Technical Disclosure Commons

Defensive Publications Series

January 19, 2016

OCCLUDING 3D MAP LABELS USING BACK FACES

Janne Kontkanen

Joshua Schpok

Recommended Citation

Kontkanen, Janne and Schpok, Joshua, "OCCLUDING 3D MAP LABELS USING BACK FACES", Technical Disclosure Commons, (January 19, 2016)
http://www.tdcommons.org/dpubs_series/114

OCCLUDING 3D MAP LABELS USING BACK FACES

ABSTRACT

A 3D map labeling system places 2D labels at a depth that is unoccluded by the labeled 3D element of the 3D map but may be occluded by a different 3D element. The 3D map labelling system renders back faces of a 3D model into a Z-buffer. The system then renders a label at a predetermined depth and performs a Z-test by testing the predetermined depth of the label against the Z-buffer. The system uses the Z-test to identify the occluded and unoccluded pixels in the label and uses this information to hide labels behind occluding elements, or style them differently, for example, with transparency.

PROBLEM STATEMENT

Placing labels on maps is recognized as a difficult problem. The problem becomes challenging in 3D, where labels are not only placed among themselves, but with other elements in front of labeled elements relative to the view, such as trees, buildings, and terrain. Furthermore, labels are often prescribed to an individual point, while the entity it labels occupies a volume. When the 3D geometry and label geometry are generated separately from each other, the resulting rendering sometimes lacks a clear distinction between what subset of the world's geometry is associated with a particular label. Accordingly, there is an opportunity to render labels in a 3D map in a manner that is more readable to a user.

3D MAP LABELING SYSTEM

The systems and techniques described in this disclosure relate to a 3D map labeling system. The system can be implemented for use in an Internet, an intranet, or another client and server environment. The system can be implemented as program instructions locally on a client device or implemented across a client device and server environment. The client device can be any electronic device such as a GPS navigation device, a mobile device, a smartphone, a tablet, a handheld electronic device, a wearable device, a laptop computer, a desktop computer, etc.

Fig. 1 illustrates an example method 100 to identify occluded and unoccluded pixels in a label for rendering of labels in 3D maps. FIG. 2 provides a conceptual example of elements from FIG. 1.

The system renders 110 back faces of a 3D model into a Z-buffer. The 3D model can be a model for a roadway which may include buildings present on both sides of the roadway, trees present on both sides of the road, traffic lights in the middle of the roadway, public conveniences present on one side of the roadway, etc. The 3D model can be generated for any three-dimensional surface of an object using specialized software in 3D computer graphics. In an embodiment, the system may render only back faces of the 3D model into the Z-buffer. The back faces of the 3D model may refer to surfaces of the 3D model facing away from a viewer. Z-buffering is also known as depth buffering and is defined as the management of image depth coordinates in three-dimensional (3-D) graphics. See FIG. 2 object cluster 210 illustrating a 3D mesh geometry of object 213 [back dark] and object 215 [back gray] behind object 217 [front light-gray]. Object cluster 220 depicts mesh backfaces of the 3D object cluster 210 rendered into a Z-buffer.

The system then renders 120 a label at a predetermined depth as shown in object cluster 230. A label can name one or more objects in the map, provide instructions to a user linked to an object, or provide other information about a map object. For example, as can be seen in Fig. 2, an object 213 can have a label such as “Capitol Building”. The predetermined z-depth may be a prefixed value set or adjusted by a system programmer or a user of the system. The predetermined z-depth may be related to the z-depth of the associated map object 213 of the label within the Z-buffer. In this FIG. 2, the label at the predetermined z-depth is “in front” or “above” the background objects 213, 215 and “behind” or “below” the foreground object 217.

The system then performs 130 a Z-test by testing the predetermined depth of the label against the Z-buffer. The Z-test in this context means comparing two z-coordinate values to each other to find out the smaller z-coordinate value. If the label’s z-value is larger than the z-value stored in the Z-buffer, the Z-test fails.

The system further identifies 140 the occluded and unoccluded pixels in the label based on the z-test. Pixels failing the z-test indicates that the depth of the label falls behind a surface and is identified as occluded whereas pixels succeeding the z-test are in front of a surface and are identified as unoccluded (as shown in block 230, Fig. 2. The pixels which are occluded are shown marked in the circle 240). By z-testing against back faces, the system allows the labels to be rendered on top of selective building geometry in front of the label. A label for a building may not be precisely located, but as long as it occurs somewhere within the building’s bounds, the building’s back faces should render behind the label, permitting the label to render unoccluded.

Fig. 3 illustrates an example showing various implementations of the aforementioned 3D map labelling system. The system can utilize this knowledge regarding occluded and unoccluded pixels of the 2D label to hide or restyle label pixels depending on occlusion relative to each occluding object. For example, block 310 shows an implementation with some label pixels of “Capitol Building”, which is associated with building 313, hidden behind a foreground building 317 based on occlusion but other pixels of the label are not hidden and are “in front of” a different background building 315. Block 320 indicates another implementation with label pixels for "Capitol Building" placed above the z-axis depth of buildings 313, 315 and which cover some part of the buildings 313, 315 based on occlusion. Block 330 provides another implementation where label background pixels are placed in front of the labeled building 313 as well as the other buildings 315, 317 and are kept transparent so that all of the building 315, 317 behind the label is still visible.

The subject matter described in this disclosure can be implemented in software and/or hardware (for example, computers, circuits, or processors). The subject matter can be implemented on a single device or across multiple devices (for example, a client device and a server device). Devices implementing the subject matter can be connected through a wired and/or wireless network. Such devices can receive inputs from a user (for example, from a mouse, keyboard, or touchscreen) and produce an output to a user (for example, through a display). Specific examples disclosed are provided for illustrative purposes and do not limit the scope of the disclosure.

DRAWINGS

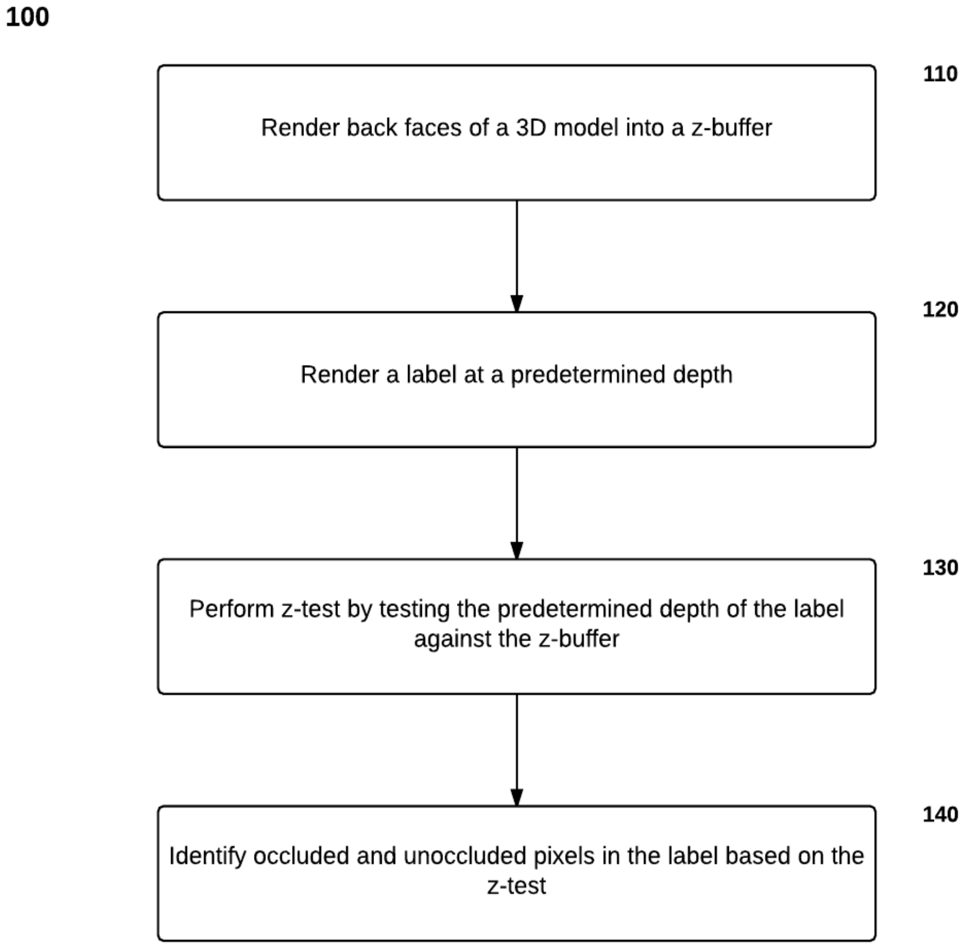
100
110
Render back faces of a 3D model into a z-buffer
120
Render a label at a predetermined depth
130
Perform z-test by testing the predetermined depth of the label against the z-buffer
140
Identify occluded and unoccluded pixels in the label based on the z-test

FIG. 1

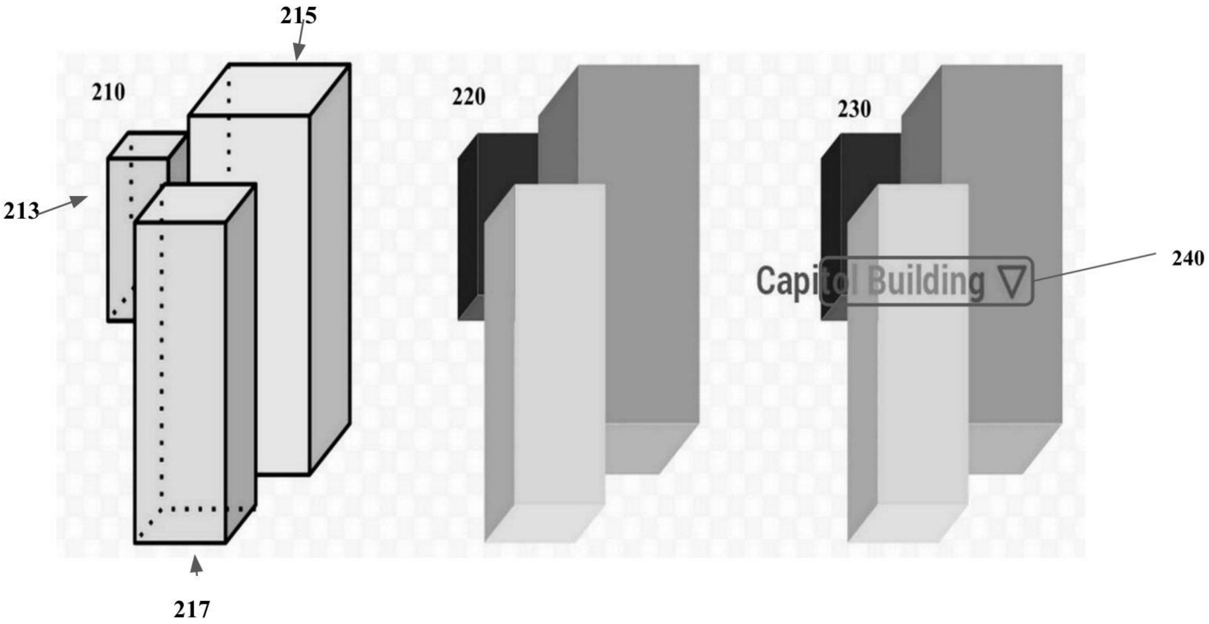
215
210
220
230
213
Capitol Building ▽
240
217

FIG. 2

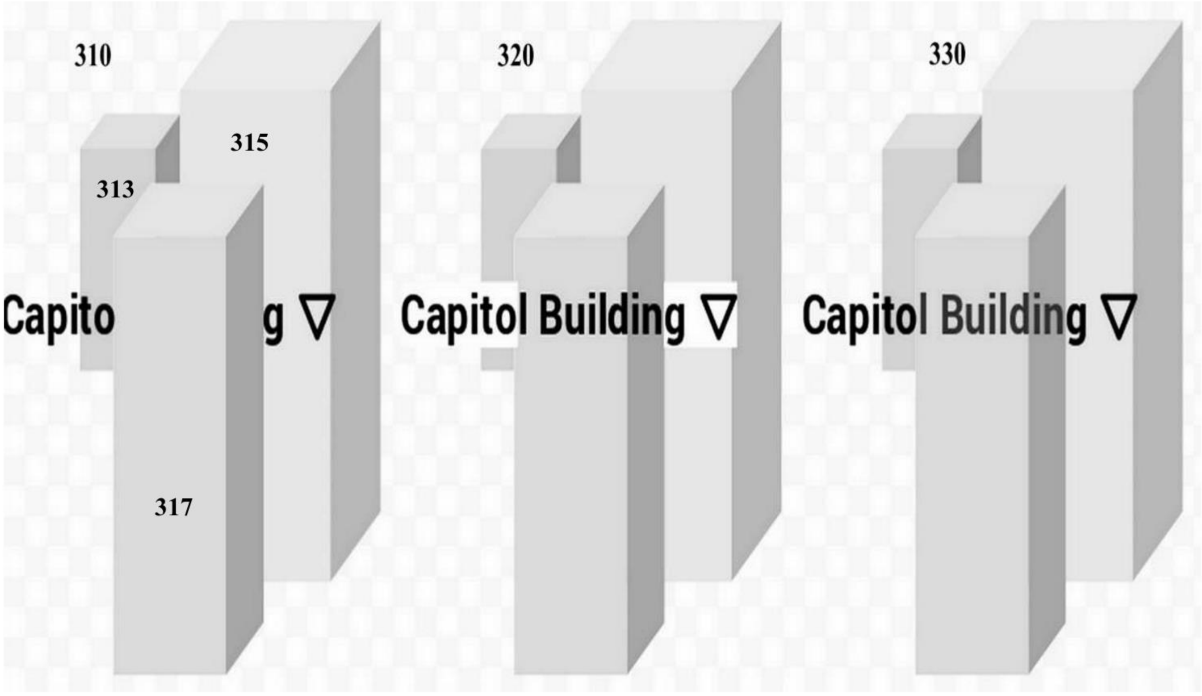
310
320
330
315
313
Capito g ▽
Capitol Building ▽
Capitol Building ▽
317

FIG. 3