Abstract

The paper presents a novel visualization technique for cracks propagating in mono-dispersed particulate material. The proposed technique is based on local space decompositions generated in fractured areas. The contact surfaces of the neighbouring particles are defined by the local Voronoi decomposition generated according to the lattice topology employed in computations of the discrete element method. The visual model validation helps to indicate the regions of a highly deformed lattice, where the defects detected between the pairs of the neighbouring particles on the lattice connections cannot be directly mapped onto the relevant edges of the Voronoi diagram. The parallel implementation of the visualization technique is based on the domain decomposition and two layers of ghost vertices and connections. The developed technique is implemented in the distributed visualization software VisPartDEM. Datasets of the elastic solid problem exhibiting non-uniform distribution of fracture force values are considered to validate the performance of the proposed technique. The parallel speed-up of the visualization software is investigated. The superior performance of the applied local technique is compared to the performance observed by using the standard global Voronoi algorithm.

Keywords: distributed visualization, propagating cracks, discrete element method, local Voronoi decomposition.

1 Introduction

The discrete element method (DEM), introduced by Cundall and Strack [1], is widely used in numerical analyses of particulate media from the perspective of science and engineering. DEM is based on a Lagrangian approach, where individual particles are calculated according to the Newton’s second law of motion. The contact forces between two particles are determined from the overlap and relative
movements of the particle pair according to a specified force-displacement law. DEM can simulate the behaviour of discrete particles, such as grain, soil and powder, as well as the fracture of non-continuous brittle mediums, such as rock, concrete and laminated glass [2-4].

Several different particle- and lattice-based approaches have been developed in the frame of DEM applied to the simulation of solids and fracture. The particle-based approach [5, 6] replaces a solid with a composition of discrete particles, where the presence of the cohesive forces acting between the particles and various mechanisms of their linkage and detachment may be allowed. The approach, where the material particles interact via the network elements, connections or bonds, can be referred to as the lattice-based model [7]. A comprehensive review of the planar elastic lattice models for micro-mechanical applications is given by Ostoja-Starzewski [8]. The lattice-based DEM [9] and the bonded particle model [10] have been extensively applied to the simulation of heterogeneous solids to study their fracture problems. The lattice-based DEM can take into account most kinds of discontinuities and material failure characterized with multiple fracture which makes it an effective tool to study fracture phenomena.

DEM computations are based on the positions of particles and forces acting between them. The lattices employed in DEM computations are assembled from the 1D springs or beams which are not well suited for the reliable interpolation and common visualization techniques used in 3D. Based on the described approach, the defects between the pairs of the neighbouring particles are identified by the broken lattice connections or bonds [9-10]. The main visualization task is to construct the surfaces of cracks from the broken lattice connections and the geometry of the neighbouring particles. The complex disjoint surfaces of cracks and the unavailability of a suitable scalar field defining the crack surfaces limit the application of the widely used surface extraction methods. Thus, there is hardly any direct technique of constructing the surfaces of propagating cracks from individual defects determined between the neighbouring particles and applying the available visualization methods. Moreover, the commercial finite element analysis software widely used by engineers can import only continuously defined crack surfaces represented by usual graphic primitives. The successful research into the problems of representation and visualization of crack surfaces can help to fill the gap between the industrial requirements and the research results currently obtained in the areas of fracture mechanics and material sciences.

The technique based on the local Voronoi decomposition is applied to visualize cracks, propagating in mono-dispersed particulate media. Local Voronoi decompositions, serving as the contact surfaces of the neighbouring particles, are generated according to the lattice topology employed in computations. The model validation is developed to identify the regions of a highly deformed lattice, to which the Voronoi decomposition-based techniques cannot be directly applied because of inconsistency between the lattice connections and the faces of the Voronoi diagram. This paper is mainly based upon the work of Kačeniauskas et al. [11], however, the following additional research was made: the visualization technique was extended to 3D, the new 3D benchmark was considered and parallel implementation of the visualization technique based on the domain decomposition was investigated.
The paper is organised as follows. The related works are discussed in Section 2. The mathematical model of the lattice-based DEM is outlined in Section 3. The visualization technique based on the local Voronoi decomposition is presented in Section 4. The details of implementation and the employed software are provided in Section 5. The obtained results and performance are discussed in Section 6, while the concluding remarks are given in Section 7.

2 The related works

Cracking is a very common phenomenon investigated by a wide research community in different scientific areas. Cracks are applied to ceramics [12], to soil [13] and, more generally, to any object represented as a 3D mesh [14]. Geometric models propose algorithms to get crack patterns close to those produced by nature. In contrast, physical approaches propose the models which tend to accurately simulate the dynamics of natural crack patterns. O’Brien et al. [14] succeeded in modelling brittle and ductile fractures by means of the finite element method. A dynamic model of crack development based on the 3D discrete shrinkage volume propagation is proposed by Valette et al. [15]. The lattice-based DEM provides a valuable insight into the fracture phenomena at the particle level [9, 10].

Computer graphics scientists focus their attention on enhancing the realism of natural scenes, while computational researchers concentrate on building accurate numerical models. Furthermore, their principal aim is not closely related with visually attracting results. In the lattice-based DEM methods, cracks and the related phenomena are often visualized in the most straightforward way. The particles coloured depending on particular scalar attributes, such as the initial high altitude or radii, are most common [16-19]. The propagation of large cracks is illustrated by using the rendered geometry of particles [19]. However, this technique can be applied only to large cracks which are of the particle size. The coloured lattice connections [9, 20] can be treated as the main alternative to the rendered particles. The broken connections between the neighbouring particles indicate the fractured regions, but do not provide any valuable information on the formation of the crack surfaces. The application of the finite elements or the Voronoi polyhedra allows us to colour the elements or cells [21]. However, the absence of the defined crack surfaces limits visual analysis of the computed results and reduces the understanding of the investigated fracture phenomena.

The problem of reconstructing a surface from a set of sample points is motivated by numerous applications and, consequently, has always been a popular field of research. Most surface reconstruction methods roughly fall into two major categories: implicit surface methods [22, 23] and Delaunay-based methods [24, 25]. The most common approach to surface reconstruction is based on the Delaunay triangulation: the underlying idea is that, when the sampling is noise-free and dense enough, points close on the surface should also be close in space. Among the Delaunay-based methods, the most well-known algorithms are, preferably, the Crust [24] and the Cocone [25] families of algorithms. Another approach that directly extracts smooth surfaces from unstructured point-based volume data without prior resampling or mesh generation is proposed by Rosenthal and Linsen [23]. In the
case of lattice-based DEM methods, the discussed surface extraction algorithms cannot be directly applied because of the absence of suitable data defining the crack surface and the complex nature of crack surfaces, defined by the scattered results of numerical computations at the micro-level. The holes and disjoint pieces of crack surfaces make surface extraction extremely complicated.

The Voronoi diagrams can be applied to post processing of particle systems as well as to lattice forming. Given a set of primitives, the Voronoi diagram partitions space into regions, where each region consists of all points that are closer to one primitive than to any other. The Voronoi diagrams have important applications in many sciences, including visualization of medical datasets, proximity queries, spatial data manipulation, shape analysis, computer animation, robot motion planning, modelling spatial structures and processes, pattern recognition, locational optimization and selection in user-interfaces [26, 27]. A concept of the Centroidal Voronoi tessellation was presented in the form of graphs in [28]. The edges of radical Voronoi diagrams were employed to construct a beam-network model for autoclaved aerated concrete [29]. Computational lattices for poly-dispersed particulate media are assembled according to the Voronoi diagrams by Cusatis et al. [19]. However, there were no attempts to describe the contact surfaces of particles and to visualize the propagating cracks by using local space decompositions based on the Voronoi cells.

3 A mathematical model of the discrete element method

The motion of the visco-elastic particle system is governed by the Newton’s second law, i.e.

\[ \frac{d^2 x_i}{dt^2} = F_i, \]  
\[ \frac{d^2 \theta_i}{dt^2} = T_i, \]

where \( m_i \) and \( I_i \) are the mass and the moment of inertia of the particle, respectively, while vectors \( x_i \) and \( \theta_i \) initiate the position of the particle centre and the orientation of particle \( i \). The vectors \( F_i \) and \( T_i \) present the sum of the contact force and the gravity force as well as the respective torques:

\[ F_i = \sum_{j=1,j\neq i}^{N} F_{ij,\text{cont}} + m_i g, \]  
\[ T_i = \sum_{j=1,j\neq i}^{N} T_{ij} = \sum_{j=1,j\neq i}^{N} d_{ij} \times F_{ij,\text{cont}}, \]

where \( g \) is the acceleration due to the gravity, \( d_{ij} \) is the vector pointing from the particle centre to the contact centre. The interparticle force vector \( F_{ij,\text{cont}} \), describing the contact between the particles \( i \) and \( j \), may be expressed in terms of normal and
tangential components. The normal component, presenting a repulsion force, comprises elastic and viscous ingredients, while the tangential component reflects static or dynamic frictional behaviour. The static force describes friction prior to gross sliding and comprises elastic and viscous ingredients, while the dynamic force describes friction after gross sliding and is expressed by the Coulomb’s law.

For evaluating the contact forces (3-4), all contacts between the particles and their neighbours must be detected. A cell-based method is used for contact detection. The numerical integration of the equations of motion (1-2) is performed to obtain the dynamical state of all particles at the time $t$, resulting from the action of the particle forces (3-4). The solution of these equations is obtained by using the Verlet scheme. The details of these procedures can be found in [6].

In the present work, the materials are modelled using the elastic perfectly brittle model of contact interaction. The lattice-based discrete element model assumes cohesive bonding between the neighbouring particles [4, 10]. These lattice connections represented by springs can be broken under excessive loading, which allows us to simulate initiation and propagation of the material fracture. When two particles are bonded, the contact force in normal direction is calculated from the linear constitutive relationships:

$$ F_{n,\text{cont}} = k_n u_n, $$

where $F_{n,\text{cont}}$ is the normal contact force, $k_n$ is the spring stiffness in the normal direction, $u_n$ is the normal relative displacement. The lattice connections are broken instantaneously, when the interface strength is exceeded in the normal direction by the tensile contact force. The failure criterion can be written as

$$ F_{n,\text{cont}} \leq R_n, $$

where $R_n$ is the interface strength in the normal direction. Although a compressive interaction force between the particles does not cause breakage of the connections, the material damage under macroscopic compression can be represented properly in the particle model. A compressive macroscopic load brings about tensile interactions at the microscopic level which may lead to connection failures.

After breakage of connections, a normal contact force is calculated, using the Hertz contact model. A frictional interaction can occur among the particles in the case of compression. The limiting friction force is evaluated, assuming the Coulomb model of friction:

$$ |F_{t,\text{cont}}| = \mu |F_{n,\text{cont}}|, $$

where $\mu$ is the Coulomb friction coefficient. The simple mathematical model (1-7) of the lattice-based DEM is provided for the sake of completeness. However, the simulation results will not be examined in detail because the presented research is concentrated on the surface extraction and the proposed visualization technique.
4 The visualization technique

The visualization technique based on the local Voronoi decomposition has been developed for the analysis of propagating cracks. The applied technique plots the crack on the contact surfaces obtained from local decompositions. During DEM computations, the defects are identified between the pairs of the neighbouring particles on the lattice connections. The broken lattice connections should be directly mapped onto the generated faces. The consistency of the generated faces of the Voronoi decomposition and the relevant lattice connections should be verified in the case of the highly deformed lattice. The standard algorithms for generating the global Voronoi diagrams are hardly applicable to visualization of cracks because they use the coordinates of points, but do not take into account the lattice topology.

4.1 A general algorithm

A general schema of the algorithm developed for generating the local Voronoi decompositions is presented in Figure 1. Data structures for loops, running through the particle neighbours, which are joined to the considered particle by the lattice connections, are prepared at the first stage of the algorithm. The next module constructs the kd-tree, which is used for visual model validation at other stages of the algorithm. A loop, running through all lattice connections, starts the main computations. The following condition checks if the current connection is broken. The cells of the local Voronoi decomposition are generated only around the end nodes of the broken connections, therefore, the next loop runs through two end nodes of the current connection. The next condition checks if the current node has not been processed yet because the Voronoi cell can be already generated around this node processing another broken connection attached to it. The following module generates the Voronoi cell around the particle centre represented by the processed node of the current connection. Each Voronoi cell is created around the node according to the available data on the node neighbours. The implemented algorithm [30] is based on the clipping of the whole domain by perpendicular planes drawn through the midpoints between the considered node and its neighbours.

All faces of the generated Voronoi cell are validated by the orange module, which will be described in the following subsection. The loop, running through the valid faces of the generated Voronoi cell, finishes processing of the generated contact surfaces. The following module maps the attributes of the lattice connections onto the relevant suitable faces of the generated Voronoi cell. In the 2D case, the attributes of the lattice connections are mapped onto the relevant edges of the cell. When the Voronoi cells are created around both end nodes of all broken connections, the main computations are finished. The last module prepares the output for visual model validation. Three orange modules (Figure 1) used in model validation will be described in the following subsection.
Figure 1: A general schema of the local Voronoi decomposition-based technique.
4.2 Visual model validation

Visualization of crack surfaces is a challenging problem, while the application of the Voronoi decomposition-based techniques is hardly possible in the regions of the highly deformed lattice. It is worth mentioning that the most of computational models also have limitations in such complex cases. In the lattice-based DEM computations, the lattice topology does not change in time, while the particles can significantly change their positions. The particles have more freedom to move and to deform the lattice in the regions containing a lot of broken connections. Thus, after a certain period of time, the lattice topology might become inconsistent with respect to the cells of a standard Voronoi diagram.

Figure 2: An explanatory schema of Voronoi decompositions: (a) the global Voronoi decomposition, (b) the local Voronoi decomposition, (c) the visual model validation.

Figure 2a illustrates the case, when the stationary lattice topology became inconsistent with a changing global Voronoi diagram. Blue tubes represent the edges of the generated decompositions, while red tubes visualize cracks relevant to the broken lattice connections. The broken yellow connection of the stationary lattice joins the centres of the particles 2 and 4, while the algorithm for generating the standard Voronoi diagram takes into account the fact that the distance between the centres of the particles 1 and 3 is shorter and does not generate the edge perpendicular to the broken connection. The available cyan edge represents the contact surface between the particles 1 and 3. The observed inconsistencies do not allow the direct mapping of the attributes of the lattice connections onto the relevant edges of Voronoi cells. Therefore, the global Voronoi diagrams cannot be applied to crack visualization in the regions of the highly deformed lattice. Figure 2b shows that, in such regions, the local Voronoi decomposition generated according to the stationary lattice topology contains the overlapped cells and introduces the numerical error.

In this work, the performed validation is based on the well-known property of the Voronoi diagrams described as follows: for each vertex $q$ of the Voronoi diagram, there exists a unique empty sphere centred on $q$, which passes through at least four nodes and is the largest empty sphere centred on $q$ [26, 27]. Figure 3 presents the algorithm for validation of each generated Voronoi cell. A loop, running through all
the vertices of the considered Voronoi cell, starts the validation. The sphere is defined by the centre located in the considered vertex and the radius, which is equal to the distance from the considered vertex to the node representing the centre of the Voronoi cell. The query to the initially prepared kd-tree structure provides the number of nodes belonging to the solid sphere. The considered vertex is marked as suitable for the Voronoi decomposition if none of the nodes of the lattice is located inside the solid sphere. In other cases, the processed vertex does not satisfy the required conditions of the Voronoi diagram. Usually, this can be observed in the regions of the highly deformed lattice, where the lattice connection cannot be directly mapped onto the Voronoi face. The end of the loop, running through all vertices of the considered Voronoi cell, finishes the consistency check.

Figure 3: Validation of the faces of the local Voronoi decomposition.
The following loop, running through all faces of the considered Voronoi cell, is used to mark the suitable faces of the Voronoi cell and the appropriate connections of the lattice. If the considered face is a boundary face, marking is not performed. The next condition checks all the vertices of the considered face. If all vertices have been marked as suitable, the entire face is marked as suitable. Otherwise, the connection, corresponding to the considered face, would be marked as belonging to the region of the highly deformed lattice, which could not be accurately visualized by using the Voronoi cells. The end of the loop, running through all faces of the considered Voronoi cell, finishes the marking procedure. Figure 3c demonstrates how the visual model validation helps to indicate the regions of the highly deformed lattice by plotting yellow tubes and to prevent the appearance of artefacts. It is worth noting that all validation modules can be removed from the pipeline to save computational time in the case of small geometric deformations in the lattice.

5 Implementation

The developed technique for visualization of propagating cracks is implemented in the distributed software VisPartDEM, developed for interactive exploration of discrete particle systems.

5.1 The distributed visualization software VisPartDEM

The distributed architecture of VisPartDEM (Figure 4) is designed for interactive visualization on different infrastructures, such as graphics workstations, Rocks clusters and gLite grids.

Figure 4: The distributed architecture of VisPartDEM.
The VisPartDEM client, implemented as Java application, does not depend on hardware and operating system, therefore, it runs on any platform. The VisPartDEM client connects to the interface computer of any infrastructure by means of JSCH library, which employs the SSH connection. The commands for job submission and monitoring are enwrapped by the Java programming language. The considered visualization pipelines, XML files and shell scripts for running visualization engine are generated automatically in order to submit job to the required infrastructure. The VTK-based [31] distributed visualization engine of VisPartDEM is platform independent and can efficiently run on different infrastructures. A high level of abstraction of the VTK graphics model makes faster the development of useful graphics and visualization applications.

GVid software [32] implemented in the Remote Viewer can transfer the video stream from any node of the investigated infrastructure to any client computer having IP address. As a result distributed visualization engine runs in parallel on working nodes while the video stream is transferred by GVid through the network from zero MPI node and displayed on the client. Thus, remote user has high interactivity provided by the Remote Viewer empowered by flexible VTK widgets.

5.2 Parallelization of the visualization technique

In VisPartDEM software, the data parallel approach of VTK is employed for visualization of large datasets resulting from DEM computations. A large dataset is partitioned into many independent subsets that are processed in parallel. Copies of the same modules run on each CPU and visualize independent subsets of data. Data parallel modules are usually followed by a data parallel merge module that gathers the independently computed results and merges them into a final result on a single processor.

![Figure 5: The employed domain decomposition including two levels of ghost cells.](image)

The proposed visualization technique is naturally parallelized by using the domain decomposition because the local Voronoi cells are generated independently in each subdomain. Domain partitions used in the DEM computations are also employed for visualization. The static domain decomposition based on the
orthogonal coordinate bisection method [33] is considered. Two layers of ghost data are generated to avoid interprocessor communication during the execution of the developed Voronoi filter. In Figure 5, black circles represent the neighbourhood information required for generating the Voronoi cell around the vertices (particles) located in the centres of the circles. The neighbourhood information of each particle includes one layer of neighbouring particles and connections. Thus, two layers of vertices (particles) and connections are required to generate Voronoi cells around the both end nodes of the broken connection, bounding the neighbouring particles that belong to different subdomains. Consequently, in each subdomain, the visualization and rendering are performed independently, while a sort-last parallel rendering class inputs a z-buffer and image pair from each process and outputs a composite result image.

### 5.3 Visualization pipeline

The proposed technique for generating the contact surfaces and visualizing cracks is implemented in the visualization engine of VisPartDEM. The general visualization pipeline based on VTK is presented in Figure 6. Initially, each process independently reads its data file containing the part of the dataset by using the developed `vtkHDF5Reader`. In the developed `vtkVoronoiFilter`, local decompositions are generated according to the input array of attributes, referred to as the connection state, which identifies the broken lattice connections.

![Figure 6: Visualization pipeline.](image-url)
The users are able to select preferable branches of the visualization pipeline by using GUI. The first branch of the pipeline plots crack surfaces. The second branch shows the local space decomposition of fractured regions. The last two branches of the pipeline are responsible for graphical output of visual model validation. The third branch plots the elongated connections, while the fourth one shows the connections that are inconsistent with the faces of Voronoi decomposition in the regions of the highly deformed lattice. All branches of the visualization pipeline can be combined and executed simultaneously. Various graphical representations of crack surfaces may be considered by users. The lattice connections, the edges of the faces and 1D cracks can be represented by tubes employing vtkTubeFilter. The condition checks if tube representation is selected by users. Finally, the values of the considered attributes are mapped onto colours by the vtkPolyDataMapper, while graphical primitives are rendered by using vtkRenderer. A parallel rendering class vtkCompositeRenderManager inputs a z-buffer and an image pair from each process by using MPI communication and outputs a single composite result image to MPI process zero.

6 Results and discussion

The developed visualization technique and the distributed VisPartDEM software are applied to visualize the geometry of cracks, propagating in mono-dispersed particle media. The visualization benchmark, performance study including the quantitative comparison with the results obtained by using other software and the attained parallel speed-up are presented in this section.

6.1 A description of the visualization benchmark

The numerical results obtained in solving the uniaxial tension problem in 2D and 3D were visualized to validate the performance of the developed technique. The considered lattice-based DEM model \([4, 9]\) is able to describe the elastic solid problem, exhibiting non-uniform distribution of the fracture force values.

Crack propagation in the rectangular plate was visualized to validate the extraction of crack surfaces in 2D. Two plate boundaries are clamped by connecting them to rigid walls, while other boundaries are free. External excitation is kinematic and it is implemented via the motion of the clamped boundaries defined by the constant velocity \((u = 0.05 \text{ m/s})\) in order to simulate tension in the specimen with the dimensions of \(0.376 \times 0.107 \text{ m}\). The simulated system consists of 4679 particles, while the lattice is formed from 13722 springs that are considered to be the 1D connections visualizing the obtained results. The initial defect defined by three broken connections is placed in the middle of the plane bottom. The dataset produced by DEM computations includes positions, velocities, accelerations, radii and material properties of particles, as well as the 1D lattice connections between the neighbouring particles and their attributes, such as the connection state, the force and the fracture force limit. The size of the visualized 2D dataset storing the results of 1284 selected time steps is equal to 1.4Gb.
Crack propagation in the rectangular cuboid (Figure 7) is considered to be a three-dimensional visualization benchmark. The tension is applied via the prescribed velocity \( u = 0.025 \text{ m/s} \) of the clamped boundaries simulating the specimen with the dimensions of \( 0.376 \times 0.107 \text{ m} \). The initial defect of the rectangular shape is defined by 121 broken connections. The simulated system consists of 46875 particles, while the lattice is formed from 267674 springs that are considered to be the 1D connections. The size of the visualized 3D dataset storing the results of 200 selected time steps in HDF5 files is equal to 2.5Gb.

![Figure 7: The geometry and the initial defect of the 3D benchmark problem.](image)

Figure 7 presents visualization of cracks propagating in the rectangular plate. The local Voronoi decomposition is generated in the vicinity of cracks to define the contact surfaces of the particles coloured in blue. The cracks are represented by red tubes that are plotted on the contact surfaces. Model validation is illustrated by rendering the coloured lattice connections. The elongated connections are represented by green tubes while the high deformation areas are shown by yellow tubes. In the regions containing a lot of broken connections of the lattice the particles have more freedom to move. Consequently, high deformations can prevent generating the Voronoi decompositions suitable for visualization purposes.

Figure 8 shows visualization of the safety margin force and the crack surfaces propagating in the three-dimensional domain. The cracks are plotted on the contact surfaces represented by the faces of the local Voronoi decomposition. The crack
surface is coloured in red, while the edges of the relevant Voronoi faces are represented by green tubes. The transparent faces of cracks help to explore the overlapping surfaces. It is worth noting that the numerical results of the lattice-based DEM at the micro-level are scattered because of the randomly distributed material properties of the individual particles. It is well-known that fractures can be of stochastic nature and have scattered defects at the micro-level [34]. The visualization confirms that the crack surface is continuous only at the location of the initial defect. The holes and disjoint pieces of the surfaces appearing due to the scattered nature of fractures can be observed in the upper part of the cracked region. The visualization of cracks on the contact surfaces facilitates the analysis of the structure and topological connectivity of crack surfaces as well as the identification of holes.

Figure 9: Visualization of the crack surfaces and the safety margin force.

The safety margin force is visualized on the lattice connections by using coloured tubes (Figure 9). The relative values of the safety margin force are computed according to the formula (6) and expressed in percent. Finally, the values are thresholded to show the lattice connections that are the closest to the breakage. The colour bar provides users with quantitative information, which accurately indicates the areas approached by the fracture.

6.2 The performance study

The benchmark tests were carried out on the HP xw4600 workstation (C1) and the personal computer (C2) to validate the computational performance of the visualization technique based on the local Voronoi decomposition. Hardware
The characteristics of the first computer (HP xw4600 named C1) are listed below: Intel® Core2Quad Q9450 2.66 GHz CPU, 8 GB DDR2 800 MHz RAM, 2x250 GB HDD, Nvidia GeForce GTX560 Ti GPU (384 CUDA cores, 1645 MHz processor clock, 822 MHz graphics clock, 1 GB GDDR5, 128 GB/sec memory bandwidth).

The hardware characteristics of the second computer (the personal computer named C2) are listed below: Intel® Core i7-3770 3.40 GHz CPU, 2x1 TB HDD, 16 GB DDR3 1600 MHz RAM and Nvidia GeForce GTX 660 Ti GPU (1344 CUDA cores, 915 MHz base clock, 980 MHz boost clock, 2 GB GDDR5, 144.2 GB/sec memory bandwidth).

Figure 10 shows the contribution of the execution time of the various visualization procedures used in performing the 3D benchmark. The columns C1_F and C2_F represent visualization of the first time step of the simulations on the computer C1 and the computer C2, respectively. The curves C1_L and C2_L represent visualization of the last time step, containing a large number of broken connections, on the computer C1 and the computer C2, respectively. The charts compare the contribution of the data reader (Reader), preparation of data structures (Prepare), generation of Voronoi decomposition (Voronoi) and rendering (Render) to the total visualization time. Mapping was not considered because it took a very short time equal approximately to 0.00005 s.

It can be observed that the impact of the data reader and rendering to the visualization time is relatively small. In the visualization of the first time step, the reader took 7.7% and 6.3% of the total benchmark time measured on the computers C1 and C2, respectively. In the case of other time steps, the time percentage for data reading decreased and did not exceed 2.5% of the total time. The preparation of data structures also took a considerable time in the case of the first time step, but later it did not exceed 6.0% on both computers. On the contrary, the longest time, consumed by the rendering, made 5.5% and 6.8% of the execution time in visualizing the last time step on the computers C1 and C2, respectively. In the case of the last time step, the Voronoi filter took the largest amount of time, because it generated the largest local decomposition. The generation of the local Voronoi decompositions took 86.6% and 85.1% of the execution time on the computers C1 and C2, respectively. The application of the hardware based on Intel® Core i7-3770 CPU reduced the time consumed by generating the local decomposition, by 45.5%.
Figure 11 presents the quantitative comparison of the execution time of filters in generating the local Voronoi decompositions and the global Voronoi diagrams on different hardware. Figure 11a shows the dependency of the execution time on the visualized time step of the 2D benchmark, while Figure 11b presents this dependency for the 3D benchmark. The curves C1_L and C2_L represent the generation of the local Voronoi decompositions on the computers C1 and C2, respectively, while the curves C1_G and C2_G represent the generation of the global Voronoi diagrams. Voro++ library [30] was employed to generate the global Voronoi diagrams for the sake of quantitative comparison.

The obtained results show that the generation of the local Voronoi decompositions was performed much faster than the construction of the global Voronoi diagrams. In the last time step of the 3D benchmark, the number of the broken connections made 4.3% of all lattice connections, while the local decompositions had 15.5% of faces of the global Voronoi diagram. Thus, the generation of the local Voronoi decomposition took 19.1% of the time required for generating the global Voronoi diagram on the computer C2. In the case of 2D benchmark, containing 4.0% broken lattice connections, the local Voronoi filter took only 7.8% of time consumed by the global Voronoi filter. The Voronoi decompositions were generated on CPU, therefore, the application of a more powerful Intel® Core i7 family processor reduced the filter execution time by 42.4% and 44.6%, generating the decompositions in the last time step in 2D and 3D, respectively.

Figure 12: The number of broken connections, generated faces and inconsistencies.
The time consumed by the global Voronoi filter does not depend on the number of the broken connections, remaining nearly constant. However, the time consumed for generating the local decompositions depends on the number of the broken lattice connections and the area of the fractured region. Figure 12 demonstrates the time variation of the number of broken connections (the curve Broken), the number of generated faces (the curve Decomp) and the number of inconsistencies (the curve Incons) in the case of the 3D dataset. Rapid changes in the number of broken connections can be observed between the time steps 80 and 100 in the case of both datasets. A considerable increase in the generated faces of the Voronoi cells was also observed in this interval. The number of the inconsistencies between the lattice connections and the faces of the local Voronoi decomposition was negligibly small compared to the number of generated faces. The number of the inconsistencies made 0.28% and 0.26% of the number of lattice connections, which was equal to the number of faces of the global Voronoi diagram, in the cases of the 2D and 3D datasets, respectively. The obtained results show that the local Voronoi decompositions are well suited for visualization of propagating cracks because the local algorithm can be very efficient for the initial stages of complex fracture problems.

6.3 Parallel speed-up

The measurements of parallel speed-up were performed on the grid site ce2.grid.vgtu.lt based on gLite middleware. The cluster, based on the multi-core architecture, was collected from HP Compaq dc7900 multi-core computers equipped with GPUs. The visualization benchmark was performed on the same nodes as those used for parallel computations in order to avoid the time consuming data transfer from the storage element.

![Parallel speed-up of visualization](image)

Figure 13 shows parallel performance of the visualization software. The curve Vis represents the visualization performed by a process per node. The special curve Ideal shows the ideal speed-up. The sufficient speed-up of visualization is achieved, employing a small number of processes. However, the scalability is limited by poor
load balancing of the employed domain decomposition. The domain decomposition used for DEM computations significantly differs from the partitioning required for efficient parallel visualization. The proper load balancing presents a great challenge because the highly changing visualization load is concentrated in a relatively small region. Moreover, the dynamic repartitioning is hardly applicable because of the disadvantageous communication to visualization ratios.

7 Conclusions

The paper presents the local Voronoi decomposition-based technique and distributed software for visualization of crack geometry on the contact surfaces between the neighbouring particles. The performed visualization benchmark based on the elastic solid problem demonstrated the ability of the proposed technique and the developed software to visualize crack propagation and fracture formation modelled in monodispersed particulate media. The performance analysis revealed that generation of the local Voronoi decompositions consumed 72.2% and 85.1% of the total visualization benchmark time in the case of the last time step of the 2D and 3D datasets, respectively. The quantitative comparison showed that the distributed software, based on the local decompositions, significantly outperformed the code based on the global Voronoi diagrams. In visualizing the last time step of the 3D dataset, the generation of the local Voronoi decomposition, containing 15.5% faces of the global Voronoi diagram, took 19.1% of the time consumed by the global Voronoi filter. The number of the inconsistencies between the lattice connections and the faces of the local Voronoi decomposition did not exceed 0.26% of the number of the lattice connections. The sufficiently high speed-up of parallel visualization was achieved by employing a small number of processes. However, the scalability and the perfect load balancing are limited because of the highly changing visualization load, which is concentrated in the relatively small fractured area occupied by the propagating crack.

Acknowledgement

The research is supported by the Ministry of Education and Science of the Republic of Lithuania through the VP1-3.1-ŠMM-08-K program, project “Research and development of technologies for virtualization, visualization and security e-services” (VP1-3.1-ŠMM-08-K-01-012).

References


