Follow Moving Things in Virtual World

Jyh-Ming Lien*
Department of Computer Science
George Mason University

Young J. Kim †
Department of Computer Science & Engineering
Ewha Womans University

Abstract

There are many scenarios in virtual reality and video games where it is desirable to assist human user and maximize his/her visibility of a group of moving targets. This paper presents a new type of space partitioning strategy based on the idea of visibility integrity to enhance the user experience in the context of solving the group following problem. This problem involves computing the motion for a mobile camera to maximize its visibility of a coherent group of targets moving about a space filled with obstacles. Current solutions to this problem involve approximation of the visibility between different areas of the space. These areas are usually defined by simple shapes such as discs, squares or rectangles, and have always been constructed somewhat arbitrarily as long as they can collectively cover the space. Consequently, such partitioning strategy usually suffers from both over partitioning and under partitioning in different regions of a given environment, thus significantly hinders the ability to provide reliable pursuit of a group. To address this challenge, we have developed a visibility-integrity roadmap data structure that provides a partition of the space into regions of similar visibility.

Keywords: virtual camera, visibility, group, motion planning

1 Introduction

There are many scenarios where it is desirable for a moving sensor to maximize its visibility of a group of moving targets. For example, in video games, a virtual camera may need to follow a group of characters as they move through complex virtual environments, such as inside and around buildings [Vo et al. 2012]. In computer graphics, it can be difficult to maintain visibility of important events occurring in a physically-based simulation, which usually involves a massive number of objects interacting with each other at high speed. In robotics, a camera may be mounted onto a mobile robot or unmanned aerial vehicle (UAV) to monitor another group of robots as they perform a cooperative task [Harrison et al. 2010; Wang et al. 2013; Vo et al. 2009], or observe groups of humans or animals as they move through complex environments [Reynolds 1987; Rodriguez and Amato 2010]. Even when a single target is followed, this target is usually modeled as a set of estimates of its actual state which is unknown due to sensing and modeling uncertainties [Schulz et al. 2003]. Such a problem of tracking one or more targets using a camera is called the group following problem, and we are usually interested in a camera motion planning method to plan the motions necessary for the camera to maximize its visibility of the targets. While camera tracking is a popular subject in the literature, the group following problem has not gained much attention despite its importance.

This paper presents a new type of space partitioning strategy based on the idea of visibility integrity to enhance the user experience in the context of solving the group following problem. In this problem, we are given $n$ points and are asked to determine the regions of the space that have visibility of all or a maximum number of these $n$ points. Since visibility computation can be expensive, and many of the applications (such as the ones mentioned above) use the information in real time, it is necessary to build a data structure that supports faster visibility queries. This data structure can be computed offline entirely before the visibility information is to be used in the application, or can be computed online to incrementally cache visibility information as it is computed. In this paper, we will explore offline computation of group visibility.

Historically, visibility methods can be classified into from-point and from-region visibility computations, which determine regions in the space visible from a point or a region, respectively. While the problem of computing the group visibility is close to from-region visibility computations, the term “region” in from-region visibility computation usually refers to static surfaces, such as walls and buildings. On the contrary, in the group visibility problem, we are interested in determining visible regions to a group of constantly moving targets. In addition, current solutions to the from-region visibility problem involve approximation of the visibility between different areas that are usually defined by simple shapes such as discs, squares or rectangles, and have always been constructed somewhat arbitrarily as long as they can collectively cover the space. Consequently, such partitioning strategy usually suffers from both over partitioning and under partitioning in different regions of a given environment, thus significantly hinders the ability to reliably pursue a group.

Main Contributions We propose a new data structure to encap-
ulate the visibility information of a known workspace in 2D. Each vertex in this graph data structure is a region in workspace with similar visibility, i.e., their visible areas are similar. Such a similarity is measured by a notation called visibility integrity. Specific definition of visibility integrity will be given in Section 3. Because each vertex in our roadmap is a region with bounded visibility integrity, we name the roadmap “visibility integrity roadmap” or simply VIR. An example of the proposed method assisting the user follow a group of 30 agents moving through a virtual city is shown in Figure 2. Our experiment results show that this new data structure allows the virtual camera to maintain better visibility of the existing strategies. A study further shows that, when users are assisted by this more intelligent virtual camera, their performances are improved noticeably in a task that involves navigating a group of coherent agents.

2 Related Work

Visibility related problems have been studied extensively in the fields of computer graphics [Sanokho et al. 2014; Galvane et al. 2013] and computational geometry [Durand 1999; Cohen-Or et al. 2003; Bittner et al. 2009]. A popular strategy to solve from-region visibility problem is to sample visibility information on a uniform grid covering the entire space. For example, Bittner et al. [Bittner et al. 2009] proposed a global visibility algorithm which uses samples of ray tracing to obtain visibility information with the goal of being able to determine the potentially visible set of objects that can be seen by a given view cell.

Most methods developed for camera motion planning have focused on maintaining the visibility of a single point [Oskam et al. 2009; LaValle et al. 1997; Li and Yu 1999; Li et al. 1999; Nieuwenhuisen and Overmars 2004; Geraerts 2009; Bandyopadhyay et al. 2006]. These methods usually explicitly or implicitly assume that maintaining the visibility of a single reference point (e.g., the center of mass) will provide the visibility to the entire target. However, most, if not all, targets in the aforementioned applications are not a single point and maintaining the visibility of the reference point usually does not provide good visibility to the entire target. To overcome this problem, we propose to model the targets as a set of points, and the camera’s task is to position itself in order to maximize the number of visible points. These target points can be a group of agents that can move coherently or split into smaller groups, a set of feature points from a rigid or deformable body, or a list joints from an articulated character.

Vo and Lien [Vo and Lien 2010] proposed a method based on the idea of monotonic tracking regions to follow a group of agents but their method is limited to 2D. Our goal is to plan the motion of a virtual camera in 3-d to maintain the visibility of a set of moving points. Our method is motivated by the visibility-aware roadmap [Oskam et al. 2009] which is designed to follow a single target. Their method creates a graph (called roadmap) to capture not only the connectivity but also the visibility of the free space, the space that is not occupied by obstacles. Their roadmap is created based on the overlapping spheres in free space and the visibility between the spheres. Vo and Lien [Vo and Lien 2010] adopted their methods to tracking multiple targets using the center of the targets. A major limitation of that method comes from the fact that the quality of the roadmap depends on the type of the environments and the size of the discs, which is currently up to the users to decide.

Recently, Christie et al. [Christie et al. 2012] proposed an approach for maintaining the visibility of one or more target objects. In contrast to existing approaches which use ray casting to compute visibility, they use the principle of soft shadow generation to quickly evaluate visibility. An advantage of this approach is the ability to aggregate the visibility of multiple targets over time, and to evaluate the visibility of the whole target rather than just single points. While their method also aims to compute visibility for multiple targets, each target in the group is processed independently. On the other hand, our method clusters the members in a group into regions (i.e., circles) of similar visibility. As a result, our approach is more suitable for computing the visibility of large groups.

3 Visibility-integrity Roadmap (VIR)

The structure of visibility-integrity roadmap encoded both visibility and navigation information. Each node in this roadmap is a visibility-integrity region, which essentially is composed of a cluster of points that see similar things in workspace. The edges in the roadmap shows the adjacency of the visibility-integrity regions. To
partition the space into visibility-integrity regions, we first sample \( n \) points \( \mathcal{S} \). As we will show later these samples can be drawn randomly or from cell centers of a regular grid.

The visibility integrity measures how likely a set of points are indeed visible by another set of points. We compute and update visibility integrity to estimate the loss after merging two clusters. Let \( C \) be a cluster of points. If \( C \) contains only a single point \( p \), its visibility \( V(C) \) is simply all points visible from \( p \). If \( C \) contains multiple points, then \( V(C) \) is the union of all points visible from the points in \( C \), i.e., \( V(C) = \bigcup_{p \in C} V(p) \).

![Figure 3: Left: a cluster of 10 points with high visibility integrity. Right: a cluster of 10 points with low visibility integrity in the same environment.](image)

Then, we define the visibility integrity of \( C \) as the average of visibility ratio of all points in \( V(C) \), i.e.,

\[
\text{vi}(C) = \frac{q \in V(C)}{\|V(q) \cap C\|} \cdot \|V(C)\| \quad (1)
\]

The value of \( \text{vi}(C) \) is also bounded between zero and one. Intuitively when visibility integrity is low, that means there are regions that are visible by some points in the cluster but are not visible from most of the points. Examples in Figure 3 illustrate the clusters with high and low visibility integrity values. For applications such as pursuit and evasion, it is desirable to obtain clusters with high visibility integrity because usually \( C \subset V(C) \) when \( \text{vi}(C) \) is close to 1. That is to say, if most of the targets are in \( V(C) \), the pursuer (the camera) only needs to stay in any position in \( C \) to maintain visibility of all the targets. Note that the visibility integrity is always 1 when \( C \) contains only one point and when \( C \) is convex.

4 Online Planning using Visibility Integrity

After the construction of the visibility integrity roadmap, each point \( s \) in the original sample \( \mathcal{S} \) now has the information about the visibility integrity regions (\( \text{vi} \) regions) that are likely to be visible from \( s \). Moreover, the visibility integrity region that has the highest visibility integrity to \( s \) must be the region that contains \( s \). When the camera is following a single target \( t \), our planner first creates \( K \) predictions of \( t \)'s potential future location around \( t \) and then identifies the visibility integrity regions that contain these \( K \) predicted locations. The camera then determines its next target configurations by selecting a cluster and a configuration in the cluster which minimizes the travel distance and maximizes the visibility. Finally, the motion of the camera is determined by a smooth and collision free path that brings the camera to the target. Details of these three steps are provided below.

Target position prediction. In each time step, the online planner samples \( K \) potential future positions for the target \( t \). Each of the these samples is scored with a likelihood \( p_k \) based on the distance and direction to the current position of the target. The likelihood \( p_k \) is a joint Gaussian distribution of distance to \( t \) and the angle from \( t \)'s heading direction. The value of \( p_k \) drops to zero if the distance or angle exceeds the maximum linear or angular velocity, respectively. In our current implementation, these \( k \) samples are drawn uniformly from the circle centered at \( t \) with radius equal to the maximum distance that \( t \) can travel in a time step. Other more sophisticated filtering techniques, such as particle filters, or predictions techniques that captures the target movement behavior using Gaussian process can be used to better predict the target position.

Assign a point to a \( \text{vi} \) region. Given a point \( s \) (either a target prediction or a camera prediction), we will need to find a way to efficiently associate \( s \) with one or even multiple \( \text{vi} \) regions. If a regular grid is used to construct the visibility integrity roadmap, we can simply look up which grid cell contains \( s \) and immediately identify the \( \text{vi} \) region that \( s \) should be associated with. On the other hand, if the visibility integrity roadmap is build from sampled points, then a more implicit approach is needed to find such an association. Therefore we propose the following strategy. We first identify \( k \) closest samples \( S_k \) to \( s \) and let \( R_k \) be the \( \text{vi} \) regions that contains \( S_k \). Each of these region is then assigned a probability that \( s \) is associated with the region. The probability that \( s \) is in a \( \text{vi} \) region \( R \in R_k \) is defined as

\[
\text{prob}(s \in R) = \frac{\text{vi}(s \cup R)}{\bigcup_{R' \in R_k} \text{vi}(s \cup R')} \quad (2)
\]

Evaluate camera positions. In the last step, the best camera position is selected from a set of potential camera positions. In our current implementation, a set of \( K \) samples are drawn uniformly from the circle centered at the camera with radius equal to the maximum distance that the camera can travel in a time step. To evaluate each camera position, we essentially count the number of target samples that can be seen from the given position. This count is approximated by the cached visibility between the associated \( \text{vi} \) regions and scored assigned to each target sample.

Follow multiple targets. While our discussion above focused on a single target, it can be extended naturally to multiple targets. When the camera is following multiple targets \( T \), the planner will identify \( K \) \( \text{vi} \) regions that contain the predicted target positions for all targets. From these \( K \) \( \text{vi} \) regions, the camera decides its next position based on the same strategy.

5 Experiment Results

In this section we evaluate the performance of virtual cameras using the proposed data structure by comparing to a state of the art method [Oskam et al. 2009]. This comparative evaluation is followed by a user study that involves navigating a group of coherent agents using a Microsoft Xbox controller.

Performance Evaluation

In our experiments, we run our algorithms on several simulated environments, shown in Figure 4. The maximum speed of the targets and the camera remain the same for all of the experiments. In all scenarios, the number of targets in a group is 30. For all camera planners, the number of samples for each target and the camera are set to 25 and 50 respectively.

Table 1 shows the visibility achieved by different methods in each environment. We note that in most cases, the planner using \( \text{vi} \) achieves better visibility than using visibility-aware roadmap \( \text{VAR} \).
learn to control a group of coherent agents and the virtual camera (i.e., the user’s view into the virtual world) using both left and right joysticks, respectively, on a typical Microsoft Xbox controller with a regular laptop as shown in Figure 5.

Once the subject is comfortable with controlling interface, three 100-second sessions of group navigation are recorded. The first and the last sessions are the same. In all sessions, the subject can control both the group and the camera, however, in the second session, the subject is assisted with the camera controller proposed in this paper. When the software detects that the subject is not controlling the camera, the autonomous camera controller takes over to maintain the visibility of the virtual agents. Table 2 reports the performances of these 12 subjects which are evaluated by (1) the number of random goals reached by the group and (2) the visibility maintained during each of the session.

The data in Table 2 shows the clear benefit provided by the proposed camera controller: users are able to reach more goals and maintain much higher visibility with the group. It is also clear that such benefit is not the results of the users gaining familiarity with the user interface or better understanding of the environment layout, because we see the performance drop in the last session when the users are asked again to take the full control of the group and the camera. The last two columns of Table 2 show exactly this: the users are able to reach, on average, 3.83 more goals than what they did in the first session while, in the last session, the users can only reach, on average, 1.94 more goals than what they did in the first session.

### 6 Conclusion

In this paper, we presented a group visibility data structure that can enhance user experience in virtual world when facing the challenge of maintaining visibility to a group of moving things. We presented preliminary experimental results from applying this data structure in a group following application. In these experimental results, the proposed method was able to achieve better visibility performance than its predecessors. There are still many limitations to the proposed approach. For example, we found that these methods still perform poorly on environments with many narrow passages. We hope to improve the performance in these types of environments and also extend our methods to 3D following.

### 7 Acknowledgment

This work is in part supported by the Korean Federation of Science and Technology Societies(KOFST) grant funded by the Korean government (MSIP: Ministry of Science, ICT and Future Planning), and by US NSF IIS-096053, CNS-1205260, EFRI-1240459, AFOSR FA9550-12-1-0238. Figure 3 is created using javascript code (sight-and-light) developed by Nicky Case. The authors would like to thank the Graphics Lab and students in the Department of Computer Science & Engineering at Ewha Womans University for their participation in the user study conducted for this paper and anonymous reviewers for their constructive comments.

### References


<table>
<thead>
<tr>
<th>Manual #1</th>
<th>Manual #2</th>
<th>Autopilot Improvement</th>
<th>Manual Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Visibility</strong></td>
<td><strong>Goal</strong></td>
<td><strong>Visibility</strong></td>
<td><strong>Goal</strong></td>
</tr>
<tr>
<td>individual</td>
<td>subjects</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.58</td>
<td>17</td>
<td>0.75</td>
<td>15</td>
</tr>
<tr>
<td>0.33</td>
<td>12</td>
<td>0.83</td>
<td>15</td>
</tr>
<tr>
<td>0.44</td>
<td>18</td>
<td>0.71</td>
<td>21</td>
</tr>
<tr>
<td>0.27</td>
<td>12</td>
<td>0.74</td>
<td>18</td>
</tr>
<tr>
<td>0.59</td>
<td>1</td>
<td>0.73</td>
<td>11</td>
</tr>
<tr>
<td>0.29</td>
<td>1</td>
<td>0.74</td>
<td>15</td>
</tr>
<tr>
<td>0.28</td>
<td>10</td>
<td>0.76</td>
<td>18</td>
</tr>
<tr>
<td>0.51</td>
<td>7</td>
<td>0.79</td>
<td>6</td>
</tr>
<tr>
<td>0.40</td>
<td>12</td>
<td>0.57</td>
<td>18</td>
</tr>
<tr>
<td>0.40</td>
<td>16</td>
<td>0.79</td>
<td>19</td>
</tr>
<tr>
<td>0.46</td>
<td>2</td>
<td>0.66</td>
<td>6</td>
</tr>
<tr>
<td><strong>average</strong></td>
<td><strong>0.44</strong></td>
<td><strong>11.08</strong></td>
<td><strong>0.75</strong></td>
</tr>
</tbody>
</table>


Vo, C., and Lien, J.-M. 2010. Following a large unpredictable group of targets among obstacles. In The Third International Conference on Motion in Games.

