IJRRECS/October 2013/Volume-1/Issue-6/1268-1275 **ISSN 2321-5461**

INTERNATIONAL JOURNAL OF REVIEWS ON RECENT ELECTRONICS AND COMPUTER SCIENCE

3D POINT CLOUD DATA OVER X3DOM

Gopi Krishna[1], B.V. Seshukumari[2]

[1]M.Tech Student, Dept of CSE, St.Peter's Eng. College, Kompally, Medchal Mandal, Hyderabad, A.P, India

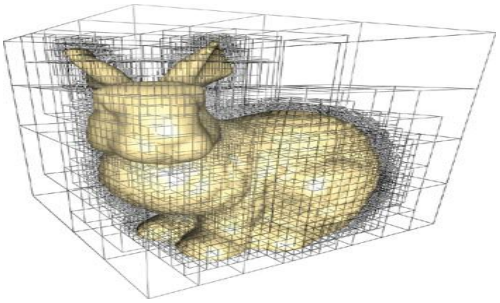
gopiapex@gmail.com

[2]Assoc. Professor & HOD, Dept of CSE, St.Peter's Eng. College, Kompally, Medchal Mandal, Hyderabad, A.P, India

ABSTRACT:

Recent developments in the area of efficient web-service architectures and the requirement to provide applications not emerging support for GPU-supported and therefore high-performance 2D and 3D graphics in modern web-client implementations and standards provide new application environments, which are especially interesting for the demands of scientific visualization solutions. Thus, I present a web application for client-side rendering as well as for a large number of processing and visualization aspects. To get high quality of 3D polygon models, we need more polygons as advantage as the disadvantage of high quality, it need more rendering, computation power, and more space. To avoid these draw backs the voxelization and point cloud data can provide the high quality visualization the data can get from Lidar devices or Kinect. But the same time it needs advanced searching algorithm and the new rendering pipeline.

Keywords: X3D, HTML5,WebGL, DOM, X3DOM, Web Integration, Transcoder, Web Services, Service-oriented Architectures, sparse tree octree, unlimited polygon.



Voxelization

Point Cloud

1. INTRODUCTION:

The X3DOM may be a climbable architecture; witch evolves from HTML X3D integration model X3DOM introduced in Beher et al.2009. The model is to integrate declarative X3D content directly in markup language DOM tree. The open supply x3dom.js design provides enchantments of X3DOM open points generic model, the outstanding idea is single declarative interface to application developers. It supports varied backend an influence full fall back models. The open supply purpose cloud library will well integrated with X3DOM, the OpenCL and CUDA conjointly proves that the capability's get advantage for X3DOM.

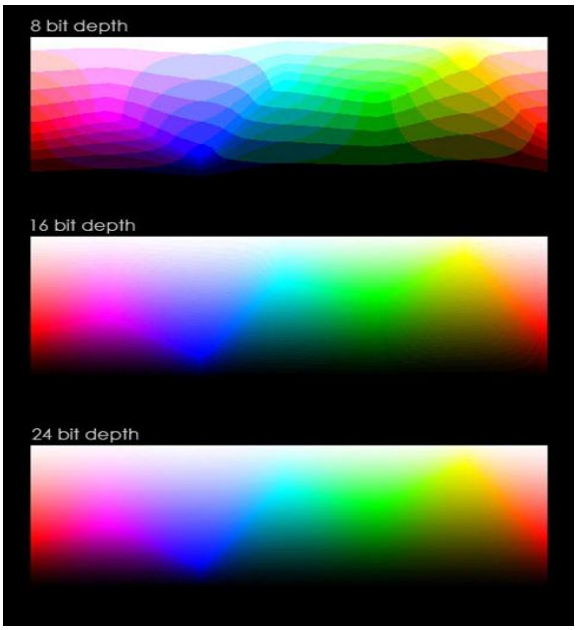
The major drawback in x3dom: It can not ready to show Brobdingnagian range of two-dimensional figure based mostly models, think about the open sim for virtual world it's manage by distributed servers, one region will live of 256X256 m as per FOV (field of vision) the open sim viewer support a minimum of one region up forty,000 objects it will support , in fly mode it will cowl a lot of regions which means a lot of two-dimensional figure has got to load in to memory . If we wish build open virtual worldor open world 3D games by sing X3DOM, we've to down load total scene, it occupies a lot of memory, and also the memory is major disadvantage until currently. The x3dom is up to currently, It will capable to render victimization Ray casting, sparse, octree as per euclideon description, we want to render 3d Atoms

Server: The up to currently the overall scene has transfer in to shopper aspect and render in browser victimization UD we want transfer camera FOV knowledge solely , this knowledge reside in server as per Bruce holler compared it with an exploration algorithmic rule, a lot of specifically compared it with an exploration algorithmic rule for words. As everyone is aware of, words kind AN ordered set, with the authorship order. Moreover, holler explains that the issue of UD is to choose from the massive info (which may well be in an exceedingly server, whereas the algorithmic rule runs on a pc elsewhere, so the algorithmic rule has got to be in an exceedingly sense an internet algorithmic rule solely the 3D atoms that square measure visible on the screen, then render solely those atoms.

Therefore, imagine that we've a large info of 3D atoms with some characteristics (like color) and a separate info manufactured from the coordinates of those atoms. The UD algorithmic rule solves a "sorting" drawback within the second info. (I place "sorting" as a result of there's no total order totally compatible with the answer of the ray-casting drawback, within the sense that it remains a similar once the POV is modified.) Once this "sorting" half is finished, the algorithmic rule asks for the characteristics of solely those points and issue to the rendering half, that is nearly trivial.By viewing sorting algorithms, it's then to be expected a time estimate of the type (for given, mounted range of screen pixels), wherever is that the range of 3D atoms. There square measure even sorting algorithms that square measure a lot of economical, like , however they're a lot of economical in observe for very Brobdingnagian numbers of atoms , like , because the AKS sorting. So, the lowest line is this: admit UD as being a sort of algorithmic program.

True Geometry

As Euclideon, is employing a specialised technique of representing purpose cloud information for the high detail 3D objects they use. In a lot of identical manner as we've the idea for what constitutes True Color, as within the higher limit of color illustration that the human eye will recognize before additional refinement goes unheeded, Euclideon possible use identical train of thought to the illustration of the 3D models themselves. There should be associate higher limit to what proportion detail are often resolved before breakdown any more would go unheeded by the human eye, regardless if the system will resolve unlimited detail additional. We'll talk over with this as a "True Geometry" technique.



Indeed, they are employing a kind of purpose cloud information illustration for the models themselves, usually times optical maser scanned into the system with nearly 1,000,000 two-dimensional figure equivalent, but this comparison is not precisely correct once were handling purpose cloud information, or moreover the Unlimited Detail state of affairs. whereas we are able to say the file originally was one,000,000 two-dimensional figure equivalent, once is not reworked into the procedural purpose cloud dataset, that equivalent remains identical whereas reducing drastically the filesize and process needs for rendering identical kind of fidelity. However, the key to sanctionative this sort of filesize reduction let alone the fractional computation for a lot of higher fidelity is not thanks to one innovation, and in result depends on variety of things happening at identical time and dealing together to realize the specified effects shown within the videos.

The Searchable Point Cloud Model

One of the vital things to contemplate on our list of innovations is that the assertion by Euclideon that they're conjointly using a sophisticated search algorithmic rule at intervals the context of this unlimited detail system. this can be extremely vital within the elementary understanding of however this sort of system is enabled, therein rather than having to access the info contained at intervals the model sort files in completeness, it's doubtless choppy into a sort of restricted info format whereby the cells within the info contain components of the pure mathematics cloud information with a prospect of relative positioning for that exact a part of the

pure mathematics in regard to the model illustration. This relative positioning data would then be accustomed puzzle out sooner than time the occlusion of the potential pure mathematics in regard to international position at intervals the model, within the world and accounting for the camera house of the user.

this system would take away the linear barrier that's value of rendering, though even this alone is simply one step within the strategies used overall joined, but spectacular. We will label this as “Non-Linear Data”.

Procedural Methods

Now that we have a beginning to our understanding, we continue on to exactly what those database lines are likely storing. Normally the geometry for something like a Collada (DAE) format would look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<COLLADA
xmlns="http://www.collada.org/2005/11/COLLADASchema" version="1.4.1">
  <asset>
    <contributor>
      <authoring_tool>Softimage|XSI
version 7</authoring_tool>
    </contributor>
    <created>2008-08-
21T17:18:12Z</created>
    <modified>2008-08-
21T17:18:12Z</modified>
    <revision>1.4.1</revision>
    <unit meter="0.1"
name="decimetre"></unit>
  </asset>
  <library_cameras>
    <camera id="cameras_0">
      <optics>
        <technique_common>
          <perspective>
            <yfov>53.638000</yfov>
            <znear>0.100000</znear>
            <zfar>32768.000000</zfar>
          </perspective>
        </technique_common>
      </optics>
      <extra>
        <technique profile="XSI">
          <XSI_Camera>
            <xsi_paramsid="std">9
</xsi_param>

<xsi_paramsid="aspect">1.333333
</xsi_param>
<xsi_paramsid="fov">0.000000
</xsi_param>
<xsi_paramsid="fovtype">1 </xsi_param>
<xsi_paramsid="proj">1
</xsi_param<xsi_paramsid="orthoheight">0
.100000 </xsi_param>
 <xsi_paramsid="interestdist">20.099751
</xsi_param>
<xsi_paramsid="near">0.000000
</xsi_param>
<xsi_paramsid="far">0.000000
</xsi_param>
<xsi_paramsid="projplane">FALSE
</xsi_param>
<xsi_paramsid="projplanewidth">0.188976
</xsi_param>
<xsi_paramsid="projplaneheight">0.141732
</xsi_param>
<xsi_paramsid="projplaneoffx">0.000000
</xsi_param>
<xsi_paramsid="projplaneoffy">0.000000</
xsi_param>

 </XSI_Camera>
  <XSI_CameraFocalLength>
 <xsi_paramsid="projplanedist">4.747280</
xsi_param>
 </XSI_CameraFocalLength

  </technique>
  </extra>
 </camera>
 </library_cameras>
```

The previous is taken from a COLLADA mesh of a rigged Man, deliberation in at around 15MB of information. so as to urge the information (XML) I tried to load the get into pad of paper, that fast up for for a while throughout the method of loading the whole 15MB of XML text. Clearly I'm not posting the whole XML information here on the journal, as a result of that might be quite vacuous. the purpose here is that actual standards like COLLADA ar dreadfully wasteful and tumid as file formats to represent model information. the difficulty is not the pure mathematics or the properties concerned, it is virtually all the way down to however they're depicted within the file and the way a system would wish to access that individual information among the file.

In our linear methodology of thinking, we have a tendency to assume that each purpose and detail would like be written get into linguistics so as to own what we want to reconstruct the model in 3D. this is often additionally supposing that we want to load the whole model in totality before we are able to run some variety of calculations on level of detail and occlusion. After all, however will be presupposed to grasp what geometries in our model are going to be occluded if we have a tendency to can not load the whole file up front to investigate it the solution might return from our original thinking within the previous section, not to mention the file format itself, and so we have a tendency to place them along side intelligent ways for more improvement.

Firstly, being a research engine algorithmic program base, the file format would seemingly be depicted sort of a information and not a tumid XML text of all pure mathematics that should be preloaded into memory. Instead we have a tendency to currently have access to individual lines or sections of the file while not having to load the whole file into memory up front as a result, effectively streaming the specified pure mathematics from the get into real time whereas actively ignoring any lines that do not match the search demand. this is often an enormous breakthrough on its own, however we have a tendency to are not quite there with a complete resolution. however that pure mathematics and property information is depicted on those lines among the information also are of concern for addressing the process necessities.

Sure, we have a tendency to not would be restrained to linear ways and not having to load a complete tumid model file into memory to figure with it, however if the pure mathematics lines themselves within the information are still 1:1 and tumid, then the full file size remains terribly massive. therefore we have resolved the process aspect and access, however were left with the file size side to unravel.

Coincidentally, this can be wherever we have a tendency to add the chance of procedural methodologies to store the purpose cloud information lines within the information file structure for simple access and compartmentalization by the search formula within the engine itself. On its own, procedural strategies will manufacture the kind of fidelity we have a tendency to see with Euclideon, however at a terrible rendering expense for computation so as to effectively solve equations to urge our results. during a linear fashion this is able to instantly be preventive to accomplish as a result of we'd have to be compelled to load presumably a 15MB file that then would wish to be consistently "inflated" in totality, or at a awfully high resolution, before having the ability to method occlusion or pure mathematics within the 3D house.

However, we're not coping with a linear methodology, and intrinsically ar free of the constraints of getting to unravel a whole pure mathematics of the model before any occlusion calculations and rendering. Instead, we will currently by selection stream solely the geometries in specific lines among that information and ignore the bulk of the file as an entire within the method. This comes as a surprise to ancient thinking in this we have a tendency to aren't any longer having to load a model up front before work with it, and instead ar currently coping with fragmentary file

segments within the rendering pipeline. The inclusion of procedural strategies of representing that pure mathematics on a per-line basis solely adds to the reduction of necessities for rendering during this case as a result of advanced calculations needn't apply as an entire (which would be computationally expensive) however instead solely terribly little snippets of calculations that ar simply solved in period ar in use.But, of course, there's even quite this occurring below the hood.

Screen Space Limiting

An addition to the current line of thinking comes within the pretence that even supposing we have a tendency to could have resolved a possible bottleneck with file size needs (Procedural purpose Cloud Data) and procedure needs (Search algorithmic program and info vogue selective streaming) we have a tendency to square measure still left with a hefty chunk of information to represent in an exceedingly 3D house. This, too, is reduced even more through the implementation of search criteria before loading the index and returning values. The screen house itself becomes a crucial issue to creating yet one more forceful improvement to the rendering value in this solely the pixels inside the viewing house of the 3D setting via the 2nd viewport canvas would like be calculated, more reducing the overwhelming quality ordinarily seen in ancient ways.

Since there's a probability that procedural geometries escort some kind of relative positioning information within the case of Euclideon's unlimited detail system, the search algorithmic program is pre-occluding pure mathematics before really loading it into memory via selective question. boost this the more refinement of calculation based mostly screen house occlusion inside the scope of a 2nd house of window resolution, and that we have more ways for a groundwork based mostly algorithmic program to ignore more individual lines of pure mathematics information once breakdown pure mathematics in an exceedingly 3D world.

As explicit by Euclideon recently, a majority of the necessities for his or her system stem from merely a computer's ability to show a bitmapped image on screen, that arguably any laptop move back to 1994 is capable of emphatically, and in software system mode alone.

Deconstructing Complexity

What we're left with could be a heap of complexness once all is alleged and done. although the individual lines of the info structure square measure non-linear access supported a extremely complicated set of search rules, we're still left with the notion that procedural strategies still come back right down to formula algorithms, that still will be quite computationally high-ticket. However, yet again, our reasoning appears to supply a solution to the current downside before it ever becomes a tangle.

Procedural purpose Cloud information features a potential to be terribly massive in filesize or procedure value, however that's simply potential and not application during this case. the particular equations themselves take up a small fraction of filesize information to store, and whereas breakdown those equations in totality will inflate a file and computation vastly, we have a tendency to're in no position to really need that we resolve those procedure algorithms hold on on a per-line basis to any intensive detail to start with. What distinction would it not build if those procedural algorithms were solved among five steps or less in formula versus a linear approach that will need a number of hundred or thousand in complexity?

The answer happens to be a procedure savings of a number of thousand %, and promptly among the realm of code computation alone. we have a tendency to aren't having to resolve quite a grip of individual points during a cloud, and one by one these computations square measure preposterously low-end, however taken along structure a robust system. Since there's a

limiting issue supported screen area and element resolution, those commonly computationally intensive resolutions for pure mathematics become downright trivial and even then square measure solely on a line-by line non-linear basis with intensive pre-culling with search strategies.

Hence, even supposing there's probably high procedure and file sizes, the particular output doesn't need that.

Conclusion:

The controversial topic of unlimited 3D point-clouddata byeuclideonhttp://www.euclideon.com/ , has been released the product geoverse (http://meixnerimaging.com/) now it no more hoax , I hope now it time to re thinking about our existing 3D rendering technology , as I have concentrate web based 3Dtechnology's X3DOM is the advanced browser based 3D rendering ,without plug-in , hard ware accelerated WEBGL based technology.

References:

1.http://www.adobe.com/products/flashplayer/.

2.Adobe systems. http://www.adobe.com/.

3.Anark Cooperation. http://www.anark.com/.

4.APPLE, 2008.3d ccs-transforms for the webkit. http://webkit.org/specs/CSSVisualEffects/CSSTransforms3D.html.

5.ARNAUD, R., AND BARNES, M. 2006.*Collada: Sailing the Gulfof 3d Digital Content Creation*, 1 edition ed. No. ISBN-13: 978-1568812878. AK Peters, 2006, August 30.
ARNAUD, R., AND PARISI, T. Developing web applications withcollada and x3d. http://www.khronos.org/collada/presentations/
Developing Web Applications with COLLADA and X3D.pdf.

6.BISHOP, C., 2008. Canvas 3d js library. http://www.c3dl.org/.

7.Cult3d by cycoresystmes. http://www.cult3d.com/.

8.ECMA. Ecma-262, ecmascript language specification. http://www.ecmainternational. org/publications/standards/Ecma-262.htm.
EXCORS, P., 2007. Canvax3d. https://labs.mozilla.com/forum/comments.php?DiscussionID=363.
GOOGLE.Googleearth. http://earth.google.com/.

9.GOOGLE, 2009. Google chrome experiments. http://www.chromeexperiments.com/.GOOGLE, 2009. O3d; anjavascript based scene-graph api. http://code.google.com/apis/o3d/.

10.GOOGLE, 2009. V8 is google's open source javascript engine.
http://code.google.com/apis/v8/.

11.KHRONOS, 2008.Opengles.Khronos Group. http://www.khronos.org/opengles/.KHRONOS, 2009.Khronos launches initiative to create open royalty
free standard for accelerated 3d on the web. Khronos Group.PANDZIC, I. S., AND FORCHHEIMER, R. 2002.*MPEG-4 Facial*
Animation: The Standard, Implementation and Applications.

12.John Wiley & Sons Ltd, West Sussex, England.PROJECT, P. Papervision3d. http://blog.papervision3d.org/.

13.STEWART, J. Freewrl, open-source vrml/x3d runtime.
http://freewrl.sourceforge.net/index.html.

14.SUN.Java. http://java.com/en/.

15.TAUTENHAHN, L., 2006. Svg-vml-3d. http://www.lutanho.net/svgvml3d/index.html.
TIM JOHANSSON, O., 2007. Taking the canvas to another dimension.

16.http://my.opera.com/timjoh/blog/2007/11/13/takingthe-canvas-to-another-dimension.
.
17.http://dev.w3.org/html5/spec/Overview.html#the-canvaselement.W3C, 2009.Scalable vector graphics.http://www.w3.org/Graphics/SVG/.
WATT, J. Svg authoring guidelines. http://jwatt.org/svg/authoring/.WEB3DCONSORTIUM. Scene access interface(sai), iso/iec cd 19775-2 ed. 2:200x.
18.http://www.atomontage.com/

19.http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter37.html

20.http://www1.eafit.edu.co/cadcamcae/documents/draft_MEDX3DOM.pdf

21.http://cdn.intechopen.com/pdfs/34474/InTech-Volume_ray_casting_in_webgl.pdf

22.http://www.semcity.net/cms/uploads/docs/3u3d2012_s6_congote_et_al.pdf

23.http://web3d2013.web3d.org/

24.http://www.crs4.it/vic

25.http://www.x3dom.org/

26.http://en.wikipedia.org/wiki/VRML

27.http://www.w3.org/MarkUp/VRML/

Mr. GOPI KRISHNA
M-TECH(CSE)
St.Peter's Eng. College

Mrs. B.V. SeshuKumari
Assoc. Professor & HOD
St.Peter's Eng. College
hemakrishna.ambati@gmail.com
Area of Interest: Natural language processing, Pattern recognition, Speech processing.