

Data model for mining

Technical Field

[0001] This disclosure relates to managing data for mining operations.

Background

[0002] Most mining operations rely on block models and a number of software solutions are available that enable geologists and miners to build, review and use a block model. The block model comprises a number of blocks and each block has a uniform property, such as ore grade or mineral content. The block parameters are calculated based on exploration drilling or other measurements. Blocks without measurements are estimated through interpolation.

[0003] One existing block model is the Datamine block model. This block model is stored in a file with a *.dm suffix. Datamine files are random access files stored as flat tables without any implied hierarchic or network relationships. The model structure is defined in a “model prototype” file and the spatial context of each block is stored as part of the record for each block using implicit positioning, which saves both storage space and processing time. This is done using the IJK indexing code, allowing rapid access by the computer program to any part of the model.

[0004] Some mathematics relating to the IJK code are:

$$IJK = NZ \times NY \times I + NZ \times J + K$$

The IJK can also be determined from the model coordinates system:

$$I = \text{ROUND}[(X_c - X_{\text{ParentINC}}/2)/X_{\text{ParentINC}}] * X_{\text{ParentINC}} - X_{\text{mORIG}}/X_{\text{ParentINC}}$$

$$J = \text{ROUND}[(Y_c - Y_{\text{ParentINC}}/2)/Y_{\text{ParentINC}}] * Y_{\text{ParentINC}} - Y_{\text{mORIG}}/Y_{\text{ParentINC}}$$

$$K = \text{ROUND}[(Z_c - Z_{\text{ParentINC}}/2)/Z_{\text{ParentINC}}] * Z_{\text{ParentINC}} - Z_{\text{mORIG}}/Z_{\text{ParentINC}}$$

[0005] Where XParentINC, YParentINC and ZParentINC are the X, Y and Z sizes of the Parent Blocks (to any subcells).

[0006] Other block model formats are Surpac models (*.mdl or *.fbm) and Vulcan block models (*.bmf). There may also be a block definition file (*.bdf).

[0007] A main concern to mining operations and their block models is how to populate the blocks based on assayed samples (such as exploration drilling). This is essentially a sampling of grades and interpolation/extrapolation into blocks. In this aspect, kriging plays an important role, which uses the spatial correlation between sample points and determines the weighting of the sample points for a prediction point. This process is similar to a spatial-based regression method to calculate an optimal weighting.

[0008] While existing block models serve well for mine planning, there are limitations to these models in other applications.

[0009] Any discussion of documents, acts, materials, devices, articles or the like which has been included in the present specification is not to be taken as an admission that any or all of these matters form part of the prior art base or were common general knowledge in the field relevant to the present disclosure as it existed before the priority date of each of the appended claims.

[0010] Throughout this specification the word "comprise", or variations such as "comprises" or "comprising", will be understood to imply the inclusion of a stated element, integer or step, or group of elements, integers or steps, but not the exclusion of any other element, integer or step, or group of elements, integers or steps.

Summary

[0011] Disclosed herein is a mining voxel net model that provides improved access to the voxel data and does not rely on an isolated data container, such as a file in which the voxel data is stored. Instead, the voxel data is accessible over a communication network and the voxel data can be retrieved voxel-wise on-demand. Such a data

structure supports global use of voxel data, which becomes useful in big data applications, such as machine learning, as well as underground navigation.

[0012] There is provided a method for managing data by an agent located within a mining operation, the data being stored as voxel data on a voxel net server, the method comprising:

- processing user input from a user controlling the agent within the mining operation;

- receiving from the agent a request for voxel data associated with one or more voxels, the one or more voxels being a subset of voxels stored on the voxel net server, wherein each of the one or more voxels is identified based on connections with voxels of previous requests;

- querying a database representing the voxel net for the one or more voxels to retrieve associated voxel data based on the connections;

- returning the voxel data to the agent; and

- displaying the voxel data on a user device to the user.

[0013] The agent may be mobile; the request may comprise location data indicative of a current location of the agent; and querying the database may be based on the location data.

[0014] The agent may have a viewing direction; the request may comprise direction data indicative of the viewing direction of the agent; and querying the database may be based on the direction data.

[0015] The method may further comprise raytracing based on the viewing direction to determine the one or more voxels to be queried from the database.

[0016] The agent may be a real world vehicle. The agent may be a virtual reality agent. The agent may be controlled by a user navigating the virtual reality.

[0017] The database may be distributed. The database may comprise a distributed hash table.

[0018] The database may be partitioned so as to provide access to smaller data sets and can thus reduce the time taken to retrieve the request.

[0019] The agent may comprise sensors to capture properties of material in the mine; the method may comprise receiving measurement data indicative of capture properties of the mine and identification data identifying one or more voxels of the voxel model, and storing the measurement data as voxel data.

[0020] The agent may host a section of the database that stores the measurement data captured by the agent. A local database server may store the measurement data captured by the agent. The local database server may perform operations on the measurement data captured by the agent.

[0021] The method may further comprise performing analytics on the voxel data across multiple mines.

[0022] A system for voxel data storage in a mining operation, the system comprising:
an agent located within the mining operation;
a voxel net server to store the voxel data and configured to:
 process user input from a user controlling the agent within the mining operation;
 receive from the agent a request for voxel data associated with one or more voxels, the one or more voxels being a subset of voxels stored on the voxel net server, wherein each of the one or more voxels is identified based on connections with voxels of previous requests;
 query a database, representing the voxel net for the one or more voxels to retrieve associated voxel data based on the connection; and
 return the voxel data to the agent, wherein

the system further comprises a user device to display the voxel data to the user.

[0023] There is further provided a method for interrogating an interconnected voxel infrastructure database to return model output data to a user, the method comprises:

receiving from an agent located within a mining operation, data to be integrated with a voxel infrastructure;
integrating the data with the voxel infrastructure to generate an interconnected voxel infrastructure database stored on a server;
partitioning the voxel infrastructure database
requesting modelling data associated with one or more of the partitions, the one or more partitions being a subset of the integrated data;
querying the interconnected voxel infrastructure database for one or more output data so as to retrieve associated data based on the interconnections; and
returning the output modelling data to a user.

Brief Description of Drawings

[0024] An example will now be described with reference to the following drawings:

[0025] Fig. 1 illustrates a system for voxel data storage in a mining operation.

[0026] Fig. 2 illustrates a quantisation of the Earth into a unified voxel segmentation.

[0027] Fig. 3 illustrates a method for accessing voxel data.

[0028] Fig. 4 illustrates an example of voxel selection

Description of Embodiments

[0029] Fig. 1 illustrates a system 100 for voxel data storage in a mining operation. The mining operation comprises a mine floor 101 and a mine wall 102 and a mine roof 103, noting that system 100 may be used in vastly different mining environments.

System 100 comprises an agent 110 located within the mining operation. In this example, the agent is an unmanned aerial vehicle (a drone) but may be a different agent, such as a land-based vehicle moving on wheels, tracks or legs, or other type of agent. The agent may also be an augmented reality device, such as Google Glass, worn by a human user. System 100 also comprises a voxel net server 111. The voxel net server 111 stores voxel data. Fig. 1 also shows a voxel net 112 but it is noted that the voxel net is not a physical object in the real-world (as shown in Fig. 1) but a virtual model stored in voxel net server 111. It is noted that the voxels of voxel net 112 are also referred to as “blocks” herein and the voxel net 112 is also referred to as a “block model”.

[0030] Fig. 2 illustrates a segmentation of the Earth into voxels of the voxel net 112. In this sense, the entire Earth is partitioned into voxels (or blocks) that are accessible using a unified access schema. This means the voxel net is a global model. While Fig. 2 only shows a surface segmentation into voxels, it is to be understood that the voxels extend into the inner volume of the Earth as well as into the space around the Earth. In one example, the default size of each voxel is 1m x 1m x 1m but the size can have different values, may be variable and may be different for each voxel based on the required spatial resolution.

[0031] While reference may be made to ‘blocks’ herein, this is understood to have a broad meaning. In that sense, ‘blocks’ can be also referred to as voxels and a block model as a voxel net due to its internal data connectivity, its interconnection and the way each voxel is identified based on connections with blocks of previous requests.

[0032] Each voxel may have a number of attributes, including longitude, latitude and altitude of the centre point that voxel. Each voxel may further have as its attribute, geological information, such as ore grade or mineral concentration. A further attribute may relate to neighbours of each voxel, such that the voxels can be queried sequentially in a computationally efficient manner without iterating over all voxels to arrive at a given voxel.

[0033] Fig. 3 illustrates a method 300 as performed by voxel net server 111. In this sense, voxel net server 111 is configured to perform the steps of method 300. For example, voxel net server 111 may have software code stored on non-transitory computer readable medium that causes the voxel net server 111 to perform the steps of method 300.

[0034] In summary, method 300 is an on-demand data provisioning method for voxel data in a mining operation. This means that the voxel net does not need to be downloaded in full but can be provided to agent 110 or multiple agents at the same time on-demand. This allows the storage of a large, complex and detailed voxel net while at the same time, enabling low latency access to the voxel data. Further, voxels can be updated and the updated voxel data disseminated to all subscribed agents on a voxel by voxel basis. In other words, if a single voxel is updated by measurement data provided by a first agent, that single voxel can be provided to a second agent as an update or a newly requested voxel. This means two agents can be measuring material properties, such as two drills operating at the same time, or two hyperspectral cameras scanning a nearby surface of rock at the same time, while both agents receive real-time updates from the other agent on their voxel net. In addition, the control centre of the mining operation represents another agent, and the voxel net available at the control centre is also updated in real time as the measurements are received.

[0035] Returning to Fig. 3, voxel net server 111 receives 301, from the agent 110, a request for voxel data associated with one or more voxels. For example, the agent 110 provides an identifier or a location or another reference, such as a neighbourhood relationship to the voxel net server 111. Typically, the requested voxels are a subset of the voxels stored on the voxel net server, which means the request is not for the entire voxel net, since that would be impractically large and complex, and not absolutely required by the agent.

[0036] In response to receiving the request, voxel net server 111 queries 302 a database 112 for the one or more voxels to retrieve associated voxel data. For example, voxel net server 111 may search for an identifier or for a latitude, longitude, altitude

(3D) coordinate. It is apparent now that a database query for a 3D coordinate may be computationally expensive since the entire database is searched for the three values in three respective columns. Therefore, it would be advantageous to have an addressing scheme that allows direct access to database elements without queries for values. For example, if each voxel stores the identifier, such as primary key, of its neighbours east/west/north/south/above/below, the database can be queried directly for that primary key. In that sense, the identifiers of neighbouring voxels represent connections between the voxels. Then, each voxel is identified based on connections with neighbouring voxels. In many applications, a query for a particular voxel would be subsequent of a query for a previous voxel and in most cases, the two queried voxels are neighbours of each other. Therefore, due to the connections, each voxel is identified based on connections with voxels of previous requests by retrieving neighbouring voxels primary keys from the previously requested voxel.

[0037] Each entry in the database relates to one voxel and the entries may be sorted by primary key, which makes a query for a particular voxel computationally efficient. This results in low latency of database queries and rapid retrieval of neighbouring voxels.

[0038] Finally, voxel net server 111 returns the voxel data to the agent. For example, the voxel data returned to the agent 111 comprises ore grade, mineral concentration etc.

[0039] As shown in Fig. 1, agent 110 is mobile, which means the agent 110 can move around the mining operation. This may be autonomous movement, remote controlled movement or directly driven, such as by an operator moving with the agent 110, such as a mining vehicle. In those examples, the request from the agent 110 to the voxel net server 111 may comprise location data indicative of a current location of the agent. For example, the request may include 3D location information, such as latitude, longitude and altitude. Server 111 then queries the database based on the location data. For example, server 111 may retrieve all voxels that are in the vicinity of the agent 110, such as all voxels that are within a given distance from the agent 110, such as 1 km. The agent 110 may also specify that distance for which the voxels are requested.

[0040] In yet another example, the location data may comprise inertial data that indicates the relative movement of the agent 110, which may be more readily available in underground mining operations. This inertial data is effectively an offset to the previous location, so server 111 can simply query for all voxels that are available at that offset from the previous location. In this sense, the inertial data directly maps to the neighbours within the voxel net, such that each update on the inertial data leads to a retrieval of one or more neighbours of the most recently queried voxels.

[0041] In a further example, the agent has a viewing direction. This may be the case where the agent 110 comprises a directional sensor, such as a camera or a LIDAR instrument. In that case, the request may comprise direction data indicative of the viewing direction of the agent. Server 111 then queries the database based on the direction data. For example, server 111 determines based on the current location of the agent 110 and the direction data, which voxels are “in view” of the agent 110. Server 111 may select the voxels that are in view based on a viewing angle, such that multiple voxels are selected.

[0042] Fig. 4 illustrates an example of voxel selection, where agent 110 has a viewing direction 401 and a 2D angle that defines a cone 402 or beam. Server 111 selects voxels that are within the cone 402 around the viewing direction 401. In Fig. 4, the white voxels, which are outside the cone, such as voxel 403, are not selected while the shaded voxels, which are at least partly inside the cone 402, such as voxel 404, are selected, queried and sent back to the agent 110.

[0043] One way of realising this determination is a raycasting technique where the server 111 calculates the trace of a ray of light cast from the agent 110 and determines where that ray of light hits a surface of the mine in the voxel net. Server 111 then returns the ‘lit’ voxels. At this point it is noted that empty space (void) in the real world may be represented by the absence of voxels in the voxel net or by voxels being annotated as ‘void’ or ‘air’.

[0044] The advantage is that agent 110 now has real-time access to the voxels that are in view, which means agent 110 can process the data from the voxel net 112 at the same time as the sensor data. For example, agent 110 can use the data from the voxel net 112 to navigate through the mine. More particularly, agent 110 can identify features in its field of view detected by the sensor, such as 3D shape features in a LIDAR point cloud or image features in an RGB or infrared image, and match those features against the features in the voxel net. In that sense, the features in the voxel net may be detected by the agent 110 based on the received voxel data or by the server 111. In yet another option, the features may be stored in the voxel net as voxel attributes. For example, each voxel may be labelled as belonging to a particular feature. This way, registration of the image or point cloud from the agent 110 is greatly facilitated because the features in the image or point cloud are only matched against features in voxels that are “in view” and not against the entire voxel net. This is particularly useful for simultaneous localisation and mapping (SLAM) of multiple agents at the same time, where detected features are shared between the agents in real time.

[0045] While the above examples relate to real-world agents, the agent may equally be a virtual reality agent. In this sense, the agent is an object in a virtual reality that is defined by the voxel net. The agent may be an avatar of a user or simply a point within the virtual reality that represents the location of the user within the virtual reality. The user may control a viewing direction of the agent within the virtual reality and the voxel net server 111 may then provide the voxel data of voxels that are in view. The user may also navigate the agent through the virtual reality by moving the agent and rotating the agent.

[0046] While server 111 is shown as a single component, the database 112 may be distributed across multiple locations. For example, each mine site may have a local server that stores and processes the voxel data locally. However, each of these servers also has access to all other servers so that mine data can be accessed globally through a unified addressing scheme.

[0047] More particularly, the database 112 may comprises a distributed hash table. This means each voxel is addressable by its unique hash and each server manages its own hash table. This way, additional mine sites can be added seamlessly to the network of hosting servers. There are a number of distributed file systems and peer-to-peer network protocols for distributed files systems, including Amazon S3/Aurora, Hadoop Distributed File System or TiDB by PingCAP Inc.. Other examples include a distributed database or data grid comprising a middleware layer that provides services and applications necessary for efficient management of the voxel data.

[0048] While the previous examples provide a method for accessing data stored on database 112, agent110 may also update the data of the voxel net. More particularly, the agent 110 may comprise sensors to capture properties of material in the mine. These sensors may include a RGB camera, infrared/thermal camera, LIDAR range finder, hyperspectral camera, radar, chemical (composition) electrical (conductivity), X-ray fluorescence, gamma ray and other sensors.

[0049] Agent 110 captures the sensor data and associates the sensor data with a particular voxel of the model. For example, agent 110 stores the sensor data together with a location and/or viewing direction of the sensor a metadata. Agent 110 then sends the sensor data and metadata to the server 111.

[0050] In turn, the server 111 receives the measurement data, which is indicative of captured properties of the mine and identification data (e.g., metadata) identifying one or more voxels of the voxel net. Consequently, server 111 stores the measurement data as voxel data. In this sense, server 111 updates the voxel data to incorporate the sensor data. The updated voxel data may also be an interpolation or extrapolation of the sensor data, which is useful where the location of the sensor data capture is not exactly identical to the location of the voxel in the voxel net.

[0051] While agent 110 and server 111 are shown as separate components, it is equally possible that the database 112 is integrated with the agent, such that the agent hosts a section of the database that stores the measurement data captured by the agent.

In this sense, agent 110 is a node in the distributed database that stores the global voxel net and other devices can access the data directly from agent 110.

[0052] A local database server may therefore be any device, entity or component where the voxel data is stored. Therefore, the local database server can perform operations on the measurement data captured by the agent and the voxel data. For example, local data base server 111 may perform the interpolation or extrapolation and machine learning or data extraction locally. This has the advantage that the extracted information and/or features are typically significantly smaller, which means less bandwidth is required between the mine site and the control centre, for example.

[0053] Since the voxel data is stored in a globally accessible format, it is possible to perform analytics on the voxel data across multiple mine sites. This is a significant advantage because the accuracy of analytics outcomes increases with the number of samples and is therefore better for multiple mine sites.

[0054] It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the above-described embodiments, without departing from the broad general scope of the present disclosure. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive.