EG UK Theory and Practice of Computer Graphics (2012)
Hamish Carr and Silvester Czanner (Editors)

Visualizing a Spherical Geological Discrete Element Model of Fault Evolution

S.M. Longshaw*, M.J. Turner† & E. Finch⊥

*School of Computer Science, The University of Manchester, UK
†Research Computing, The University of Manchester, UK
⊥Basin Studies and Petroleum Geoscience, School of Earth, Atmospheric and Environmental Sciences, The University of Manchester, UK

Abstract

Discrete Element Modelling (DEM) is a numerical technique that uses a system of interacting discrete bodies to simulate the movement of material being exposed to external forces. This technique is often used to simulate granular systems; however by adding further elements that inter-connect the bodies, it can be used to simulate the deformation of a large volume of material. This method has precedent for use in the Earth Sciences and recently, with the increase of available computing power, it has been put to good use simulating the evolution of extensional faults in large scale crustal experiments that involve over half a million individual spherical bodies.

An interactive environment that provides high quality rendering is presented, showing that interactivity is key in allowing the intelligent application of visualization methods such as colour-mapping and visibility thresholds in order to extract fault information from a geological DEM. It is also shown that glyph representation alone is not sufficient to provide full insight into the complex three dimensional geometries of the faults found within the model. To overcome this, a novel use of the MetaBall method is described, which results in implicit surface representations of sphere sub-sets. The surfaces produced are shown to provide greater insight into the faults found within the data but also raise questions as to their meaning.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.4]: Application packages—Computer Graphics [I.3.7]: Visible line/surface algorithms—Computer Graphics [I.3.5]: Curve, surface, solid, and object representations—Simulation and Modeling [I.6.6]: Simulation Output Analysis—

1. Introduction

Understanding the initiation of geological faults, and their interaction with each other through time, is key to improving our interpretation of the present landscape and determining the evolution of previous extensional environments, such as under the North Sea. Better understanding of these phenomena can help improve seismic hazard analysis through examination of the spatial and temporal distribution of faults in order to theorise likely magnitudes of earthquakes and localisation of fault activity. It can also aid in the exploration for hydrocarbons, where being able to interpret how faults evolve and interact through time can be employed to resolve the landscape response to these changes and generate probabilities on the location of sediment deposition in evolving basins.

The data being focused on in this paper is a result of Discrete Element Modelling (DEM) work carried out by Finch et al. [Fin98, FHG03], which aims to simulate the evolution of extensional fault systems in models spanning hundreds of kilometres over periods of millions of years. The DEM is comprised of spheres of equal volume, initially arranged in a hexagonally close packed (HCP) lattice and inter-connected by a bond (represented in the model by a spring-damper). Each bond has a unique breaking strain, with the distribution of these strains chosen to represent an average volume of crustal material. The spheres are subjected to a linear force through time, and as the spheres move their bonds become stretched until they reach a pre-defined length, at which point they break. The expected result of this simulation is that the weaker bonds act as seed points from which cascades of further broken bond systems will emanate, resulting in cracks (or faults) forming in the continuous volume. Data is output at a fixed rate based on the number of time-steps that the simulation takes. Every sphere is represented as a tuplet of values for its position and then as associated variables,

DOI: 10.2312/LocalChapterEvents/TPCG/TPCG12/077-084

including: radius, strain, number of broken bonds and the body's initial position in the model according to its layer in the HCP lattice.

Interest in this data is notable for the geologist as current simulation techniques for fault evolution either rely on small scale laboratory experiments, which contain approximations in order to make them feasible as real-world endeavours, or on numerical methods which are fundamentally unsuited to the problem of simulating the formation of discontinuities. An example of this would be the Finite Element Method (FEM) which has precedent for use in solving this problem [MOB95, MBS05], but suffers from the fact that its mesh-based nature means it is unable to form true discontinuities. Finch et al. have shown DEM to be successful in simulating faults [FHG03, HF06], however due to the nature of the data being that of a densely packed collection of spheres, the process of actually finding and disseminating information about the fault structures contained within has proven difficult.

DEM as a technique was borne from geology in the 1970s [Cun71], however as with many particle based numerical simulation methods, it is only in the past decade that it has begun to receive a more widespread interest in its applicability to useful simulation both inside and out of academia. Primarily this is due to available (and accessible) computational power as n-body techniques present extensive computational overhead due to the necessity of calculating neighbouring bodies. At worst this represents a problem with a complexity of $O(n^2)$ and at best $O(n)$ using current techniques such as the Sweep and Prune broad-phase method [Bar92]. Modern processing capabilities have now made it possible to define DEMs that contain enough individual bodies such that they are able to represent a discretised version of a real-world system.

The act of visualizing $500,000$ or more three dimensional geometries in a manner that allows those of interest to be highlighted, while maintaining the visibility of those around them so that the context they provide is not lost, is a difficult problem. Currently, the visualization methods that best solve these problems in a manner suitable for the geologist have not been well defined. The work presented in this paper documents the process of applying multiple techniques to discrete element data, showing problems that obvious solutions introduce and exploring the suitability of less obvious methods to extrapolate continuous three dimensional structures from a set of discontinuous three dimensional bodies.

Visualization packages utilised through the sciences [Hen04, Sch12] offer mature environments in which to view and analyse large particle datasets. However when asked to represent the same data using high quality three dimensional glyphs they can struggle on modern commodity grade hardware. Efforts have therefore been made to apply the common visualization techniques these packages provide, using high performance rendering methods more commonly found in the likes of computer games. These include methods like favouring OpenGL Vertex Buffer Objects (VBO) over Display Lists and utilising modern OpenGL API functionality for manipulating and duplicating the vertices that represent a single object. This was done with the mind-set of allowing upwards of a million three dimensional glyphs to be rendered and interactively analysed.

Appropriate visualization methods, such as colour-mapping plus opacity variance and a visibility threshold on both the simulation domain and individual glyphs, were combined with renderings of each sphere in the model. When combined with interactivity these methods were found to allow greater understanding of the nature of the model, with a controlled set of geologists (henceforth referred to as experts) able to determine areas of the model that represented faults as they understood them. These results can be found in the thesis of Longshaw, chapter five, section 5.6.2 [Lon11]. In the thesis it is shown that when asked to rate the question *"You can definitely see at least one fault within the data"* with a value between 1 and 10 (with 1 being a negative response), 80% of those asked gave a perfect score of 10, with the remaining 20% giving a score of 9. This shows a positive response from the tested group, pointing towards the success of the glyph based visualizations.

Despite the success of utilising glyphs that is reflected in the aforementioned results, a problem with viewing the data like this was cited by a portion of the experts. While they could see where faults lay, they struggled to grasp the three dimensional nature of the structures that the collections of spheres represented, especially when considering more complex factors such as the interaction of multiple faults. In order to overcome this, Blinn's *MetaBall* technique [Bli82] was utilised to produce an implicit surface that represented the space inhabited by a set of spheres that defined a fault.

The implicit surfaces produced are compelling representations of three dimensional fault structures. There are numerous cases where the use of a single continuous surface better reveals complex detail in the structure of the faults than when viewed using glyphs; a selection of these are presented later in this paper.

The results of this work have also been evaluated by the same group of experts and can be found in the same thesis, chapter six, section 6.4.2, where each expert was shown the same five images of visualization comparisons containing glyph only representations of faults and an implicit surface alternative. For each case the expert was asked to assign a value between 1 and 10 that defined how much more effective they felt the implicit surface was in revealing the fault when compared to the glyph representation. Taking results from across all test images through the entire expert group an arithmetic mean of 6.4 was reached, showing that the method was generally favoured.

2. Related Work

Papers showing visualizations of the results from DEM simulations tend to use high quality renderings produced using specialist ray-tracing software. The result is that the images shown in publications are attractive but often provide little in a visualization sense beyond a view of the location and physical nature of the elements. There is some notable work toward utilising modern rendering techniques to provide an interactive visualization environment for datasets that contain large numbers of individual bodies [Gri09, GRDE10], however most has not yet made it into the mainstream visualization packages. In order to facilitate the production of publication grade visualizations which reveal underlying information within large DEM datasets it is necessary to combine an interactive frame-rate with the techniques currently available for particle datasets, while producing accurate three dimensional representations of the elements that define the model [LLW*11].

When a DEM is used to simulate the fracture of a block of material, the theoretical meaning of each element is as a portion of the overall volume of the material with connecting structures such as spring-dampers representing its connected nature. Changing the properties of the inter-connect makes it possible to change the overall materialistic nature of the simulated volume [MP93]. In these cases it becomes important to consider whether representing each individual element as a glyph is still the most appropriate method of visualization, or whether using the individual geometries to form a single, continuous surface is now more applicable.

The MetaBall technique, originally defined by Blinn [Bli82], can be described as the process of drawing an isosurface through a pre-computed scalar field. It is often used in molecular visualization and within animation packages for the production of smoothed conglomerate objects and has precedent for use in presenting discrete datasets [NIDN97]. In his original paper, Blinn defined the technique as a way to draw isosurfaces that showed the electron field produced by molecular structures. As a result of this, he specified a field function based on the formulation used to calculate electron density maps. The important aspect of this function, however, was that it behaved as an inverse square law. Subsequent implementations of the technique refer to various field functions, such as gravitation or radiation but this can be generalised to state that a suitable field function to produce a natural looking drop-off rate between two connected spheres can be achieved using a function $f(x)$ in the form of

$$f(x) = \frac{\omega}{||x - c_i||^2} \quad (1)$$

where ω is a weighting associated with the shape currently being tested and $||x - c_i||^2$ is the squared distance between the current field point x and the current sphere being tested c_i.

3. Visualizing Faults Using Glyphs

The goal when interrogating the DEM data presented is to pick out the elements which collectively form a fault and then see how they evolve though time. This includes factors such as their broken bond seed point, their growth directions and how they interact with each other.

The data includes multiple variables per element, meaning that their position and volume in three dimensional space is known at all points through time as well as aspects of their physical state. These include the current number of broken bonds, the shear strain that the element has experienced and its original position in the HCP lattice at the start of the simulation. Using high quality spherical glyphs to represent each element, it is possible to utilise the extra information that these variables provide in order to differentiate them. An example of this is shown in figure 1 where the final state of evolution of a model containing $525,000$ spherical elements is shown. The top figure shows each glyph rendered according to the underlying strain variable while the bottom figure shows them rendered using the same linear colour-mapping but according to the broken bond variable. As one would expect, the use of colour enhances contrast within the visualization, drawing the eye towards fractured areas that represent faults. The examples shown here utilise a simple blue to red linear colour-mapping. This scheme has been found to be effective in providing a high contrast between important elements that represent a fault and those which do not but provide context. However, there is future scope for refinement of the colour-mappings used, with an initial suggestion being the use of a non-linear scheme between two contrasting pastel shades.

The most obvious problem that figure 1 highlights is that, due to the densely packed nature of the data, looking below the surface is impossible. The three dimensional nature of the shape and size of the faults is a critical outcome of this type of modelling; therefore the ability to assign a threshold on which elements are visible is an important one. One way this can be achieved is by assigning elements with low importance (closer to the blue end of the colour-mapping in figure 1) a different opacity value, meaning they do not completely obscure elements of higher importance. This was attempted but found to produce a confusing visualization due to the fact that the opacity gradient through the elements was not typically smooth in any one direction; a better solution therefore was to completely hide the elements deemed unnecessary.

The definition of a visibility threshold can occur in two ways, the first being according to position in space or a *cutting threshold*, the second being by an associated variable or a *variable threshold*. The definition of a cutting threshold is commonly associated with volume visualization and involves the definition of a plane taken through the volume which defines a new outer surface for the rendering algorithm to calculate. The basic premise here is similar but

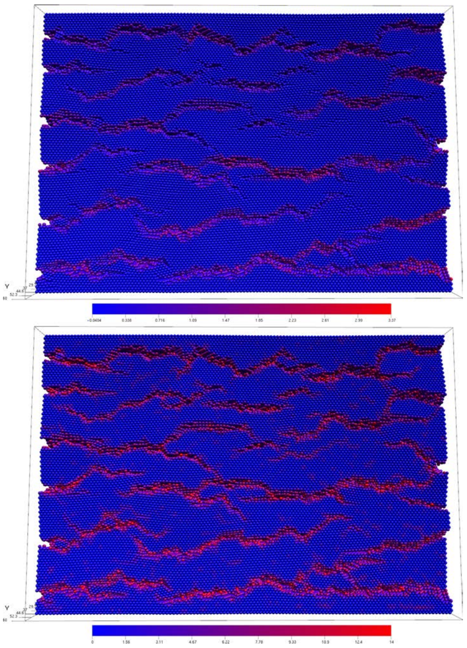

Figure 1: *Two views of the same DEM dataset containing* 525,000 *spheres. The top figure shows each glyph rendered according to a linear blue to red colour mapping defined by the underlying strain variable while the bottom figure uses the same mapping assigned to the broken bond variable. In both cases the use of colour offers visual differentiation between the glyphs, revealing the faults more easily.*

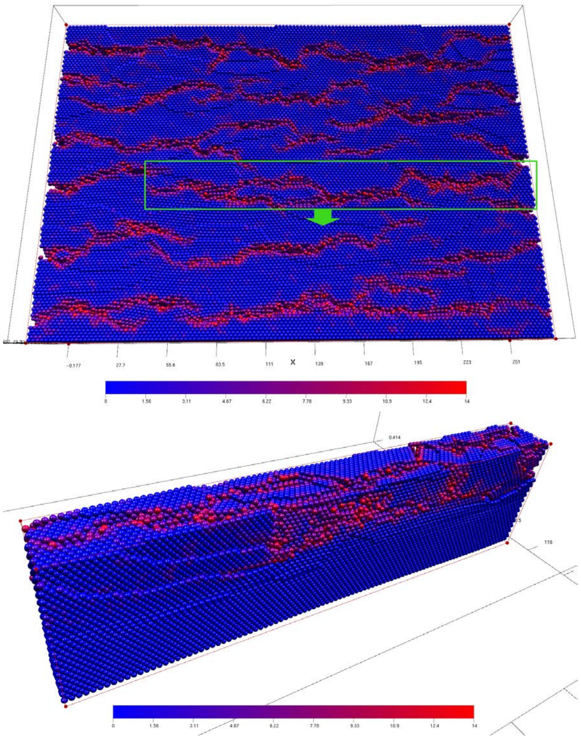

Figure 2: *The process of interactively defining a cutting threshold in order to reveal and then explore the nature of a single fault. The top of the figure shows a view of the initial dataset using the broken bond based colour-mapping (see figure 1 for an overall view) with a cutting threshold applied through its height. The bottom then shows the effect of applying further cutting thresholds to refine the view to a single fault.*

slightly different in that the data being viewed is not a volume but a collection of discrete bodies. A *cutting threshold* in this case therefore refers to the definition of a bounding box that sits within the confines of the overall simulation space and any elements that fall within its catchment are visualized. This simple premise allows the DEM data to be dissected and considered as sub-sets of elements. Combined with the ability to differentiate elements by colour, the researcher is able to spot an area of interest and remove surrounding elements so they can consider the structure.

Figure 2 shows an example of this, showing two views of the model, the view at the top of the figure shows a cutting threshold defined through the height of the model, revealing underlying fault information, while the view at the bottom shows a further refinement made to the width and depth of the model and a view rotation applied.

Refining the visible data by cutting threshold, as per figure 2, suffers from the problem that it may not always be desirable to only be able to view one area of the model at one time. One solution for this is to allow the definition of multiple thresholds; another is to follow the aforementioned technique of assigning opacity to an element according to its variables, only this time assign a simpler visibility state. This process of assigning a *variable threshold* is easy to grasp for the researcher as it allows them to step through their chosen variables data range, either defining an upper or lower limit for the data that is visible. The most useful case for this technique was when considering a variable with a decimal data type, i.e. the strain variable, which is floating point and therefore has a potentially large range of values that dictate the number of discrete threshold levels available.

The effect of refining the glyph visibility according to a variable threshold is compelling, with a carefully selected threshold in place it becomes possible to start to see the three dimensional nature of the faults as well as which clusters of elements belong to which structure. Figure 3 shows a variable threshold being applied to the strain variable, the faults are revealed and extraneous surrounding elements are removed.

The combination of both types of threshold, when intro-

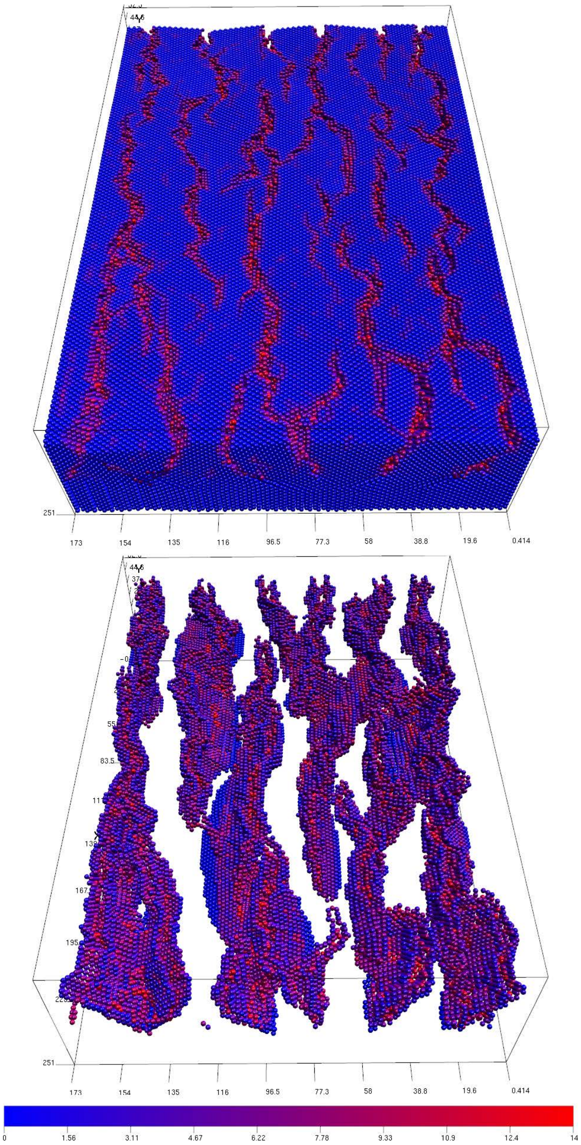

Figure 3: *Two views of the DEM data showing the effects of applying the variable threshold according to the strain variable. The top figure shows a view of the data chosen to allow good understanding of the extensional nature of the data as extension occurred from the left to the right. The bottom shows the same data from the same view but with a variable threshold applied, revealing the underlying faults.*

duced in an interactive environment, allows the researcher to refine a large and densely packed DEM dataset down to a smaller sub-set representative of an area of interest. They can then further strip away elements within that area that have not undergone failure on a fault plane in order to reveal the underlying three dimensional structure.

While the techniques considered so far have allowed the glyph representation of the DEM data to be refined, it is true to say that the discrete nature of the DEM data produces, at best, an abstract representation of something which in the real world would be seen as a three dimensional surface. The remainder of this paper is therefore devoted to the process of utilising these sub-sets of elements to produce a more tangible three dimensional surface representation of the faults. This starts by looking at the technical aspects of the MetaBall technique, followed by a presentation of results that applying this method has allowed and then a conclusion of its successes and areas for further development.

4. From Spheres to a Surface: Visualizing a Scalar Field Representation of a Collection of Geometries

When considering the best method to produce a continuous surface representation of an underlying set of spheres, one solution is to wrap the outermost spheres from the set in a manner akin to the vacuum forming process. Theoretically this process can be achieved using various techniques with the most likely being a form of ray-tracing in order to detect outermost geometries; alternatively a mesh generation technique could be used to join vertices placed on the outermost point of the outermost spheres from the set.

The problem with both of these techniques is that, taken literally, they will not produce a smooth surface between each sphere, instead producing a surface that looks like it is made from many connected spheres. A solution for this is to introduce a smoothing factor for areas of the surface that fall between spheres, however finding a natural balance between a smooth surface and ensuring data is not lost is non-intuitive. An answer is to utilise a technique which first calculates an underlying scalar field which can then have an isosurface extracted from it at differing threshold values, such as the MetaBall process [Bli82].

4.1. The MetaBall Technique

The process of producing a MetaBall has two stages, the calculation of a scalar field that represent the underlying phenomena being investigated and then rendering an isosurface through the field using a technique such as Marching Cubes [LC87]. Calculation of the scalar field can be a computationally intensive process, however it only needs to be completed once and its results can be stored; therefore it is a feasible method for this application.

From a technical point of view, the resolution of the field has a direct effect on how smooth the implicit surfaces that can be drawn through it are, therefore it is important to select a value which balances processing time with rendered surface quality. The function used here is as per equation 1, with the weighting variable, ω, set equal to the radius of the sphere being tested. The algorithmic layout of this function is shown in algorithm 1, the structure of the pseudo-code is simplified here using nested loops; this is to aid the clarity of the description and does not represent the most efficient implementation.

Algorithm 1 describes the field function $f(\mathbf{x})$ used to generate the scalar field given a field of points (FP), a list of spheres (s) and a distance store d.

```
for $i = 0$ to $i = s_{max}$ do
    for $j = 0$ to $j = FP[x]_{max}$ do
        for $k = 0$ to $k = FP[y]_{max}$ do
            for $l = 0$ to $l = FP[z]_{max}$ do
                $d = Distance(FP[j][k][l], s[i])$
                if $d > 0$ then
                    $FP[j][k][l].value + s[i].radius/d$
                    $FP[j][k][l].norm + ((FP[j][k][l].pos - s[i].pos) * s[i].radius)/d^2$
                end if
            end for
        end for
    end for
end for
```

At this stage it is necessary to impart information to each vertex that can be used to create a colour at that point on the surface. Visualized surfaces typically have colour-mappings assigned according to the relative location of each vertex; in this case however the relative location is not a useful metric. Instead information regarding an associated variable (e.g. broken bond count or strain) is more useful. This process is depicted in a simplified two-dimensional manner in figure 4. It involves assigning each field point an initial colour, then those points which fall within the catchment of a sphere are assigned that glyph's colour. In each case, the radius (r) of each glyph is extended by a nominal amount (w), which has the effect of allowing some points to fall within the catchment of more than one glyph, where an average value is then taken. This allows for a smooth transition of colour at areas of the surface where two spheres are next to each other. This technique can be likened to others such as *splatting* [Lar03], however it is simpler in concept due to the fact no alpha-map needs to be defined (as this is intrinsically defined by the underlying geometries).

4.2. Implicit Surface Representations

Once a scalar field has been defined that is representative of an underlying set of spheres, it is then possible to render iso-surfaces through the field at differing threshold values. With an appropriate threshold selected, these surfaces are able to provide a reasonable representation of the actual DEM data.

Figure 5 shows an implicit surface drawn through a scalar field representative of the sub-set of elements selected in figure 3, viewed from an angle rotated 180° in the X/Z plane. In viewing the discrete data as a single surface, the three dimensional nature of the faults becomes more visually apparent. This conclusion is backed up by a panel of experts who, when interviewed, agreed that this was the case [Lon11], although this was not unanimous.

When viewed using only glyphs it can sometimes be difficult to see detail regarding the geometry that the spherical elements combine to make. However, when the same data is viewed as an implicit surface these problems become

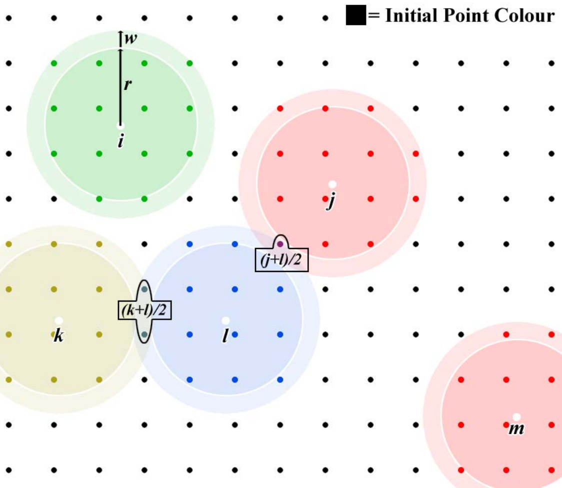

Figure 4: *A two-dimensional simplification of the process of assigning colour to each point in the scalar field, according to the spherical glyphs that encapsulate them.*

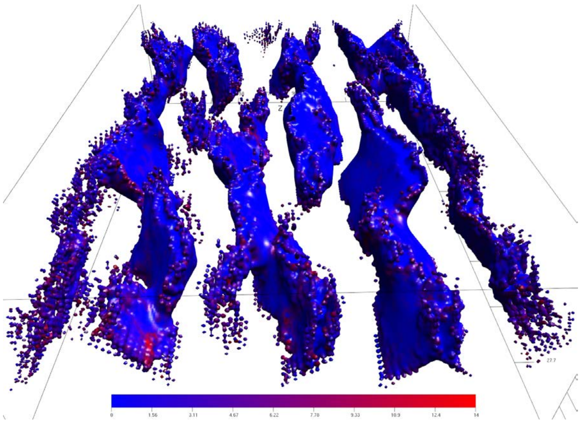

Figure 5: *An implicit surface representative of a set of spheres.*

less apparent. Figure 6 shows this effect, where a sub-set of spheres has been refined and is believed to represent a fault.

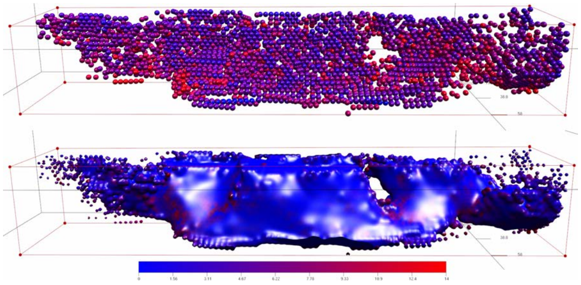

Figure 6: *Two views of the same data. The top view uses glyphs only while the bottom view uses an implicit surface representation of the spheres from the top view. When viewed as a surface details about the geometry of the structure that the spheres represent become more obvious with the effects of lighting and shading highlighting the surface shape of the fault better than using glyphs.*

The bottom view shows that by implementing an implicit surface it is now possible to gain a greater understanding into the nature of the fault, with the effects of shading and lighting in the rendering meaning that undulations that cause indentations can now be seen.

While the use of an implicit surface to represent a collection of spheres has been shown to be a useful visualization technique for the analysis of geological DEM data, it is not true that it can totally replace the more usual methods of glyph representation. This is primarily due to the fact that the surface representation hides the underlying detail provided by the individual elements. While this is by design, it also means that only viewing the discrete data as a surface may lead to the researcher not gaining an understanding of the true nature of the model, potentially leading to misinterpretation and presentation of faulty assumptions as scientific fact.

4.3. Combined Visualization

Building on the successes of the aforementioned methods, a logical step was to create a single scene, containing both glyph and implicit surface representations, which would allow for a more informative visualization. Figure 7 shows a sub-set of spheres represented using glyphs that have been coloured in alternating bands according to their original height in the model. Seeing a cross-section through the model using this method is useful in showing how the material has deformed compared to its starting formation. An implicit surface that represents a selected fault through the entire width of the model is also shown, this is coloured according to the broken bond count of the underlying elements.

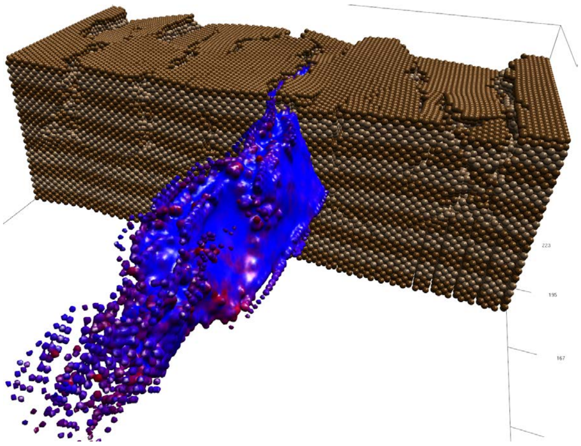

Figure 7: *A visualization showing the effect of combining glyph representation with an implicit surface. The two techniques combine with each bolstering the information provided by the other. However their combination also has the effect of hiding detail formerly visible when viewed individually.*

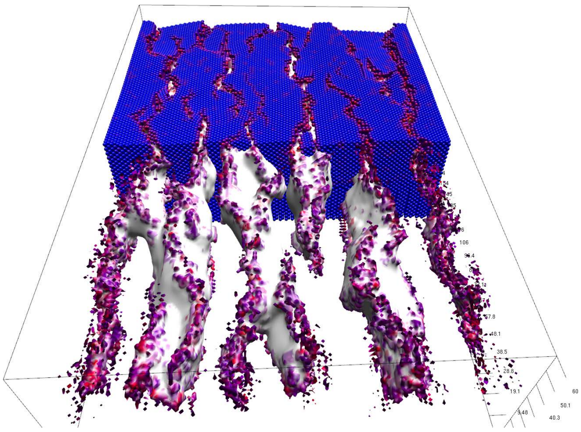

Figure 8: *A visualization showing the effect of combining glyph representation that uses a complex colour assignment scheme to highlight important glyphs with an implicit surface representation of underlying faults, rendered according to a vertex colour-mapping derived as per section 4.1, which highlights areas of the surface that does not physically fall within the space occupied by the spheres that it is representing by initialising each vertex to white rather than blue (as seen in figures 5, 6 and 7).*

Figure 8 shows a similar sub-set of spheres to figure 7, however this time the glyphs are coloured according to their broken bond variable. An implicit surface representation is also rendered in the visualization, however in this case a cutting threshold has not been applied to the sub-set of elements used in its creation; therefore faults can be seen emanating from more areas of the block. The vertex colour-mapping used for the surface has also been changed such that each vertex has been given an initial colour of white rather than blue. The choice of starting colour alters the clarity of areas of the surface which fall physically outside of space occupied by the spheres.

In combining the methods, the figure is able to show more information than either technique would be able to when used on their own. When these techniques are provided in an interactive environment, they provide a powerful visualization pipeline for the researcher trying to understand the complex nature of the DEM fault data. As with any visualization it is still possible to introduce too much visual clutter and hinder the usefulness of the individual methods.

5. Conclusions

This paper has outlined work undertaken to allow the analysis of data produced by a geological DEM through visualization. It has covered the use of interactive glyph representation and introduced a novel application of the MetaBall technique to produce implicit surface representations of faults defined by the placement of elements within the data.

Using glyph representations of each element and applying visualization techniques such as colour-mapping and visibility thresholds, within an interactive environment, allowed the faults to be revealed from within the block-like structure output by the DEM. It was found, however, that it could be difficult to gain an insight into the three dimensional nature of the faults when viewed using discrete glyph representations alone.

Implicit surfaces were therefore produced via the MetaBall technique, which were approximations of the overall volume occupied by a set of spheres that represented a fault. The surfaces were shown to produce a useful visual alternative to glyph representation and, in some cases, combined well into a single visualization with the glyph technique. The surfaces correlate visually with their underlying spherical glyphs but a further question to ask is what meaning do the surfaces hold. As a visual approximation of the underlying DEM data they have been shown to be a successful alternative to discrete glyph representation; however the important question of whether they can be considered as true manifestations of the modelled faults remains open.

References

[Bar92] Baraff D.: *Dynamic Simulation of Non-Penetrating Rigid Bodies*. PhD thesis, Computer Science Department, Cornell University, 1992. 2

[Bli82] Blinn J. F.: A generalization of algebraic surface drawing. *ACM Transactions on Graphics 1*, 3 (1982), 235–256. 2, 3, 5

[Cun71] Cundall P. A.: A computer model for simulating progressive large scale movements in blocky rock systems. In *Proc. Sympo. Int. Soc. Rock Mech.* (1971), pp. 129–136. 2

[FHG03] Finch E., Hardy S., Gawthorpe R.: Discrete element modelling of contractional fault-propagation folding above rigid basement fault blocks. *Journal of Structural Geology 25*, 4 (2003), 515–528. 1, 2

[Fin98] Finch E.: *A Crustal Lattice Solid Model: The Evolution, Geometry and Scaling of Tectonic Extension*. PhD thesis, School of Environmental Studies, University of Ulster, 1998. 1

[GRDE10] Grottel S., Reina G., Dachsbacher C., Ertl T.: Coherent culling and shading for large molecular dynamics visualization. *EG Computer Graphics Forum 29*, 3 (2010), 953–962. 3

[Gri09] Gribble C. P.: Interactive particle visualisation. In *Trends in Interactive Visualization*, Jain L., Wu X., Liere R., Adriaansen T., Zudilova-Seinstra E., (Eds.), Advanced Information and Knowledge Processing. Springer London, 2009, pp. 1–19. 3

[Hen04] Henderson A.: *The ParaView Guide: A Parallel Visualization Application*. Kitware, 2004. 2

[HF06] Hardy S., Finch E.: Discrete element modelling of the influence of cover strength on basement-involved fault-propagation folding. *Tectonophysics 415*, 1-4 (2006), 225–238. 2

[Lar03] Laramee R. S.: First: A flexible and interactive resampling tool for cfd simulation data. *Computers and Graphics 27*, 6 (2003), 905–916. 6

[LC87] Lorensen W. E., Cline H. E.: Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph. 21*, 4 (1987), 163–169. 5

[LLW*11] Lipşa D. R., Laramee R. S., Walker R., Roberts J. C., Cox S.: Visualization for the physical sciences. In *State of the Art Reports* (2011), EUROGRAPHICS 2011, pp. 49–73. 3

[Lon11] Longshaw S. M.: *Numerical Modelling and Visualization of the Evolution of Extensional Fault Systems*. PhD thesis, School of Computer Science, The University of Manchester, 2011. 2, 6

[MBS05] Megna A., Barba S., Santini S.: Normal-fault stress and displacement through finite-element analysis. *Annals of Geophysics 48*, 6 (2005). 2

[MOB95] Munjiza A., Owen D. R. J., Bicanic N.: A combined finite-discrete element method in transient dynamics of fracturing solids. *Engineering Computations 12*, 2 (1995), 145–174. 2

[MP93] Mora P., Place D.: A lattice solid model for the nonlinear dynamics of earthquakes. *International Journal of Modern Physics C 4*, 6 (1993), 1059–1074. 3

[NIDN97] Nishita T., Iwasaki H., Dobashi Y., Nakamae E.: A modeling and rendering method for snow by using metaballs. *EG Computer Graphics Forum 16*, 3 (1997), 357–364. 3

[Sch12] Schlumberger: Petrel. `http://www.slb.com/services/software/geo/petrel/`, 2012. 2