

# Interactive Visualization through Sparse Multidimensional Data Spaces

Maxim Bogdan, Dorian Gorgan

Technical University Cluj-Napoca  
Memorandumului 28, Cluj-Napoca

*E-mail: bogdy.maxim94@yahoo.com, dorian.gorgan@cs.utcluj.ro*

**Abstract.** The acquisition, storing and processing of massive multidimensional spatial data requires data models able to support search, access, and fetch particular data from sparse data value space. In order to understand the meaning of complex data, the visual analysis has to integrate the complex algorithmic processing over high performance processing infrastructures with interactive visual navigation throughout the multidimensional data space. The visualization throughout the sparse value spaces is not a trivial task. This paper presents and experiments a solution for sparse multidimensional spatial data models, and two new user interaction techniques to navigate and visualize data model. The paper highlight the main issues of visualization by intersection plane, and projection plane by deeper visual penetration within the data value space.

**Keywords:** Visual analytics; sparse data; multidimensional spatial data; Rasdaman technology; visualization technique.

## 1. Introduction

Multitude types of sensors (e.g. environment, satellite, etc.) and many other data sources from the domains of traffic and transportation, industry, agriculture, health, social life, and smart homes, produce huge quantities of data (Figure 13). While the capacity to collect and store new data rapidly grows, the ability to analyze these data volumes increases at much lower rates (Keim et al., 2008). The analysis of these big data leads to new challenges in the data mining process, since analysts, decision makers, engineers, or emergency response teams depend on information hidden in the data. The visual analytics focuses on handling these massive, heterogeneous, and dynamic volumes of information by integrating human analytical reasoning by means of visual representations and interaction techniques in the analysis process.

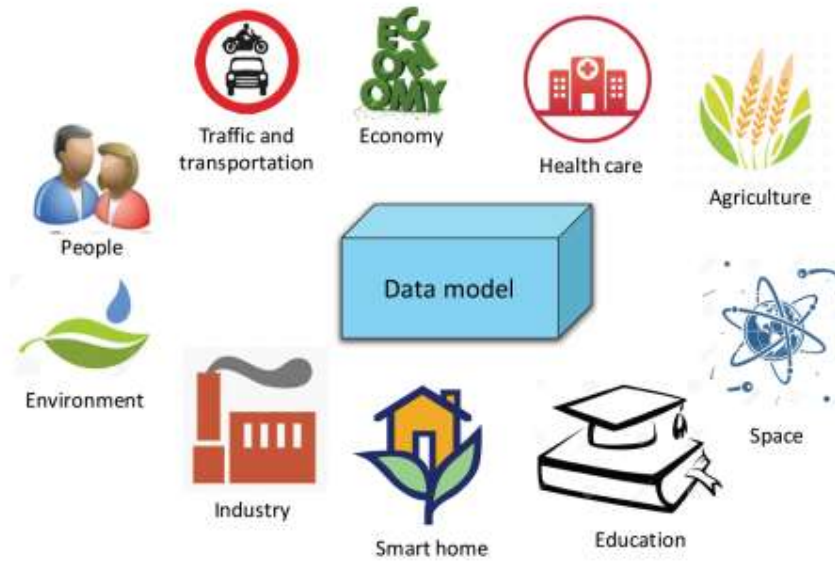


Figure 13. Multidimensional data sources.

The heterogeneous data are supplied by various sources such as computed data, sensor networks, simulated data, measured data in the field, or created and edited data. To store and process these data require very expensive storage and high performance processing resources. The main idea is not to just store data but analyze and valorize them. This a big challenge of transforming data in knowledge, as reduced and encoded form, and publishing data, as information (Gorgan, 2017). The automatic data mining uses performant and adaptive algorithms that could be executed quite fast on the actual computing infrastructures (e.g. cloud, GPU clusters, grid, HPC architectures). Nevertheless, the creation, development and implementation of these algorithms needs creativity of the specialists by using flexible and performant tools.

One of such approach of exploring the data value space is the interactive navigation and visual analysis of the multidimensional data spaces. The specialists can identify by interactive visual navigation and analysis the particular values, critical regions, tendencies in static and dynamic data value space.

This paper presents the results of scientific research on exploring the

interactive visual analysis on massive and multidimensional data, by using the high performance computation infrastructures. The paper highlights the advantages and the issues of such interactive navigation and visualization techniques.

The paper is structured by six sections. This section is Introduction and describes the domain, the challenge, and the objectives of the paper. The next section presents the related works reported in the scientific literature. The third section describes the Rasdaman data model. The fourth section describes the interactive navigation and visualization techniques. The paper follows by the experimental validation of the proposed visual analysis techniques. The last section concludes on the achievements and sketches the future works.

## 2. Related works

The visual analytics is an important subject in the scientific literature in the last decade. (Keim et al., 2008) provides an overview of visual analytics, its scope and concepts, addresses the most important research challenges and presents use cases from a wide variety of application scenarios.

Visualizations of multidimensional data are quite complex, and the interactive techniques support users to explore and select the presentation for better understanding (Siirtola, 2007). Siirtola presents three conceptually different visualization techniques for multidimensional data: reorderable matrix visualization, enhancements to the user interface of parallel coordinate browsers, and a new visualization technique based on data glyphs and small multiple visualizations.

Sometimes multivariate data are not available as sets of equally multivariate tuples, but only as sets of projections into subspaces spanned by subsets of these attributes. (Xie et al., 2017) proposes a visual analytics approach that helps the user understand the patterns and the uncertainties. Constraints that narrow the solution space can then be added by the user interactively during the iterative exploration.

Seo and Shneiderman offer a set of principles and a novel rank-by-feature framework that could enable users to better understand distributions in one (1D) or two dimensions (2D), and then discover relationships, clusters, gaps, outliers, and other features. Users of our framework can view graphical presentations (histograms, boxplots, and scatterplots), and then

choose a feature detection criterion to rank 1D or 2D axis-parallel projections. By combining information visualization techniques (overview, coordination, and dynamic query) with summaries and statistical methods users can systematically examine the most important 1D and 2D axis-parallel projections.

In interactive data analysis processes, the dialogue between the human and the computer has to be not interrupted by slow computational mechanisms. Such integrated computations could be, for instance to reduce the dimensionality of the data or to perform clustering. Turkay et al. experiment in (Turkay et al., 2017) the integration of such progressive methods within interactive analysis processes that involve high-dimensional data.

In (Catana and Gorgan, 2015) and (Gorgan et al., 2015) the authors presents the visual analytics techniques in exploring the solutions for strategic games.

(Pahins et al., 2017) proposes the notion of Hashedcubes, a data structure that enables real-time visual exploration of large datasets that improves the state of the art by virtue of its low memory requirements, low query latencies, and implementation simplicity. In some instances, Hashedcubes notably requires two orders of magnitude less space than recent data cube visualization proposals.

### **3. Spatial Massive Data**

The multidimensional space of data (Figure 14) is such a cloud that the user needs to analyze, process and discover their meaning. The data mining process integrated within the visualization loop needs optimal organization of the data model, rather than the classical relational database systems.

One such a solution of organizing the multidimensional spatial data is Rasdaman (Baumann, 1994).

#### **3.1 Rasdaman Data Model**

Rasdaman is a domain-neutral array database system. It organizes data into a multidimensional array of homogenous, atomic elements, usually pixels or voxels. It has no limitation on the number of dimensions. It can serve, for example, for 1D measurement data, 2D images (like: satellite, tomographic, typographic, cartographic, infrared images), 3D x/y/t time series and x/y/z

exploration space, 4D ocean and climate data, and can even exceed spatio-temporal dimensions (Baumann, 1994). The array managed by Rasdaman is Big Data, often single objects ranging into Terabyte and soon, into Petabyte sizes. Usually, this size is achieved by Earth and space observations.

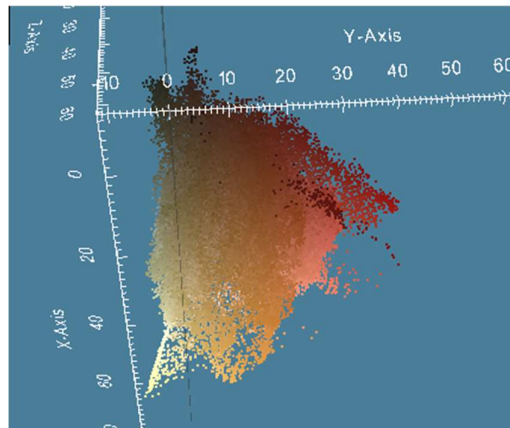


Figure 14. Value space of the multidimensional data

The main concept on which Rasdaman arrays are based on, is called tiling. When an array is inserted into Rasdaman, it is first subdivided into smaller sub-arrays, called tiles, and then the tiles are being stored in a conventional database management system. Internally, Rasdaman uses PostgreSQL as its primary database management system. The tile is the main storage unit. Before inserting the tiles into the database, the system assigns them an index. This index is useful for the later operations which will be executed on them. Different tiling strategies (like: regular tiling, caveat, aligned, directional, area of interest, statistic and user generated) can be used when inserting data. Rasdaman provides a way to specify the index type used for tiling. It provides a general-purpose index, a directory index specific for interval objects and a special index which works with regular tiling. In its core, Rasdaman stores both the bulk array and the auxiliary geo-semantics for the real-world mapping of the arrays (Campalani et al., 2014), (Baumann et al., 2016).

Two special ways of storing the data in Rasdaman are `RectifiedGridCoverage` and `ReferenceableGridCoverage`. The term "coverage" refers to any data representation that assigns the values directly

to the spatial position. A coverage is a function which from a spatial, temporal or spatiotemporal domain to an attribute range.

The `RectifiedGridCoverage` is a coverage which geometry is represented by a rectified grid. A rectified grid is a grid for which the function which transforms the grid coordinates is affine.

The `ReferenceableGridCoverage` is a coverage of which geometry is represented by a referenceable grid. A referenceable grid is a grid for which the function that transforms the grid coordinates are not affine (Baumann et al., 2016).

### 3.2 Multidimensional data model

As the first important step of data organization, Rasdaman permits defining the data type which is going to be used. Each instance within the database is described by its type. A Rasdaman schema contains three categories: cell type definitions (which can be base types like floats or struct types like `r/g/b`), multiple dimensional data type definition (which can be arrays over a base type and with some spatial domain) and collection type definitions which describe sets over some multiple dimensional data. Rasdaman supports adding new data types at runtime (Baumann, 2013).

The basic unit of storage in Rasdaman is the cell. It may hold from a basic value, like integer, or float, to a much more advanced data structure, like `r/g/b` structs and anything that a user may think of. In general, any struct which is available in C/C++ can also be used in Rasdaman type definition, with the except of pointers and arrays. Nested structures are also supported.

By grouping multiple cells together, a more advanced data model is constructed. This is the multidimensional data array. Rasdaman allows the user to specify the cell type and the domain for the array. The domain can vary, and even boundless arrays are allowed. It is discrete and only integer values are allowed (Figure 15). Rasdaman allows unlimited number of dimensions for the arrays (Campalani et al., 2014).

Over multiple typed arrays, collections are being build. Collections are data types which contains two specific columns: a system-maintained object identifier and the array itself. Using this type of data organization, Rasdaman is able to handle such data in the relational way: foreign keys in conventional tables allow to reference particular array objects (Baumann, 1999).

By considering time as one of the dimension, Rasdaman can easily take a

series of images and store them into its internal database management system. The user defines the coverage for the collection, and the system is inserting the images based on those properties. The user is also able to specify the time increment for the series. By using the coverage, Rasdaman can easily handle multidimensional geographical or satellite data which comes in the form of time series images (Campalani et al., 2014).

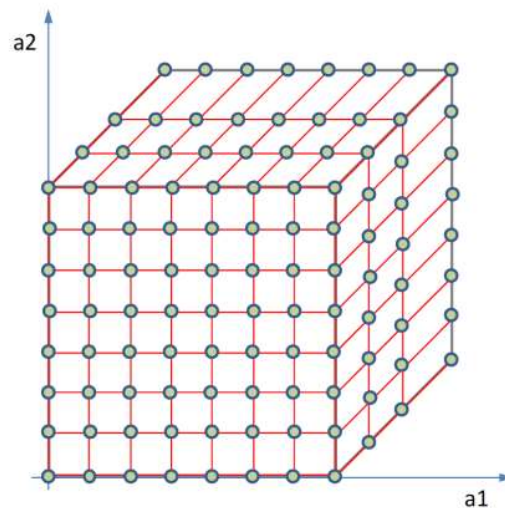


Figure 15. Multidimensional data space as a sparse volume

### 3.3 Data model interrogation

By using a relational model, the main data interrogation is being achieved using a query language. The Rasdaman query language embeds itself into the standard SQL and it is set-oriented processing. Besides the classic CRUD operations, the Rasdaman query language adds a new set of operations for the multidimensional data arrays. These operations are based on a minimal set of algebraically defined core. This set provides an array constructor (which establish a new array and fills it with values) and an array condenser (which provides scalar summary information from an array, like the sum, average, minimum, maximum, some and all cells).

Queries are parsed, optimized and executed in the Rasdaman server. Parallelism execution is also implemented. Different threads will process

different parts of the array and the final result is assembled in the main thread.

Whenever a query is processed, the tiles of the selected array are processed sequentially, giving the possibility to query data with a dimension far bigger than the server memory.

When using the coverages, the user is able to manipulate data in an efficient way. Internal queries are optimized for the coverage section, giving a boost of performance for both the classic operations, and for some special ones. The user can select the domain of the dimensions which will be processed (both time and spatial) and a specific portion of the image to be displayed. The user is able to manipulate the images by zooming in and out, rotate the images and even process the images like addition, subtraction, multiplication and even user defined functions of the images. Those queries can be executed using the array constructor query which is a general-purpose operator. For a better performance, specific queries should be used when working with the coverages. Those queries include: for operator and array condensers. By having a data model like this, the availability of the relational model is achieved. The information can be distributed in the traditional way, which ensure high replicability (Baumann, 2013).

## 4. Data visualization

### 4.1 Taxonomy of interactivity

Heer and Shneiderman present in (Heer and Shneiderman, 2012) the set of functionalities providing interactive dynamics in visual analytics:

- *Data and View specification;*
- *Visualize data by choosing visual encodings;*
- *Filter out data to focus on relevant items;*
- *Sort items to expose patterns;*
- *Derive values or models from source data;*
- *View manipulation;*
- *Select items to highlight, filter, or manipulate them;*
- *Navigate to examine high-level patterns and low-level detail;*
- *Coordinate views for linked, multidimensional exploration;*
- *Organize multiple windows and workspaces;*



- *Process and provenance;*
- *Record analysis histories for revisiting, reviewing, and sharing;*
- *Annotate patterns to document findings;*
- *Share views and annotations to enable collaboration;*
- *Guide users through analysis tasks or stories.*

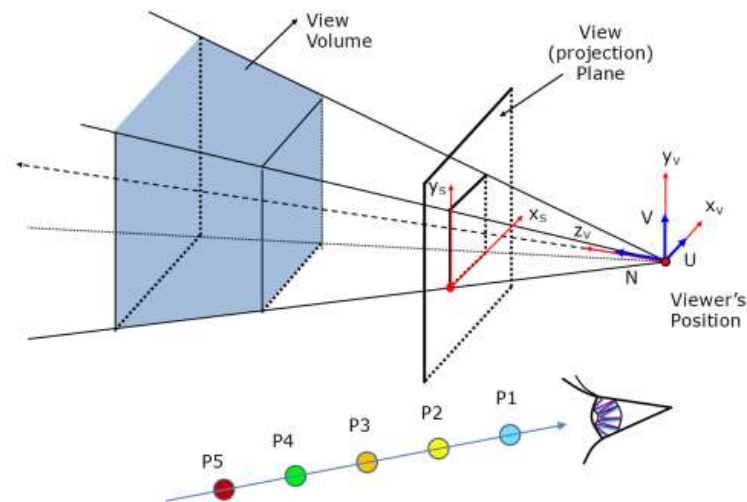


Figure 16. Viewing parameters throughout the data space

## 4.2 Data space navigation

In order to support the understanding of data by visualization the proposed system provides the user with the following operations:

- *Navigate throughout the value spaces;*
- *Update the viewing parameters (e.g. camera position, orientation, distance, zoom in and out, etc.);*
- *Adapt the transparency in order to visually penetrate the value space throughout soft values till the hard values;*
- *Identify continuous objects from sparse data;*
- *Identify critical points, convergence, tendencies, surfaces, thresholds, hard edges;*

- *Identify the limits of the homogeneous domains.*

### **4.3 Viewing system**

For the viewing system, we used the classic camera system, adapted to our data model. The main involved parameters are such as position, up vector, look vector and the frustum defined by the near and far values and orthographic projection.

The position specifies the world coordinates of the camera. The space is discrete so the position coordinates will be integers (Figure 15). The up vector orients the camera about the look direction (Figure 16). It is specified in the camera coordinates. The look vector orients the camera into the space. It is specified relative to the camera position. The near and the far values are specified as integers. They are used to cut the frustum of the view cone. Object inside the frustum will be displayed, and those outside will not. They are specified relative to the position and the look of the camera.

The projection used in the system is the orthographic one. It is a form of parallel projection in which all projection lines are orthogonal to the projection plane.

Another important parameter of our visualizing system is the visibility of the voxel. It is held in the database and it is read at processing time. The visibility is used to define which layer will be visible from the user's perspective (Gorgan, 2017).

Another parameter which has a big impact on the quality of the rendered data is the resolution of the image and the resolution of the world space. Having different resolutions (especially smaller screen resolution than the space one) affects the output of the data in a significant way. The reason of the impact is the interpolation which is being done to achieve a continuity in the image.

### **4.4 Data visualization techniques**

#### **Intersection plane technique**

The first visualization technique that we propose is the section plane technique. It consists of an intersection plane which "cuts" through the world.

This intersection plane is going to be visualized by the user. The world

space is heterogeneous and the objects are built of many layers. By using the section plane technique, the user is able to see the different layers from which the objects are being made of and identify different patterns and new data out of it. The user is able to select the three dimensions which he wants to visualize.

The section plane is defined at the camera position. The camera look direction is the normal to the plane. By moving the camera around, the user is able to inspect the different layers of the object.

The intersection plane is achieved by using the array constructor of the Rasdaman query language. Having the camera position and the normal to the plane, we compute the direction which is going to be used to make the cut on the x and z axis. We then compute the starting and ending point, based on the camera frustum. Having the starting and the ending point, we apply the Bresenham's line algorithm on the plane between those points. For every x and z points, we select all the y positions and add them to the array, as a column. After this, we display the newly created array to the user (Figure 20).

For the special cases of vertical and horizontal plane, we are using the select statement of the Rasdaman query language. The select statement lets us specify which dimension is constant and the domain of the other dimensions, and it will automatically build the new two-dimensional arrays based on our selection.

The newly created array is going to be send to the client and to be displayed there.

The main advantage of this technique is the presentation of the layers from which the objects are made of. By moving around, the user is able to identify different "extreme points" of the data, which allows into a better understanding of the composition of the object. Also, the user can identify different abnormalities which can be present. The user can also specify time as one of the dimensions. By doing so, the user may follow the abnormality and see its evolution, and, in some cases, it can even predict what it is going to happen there.

By knowing all the layers from which an object is made, the user can specify the layer's visibility. This is extremely useful in the case of projection plane technique. In that case, the user is able to see a different point of view for the abnormality and deduct more results about its origin and evolution. By specifying the visibility, the user can see different layers

of the object in a projection plane, making different combinations of layer's visibility to achieve a much more complete view over the model.

### Projection plane techniques

The second technique that we propose is the projection plane technique. This allows the user to visualize the space in a 3D manner. The user is able to select which dimensions he wants to visualize, the resolution of the image and the camera parameters. He is also able to manipulate the camera parameters during the runtime of the visualization. Using this method, the user gains a better understanding of the outside view of the object and the way the object resides in the world.

The user selects the three dimensions he wants to visualize them.

The main technique implemented here is the Ray-casting method. Given the projection method which we want to use, we are sending rays from the user's camera position into the world, based on the look vector. At runtime, we compute the frustum coordinates in the world (Hughes et al., 2014).

The rays compute the visibility of the layers as they pass on and will give the final color of the given pixel. Here, the resolution of the image is the most important parameter. Given a smaller resolution than the space, only fragments of the object will be seen. This is resolved by triangulating (Delaunay triangulation), the points returned by the ray-caster and interpolating the triangles (triangle rasterization). If the resolution is the same, no interpolation will be executed, as the rays hit all the visible point of the object. If the resolution of the image is bigger than the resolution of the space, more rays will hit the same points of the surfaces (Figure 21).

When the rays are send from the camera position to the objects, they compute the final pixel value based on its visibility and the other points' visibility. It is used the iterative visualization formula for multiple transparency layers:

$$C_0 = \alpha_1 C_1 + \alpha_2 (1 - \alpha_1) C_2 + \alpha_3 (1 - \alpha_1) (1 - \alpha_2) C_3 + \alpha_4 (1 - \alpha_1) (1 - \alpha_2) (1 - \alpha_3) C_4 + \dots$$

where  $C$  represents the layer color, and alpha represents the layer's visibility.

By doing so, different effects can be achieved. Some layers can be fully transparent, some layers can be partially transparent, and some layers can be fully opaque. This is extremely useful in the case where the user wants to

see exactly a layer, without being distracted by other. The user is able at runtime to select which layers should be visible and which one should not. The background is always a transparent layer. The ray stops whenever the next colors have no influence over the final pixel color (for example, when it hits an opaque layer).

The main advantage of this technique is the visualization of the object from an outside point of view using the visibility method. In this way, the user which has identified an anomaly in the object, is able to see the way it looks from outside and gain different information which, combined with the section plane visualizing, gives a whole new perspective over the problem. The visibility mechanism is extremely useful when the user needs to be focused on a specifically part from the object and does not want other layers to distract them. This is also useful to see what lays after an opaque object and also to see what happens if a given object is transparent, rather than opaque. Different optical tests can be developed this way. Another advantage of this method is letting the computer to search through the space, based on a given animation of the camera, and visualize different results returned by this.

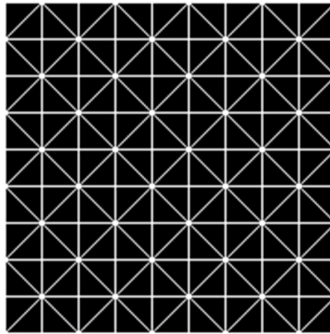


Figure 17. Regular grouper used for triangulation

Another advantage of this method is letting the computer to search through the space, based on a given animation of the camera, and visualize different results returned by this. The user is able to pause the animation at any time, and also guide the computer to follow specific paths. In this way, useless search paths (and computations) can be avoided just by the user's intervention. This automation makes it easier for the user too, as he doesn't

have to search by hand all the possibilities. His only intervention is wherever he wants to deviate the normal search path as the old one can lead to nowhere.

### Interpolation techniques

When the computation window has a different resolution relative to the display window (mainly, when the computation resolution is smaller than the display resolution), the system computes the ratio. Then it arranges the points from the computation window to the display window based on the ratio computed before. The points will be grouped based on the pattern represented by the Figure 17. This pattern will generate triangles. After that, the system will interpolate the colors in the middle of the triangle based on the linear interpolation method. Figure 18 represents the way the interpolation is computed.

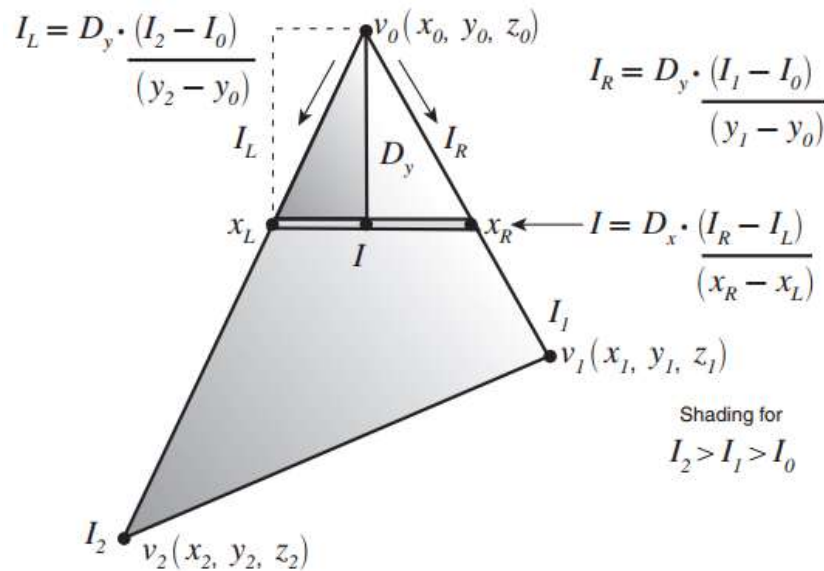


Figure 18. Representation of the scan line interpolation applied on the triangle

The system gets the color from the vertices of the triangles. Then, it will iterate from the first vertex y value to the last vertex y value. For every y value, the system computes the x value for the left and right edge. Then, it

will compute the color values for the values in between. The color is computed in a linear interpolation value. 0 represents the first color value, and 1 represents the last color value. The current value is computed based on the formula :

$$(1 - t) * c_i + t * c_f$$

where  $t$  is a value from 0 to 1,  $c_i$  is the value from the left edge of the triangle and  $c_f$  is the color from the right edge of the triangle.  $t$  value is computed as follows

$$\frac{pos_c - pos_i}{pos_f - pos_i}$$

where  $pos_c$  represents the current position on the scan line,  $pos_i$  represents the initial position on the current scan line and  $pos_f$  represents the final position on the scanline. By doing this, we bring the positions into [0,1] interval. This is used for both  $y$  and  $x$  values.

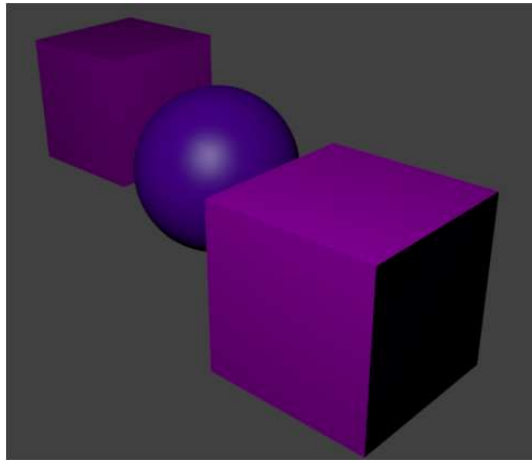


Figure 19. Scene of objects stored within the rasdaman data model

At first, we compute the slope from the second and first point and from the third and first point, we take the initial point as being the first point and for each  $y$  scan line, we adjust the values according to the slope.

The triangles have the vertices sorted after the  $y$  value, and if the  $y$  value is the same,  $x$  value is took into account. Those are window coordinates, with the top left corner being the (0, 0) coordinates, and the bottom right

corner being the (width, height) coordinates.

To make a general interpolator for triangles, the system classifies the triangle as follows :

- Top flat triangle – this means that the y coordinates of the first two vertices is the same.
- Bottom flat triangle – this means that the y coordinates of the last two vertices is the same
- Normal triangle – all the y values are different

In the case of the normal triangle, a change in the slope is needed. If the second point is on the left of the first point, then when we iterate from this point to the third, the slope is going to be different. When we arrive there, we are going to compute the new slope accordingly and then carry on with the interpolation. The same is applied in the case of a right adjustment. This generalization is useful for other kinds of triangularizations (Xie et al., 2017).

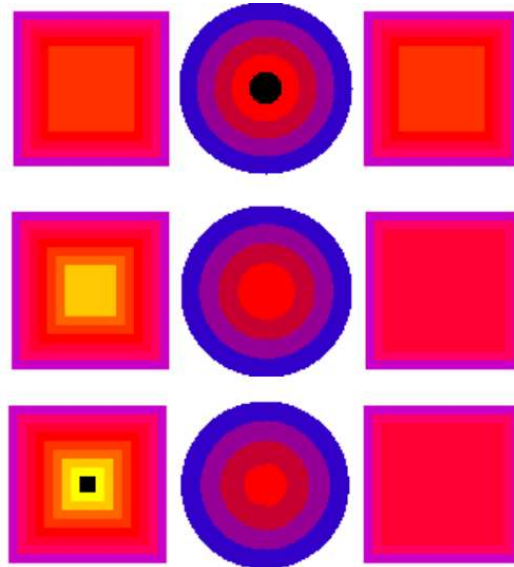


Figure 20. Interactive visualization by intersection plane: (top) vertical plane cuts through three objects - two cubes and one sphere. The objects are made from different colored layers; (middle) oblique plane; (bottom) closer oblique plane.



## 5. Experimental validation

Practical experiments have been defined and developed in order to prove the efficiency of interactive visualization of the sparse multidimensional data model.

The data model consists of spatial definition of a scene of 3D objects consisting of two cubes and one sphere (Figure 19). All objects are defined as concentric layers of various materials. The material of each layer is defined by its particular color and transparency.

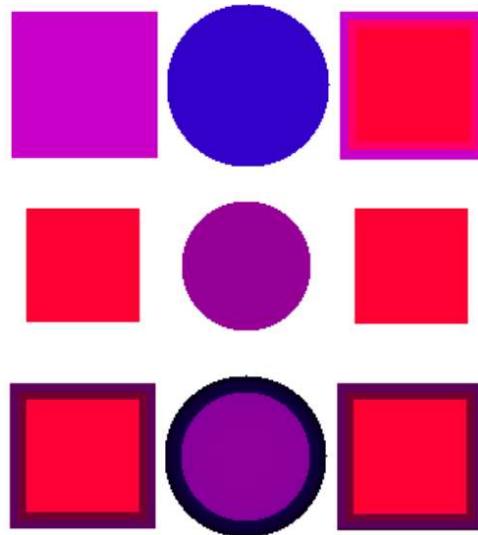


Figure 21. Interactive visualization by controlling the visual penetration: (top) outside the objects; (middle) to inner levels; (bottom) 70% is the level transparency. Two layers are transparent for the cubes, and one layer for the sphere.

The 3D scene of objects have been discretized and modeled in Rasdaman as multidimensional data with five attributes (i.e. x, y, z coordinates, color, and transparency). The discrete values generate a sparse description of the space. In order to visualize the objects, on the screen by high graphics quality at various high resolution, the system interpolates the screen space. It is not the scope of this paper to detail the discretization and interpolation of the objects. This paper concerns with interactive navigation and visualization by section and projection planes.

In such a manner the objects are viewed obviously by their exterior material. To visualize the inside objects, the systems offers two techniques: (1) by intersecting the objects through a section plane, and (2) by visually penetrating through outer layers to visualize the inner layers.

### **5.1 Intersection plane use cases**

Figure 20 presents the visualization cases by intersecting the scene of three objects by one vertical plane. In Figure 20.top the intersection plane passes by outer layers of the cubes and the middle of the sphere. Therefore, three layers are visible throughout the cubes and five layers throughout the sphere.

In Figure 20.middle the intersection plane is moved deeper in the left cube, and in In Figure 20.bottom the intersection plane passes through the center of the left cube.

By this technique are visualized the values along an intersection plane. The technique is less appropriate for volumetric visualization (e.g. cloud of values). For instance, instead of viewing by this technique the anatomical femoral bone by section, it is viewed as individualized 3D object by the projection plane technique.

### **5.2 Projection plane use cases**

Figure 21 exemplifies the visualization on the projection plane. Figure 21.top considers the outer space as a transparent space, and the three objects are viewed from outside.

Figure 21.middle presents the visualization of the inner layers of the three objects, by penetrating visually the outer layers. The visual penetration is controlled interactively by the user. The user can specify the viewing parameters, camera position, and the visibility of the outer space and object layers.

Figure 21.bottom considers 70% the level transparency for two outer layers of the cubes, and one outer layer of the sphere.

By these two visualization techniques the user can navigate throughout the value space. By setting the viewing parameters (e.g. camera position, orientation, distance, zoom in and out, etc.) the user can move around the value space, and visualize the two cubes and the sphere from any position.

By the second technique, the user can adapt the transparency in order to

visually penetrate the value space throughout soft values till the hard values. For instance, penetrate the skin, and flash of the human body, in order to visualize the bones.

By the projection plane technique, the user can identify continuous objects from sparse data. For instance, even the temperature of the atmosphere, have been provided just for some measurement positions, the system is able to visualize the same temperature spaces as contiguous sub-spaces (e.g. isothermal clouds).

By controlling the viewing penetration, the user could identify critical points, convergence, tendencies, surfaces, thresholds, and hard edges within the value space. The user could identify the limits of the homogeneous domains.

The main utility is in the cases when the value space in Rasdaman is filled in with data from the sensors, as sparse positions and various moments. The visualization techniques support the visualization of the multidimensional space as a contiguous value space.

## 6. Conclusions

The scientific research and the practical experiments have proved the high performance computation can be integrated within the user controlling visualization loop. New processing and visualization use cases, in the domains of Earth Science, Earth Observation, medicine, climate change, and hydrological models, will be analyzed and experimented in order to better integrate the processing algorithms on high performance computation infrastructures, and visualization platforms.

The research and the presentation through this paper have proved the massive data can be understood by visual analytics techniques. This paper concerned mainly with two techniques of interactive visualization, which provide user with better access to data value space: intersection plane and projection plane.

The Rasdaman technology will be experimented with different heterogeneous and sparse environmental data.

## References

- Baumann, P., (1994), Management of multidimensional discrete data, *International Journal on Very Large Data Bases* 3.4: 401- 444.

- Baumann, P., (1999), *A database array algebra for spatio-temporal data and beyond*. International Workshop on Next Generation Information Technologies and Systems. Springer Berlin Heidelberg.
- Baumann, P., (2013), Query language guide. URL: [http://rasdaman.eecs.jacobsuniversity.de/trac/rasdaman/browser/manuals\\_and\\_examples/manuals/pdf/ql-guide.pdf](http://rasdaman.eecs.jacobsuniversity.de/trac/rasdaman/browser/manuals_and_examples/manuals/pdf/ql-guide.pdf).
- Baumann, P., Merticariu, V., Dumitru, A., Misev, D., (2016), Standards-Based Services for Big Spatio-Temporal Data. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLI-B4, pp. 691-699.
- Campalani, P., Xinghua G., and Baumann P., (2014), *Spatio-Temporal Big Data*.
- Catana M.C, Gorgan D., (2015), Analyzing Computer Game Strategies through Visual Techniques. *Proceedings of the 12th Romanian Conference on Human-Computer Interaction*, ISSN 2344-1690, pp.41-48.
- Gorgan D., Catana M.C., Stefanut T., (2015), Visual analytics on multidimensional massive data (Tehnici vizuale de analiza a datelor masive multidimensionale). *Revista Romana de Interactiune Om-Calculator, Vol.8(3)*, 237-255.
- Gorgan D., (2017), Visual analysis on multidimensional massive data. *Journal of Aeronaut Aerospace Engineering 2017, Vol.6(2)*, ISSN: 2168-9792 JAAE.
- Heer J., Shneiderman B., (2012), Interactive dynamics for visual analysis. *Communications of the ACM, Vol.55(4)*, pp.45-54.
- Hughes, John F., et al., (2014), *Computer graphics: principles and practice*. Pearson Education.
- Keim D.A., Mansmann F., Schneidewind J., Thomas J., and Ziegler H., (2008), Visual Analytics: Scope and Challenges, chapter in *Visual Data Mining, Lecture Notes in Computer Science, Vol. 4404*, pp.76-90.
- Pahins C.A.L., Stephens S.A., Scheidegger C., Comba J.L.D., (2017), Hashedcubes: Simple, Low Memory, Real-Time Visual Exploration of Big Data. *IEEE Transactions on Visualization and Computer Graphics Vol. 23(1)*, pp.671-680.
- Siirtola H., (2007), *Interactive Visualization of Multidimensional Data*. Academic dissertation in interactive technology at the University of Tampere, Finland.
- Seo, J., Shneiderman, B., (2005), A rank-by-feature framework for interactive exploration of multidimensional data. *Information visualization, 4(2)*, 96-113.
- Turkay C., Kaya E., Balcisoy S., Hauser H., (2017), Designing Progressive and Interactive Analytics Processes for High-Dimensional Data Analysis. *IEEE Transactions on Visualization and Computer Graphics Vol. 23(1)*, pp.131-140.
- Xie C., Zhong W., and Mueller K., (2017), A Visual Analytics Approach for Categorical Joint Distribution Reconstruction from Marginal Projections. *IEEE Transactions on Visualization and Computer Graphics Vol.23(1)*, pp.51-60.