Information Systems 42 (2014) 89–106

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Scalable visibility color map construction in spatial databases

Farhana Murtaza Choudhury [a], Mohammed Eunus Ali [a,*], Sarah Masud [b], Suman Nath [c], Ishat E. Rabban [a]

[a] Department of CSE, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
[b] Department of Computer Science, University of Virginia, VA, United States
[c] Microsoft Research, Redmond, WA, United States

ARTICLE INFO

Article history:
Received 14 November 2013
Received in revised form
10 December 2013
Accepted 14 December 2013
Recommended by: D. Shasha
Available online 21 December 2013

Keywords:
Spatial databases
Query processing
Three-dimensional (3D) objects
Visibility color map

ABSTRACT

Recent advances in 3D modeling provide us with real 3D datasets to answer queries, such as "*What is the best position for a new billboard?*" and "*Which hotel room has the best view?*" in the presence of obstacles. These applications require measuring and differentiating the visibility of an object (target) from different viewpoints in a dataspace, e.g., a billboard may be seen from many points but is readable only from a few points closer to it. In this paper, we formulate the above problem of quantifying the visibility of (from) a target object from (of) the surrounding area with a *visibility color map* (VCM). A VCM is essentially defined as a surface color map of the space, where each viewpoint of the space is assigned a color value that denotes the visibility measure of the target from that viewpoint. Measuring the visibility of a target *even* from a single viewpoint is an expensive operation, as we need to consider factors such as distance, angle, and obstacles between the viewpoint and the target. Hence, a *straightforward approach* to construct the VCM that requires visibility computation for every viewpoint of the surrounding space of the target is prohibitively expensive in terms of both I/Os and computation, especially for a real dataset comprising thousands of obstacles. We propose an efficient approach to compute the VCM based on a key property of the human vision that eliminates the necessity for computing the visibility for a large number of viewpoints of the space. To further reduce the computational overhead, we propose two approximations; namely, *minimum bounding rectangle* and *tangential* approaches with guaranteed error bounds. Our extensive experiments demonstrate the effectiveness and efficiency of our solutions to construct the VCM for real 2D and 3D datasets.

1. Introduction

Recent advances in large-scale 3D modeling have enabled capturing urban environments into 3D models. These 3D models give photo-realistic resemblance of urban objects such as buildings, trees, and terrains and are widely used by popular 3D mapping services, e.g., Google Maps, Google Earth, and Bing Maps. The increasing availability of these realistic 3D datasets provides us an opportunity to answer many real-life queries involving visibility in the presence of 3D obstacles. For example, an advertising company may wish to determine the visibility of their existing billboards from the surrounding areas in order to find a suitable location to place a new billboard; the police may check the visibility of a surveillance camera to find how well it covers its surrounding areas; and an apartment buyer may want to check the visibility of the nearby sea-beach and mountains from various available apartments.

---

[*] Corresponding author. Tel.: +88 029665612; fax: +88 029665612.
E-mail addresses: farhanamc@gmail.com (F.M. Choudhury), eunus@cse.buet.ac.bd (M.E. Ali), sp9cx@virginia.edu (S. Masud), sumann@microsoft.com (S. Nath), ieranik@yahoo.com (I.E. Rabban).

0306-4379/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2013.12.002

In this paper, we investigate efficient techniques to answer the underlying query required by the above applications: computing visibility of an object (e.g., billboard) from the surrounding continuous space, or that of the surrounding space from a source viewpoint (e.g., camera). Our target applications treat visibility as a continuous notion—e.g., a billboard may be more visible from one location than another, depending on factors such as distance, viewing angle, and obstacles between the viewpoint and the target. Such quantification of visibility is important, because a billboard can be visible from a number of places, but may not be readable from all of them. We therefore use a *visibility function* that provides real-valued visibility measures of various points in the (discretized) 3D space, where the visibility measure of a point denotes its visibility from the viewpoint or to the target object. Thus, the answer to our target query is essentially the visibility measures for every point in the 3D space. The result can be graphically represented as a *heat map*, by assigning colors to various points according to their visibility measures. We call this a *visibility color map* (VCM) of the space for a given target or for a given viewpoint.

Recent works have shown how database techniques can enable efficiently answering various types of visibility queries in the presence of obstacles. Various nearest neighbor (NN) queries consider visibility of objects [1–3]; for example, the visible nearest neighbor query [1] finds the nearest neighbors that are visible from the source. However, these works, like various other computer graphics works [4–9], treat visibility as a binary notion: a point is either visible or not from another point. In contrast, in our target applications, visibility is a continuous notion. Recently, Masud et al. proposed techniques for computing continuous visibility measure of a target object from a particular point in 3D space (e.g., computing visibility of a billboard from a given location) [10]. On the contrary, our target applications require visibility calculation from or of a continuous space, not from a user specified location.

One straightforward way to generate a VCM is to discretize the 3D space and to use the techniques in [10] to compute visibility measure for each discrete point in the space. However, this can be prohibitively expensive. For example, discretizing the surrounding space into 1000 points in each dimension would give a total of 10^9 points in the 3D space; and computing visibility measure for each point by using techniques in [10] would take 128 days! The huge cost comes from two sources: (i) computing the visibility measure based on the distance and angle from all viewpoints, which is computationally expensive and (ii) accessing a large set of obstacles from the database, which is I/O expensive.

We address the above challenges with a three-step solution that uses several novel optimizations to reduce computational and I/O overhead. First, we partition the dataspace into a set of *equi-visible cells*, i.e., all points inside a cell have equal visibility of the target object in terms of visual appearance. We exploit the key observation that when a lens (e.g., a human eye) sees an object without any obstacles, it cannot differentiate between its visual appearances from a spatially close set of points within an angular resolution (or spatial resolution) of ≈ 4 arcmin ($\approx 0.07^\circ$) [11]. Thus, we can safely prune the visibility computation for a large number of viewpoints within the angular or spatial resolution without affecting viewer's perception. This optimization significantly reduces the computation cost, as we can compute only one visibility measure for each cell.

In the next step, we consider the effect of obstacles. We compute *visible regions*, the regions in the space from where the target object is completely visible in the presence of obstacles. In the final step, we assign visibility measures to these regions from the corresponding cells by spatial joins. Both steps are I/O and computation intensive. For example, they both require retrieving a large number of cells and obstacles from the spatial database. To reduce I/O costs, we employ various indexing techniques to incrementally retrieve a small number of obstacles and cells near the target object. These steps also require performing many computationally expensive operations such as polygon intersections of irregular-shaped regions and cells. To reduce such costs, we represent regions with regular shaped polygons in a quad-tree. We also propose two approximations that further reduce the cost while providing guaranteed small error bounds.

We have evaluated the performance of our solution with real 3D maps of two big cities. We compare our solution with a *baseline* approach that divides the space into a regular shape grid of 500 cells in each dimension and computes visibility from each grid cell. The baseline approach results into more than 30% error while requiring about 800 times more computation time and six orders of magnitude more I/O than our solution. Hence, in the baseline approach, dividing the space into more cells for more accuracy is not feasible for practical applications. On the other hand, our approach provides efficient and effective solution.

In summary, we make the following contributions:

- We formulate the problem of efficiently constructing a visibility color map (VCM) in the presence of obstacles in 2D and 3D spaces.
- We propose an efficient solution to construct a VCM. The solution uses various novel optimizations to significantly reduce the computational and I/O overhead compared to a baseline solution.
- We propose two approximations with guaranteed error bounds and reduced computation to construct the VCM.
- We conduct extensive experiments in two real datasets to show the effectiveness and efficiency of our approaches.

2. Problem formulation and preliminaries

2.1. Problem formulation

The construction of a *visibility color map* (VCM) can be seen from two perspectives: *target-centric VCM* and *viewer-centric VCM*. The construction of a *target-centric*