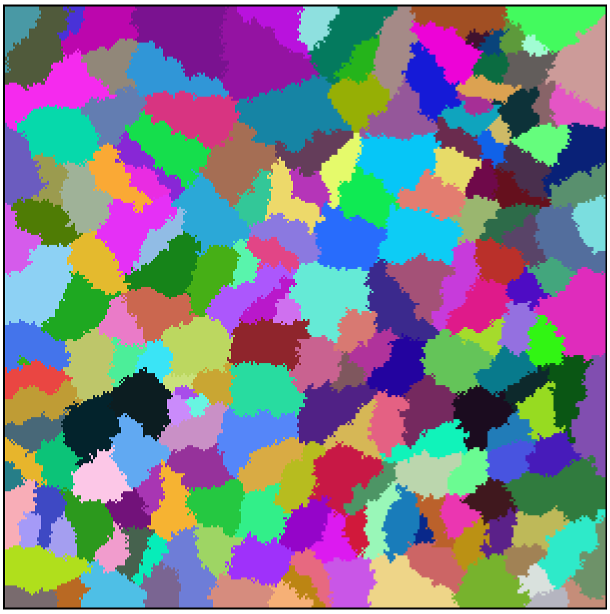


*Fig. 14: Application output - Extended Moore – huge (probability 10%)*

| | |
|---|---|
| Number of grains: 10<br>Neighborhood: Moore 2<br>Probability: 90% | Number of grains: 200<br>Neighborhood: Moore 2<br>Probability: 90% |

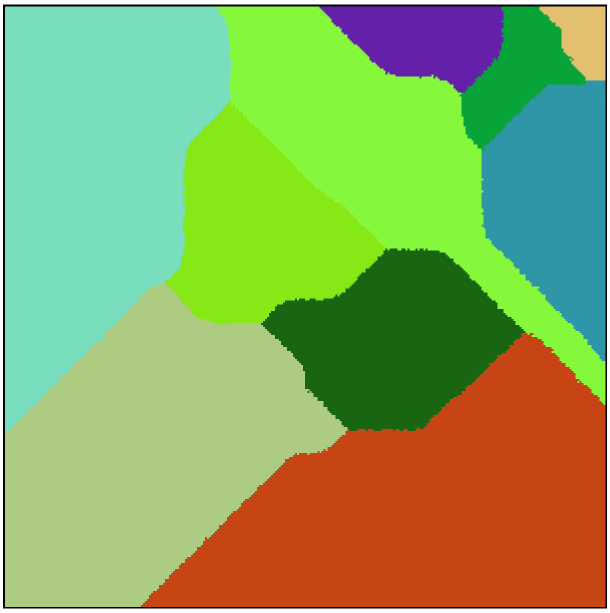


*Fig. 15: Application output - Extended Moore - small (probability 90%)*

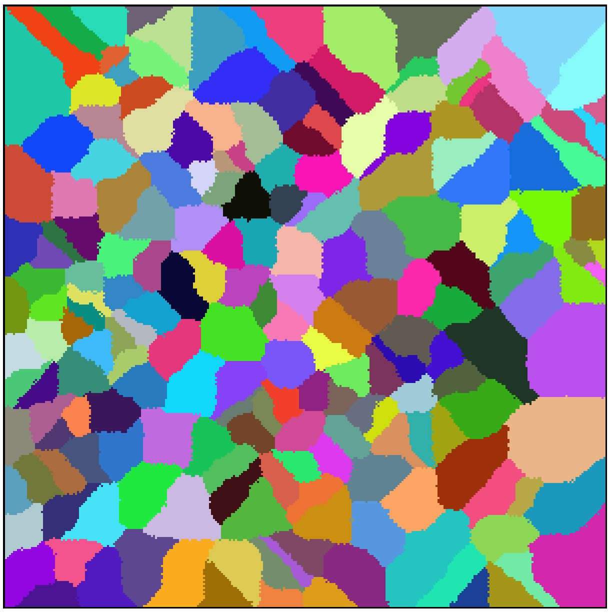


*Fig. 16: Application output - Extended Moore - huge (probability 90%)*

## 3.2. Inclusions

Application is implementing two types of inclusions: circle and square. User have possibility to change size of inclusions and change amount of them to add [Fig. 5]. We have two options for add inclusions – before and after simulation. Inclusions added after simulation should be generated only on boundaries of grains. Results of test simulations you can see on bitmaps Fig. 17-20.

| | |
|---|---|
| Number of grains: 20<br>Neighborhood: Von Neumann<br>Type of inclusion: Circle<br>Inclusions generate before simulation | Number of grains: 20<br>Neighborhood: Von Neumann<br>Type of inclusion: Circle<br>Inclusions generate after simulation |

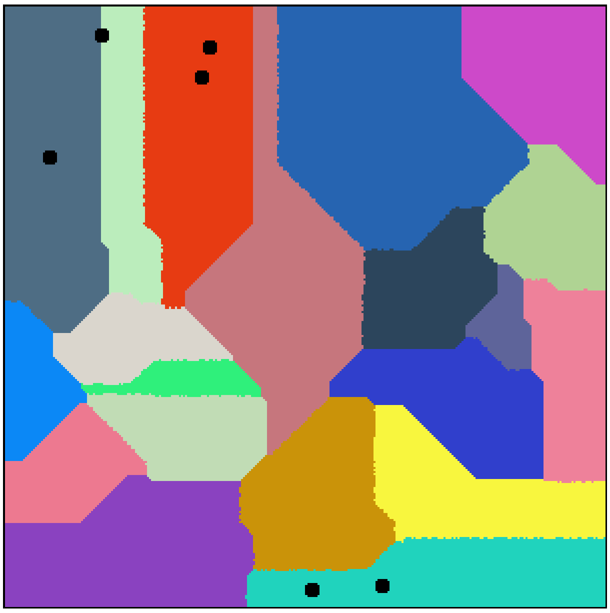


*Fig. 17: Application output - circle - before*

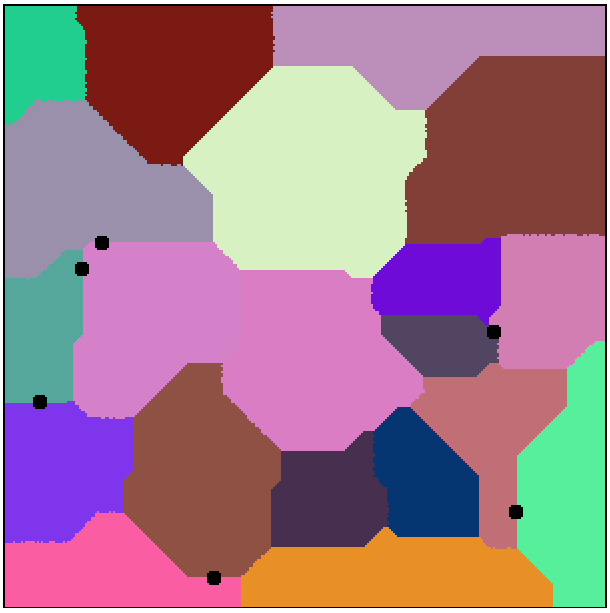


*Fig. 18: Application output - circle - after*

| | |
|---|---|
| Number of grains: 20<br>Neighborhood: Von Neumann<br>Type of inclusion: Square<br>Inclusions generate before simulation | Number of grains: 20<br>Neighborhood: Von Neumann<br>Type of inclusion: Square<br>Inclusions generate after simulation |

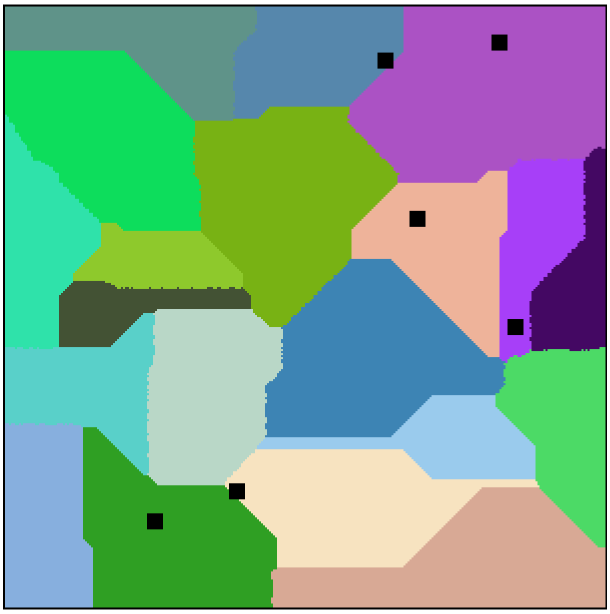


*Fig. 19: Application output - square - before*

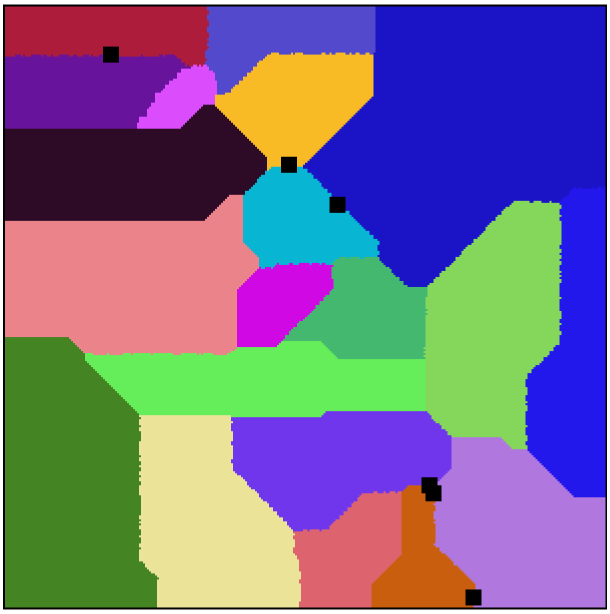
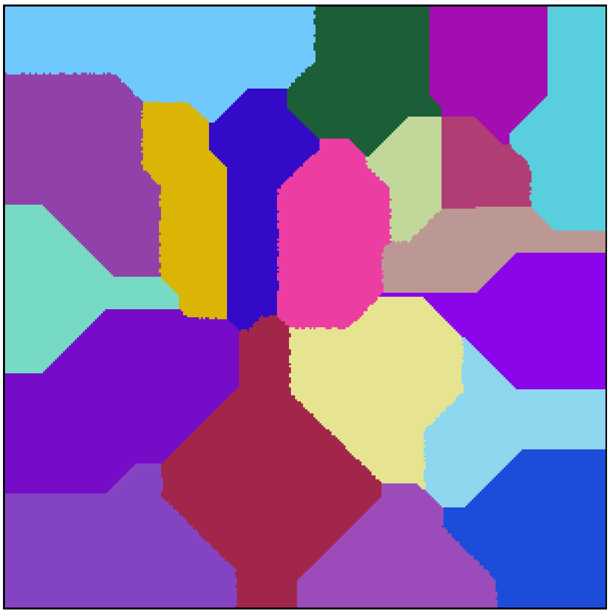


*Fig. 20: Application output - square - after*

## 3.3. Different micro structure types

After generate grains structure, will be possible to generate substructure or dual phase. User need to select remaining grains, write amount of grains to generate and select type of new structure [Fig. 6]. Next step is click generate button and start simulation. Results of test simulations you can see on bitmaps Fig. 21-23.

<table>
<tr><td colspan="2">Number of grains: 20<br>Neighborhood: Von Neumann</td></tr>
<tr><td colspan="2"><br>Fig. 21: Application output - micro structure basic bitmap</td></tr>
<tr><td>Number of grains: 20<br>Neighborhood: Von Neumann<br>Subsctucture type: Substructure<br>Amount of subgrains: 50<br>Amount of selected grains: 5</td><td>Number of grains: 20<br>Neighborhood: Von Neumann<br>Subsctucture type: Dual Phase<br>Amount of subgrains: 10<br>Amount of selected grains: 8</td></tr>
</table>

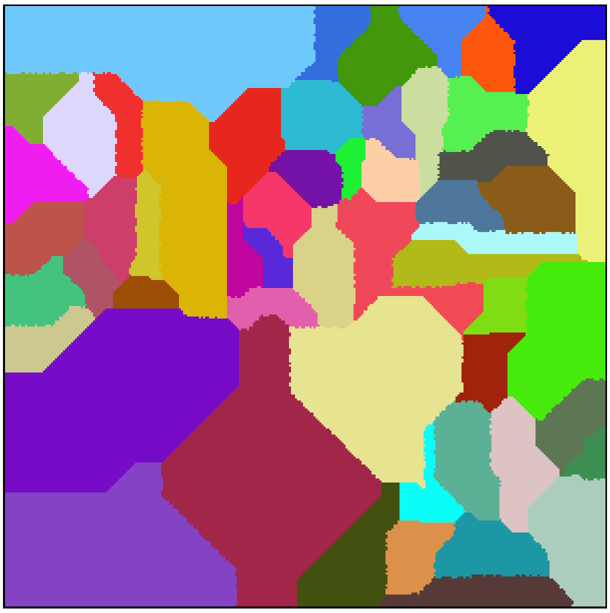


*Fig. 22: Application output - substructure*

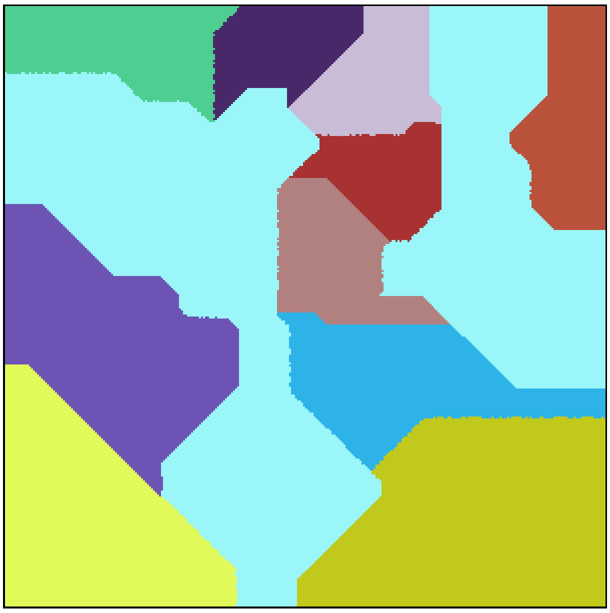


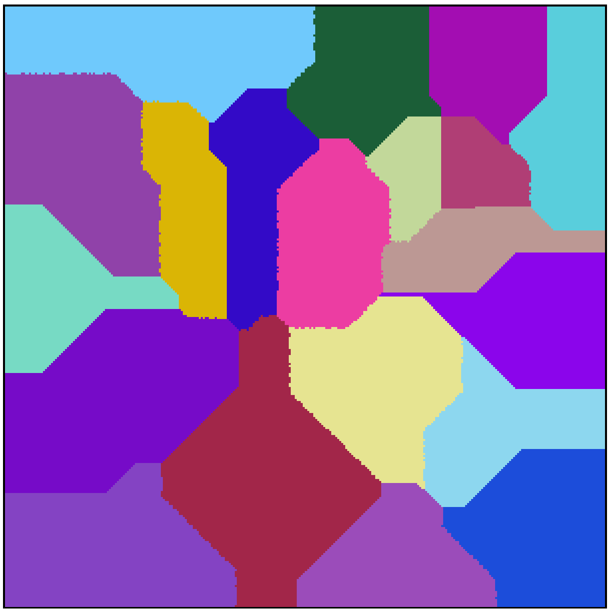
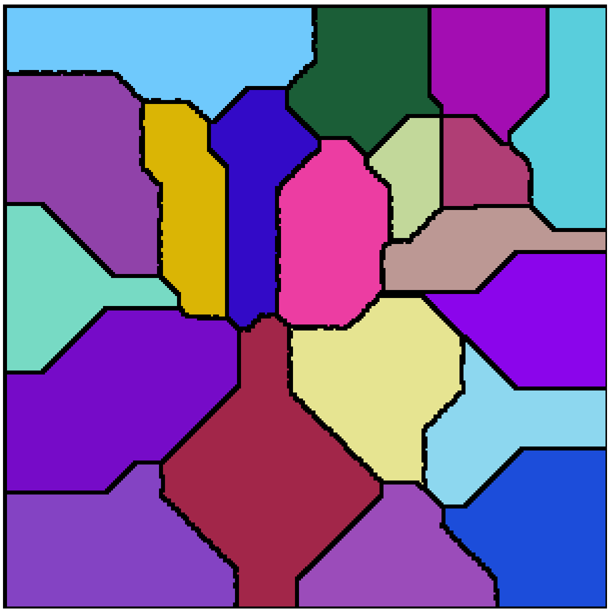
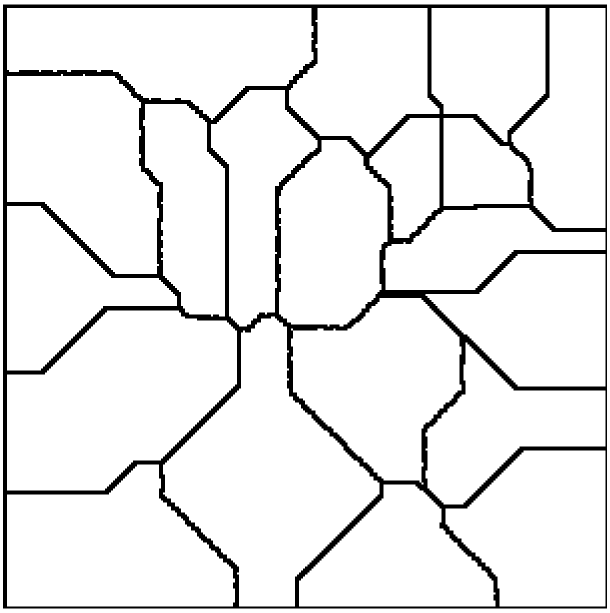
*Fig. 23: Application output - Dual-Phase*

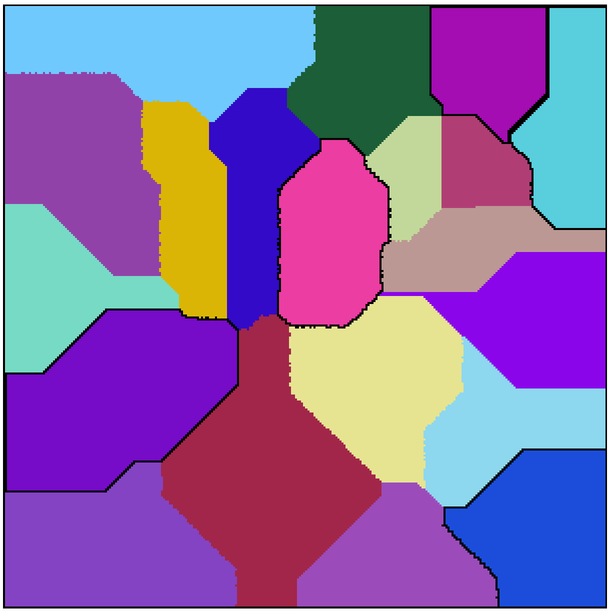
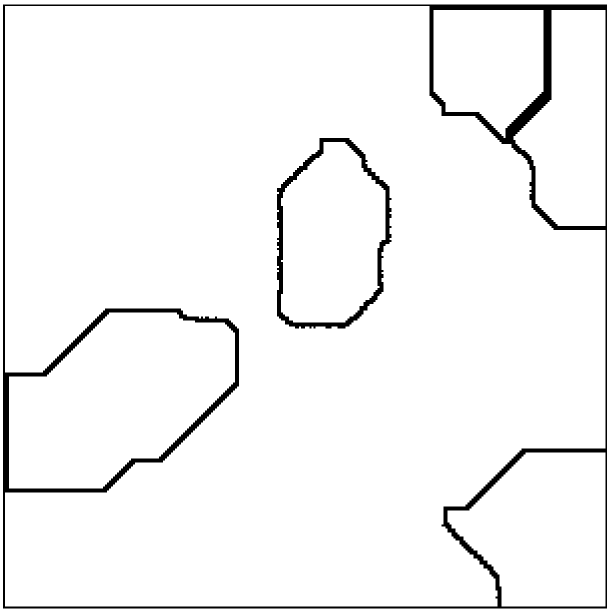
## 3.4. Grain boundaries

Another implemented feature is grain boundaries. After generate basic structure is possible to generate grain boundaries. User can set border size [Fig. 7] and select grains to process, if user won't select any grains, should be add for each grains. After process is possible to remove colors from grains.

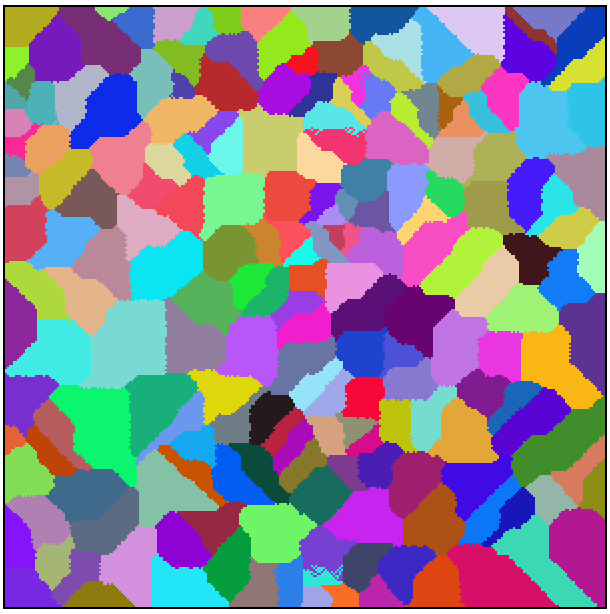
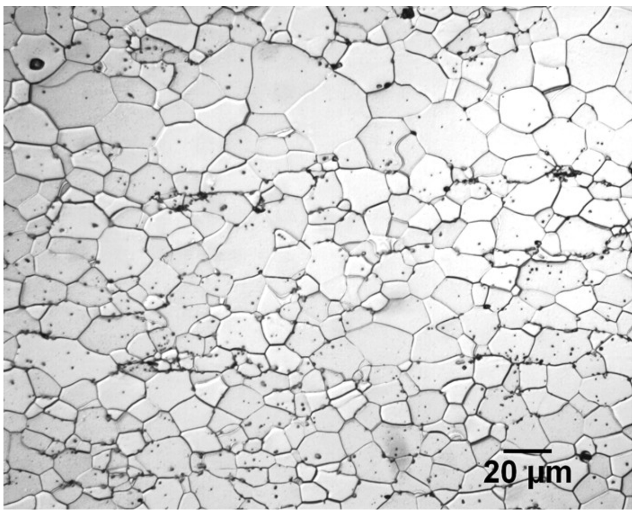
If user add boundaries for grains, will be possible to see label with percent of relation between boundaries and other structures. Results of test simulations you can see on bitmaps Fig. 24-28.

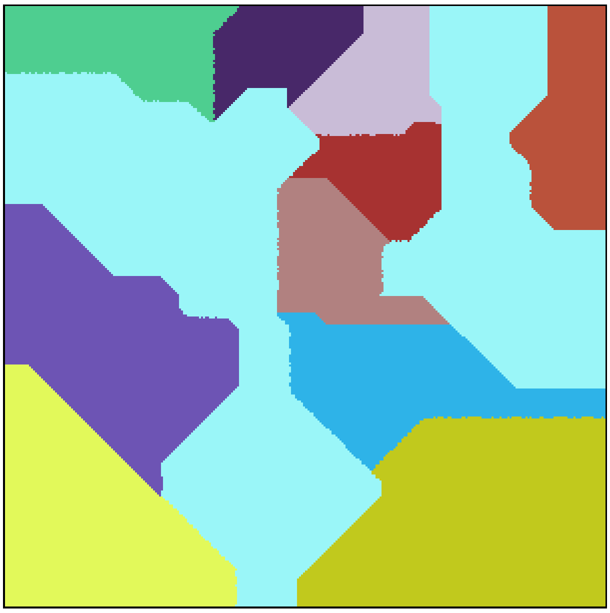
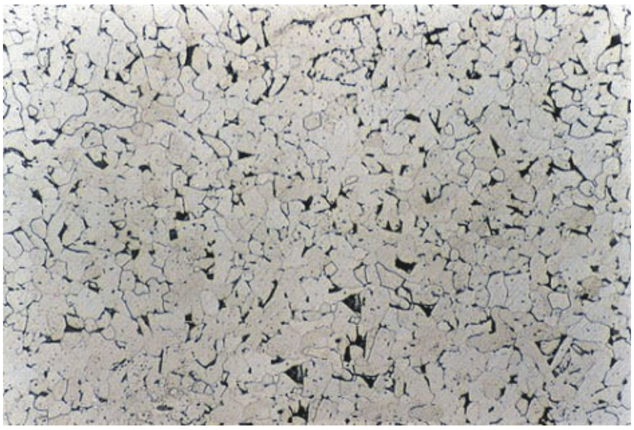
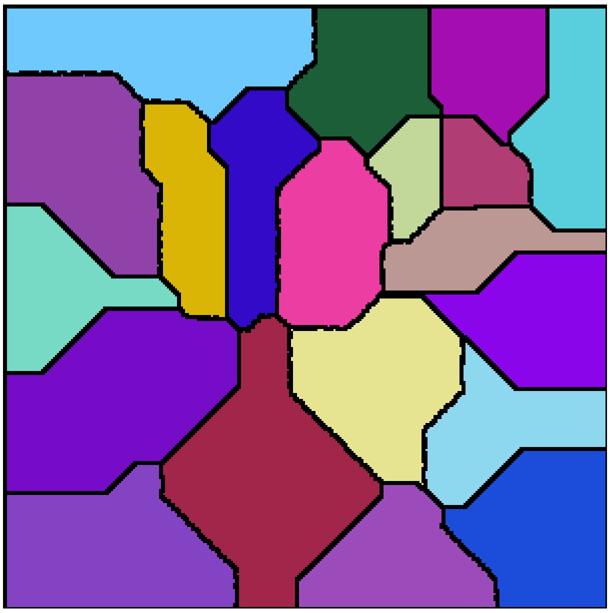
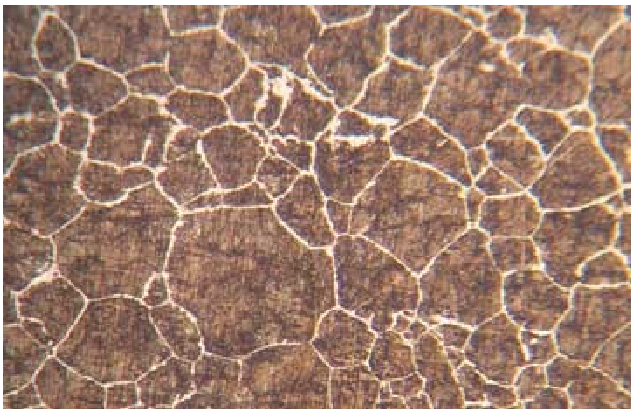
Number of grains: 20
Neighborhood: Von Neumann

<table>
<tr><td colspan="2"><br>*Fig. 24: Application output - borders baseline bitmap*</td></tr>
<tr><td>Number of grains: 20<br>Neighborhood: Von Neumann<br>Border size: 1<br>All grains<br>GB: 6,98%</td><td>Number of grains: 20<br>Neighborhood: Von Neumann<br>Border size: 1<br>All grains<br>Without grains colors<br>GB: 6,98%</td></tr>
<tr><td><br>*Fig. 25: Application output - all grains borders*</td><td><br>*Fig. 26: Application output - all grains borders after clear*</td></tr>
<tr><td>Number of grains: 20<br>Neighborhood: Von Neumann<br>Border size: 1<br>Selected grains<br>GB: 1,56%</td><td>Number of grains: 20<br>Neighborhood: Von Neumann<br>Border size: 1<br>Selected grains<br>Without grains colors</td></tr>
</table>

| | GB: 1,56% |
|---|---|
|  *Fig. 27: Application output - selected grains borders* |  *Fig. 28: Application output - selected grains borders after clear* |

# 4. Real structure compare

| Microstucture | |
|---|---|
|  *Fig. 29: Application output - microstructure* |   *Fig. 30: Motor lamination steel before decarburization etched with (a) Marshall's reagent* [ref1] |

| Dual Phase | |
|---|---|
| <br>*Fig. 31: Application output - Dual-Phase* | <br>*Fig. 32: Microstructure of as-Received Materials (a) Stainless Steel (b) Low Carbon Steel* [ref2] |
| Grain Borders | |
| <br>*Fig. 33: Application output - borders* | <br>*Fig. 34: Cementite or iron carbide (Fe3C)* [ref3] |
| Inclusions | |

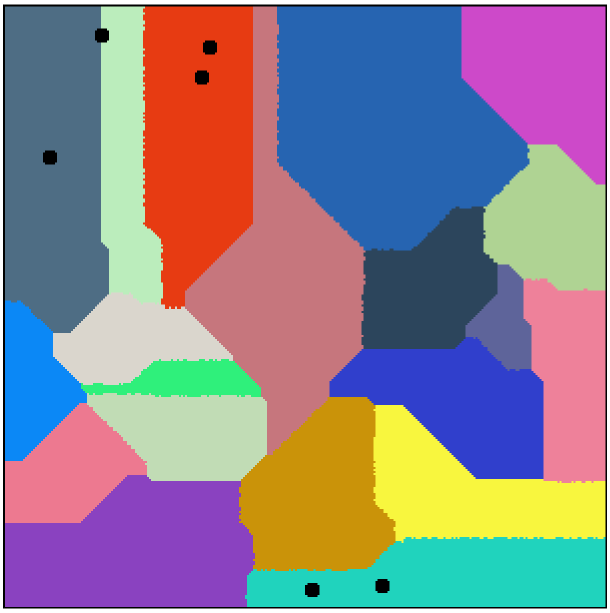


*Fig. 35: Application output - Inclusions*

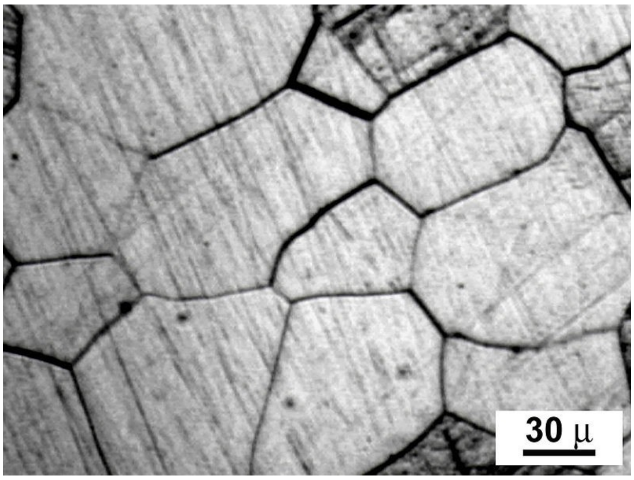



*Fig. 36: Micrograph of a polycrystalline metal; grain boundaries evidenced by acid etching* [ref4]

Substructures

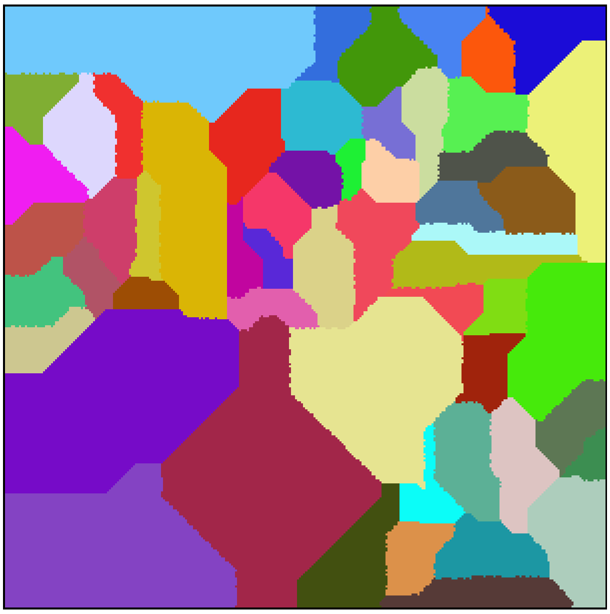


*Fig. 37: Application output - Substructures*

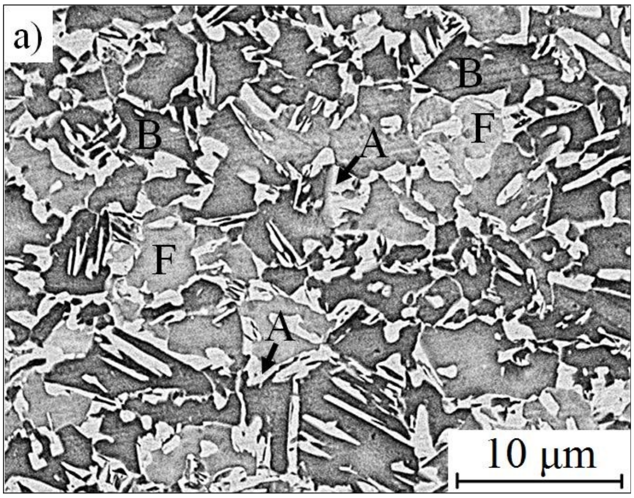



*Fig. 38: Abnormal grain growth observed in Rutile TiO2, induced by the presence of a zircon secondary phase* [ref5]

The application allows to generate structures similar to real [Fig. 29-38], but still it will be very simplified structure. Grains in real are much complex it it is hard to simulate all process for generate exactly this same structure. The best results you can see in basic grains structure and with inclusions.

The biggest problem is with compare substructures effect with real photo, but probably exist better example, where structures will be similar.

# 5. Conclusions

Grains in nature in material often are very complex, so it is huge problem for generate simulation for generate simple results. In this project used simple methods for create simulation. Results aren’t perfect, but after compare with real photos we can see similarities for generated structures.

Application implemented few neighborhoods method, more complex algorithm like Extended Moore, give us more natural grains form, but it complicate simulation, so we need more powerful machine to simulation bigger bitmaps.

Generated bitmaps can be used to next analysis, without specialist machine like expensive microscopes. Else application is generating clear output bitmaps, we don’t need to use filters and other graphic processing, like for data from microscope.

The biggest problem with application from technology side is about performance. Grains growth is very complex process and need optimal algorithm and technology. Created application need a lot of system resources and still need a lot of time for bigger simulation.

# Illustration Index

i Url for repository: https://github.com/Micp95/MultiscaleModelling (active for day: 14.11.2018)

ref1. url: https://vacaero.com/information-resources/metallography-with-george-vander-voort/1418-delineation-and-measurement-of-grain-size-by-ebsd.html (active for day: 14.11.2018)

ref2. url: https://www.researchgate.net/figure/Microstructure-of-as-Received-Materials-a-Stainless-Steel-b-Low-Carbon-Ste_fig1_310766913 (active for day: 14.11.2018)

ref3. url: http://www.cashenblades.com/metallurgy.html (active for day: 14.11.2018)

ref4. url: https://en.wikipedia.org/wiki/Grain_boundary (active for day: 14.11.2018)

ref5. url: http://www.wikiwand.com/en/Abnormal_grain_growth#/citenoteref14 (active for day: 14.11.2018)