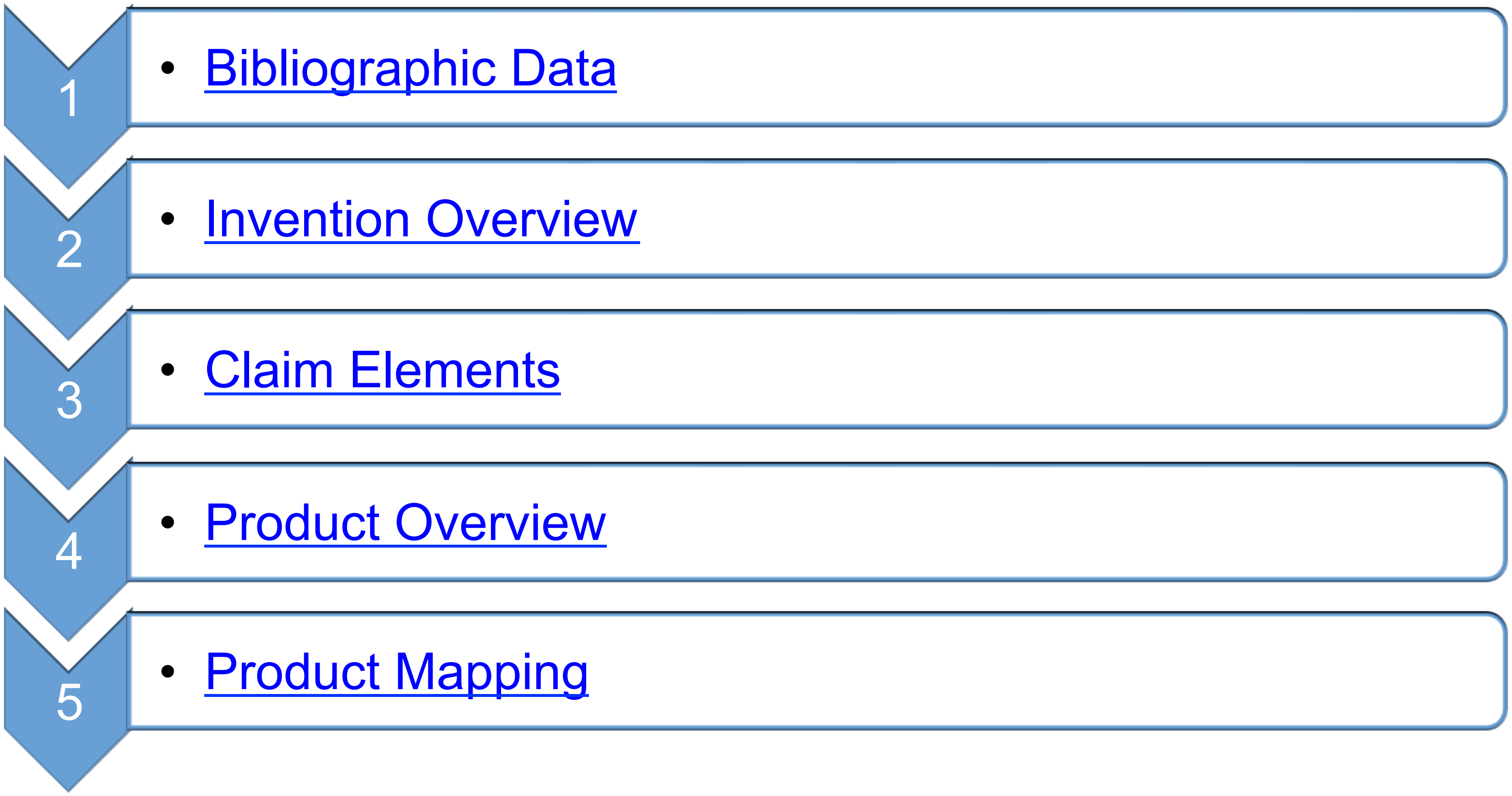
US6538654 Claim Chart

Microsoft Silverlight

Disclaimer

The patent right(s) referenced in these materials are offered for acquisition through a sale as of the date of this publication. The patent owner may, at any time, in its sole discretion and without prior notice, modify, substitute or withdraw this offering, and may modify any and all terms and conditions related to the acquisition including, but not limited to, pricing. Neither these materials nor the patent-related information referred to in these materials is intended to constitute legal analysis, notice or accusation of infringement, or to claim or assert that any party has violated any intellectual property rights or law. Prospective purchasers must rely on their own examination and evaluation of the patent(s), including but not limited to review of associated patent applications, patent family members, file histories, and other relevant information, in determining the value or applicability of the patent(s) with respect to any activities, products, or services. Any purchase is to be made independent of these materials. Neither the patent owner nor Drakes Bay Company makes any representation or warranties regarding the patent(s), the information contained in or referred to in these materials, or any other information supplied in connection with this offering. The patent owner and Drakes Bay Company expressly disclaim any warranty of merchantability or fitness for a particular purpose in connection therewith. The patent owner, in its sole discretion, may reject any and all offers for any reason. No obligations or commitments, whether express, implied or otherwise, shall be enforceable against the patent owner unless and until a final, written agreement is executed by and between the patent owner and a prospective purchaser.

Contents



1
- Bibliographic Data

2
- Invention Overview

3
- Claim Elements

4
- Product Overview

5
- Product Mapping

Bibliographic Data

US6538654: System And Method For Optimizing 3D Animation And Textures

Issue Date: March 25, 2003

Filing Date: December 23, 1999

Priority Date: December 24, 1998

Inventors: Anthony Rose, Andrew D. Davie, Alexis Vuillemin.

Original Assignee: B3D Inc.

Current Assignee: B3D Inc.

Independent Claims: 9 (Claim #1, #6, #7, #8, #10, #11, #12, #14 and #15)

Total Claims: 15

Applications: The invention can be used in following application areas :

- 3D animation object development and rendering software
- major animation film making studios.

Summary: The subject patent relates to 3D animation software. It discloses a method for determining and removing the animation data of 3D objects (actors) which are outside the view frame.

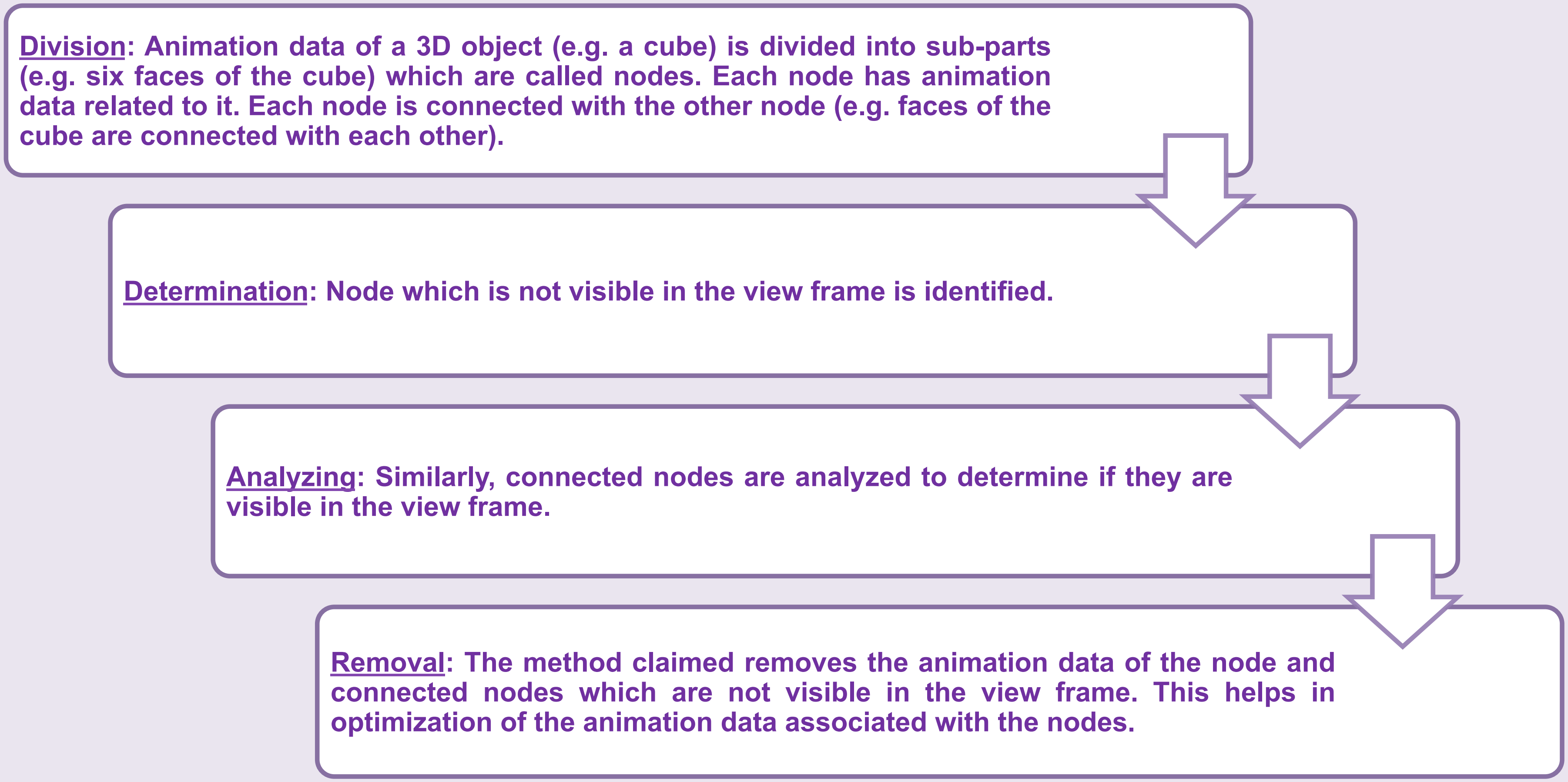
U.S. Family Members: 5

Non-U.S. Family Members: 0

Invention Overview

Subject patent describes the process of creating the animation data for 3D objects (Actors, Props, Cube etc.) in an optimized manner.

Below is a flow chart illustrating the method in the claimed invention:



Division: Animation data of a 3D object (e.g. a cube) is divided into sub-parts (e.g. six faces of the cube) which are called nodes. Each node has animation data related to it. Each node is connected with the other node (e.g. faces of the cube are connected with each other).

Determination: Node which is not visible in the view frame is identified.

Analyzing: Similarly, connected nodes are analyzed to determine if they are visible in the view frame.

Removal: The method claimed removes the animation data of the node and connected nodes which are not visible in the view frame. This helps in optimization of the animation data associated with the nodes.

Note: Steps are provided by the analyst for better visualization of the claimed invention.

CLAIM ELEMENTS

This section presents the shortlisted claim used for mapping. It also provides an overview on what information has been identified and to what extent. Below is the color-coding used for this purpose –

- **Green** text represents a sub-element, relevant information for which is directly and explicitly available in the product.
- *Blue* text represents additional important information related to a topic/subtopic added by the analyst team.

Elements and Sub Elements of Claim 1

Preamble:

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, the 3D object having a hierarchy of parent nodes and children nodes, each node being associated with animation data, a method for optimizing the animation data associated with each node comprising:

Clause 1:

determining if the node is outside a view frame;

Clause 2:

determining if any child node associated with the node is outside the view frame; and

Clause 3:

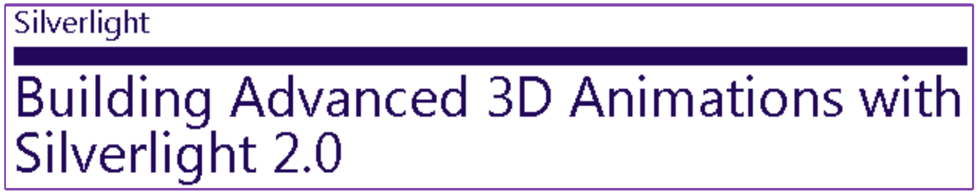
removing the animation data associated with the node if the node and any associated child node are outside the view frame.

PRODUCT OVERVIEW

This section provides an overview of the product mapped with the subject patent. This information is provided to aid understanding of the mapping by providing a glance at the product beforehand.

Purple text and boxes represent comments added by the analyst team to improve understanding.

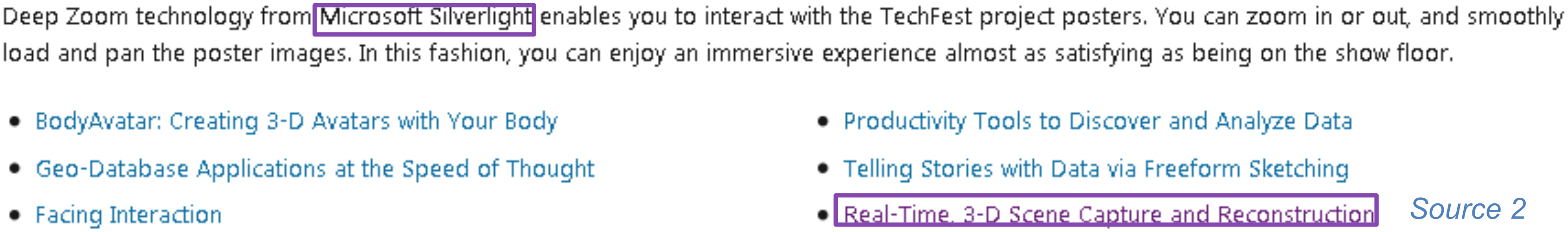
Company: Microsoft
Product: Silverlight



Silverlight

Building Advanced 3D Animations with Silverlight 2.0

Source 1



Deep Zoom technology from Microsoft Silverlight enables you to interact with the TechFest project posters. You can zoom in or out, and smoothly load and pan the poster images. In this fashion, you can enjoy an immersive experience almost as satisfying as being on the show floor.

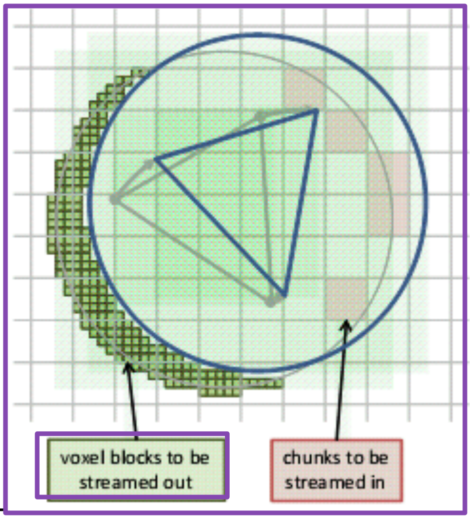
- BodyAvatar: Creating 3-D Avatars with Your Body
- Geo-Database Applications at the Speed of Thought
- Facing Interaction
- Productivity Tools to Discover and Analyze Data
- Telling Stories with Data via Freeform Sketching
- Real-Time, 3-D Scene Capture and Reconstruction

Source 2

Real-time 3D Reconstruction at Scale using Voxel Hashing

Source 3

Analyst Comment: Microsoft Silverlight is software used for creating 3D animation content. Silverlight optimizes 3D animation data by removing voxel blocks (portions of 3D objects) which are not in the field of view.



Source 3

Source1: http://msdn.microsoft.com/en-us/magazine/cc500570.aspx

Source2: http://research.microsoft.com/en-us/um/redmond/events/techfest2013/projects.aspx

Source 3: http://research.microsoft.com/pubs/217574/voxelhash.pdf (Page 1 and 6 of 11)

Company: Microsoft
Product: Silverlight

Market information:

The U.S. registers the maximum number of patents due to government support and the presence of **industry giants such as Microsoft Corporation**, Electronic Arts Inc, and Disney among others.

The global animation and gaming market is expected to grow from $122.20 billion in 2010 to $242.93 billion by 2016. This represents a compound annual growth rate (CAGR) of 12.94% from 2011 to 2016.

Analyst Comment: Microsoft is one of the industry giants of global animation.

Note: Market information is taken from unofficial sources as the same is not available on official website of Microsoft

Source: http://www.marketsandmarkets.com/Market-Reports/animation-gaming-market-514.html

PRODUCT MAPPING

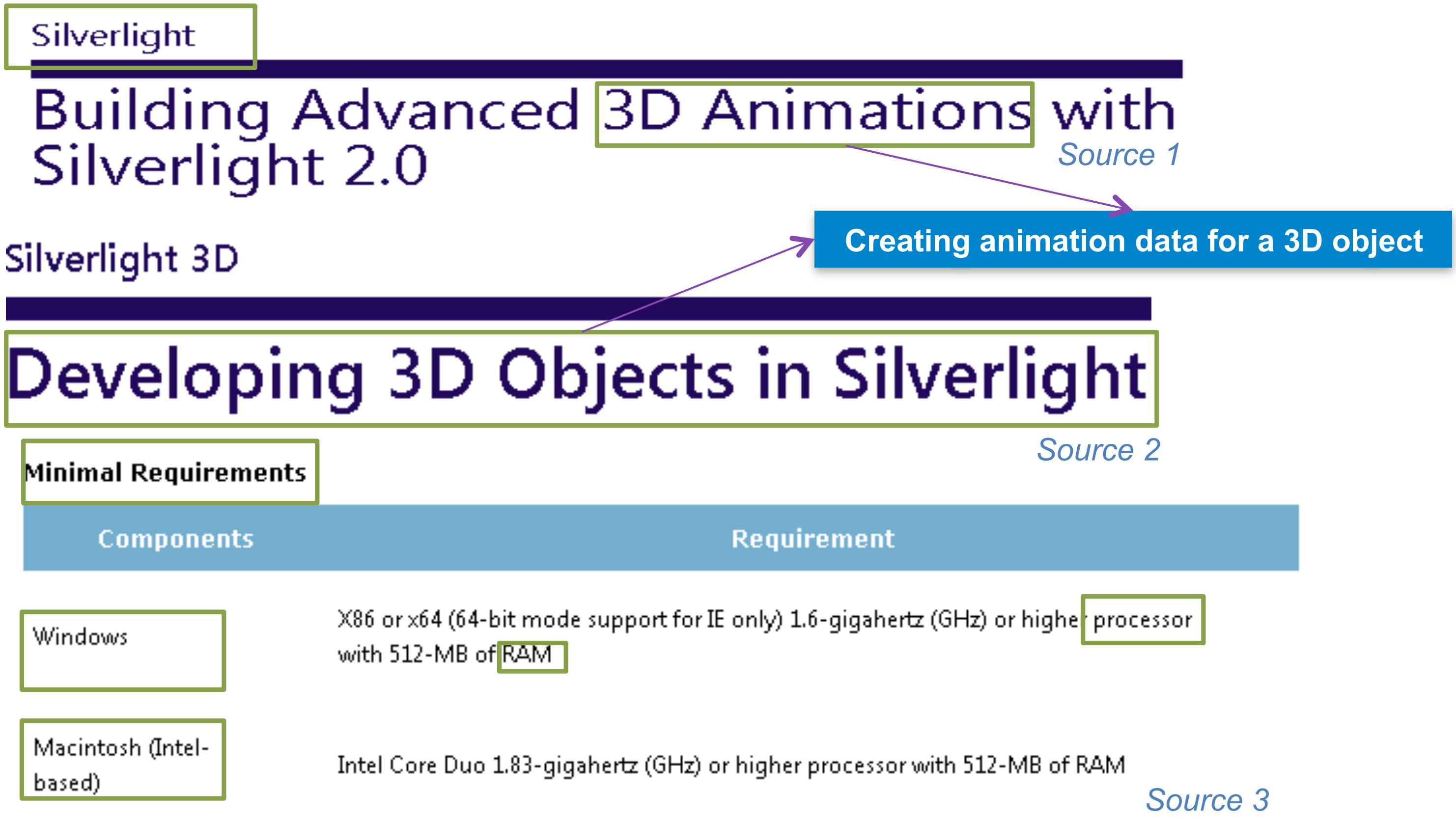
This section provides element-by-element mapping of the subject claim with the product identified. Below is the color-coding used for this purpose –

- **Green** text represents a sub-element for which relevant information is explicitly available.
- **Purple** text represents comments and notes added by the analyst team to improve understanding.
- **Blue box** represents the claim element that is being mapped with the product.

Note: To explicitly show presence of some claim elements, we have used references/ text which are not from official website of Microsoft.

Claim 1 (Preamble) (I)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, the 3D object having a hierarchy of parent nodes and children nodes, each node being associated with animation data, a method for optimizing the animation data associated with each node comprising:



Silverlight

Building Advanced 3D Animations with Silverlight 2.0

Source 1

Creating animation data for a 3D object

Silverlight 3D

Developing 3D Objects in Silverlight

Source 2

Minimal Requirements

Components	Requirement
Windows	X86 or x64 (64-bit mode support for IE only) 1.6-gigahertz (GHz) or higher processor with 512-MB of RAM
Macintosh (Intel-based)	Intel Core Duo 1.83-gigahertz (GHz) or higher processor with 512-MB of RAM

Source 3

Analyst Comment: Microsoft Silverlight is software which requires a computer system (windows, processor, RAM etc.) to run. Silverlight creates animation data for 3D objects (the objects occurring in 3D content).

Source1: http://msdn.microsoft.com/en-us/magazine/cc500570.aspx

Source 2: http://msdn.microsoft.com/en-us/magazine/hh547098.aspx

Source 3: http://www.microsoft.com/getsilverlight/Get-Started/Install/Default.aspx

Claim 1 (Preamble) (II)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, **the 3D object having a hierarchy of parent nodes and children nodes,** each node being associated with animation data, a method for optimizing the animation data associated with each node comprising:

*Description of “**3D object having a hierarchy of parent nodes and children nodes”** from the subject patent.*

> **Actors and props (collectively referred to as actors) are composed of a hierarchy of nodes.** The hierarchy begins with a root node and proceeds down to other nodes, each node being associated with a discrete piece of 3D geometry (group of polygons) making up the 3D object. Each node is further identified by a node name. For example, a node representing an object's head might be named a “head” node.
>
> Each node has zero or more parent and child nodes, with the restriction that the linkages cannot form a loop. Thus, **a trunk node may have a leg node as one of its children, and a head node as one of its parents.**

Analyst Comment: According to the subject patent, a 3D object e.g. an actor, is made up of various 3D geometries and each 3D geometry is associated with a unique node. A node has other nodes connected to it, called parent and child nodes. For example, a trunk node may have a head node as a parent and a leg node as a child connected to it.

Source: http://patentimages.storage.googleapis.com/pdfs/US6538654.pdf (Column 5, Line 58)

Claim 1 (Preamble) (III)

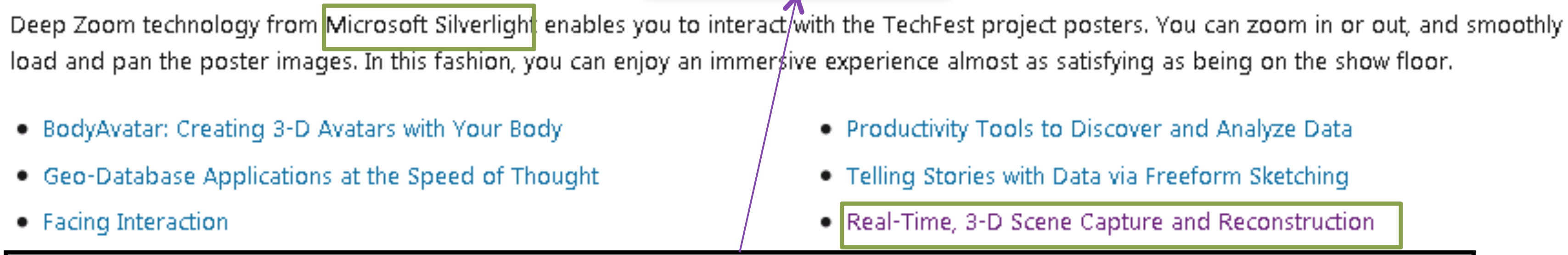
In a computer system for creating animation data for a 3D object appearing in a 3D animated content, **the 3D object having a hierarchy of parent nodes and children nodes,** each node being associated with animation data, a method for optimizing the animation data associated with each node comprising:

Silverlight 3D

Developing 3D Objects in Silverlight

Source 1

3D Objects



Deep Zoom technology from Microsoft Silverlight enables you to interact with the TechFest project posters. You can zoom in or out, and smoothly load and pan the poster images. In this fashion, you can enjoy an immersive experience almost as satisfying as being on the show floor.

- BodyAvatar: Creating 3-D Avatars with Your Body
- Geo-Database Applications at the Speed of Thought
- Facing Interaction
- Productivity Tools to Discover and Analyze Data
- Telling Stories with Data via Freeform Sketching
- Real-Time, 3-D Scene Capture and Reconstruction

This project is a novel method for real-time, **3-D scene capture and reconstruction.** Using several live color and depth images, this technology builds a high-resolution **voxelization of visible surfaces**.

Source 2

We represent voxels as index triples corresponding to the 3D spatial location of a cell within a discrete voxel lattice (grid). **A voxelization is an array (list) of such voxels.**

Source 3

Analyst Comment: In Silverlight 3D, object surfaces are made up of voxels. A voxel can be considered analogous to nodes as described on the next slide.

Information on Next Slide

Source 1: http://msdn.microsoft.com/en-us/magazine/hh547098.aspx

Source 2: http://research.microsoft.com/en-us/um/redmond/events/techfest2013/projects.aspx

Source 3: http://research.microsoft.com/en-us/um/people/cloop/HPG2013.pdf (Page 1 of 7)

Claim 1 (Preamble) (IV)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, **the 3D object having a hierarchy of parent nodes and children nodes,** each node being associated with animation data, a method for optimizing the animation data associated with each node comprising:

*Description of "**voxel**" from general literature.*

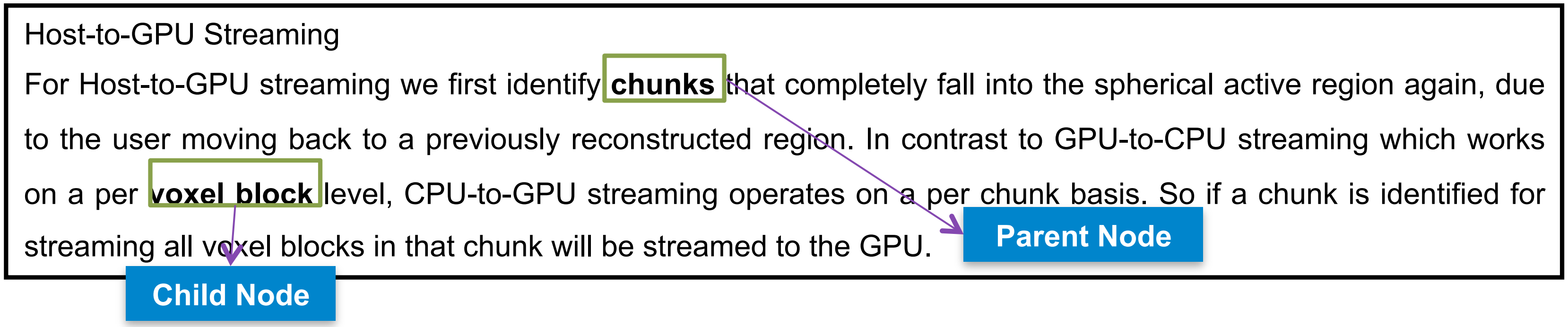
(In computer-based modelling or graphic simulation) each of an array of elements of volume that constitute a notional three-dimensional space, especially each of **an array of discrete elements into which a representation of a three-dimensional object is divided.**

Source 1

<u>Analyst Comment</u>: According to general literature, voxels are array of elements representing 3D objects.

Relevant information from product literature.

Voxel blocks are appended to these chunks using a linked list.



Host-to-GPU Streaming

For Host-to-GPU streaming we first identify **chunks** that completely fall into the spherical active region again, due to the user moving back to a previously reconstructed region. In contrast to GPU-to-CPU streaming which works on a per **voxel block** level, CPU-to-GPU streaming operates on a per chunk basis. So if a chunk is identified for streaming all voxel blocks in that chunk will be streamed to the GPU.

Source 2

<u>Analyst Comment</u>: A 3D object created in Silverlight is made up of chunks and voxel blocks which correspond to parent and child node respectively.

Source 1: http://www.oxforddictionaries.com/definition/english/voxel

Source 2: http://research.microsoft.com/pubs/217574/voxelhash.pdf (Page 7 of 11)

Claim 1 (Preamble) (V)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, the 3D object having a hierarchy of parent nodes and children nodes, **each node being associated with animation data,** a method for optimizing the animation data associated with each node comprising:

*Description of “**nodes being associated with animation data”** from the subject patent.*

Accordingly, the system and method includes retrieving **the texture map of the polygon making up the node**, and rendering the polygon according to the retrieved texture map onto a screen space.

Analyst Comment: According to the subject patent each node is associated with the texture map of the polygon (i.e. animation data).

Source http://patentimages.storage.googleapis.com/pdfs/US6538654.pdf (Column 3, Line 42)

Claim 1 (Preamble) (VI)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, the 3D object having a hierarchy of parent nodes and children nodes, **each node being associated with animation data,** a method for optimizing the animation data associated with each node comprising:

Analyst Comment: 3D objects animated in Silverlight are made up of voxels blocks. Thus each voxel (as being a part of 3D object) is also associated with some animation data.

Figure 1: *Several views of a model, with and without texture reconstructed from eight cameras. This surface contains 2.5 million out of* 1024^3 *voxels. Computation plus rendering took 24 milliseconds.*

Node

Animation data

Analyst Comment: In Microsoft Silverlight, texture (i.e. animation data) is constructed for all voxels. It shows each voxel is associated with animation data.

Source : http://research.microsoft.com/en-us/um/people/cloop/HPG2013.pdf (Page 1 of 7)

Claim 1 (Preamble) (VII)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, the 3D object having a hierarchy of parent nodes and children nodes, each node being associated with animation data, **a method for optimizing the animation data associated with each node** comprising:

Description from the subject patent.

> In accordance with one aspect of the invention, **the animation and texture data of the 3D animation content is optimized to reduce the size of the animation data to be streamed over the Internet.** The system and method according to this second aspect of the invention gathers statistical information about the nodes and textures being utilized in each frame of the animated content. In doing so, **the system and method determines if a particular node and any of its children nodes are outside the view frame. If they are, then the animation data associated with the node is removed.**

Analyst Comment: According to the subject patent, optimization of animation data includes removing the data associated with a node which is outside the field of view.

Source: https://patentimages.storage.googleapis.com/pdfs/US6538654.pdf (Column 3, Line 17)

Claim 1 (Preamble) (VIII)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, the 3D object having a hierarchy of parent nodes and children nodes, each node being associated with animation data, **a method for optimizing the animation data associated with each node** comprising:

*Definition of "**culling**" from general Literature:*

> **In 3D rendering the term culling describes the early rejection of objects of any kind (objects, draw calls, triangles and pixels) that don't contribute to the final image.**

Source 1

> Host-to-GPU Streaming
>
> For Host-to-GPU streaming we first identify chunks that completely fall into the spherical active region again, due to the user moving back to a previously reconstructed region. In contrast to GPU-to-CPU streaming which works on a per voxel block level, CPU-to-GPU streaming operates on a per chunk basis. So if a chunk is identified for streaming all voxel blocks in that chunk will be streamed to the GPU. **This enhances performance, given the high host-GPU bandwidth and ability to efficiently cull voxel blocks outside of the view frustum.**

Optimizing the animation data

Source 2

Analyst Comment: Microsoft Silverlight provides a method of optimizing performance by removing or culling the voxels which are outside view of user.

Source 1: http://docs.cryengine.com/display/SDKDOC4/Culling+Explained

Source 2: http://research.microsoft.com/pubs/217574/voxelhash.pdf (Page 7 of 11)

Claim 1 (Clause 1) (I)

determining if the node is outside a view frame;

*Description of "**view frame**" from the subject patent.*

> The animation inaccuracy of the child node does not affect the quality of the 3D movie if the child node is **not visible (i.e. outside the view frame).**

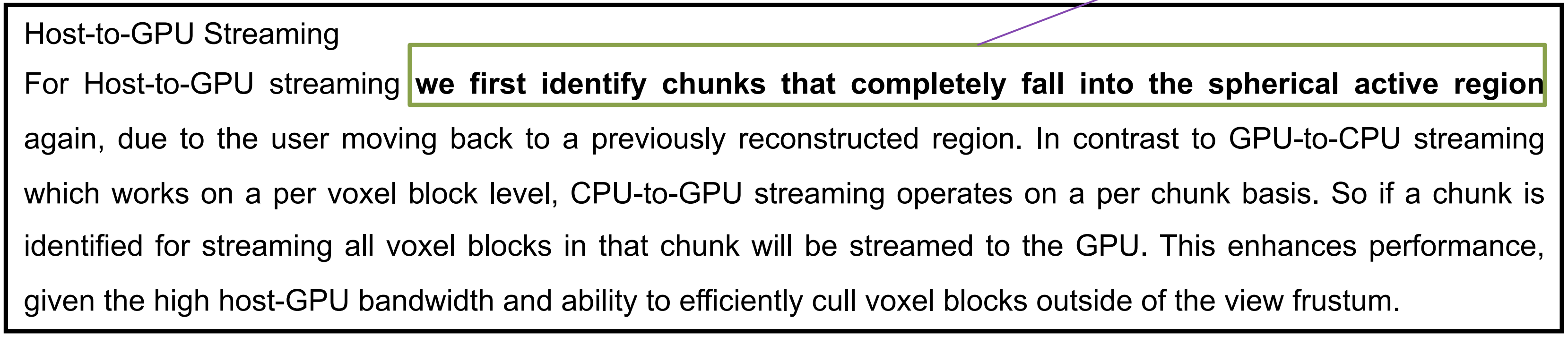
Analyst Comment: According to the subject patent, view frame refers to the field of view in which the node is visible and is not obscured by the other nodes.

Source: http://patentimages.storage.googleapis.com/pdfs/US6538654.pdf (Column 65, Line 52)

Claim 12 (Clause 1) (II)

determining if the node is outside a view frame;

Determine if the node is outside the view frame



Host-to-GPU Streaming

For Host-to-GPU streaming **we first identify chunks that completely fall into the spherical active region** again, due to the user moving back to a previously reconstructed region. In contrast to GPU-to-CPU streaming which works on a per voxel block level, CPU-to-GPU streaming operates on a per chunk basis. So if a chunk is identified for streaming all voxel blocks in that chunk will be streamed to the GPU. This enhances performance, given the high host-GPU bandwidth and ability to efficiently cull voxel blocks outside of the view frustum.

Analyst Comment: Microsoft Silverlight identifies the chunks (i.e. parent node) that are in the spherical region or view frustum or view frame. Thus, reciprocally it identifies the chunks which are outside the view frame.

Source: http://research.microsoft.com/pubs/217574/voxelhash.pdf (Page 7 of 11)

Claim 12 (Clause 2)

determining if any child node associated with the node is outside the view frame; and

An oracle kernel decides, for each voxel in parallel, whether to keep or cull its voxel from the list based on an image consistency criterion of its projection across cameras.

Source 1

Determine if the child node is outside view frame

Each voxel in the grid contains two values: a signed distance and weight. For a single depth map, data is integrated into the grid by uniformly sweeping through the volume, **culling voxels outside of the v ew frustum,** projecting all voxel centers into the depth map, and updating stored SDF values.

Source 2

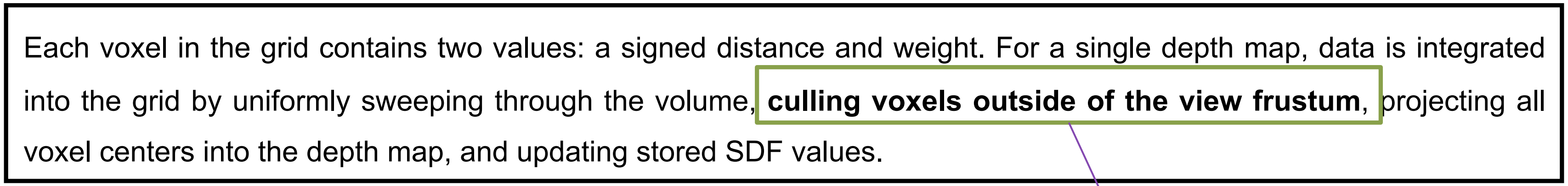
Analyst Comment: Microsoft Silverlight software determines whether to keep or cull a voxel block (i.e. child node) based on criterion of its projection across cameras (i.e. view frame).

Source 1: http://research.microsoft.com/en-us/um/people/cloop/HPG2013.pdf (Page 1 of 7)
Source 2: http://research.microsoft.com/pubs/217574/voxelhash.pdf (Page 3 of 11)

Claim 1 (Clause 3)

removing the animation data associated with the node if the node and any associated child node are outside the view frame.

Relevant information from product literature.



Each voxel in the grid contains two values: a signed distance and weight. For a single depth map, data is integrated into the grid by uniformly sweeping through the volume, **culling voxels outside of the view frustum**, projecting all voxel centers into the depth map, and updating stored SDF values.

Removing animation data associated with the node

Analyst Comment: In Microsoft Silverlight, chunk (i.e. parent node) and voxel blocks (i.e. child node) are culled and animation data is removed if they are outside the view frame.

Source: http://research.microsoft.com/pubs/217574/voxelhash.pdf (Page 3 of 11)

For additional information, please contact:

- Joseph W. Jennings
  - 1-415-927-2716
  - jjennings@drakesbaycompany.com

- Marisa Bracoloni
  - 1-415-927-2716
  - mbracoloni@drakesbaycompany.com