Report

Karol Kiełbasa

1. Technology

Application was implemented in Java 8. One of the reasons why this programming language was used is that it has built-in support for graphic library - Swing. This library provides a set of complex and well looking graphic components. Next reason that stands for Java is platform independence. This application can be launched on any available platform.

2. Interface

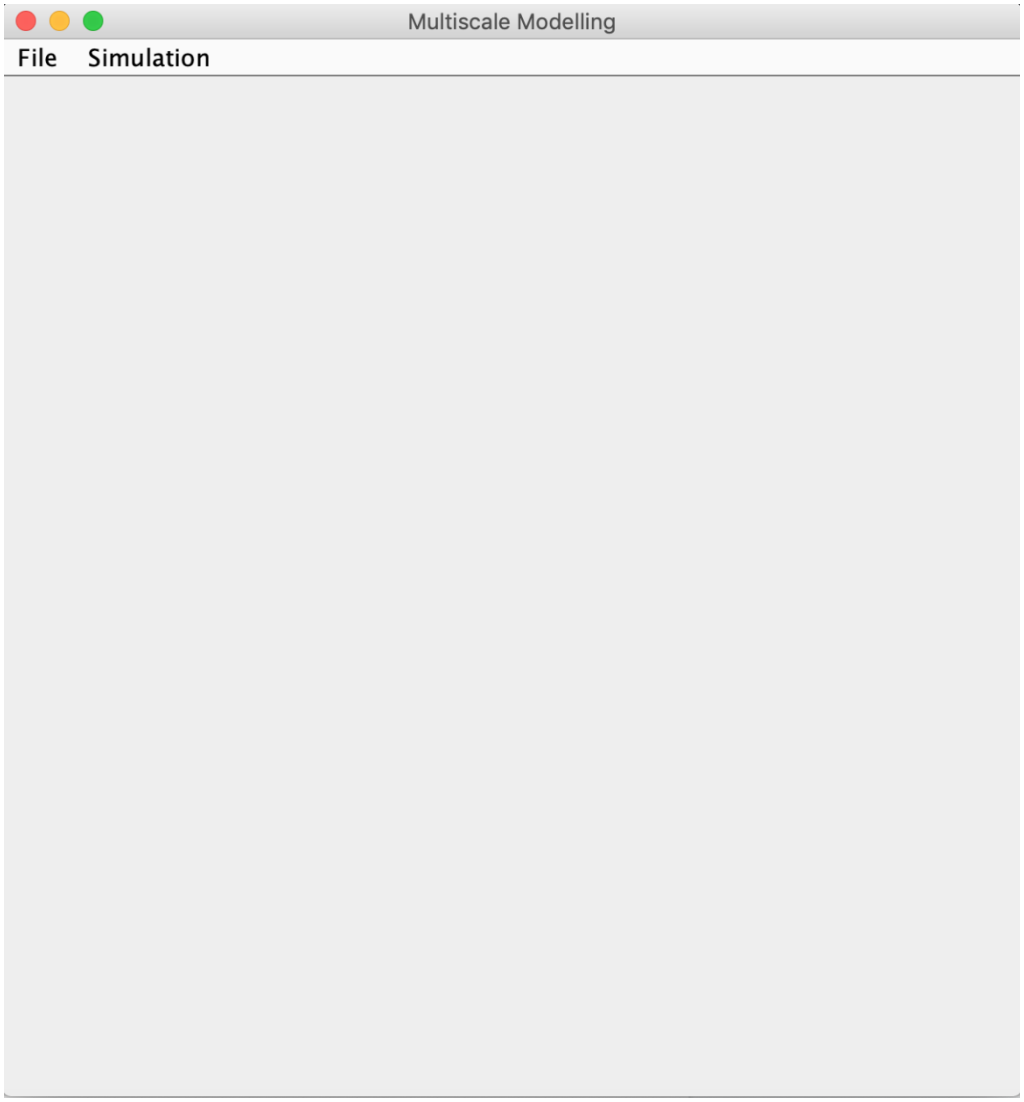

Image 1. Main window of application.

Right after launching the program main window appears as Image 1 shows. On the top there are two options: **File** and **Simulation**. In file menu there is a possibility of importing and exporting results of simulation in *.bmp* and *.txt* formats and also there is a button to exit the program. Menu is illustrated on Image 2.

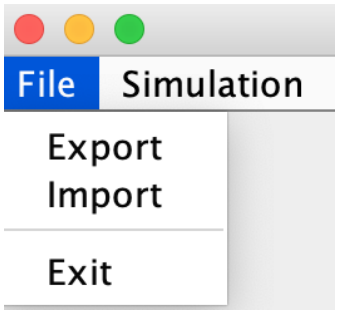

Image 2. File menu

Image 3 shows window for importing/exporting. To export/import user needs to firstly specify a format - if it is *.bmp* or *.txt*.

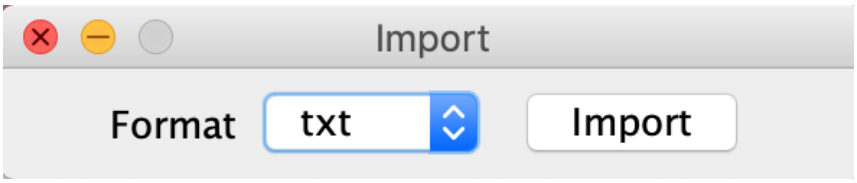

Image 3. Selection of format

Then a path where to export/from import a file. As on Image 4.

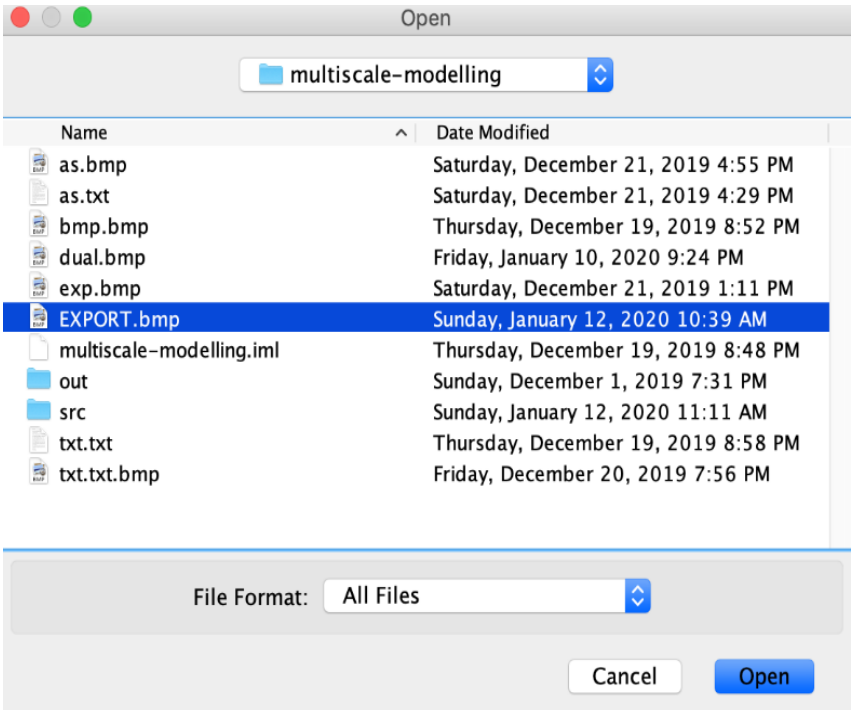

Image 4. Selection of a path to export/import

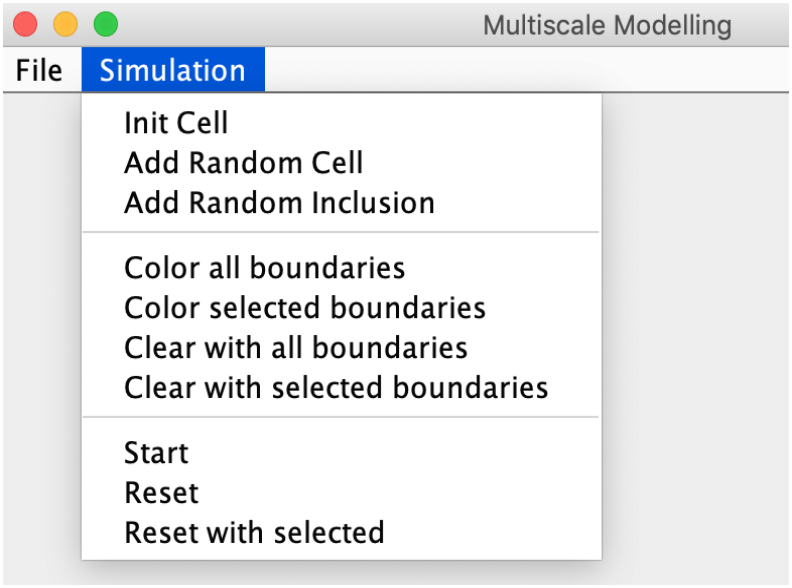

Image 5. *Simulation* menu

In Simulation menu, which is shown on Image 5 following options are available:

- *Init Cell* - initialize a graphic components where cells will be visible. User can provide size of a component and percentage rate applied for growth rules (Image 6).

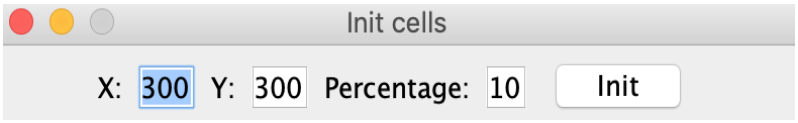

Image 6. *Init Cells* menu

- *Add Random Cell* - adds a given number of cells to component at random position (Image 7).

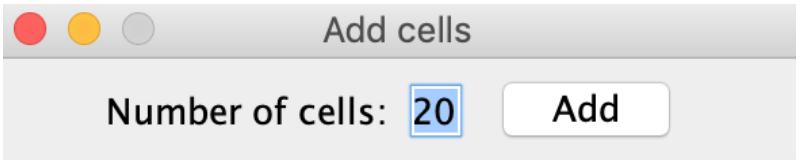

Image 7. *Add cells* menu

- *Add Random Inclusion* - adds inclusion to component at random position. User can provide a number of inclusions, size and type. Type might be circle and square (Image 8).

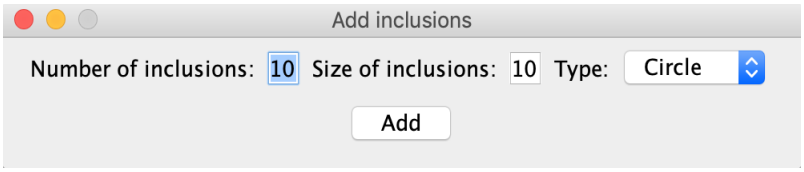

Image 8. *Add inclusions* menu

- *Color all boundaries* - colors all boundaries between all nucleons.
- *Color selected boundaries* - colors boundaries between selected nucleons.
- *Clear with all boundaries* - clear all cells except colored boundaries. User can provide a boundary size (Image 9).

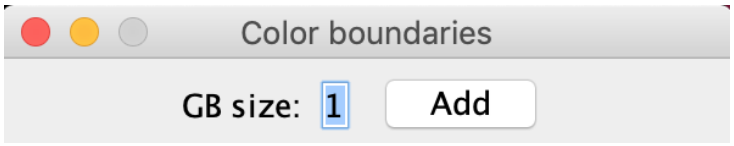

Image 9. *Color boundaries* menu

- *Clear with selected boundaries* - clear all cells except colored boundaries. User can provide a boundary size.
- *Start* - starts simulation.
- *Reset* - resets simulation.
- *Reset with selected* - reset all cells except selected nucleons. Selected nucleon's microstructure may be changed to dual phase or substructure (Image 10).

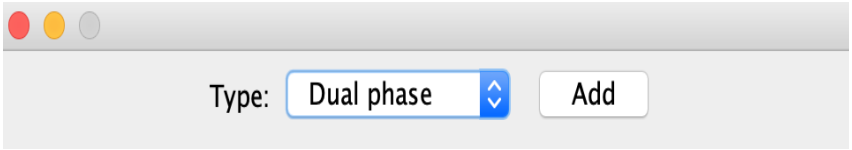

Image 10. *Reset with selected* menu

3. Results

First simulation started with 20 cells. After simulation, four inclusions in the shape of a circle was added. After that all boundaries between inclusions have been colored. Results are visible on Image 11.

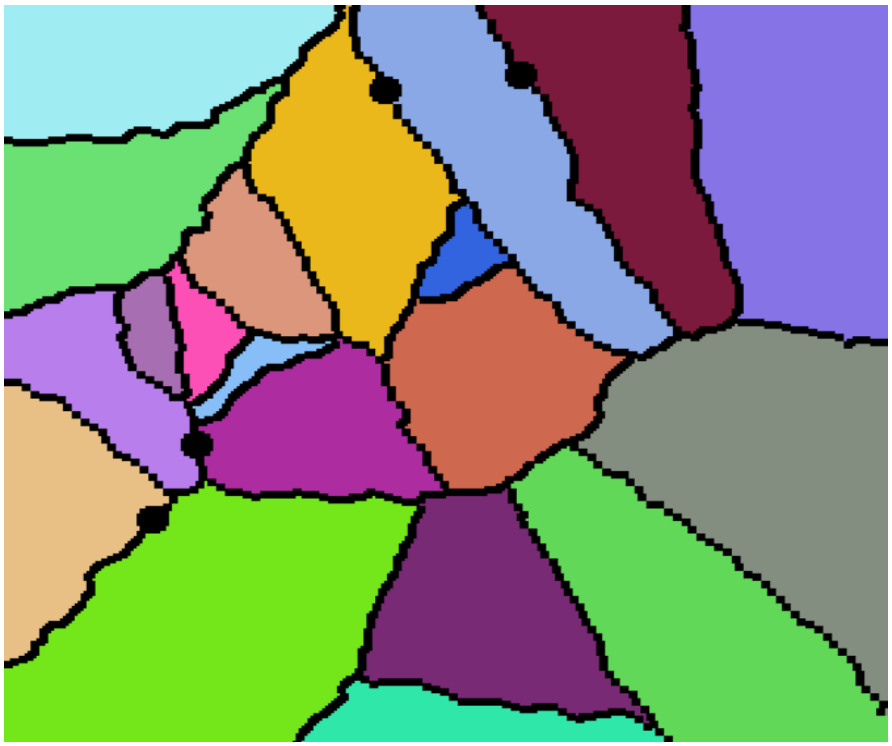

Image 11. Results of first simulation

Second simulation started with 20 cells. All boundaries was colored and then nucleons were cleared. Results are visible on Image 12.

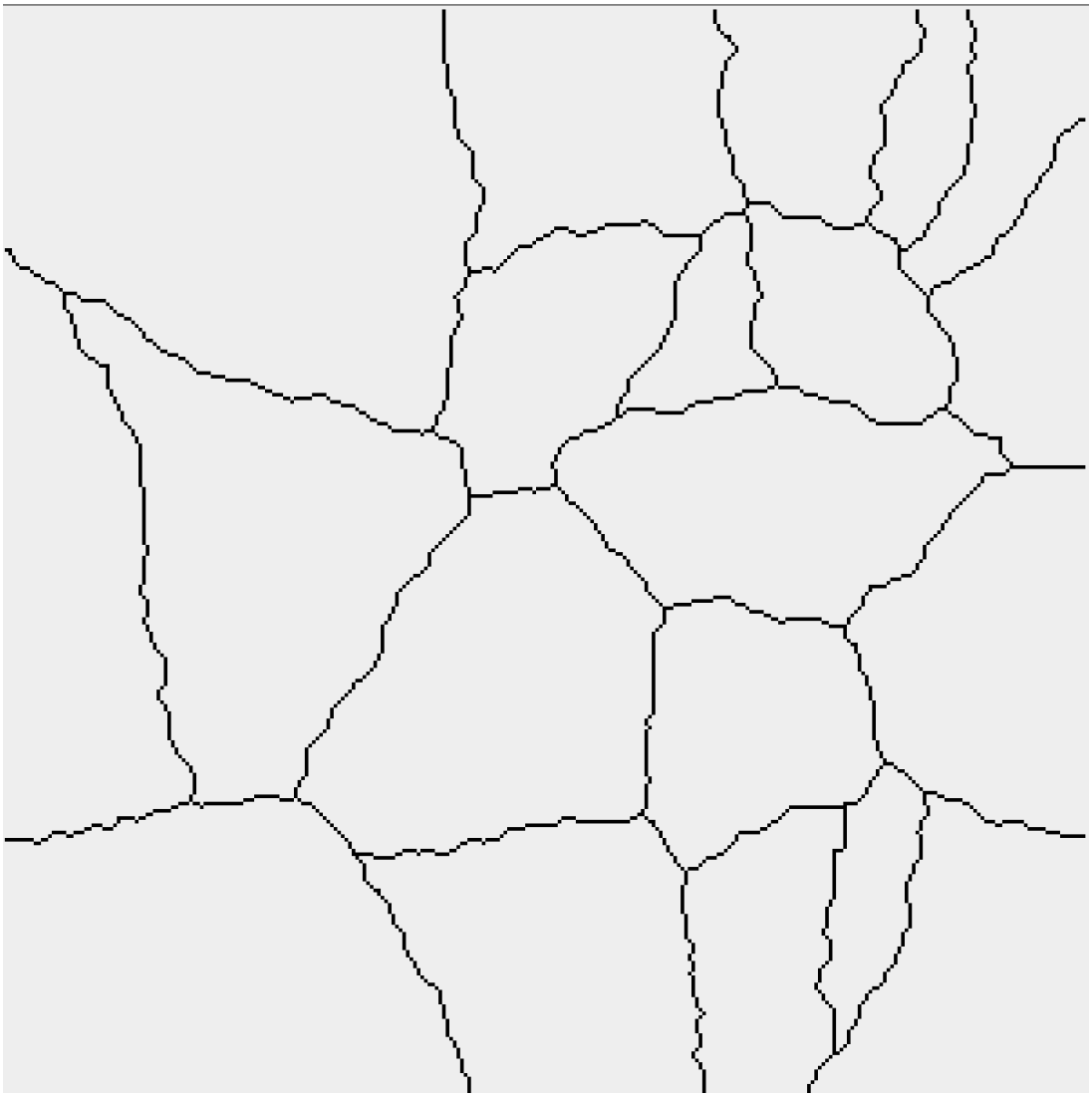

Image 12. Results of second simulation

Third simulation started with 20 cells. After simulation was finished, few nucleons were selected on the corners of image then changed a substructure with dual phase. In order to populate gaps between ‘big’ nucleons 90 cells were added and simulation started once more. Last step was adding inclusions with different size and shape. Results are shown on Image 13.

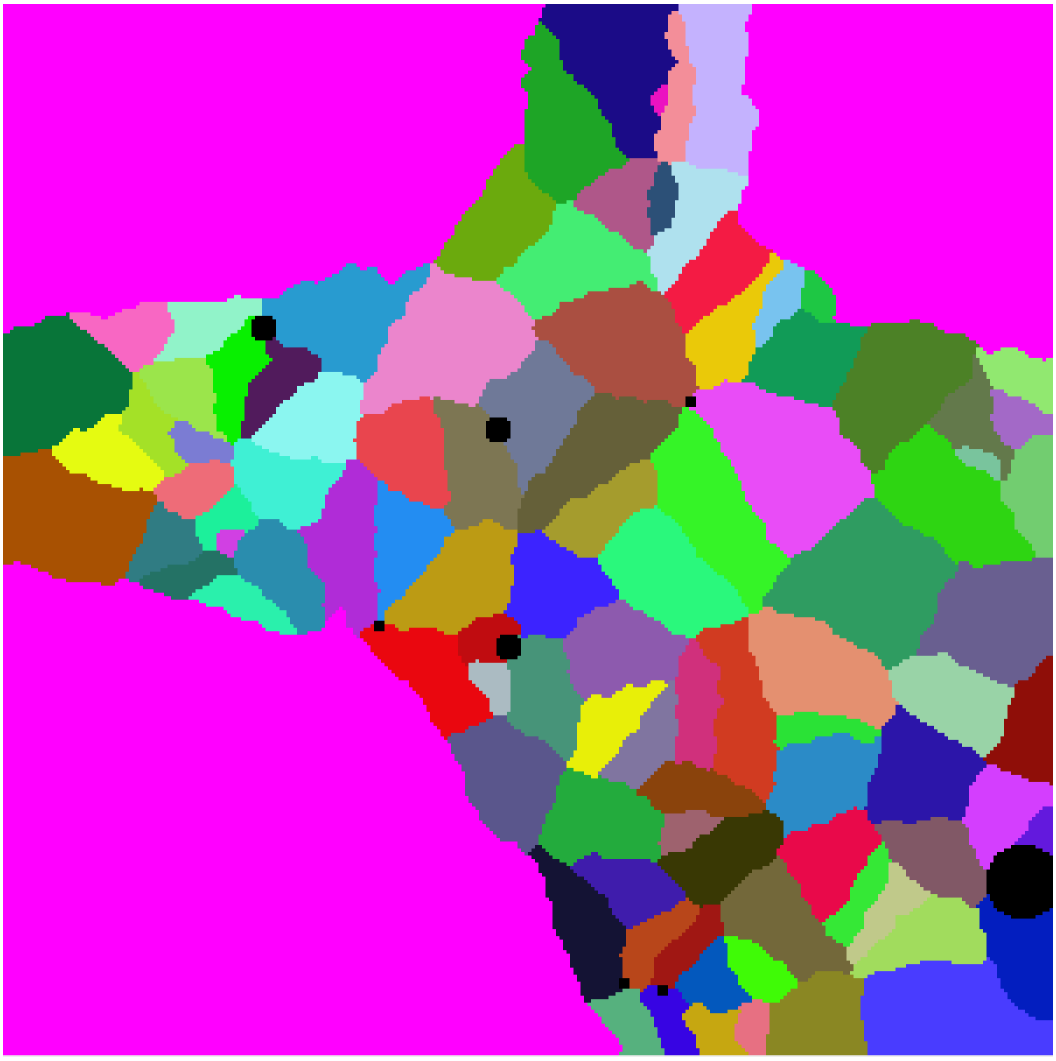

Image 13. Results of third simulation

4. Comparison with real microstructure

First attempt of reproducing real microstructure was quite successful. Similarity between Image 14 and Image 15 is visible.

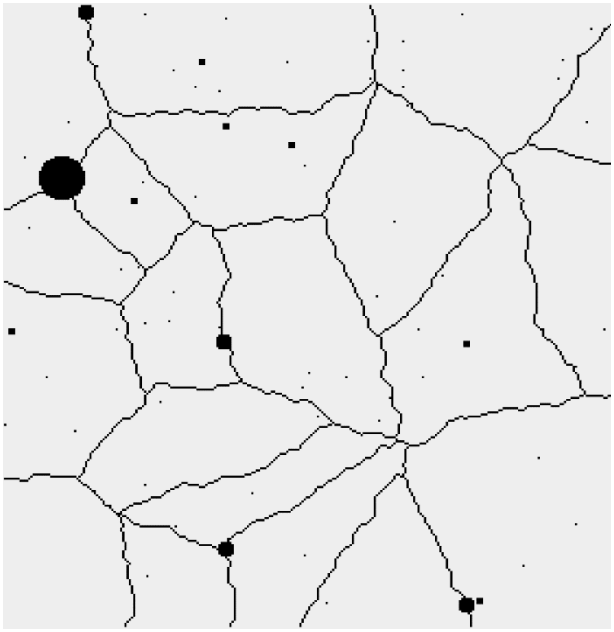

Image 14. Own results of simulation done in application – an attempt to reproduce picture of real microstructure.

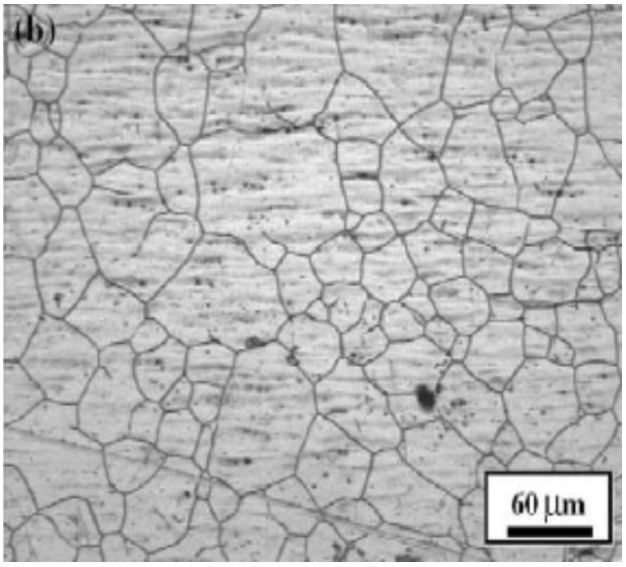

Image 15. Microstructure of austenitic stainless steel samples annealed in 1100 ° C for 15 min; the resulting average grain size is 40 l m (coarse grain – CG)
source: G. Bregliozzia, S.I.-U. Ahmed a, A. Di Schinob ,J.M. Kennyb, and H. Haefkea *Friction and wear behavior of austenitic stainless steel: influence of atmospheric humidity, load range, and grain size* Tribology Letters, Vol. 17, No. 4, November 2004

Second attempt of reproducing microstructure wasn't so successful. There are some differences between Image 16 and Image 17. For instance, application do not support nucleons which have shape of a rectangle.

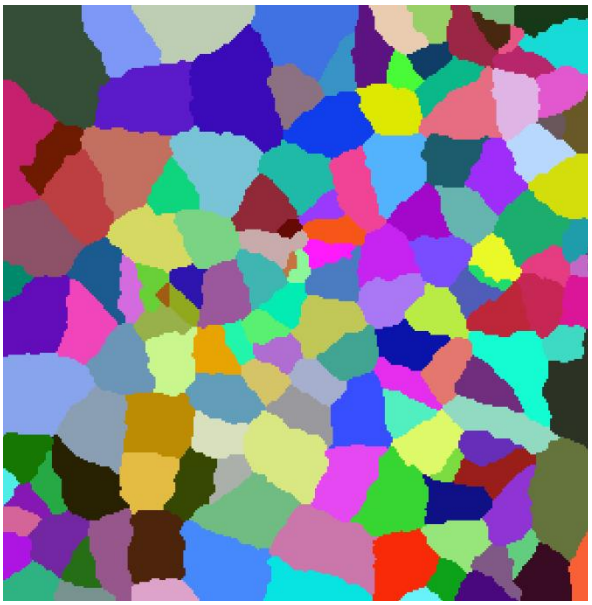

Image 16. Own results of simulation done in application – an attempt to reproduce picture of real microstructure.

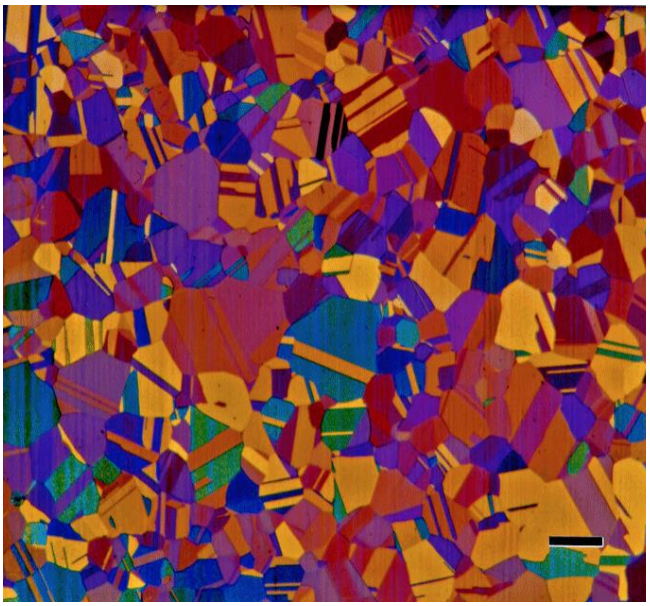

Image 17. Microstructure of Custom Flo 302-HQ austenitic stainless steel (Fe - <0.08% C – 18% Cr -9 % Ni 3,5% Cu) in the hot rolled and solution annealed condition after tint etching with Beraha's BI reagent.
Source:https://vacaero.com/information-resources/metallography-with-george-vander-voort/894-microstructure-of-ferrous-alloys.html

5. Conclusions

Application is able to handle basic operation like simple cell growth, adding inclusions, changing substructure, coloring boundaries of nucleons. Despite application simpliness results of simulation can be quite similar to real microstructure as first attempt shows (Image 14 and Image 15) . However, the goal of such applications is to reproduce a microstructure as similar as it would be in reality. Achieving this goal requires more advanced features like drawing boundary which is not connected to any other boundary. In addition second attempt (Image 16 and Image 17) shows that application do not handled reproducing microstructure very well. Main reason for that is application doesn't support drawing nucleons, which have the shape of a rectangle.

To sum up, implementing an application which is able to reproduce real microstructures is very complex and challenging task, which requires knowledge not only about programming but also

metallurgy. Application described above supports only trivial operations, which do not allow to fully reproduce real microstructures.