

# Quantitative Analysis of Visibility Determinations for Networked Virtual Environments

Beomjoo Seo<sup>a</sup>, Roger Zimmermann<sup>a</sup>

<sup>a</sup>Department of Computer Science, National University of Singapore, Singapore

---

## Abstract

The Area Of Interest (AOI) model is a simple and popular technique used in many applications to determine the region which needs to be considered and processed for each entity (e.g., user). One example application is object visibility determination around user-representing avatars in virtual environments or networked games. There exist a number of variations of the AOI model and in our prior work we have demonstrated how object-oriented visibility determination is more suitable for networked virtual environments than conventional user-oriented visibility determination. Here we extend our work to study a unified and comprehensive analytical model that reveals fundamental properties about the different visibility determination techniques under a variety of virtual environment settings. We also present what the best operational scenarios are for each different approach. Although our discussion and analytical results are focused on the visibility domain, the arguments and conclusions can be extended to various applications or services where spatial attributes are required.

*Keywords:* Visibility Determinations, Interest Management, Area-Of-Interest, Virtual Reality

---

## 1. Introduction

In computer graphics, visibility determination, *i.e.*, the elimination of visibly irrelevant geometry objects from a huge object database, has been extensively studied. Many well-versed efficient algorithms such as view-frustum culling, back-face culling, and occlusion culling algorithms [1] are, however, applicable only for local rendering applications, assuming all objects are stored locally.

If geometry objects are stored on a remote node, a client will receive objects to be rendered from the remote node [2]. In such a streaming approach, existing visibility algorithms are no longer serviceable. Due to the simple design and low computational complexity, Area-Of-Interest (AOI) filtering has been adopted in remote rendering applications - *i.e.*, a type of Massively Multi-player Online Game (MMOG) where a user downloads objects on demand within her view frustum from a server [3, 4].

Although user-centric AOI filtering is popular, it can lead to undesirable artifacts. For example, *Second Life*, one of the most successful client-server based MMOG systems, employs an AOI-filtering based visibility determination model but it sometimes suffers from “object popping” discontinuities, as illustrated in Fig. 1. In the left image, a house outside the user’s visible area is not rendered at the present time  $t$ . As the user moves forward, the house enters into the user’s AOI and then appears unexpectedly, hence likely disrupting the user’s navigational experience. This visual aberration is caused by the inherent properties of user-centric AOI filtering, which in this

case neglected the spatial contribution of a large in-world object in order to reduce the search scope and improve performance.

To mitigate such problems, several virtual reality systems have amended user-centric AOI filtering by additionally taking the “visual scope” of target objects into account during their visibility determination [5, 6, 7]. In these enhanced AOI algorithms, an object is determined to be visible to an observer only when its visual scope is inside the observer’s visual scope. Although these models identify more relevant objects than the original AOI model, they are exposed to several challenging issues. First, their intersection-of-scope based visibility determination does not reflect physical visibility properties correctly. The reason is that these models were originally designed to detect collisions between spatial objects and then the collision was intuitively extended to other spatial relationships such as interest detection and visibility determination. Second, these algorithms are computationally very expensive, since they do not effectively narrow the search scope to the extent that the user-centric AOI model does. Thus, they are impractical for real-time environments.

Because of these limitations we earlier investigated an alternative approach, called the *object model* [8], which accounts for the spatial scope of target objects and therefore exhibits certain advantages over the existing user-centric AOI filtering paradigm. In our prior work we successfully demonstrated the superiority of the object model by quantifying its determination results in a static environment, where all users and objects are stationary. Additionally, we investigated the real-time capability of the object model by proposing an efficient spatial indexing technique based on the latest grid-based sub-partitioning method. The simulation results showed that our indexing ap-

---

*Email addresses:* beomjoo90@gmail.com (Beomjoo Seo),  
rogerz@comp.nus.edu.sg (Roger Zimmermann)

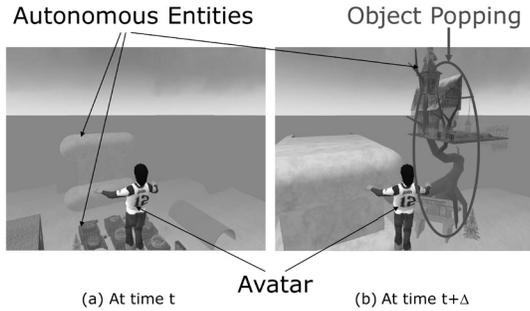


Figure 1: An illustration of the visual artifact caused by AOI filtering in Second Life. (Image courtesy of Linden Lab)

proach can determine visibly relevant objects among one hundred thousand moving objects for ten thousand moving users in less than two seconds.

Although we verified the real-time capabilities of the object model in a simulation environment with moving objects, the theoretical analysis of the previous work was only valid for stationary environments. Moreover, we did not clearly associate the visibility determination models with physical phenomena. In this study, we present systematic and comprehensive analytical results of the features and limitations of three different AOI management models: the traditional *user AOI model*, the recently introduced *object AOI model*, and a combined *hybrid AOI model*. To compare the optimality of each determination algorithm, we also present a very simple, yet effective, visibility distance model.

Even though our study focuses on the analysis of the visibility determination, it has implications in various spatial areas where spatial relationships among objects are of importance, such as audibility determination or the detection of mobile devices in the wireless coverage area of an access point.

This study results in the following important contributions related to the spatial (especially, in terms of visibility awareness) interaction among users and objects in networked virtual environments:

- We provide a mathematical foundation of different AOI filtering methodologies that associates spatial relation among users and target objects. Although all the analyses and evaluations in this study are focused on visibility awareness problem in networked environments, their conclusions can be extended to any type of spatial properties which are transformable to a measurable quantity.
- We establish a new evaluation methodology that quantifies the retrieval quality of static (or moving) search queries in static or moving environments in terms of visible relevance among users and objects. While our methods are based on the traditional definitions of Precision/Recall metrics, their extensions allow measuring the search quality of static or moving queries. We also show that they can account for the effect of the use of client-resident local storage.
- We present the fundamental limitation of the user-centric

AOI filtering mechanism. That is, it cannot achieve optimality search results in terms of visible relevance. In other words, the user-centric AOI model inevitably retrieves visibly irrelevant objects and not all visibly relevant objects are retrieved. This results in unwanted visual artifacts (object popping problem) in a client/server-based object streaming domain.

- We propose a new filtering methodology that achieves optimality. We also discuss its potential limitations and justify its practicality in real environments through the extensive analyses and experiments. Additionally, we present its optimal usage scenario.
- Finally, we present accurate analytical AOI filtering models, which reflect the performance characteristics of individual filtering approaches. Therefore, system administrators who may choose one of the filtering mechanisms can easily evaluate them without any experimentation.

The organization of this article is as follows. Section 2 introduces background information and presents the target visibility determination models that will be compared. In subsequent sections we provide analytical results of the visibility determination algorithms for three different cases: a stationary user in a static environment (Section 3); a moving user in a static environment (Section 4); and a stationary user in a moving environment (Section 5). In Section 6, we present the evaluation results of the analytical models and discuss their implications. Finally, Section 7 summarizes the conclusions of this study.

## 2. Background

Our analysis targets client/server-based game applications, where a game server stores a large number of virtual entities and transmits visible entities to their associated client nodes continuously. In this section, we review a useful visible distance model that is associated with visible relevance and classify visible determination models into three categories. For the remainder we use the term **object entity** (or shortly object) to refer to an ordinary moving or stationary object (or player), while we use **user entity** (or shortly user) to denote one specific player.

### 2.1. Modelling Visible Distance

Quantifying the degree of visibility of an object is a very broad research topic. The research goes back to the early 18th century when scientists first studied visibility such as the Beer-Lambert-Bouguer Law<sup>1</sup>. In this article, since covering all visual effects caused by various atmospheric conditions is an extremely challenging problem, we do not attempt to use an omnipotent visibility model that accounts for such physical phenomena.

Instead, we use a simplified model, called *Visual Acuity Model*, that has been acknowledged in several virtual reality

<sup>1</sup>It approximately derives the travelling distance of a light through the atmosphere from its absorption coefficient and scattering of radiation.

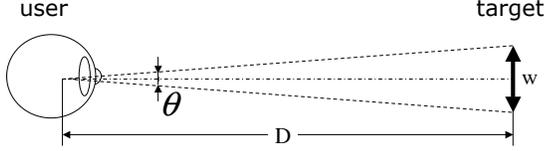


Figure 2: Illustration of visual acuity model. In this model, the visible condition between a user and a target object is determined by three parameters : the width of the target  $w$ ; the distance between the two  $D$ ; and the angle subtended at the user's eye by the target  $\theta$ .

systems [9, 10]. In this model, the visual acuity number of a user ( $A$ ), a comparative measure of the user's ability to identify a given spatial pattern at a given distance, is defined as the ratio of the user's maximum visible distance ( $V$ ) to that of an ordinary user with normal vision ( $V_{norm}$ ) - i.e.,  $A = V/V_{norm}$  [11]. As illustrated in Fig. 2,  $V_{norm}$  is computed from the length of a target object  $W$  and its visual angle of one minute of arc (1MOA) threshold angle  $\theta_{norm}$  - i.e.,  $V_{norm} = \frac{W}{2} \cot \frac{\theta_{norm}}{2}$ . Therefore, if a user's visual acuity number is given,  $V$  can be obtained by  $A$  and  $W$  (Eq. 1).<sup>2</sup>

$$V = C \cdot A \cdot W, \quad \text{where } C = \frac{1}{2} \cot \frac{\theta_{norm}}{2} \quad (1)$$

Note that this model is valid under the following assumptions. First, we assume that the threshold visual angle,  $\theta_{norm}$ , is applicable to every object. Thus, if a user's subtended angle to a target is greater than  $\theta_{norm}$ , the target will be perceived by the user. Second, the user's visual sensitivity is assumed to be constant regardless of target size.<sup>3</sup> It means that a user can equally identify target objects as long as they are within the range of  $\theta_{norm}$ .

## 2.2. Visibility Determination Algorithms

To begin with, we briefly sketch a sample virtual space and useful notations. First, imagine the following two-dimensional disk of radius  $D$ , as illustrated in Fig. 3. In the space, a user of interest (shown as a black square) is centered and objects (dashed circles) are evenly dispersed; a specially chosen object among the objects is shown as a bigger dashed circle. In the figure, the visible range of the user is drawn as a dotted circle (left), which is associated with user's visual acuity  $A$ , and that of the object, derived from object size  $W$ , is depicted as the dotted circle (right).

Then we introduce three random variables,  $X$ ,  $Y$ , and  $Z$ .  $X$  is a distance metric that represents the visible range of an object. In Fig. 3, it corresponds to the radius of the right dotted circle.  $Y$  denotes the Euclidean distance between a user and an object. The  $Z$  distance metric symbolizes the visible range of the user, depicted as the radius of the left dotted circle. Their probability

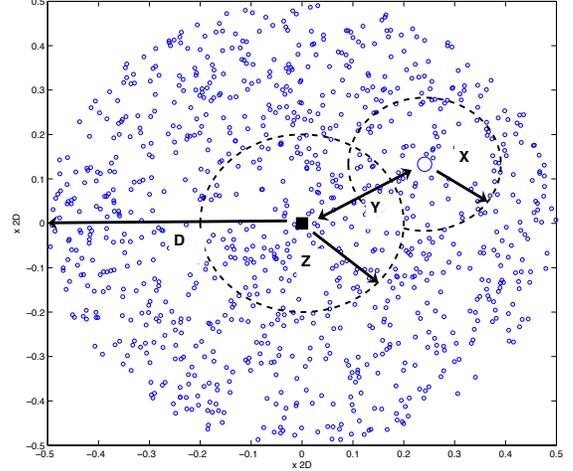


Figure 3: A sample 2D space where one thousand sample objects are uniformly distributed around a user at center.

density functions are denoted as  $f_X$ ,  $f_Y$ , and  $f_Z$ , accordingly. For convenience purposes, we assume that every random variable has a maximum value, whose symbolic notations are  $D_X$ ,  $D_Y$ , and  $D_Z$ . From a user's standpoint,  $D$  and  $D_Y$  are functionally identical, since the distance of the farthest object from the user ( $D_Y$ ) does not exceed the user's search domain ( $D$ ) and any objects beyond the domain do not contribute to any improvements on search quality, while increasing the search scope unnecessarily. To include all visibly relevant objects,  $D_Y$  should be large enough to cover any combinations of  $X$  and  $Z$ . At the same time, the domain scope should be carefully chosen to be as compact as possible, such as not to extend more than required. During our analysis, a user's visual range  $z \in Z$  will be chosen arbitrarily, but won't exceed the search scope. Otherwise, it may lead to over-estimation, resulting in significant quality degradation. The side-effect of such over-estimation will be fully discussed later.

Now, we introduce an indicator function,  $I_S(Y)$ , which simplifies spatial relevance between a user and a target object. It is defined on a set of Euclidean distance ( $Y$ ) between two virtual entities,  $I_S : Y \rightarrow \{0, 1\}$ , where  $S$  is a distance unit. For any given actual distance  $y \in Y$ , it is formally defined as

$$I_S(y) = \begin{cases} 1 & : y \leq S \\ 0 & : \text{otherwise.} \end{cases} \quad (2)$$

If the given distance is smaller than  $S$ , the indicator function will return one; otherwise, it will return zero.

### 2.2.1. Classification

Existing visibility algorithms can be classified by different usage conventions of the following visibility mapping functions,  $\phi_A$  and  $\phi_W$ . They define how a user's visual acuity number  $A$  and an object length  $W$  are mapped to distance metrics,  $Z$  and  $X$ , respectively. Intuitively, these are regarded as non-decreasing functions. For example, a user who possesses a higher visual acuity number can distinguish farther objects.

<sup>2</sup>In this equation, we neglect the effect of scaling correction according to the video resolution of a display device [10].

<sup>3</sup>In fact, the visual sensitivity of target size is not always constant. It degrades after a specific distance [10]. Thus, far-away objects tend to be less accurately perceived by ordinary human.

Table 1: Summary of visibility mapping conventions of different visibility determination types.

	transform functions		visible when
	user’s visual scope ( $\phi_A$ )	object’s visual scope ( $\phi_W$ )	
user model	$A \rightarrow z$ (3)	$W \rightarrow 0$ (4)	$Y \leq z$ (5)
object model	$A \rightarrow 1$ (6)	$W \rightarrow C \cdot W$ (7)	$Y \leq C \cdot W = C \cdot A_{norm} \cdot W = C \cdot W$ (8)
hybrid model	$A \rightarrow C_A \cdot A$ (9)	$W \rightarrow C_W \cdot W$ (10)	$Y \leq X + Z = z + C_W \cdot W$ (11)

Similarly, bigger objects can easily be perceived by the user at a farther distance. Using the different conventions (summarized in Table 1), we categorize the algorithms into three types: user, object, and hybrid model. In the rest of this article, we may omit AOI for simplicity when referring to these models.

#### User AOI Model

This type of application determines viewable objects inside a user-centered visible area (Eq. 3), while ignoring the visual attribute of objects (Eq. 4). Due to its simplicity, it has been popularly used by many location-based service applications to process nearest neighbor queries, where the number of objects is typically larger than that of users, so it appears that the user model is a reasonable choice to lower the system load.

In the model, the visible condition of an object is defined as the spatial condition that the Euclidean distance of the object location from the user location is less than or equal to a given distance value ( $z \in Z$ ) that is specific to a user (Eq. 5). In the rest of the article, the given distance is termed the *user’s visible threshold*, or shortly *threshold*. Its normalized value  $z' = \frac{z}{D_z}$  is called *threshold ratio*.

#### Object AOI Model

In this model, users are simplified as points and objects possess their proprietary viewable region (Eq. 7). Without loss of generality, every user’s visual acuity number is assumed to be one, meaning that the user has normal vision (Eq. 6).<sup>4</sup> An object is determined to be visible to a user if the user is inside the visible range of the object (Eq. 8).

#### Hybrid AOI Model

The hybrid model uses two visual scopes stemming from users and objects. Existing methods such as the Aura and Nimbus model [5] and CyberWalk [7], however, fail to provide any specific mapping rules to assign the visible scope of objects and users. To analyze them, we employed linear mapping conventions shown in Eqs. 9 and 10 by constant factors  $C_A$  and  $C_W$ , respectively. That is, the visual scope of a user is proportional to the user’s visual strength and that of an object also proportional to its size. With this model, an object is computed to be visible to a user if two visible scopes intersect with each other (Eq. 11).

<sup>4</sup>It does not mean that every user has no visual strength, but that every user possesses equal sight vision.

In the following sections, we present our analytical models of each type that quantitatively estimate the search quality under different object distribution patterns. To make the analysis solvable, we narrow the problem scope by constraining the density pattern on  $Y$ , assuming that two objects that are equidistant from an observer have the same density probability – *i.e.*, the uniform density function of  $Y$  is  $f_Y(Y = y) = \frac{2y}{D_y^2}$ . The uniform distribution is one example of such distribution patterns. This assumption, although not reflecting all real-world phenomena, is powerful enough to express different degrees of crowdedness.

While  $f_Y(y)$  is limited to a uniform distribution,  $f_X(x)$  still remains undetermined. Therefore, the prediction model of each visibility determination approach is presented as a function of  $f_X(x)$  and  $f_Y(y)$ .

In the next section, we start with presenting a simple analysis model for a static user in a static environment. We then continue to cover a more complicated case (a moving user in the same static environment) and end with discussing its reverse case (a static user in a dynamically moving environment).

### 3. A Stationary User in a Stationary Environment

In this section, we evaluate the performance of three different types of visibility models in a stationary environment whose behavioral scenario is as follows:

**Scenario:** A stationary user of interest is located at the center and surrounded by  $n$  stationary objects, which are uniformly distributed in a two-dimensional virtual space, a disk of radius  $D_Y$ . Assuming that every user possesses the same visible strength ( $A=1$ ), the optimal visible distance between the user and an object  $i$  is denoted by  $V_i$  and then computed as  $C \cdot W_i$ , where  $W_i$  is the size of object  $i$ .

#### 3.1. Performance Metrics

For quantitative evaluations, we use two well-known performance metrics *Precision* ( $P$ ) and *Recall* ( $R$ ). They, respectively, estimate the degree of accuracy and comprehensiveness of the search results [12].  $P$  is the ratio of visibly relevant items among retrieved items. A lower  $P$  means that an algorithm computes visibly irrelevant objects more often.  $R$  is the ratio of retrieved items among visibly relevant items. An algorithm with a low  $R$  score neglects visibly relevant objects falsely, leading to object popping more frequently.

$$P = \frac{\text{relevant and retrieved}}{\text{retrieved}}, R = \frac{\text{relevant and retrieved}}{\text{relevant}}$$

In addition to  $P$  and  $R$  metrics, we use a standardized single-valued query estimation metrics that combines  $P$  and  $R$ , called  $E$ -measure [12]. The  $E$ -measure is defined as:

$$E = 1 - \frac{(\beta^2 + 1)PR}{\beta^2 P + R},$$

where  $\beta$  is the relative importance of  $P$  or  $R$ . If  $\beta$  is equal to 1,  $P$  and  $R$  are equally important. If  $\beta$  is less than 1,  $P$  becomes more important and vice versa. A lower  $E$ -measure value implies that a tested visibility algorithm has a higher determination quality. The best  $E$ -measure value is 0 where  $P$  and  $R$  are both 1. Throughout the rest of the article, we use the  $E$ -measure parameter of  $\beta = 1$ .

To compute these metrics, we analyze three probabilistic quantities: the expected number of retrieved objects; the expected number of visibly relevant objects; and the expected number of retrieved and visibly relevant objects.

### 3.2. Analysis of User AOI Model

In the user model, every user only recognizes objects within its visible threshold  $z$ , regardless of the different visible properties of individual objects. The expected number of retrieved items by the user model ( $Ret_u$ ) is the cardinality of the set of objects whose Euclidean distance to the user is less than or equal to  $z$ . Assuming  $Y_i$  is the Euclidean distance between the user and an object  $i$ , the number of retrieved items is expressed as the sum of the binomial indicator function  $I_z(Y_i)$ .

$$Ret_u = \sum_{i=1}^n I_z(Y_i)$$

The above equation is then finally converted to its equivalent probabilistic form.

$$\sum_{i=1}^n I_z(Y_i) = n \cdot E[I_z(Y)] = n \cdot Pr(Y \leq z) = n \cdot F_Y(z) \quad (12)$$

The expected number of visibly relevant items ( $Rel_u$ ) is formulated as the number of items whose Euclidean distance to the user is less than or equal to its optimal visible distance  $V_i$ , which is expressed as  $Y_i \leq V_i$  for a given object  $i$ . Thus, it is the sum of  $I_{V_i}(Y_i)$ . Since  $V_i$  is set to  $X_i$ , its probabilistic expression is then

$$Rel_u = \sum_{i=1}^n I_{X_i}(Y_i) = n \cdot Pr(Y \leq X).$$

Using the total probability theorem, we expand the probabilistic model as an integral of  $X$ , resulting in

$$Rel_u = n \int_X f_X(x) F_Y(x) dx. \quad (13)$$

The expected number of relevant items among retrieved items ( $Ret \cap Rel_u$ ) is the sum of the probability that an object is not only within the user's visible threshold but also that its distance to the user is within the optimal visible distance. The probability can also be similarly expanded as the sum of two disjoint cases:  $X \leq z$  and  $X > z$ :

$$\int_0^z f_X(x) Pr(Y \leq z, Y \leq x) dx + \int_z^{D_X} f_X(x) Pr(Y \leq z, Y \leq x) dx$$

The left term is further simplified as  $\int_0^z f_X(x) Pr(Y \leq x) dx$ , since the given  $x$  belonging to the first case is always less than  $z$ . Likewise, the right term is reduced to  $\int_z^{D_X} f_X(x) Pr(Y \leq z) dx$ . As a result, the final form of  $Ret \cap Rel_u$  is obtained as

$$Ret \cap Rel_u = n \cdot \left( \int_0^z f_X(x) F_Y(x) dx + \int_z^{D_X} f_X(x) F_Y(z) dx \right). \quad (14)$$

If we apply uniform distribution patterns of  $X$  and  $Y$ , the precision and recall metrics are expressed as a function of the normalized visible threshold  $z' = \frac{z}{D_X}$ . Fig. 4(a) illustrates the prototypical performance trend of the user model: as the user's visible threshold increases, the recall increases rapidly at the beginning and decreases later, while the precision decreases linearly. The  $E$ -measure curve depicts the combined effect of the two metrics; as the user increases his visible region, the  $E$ -measure value moves downwards to reach a local minima and then increases afterwards. Therefore, the optimum usage scenario of the user model is when the user's visible threshold is set to 68 % of the maximum distance  $D_X$ . Nevertheless, the user model cannot achieve any optimality, *i.e.*, a value of zero.

### 3.3. Analysis of Object AOI Model

The object model is designed to identify all visibly relevant objects by mapping the visible distance of every object to its ideal visible distance, so that its performance metrics is naturally guaranteed to achieve optimality.

In this analysis, we further investigate a potential issue of the object model – how much an incorrect mapping on the visible distance influences the performance of a system. To do this, we first introduce two notations that represent different aspects of the visible distance of an object  $i$ : one configured by an actual system  $X_i$  and the other by an ideal method  $V_i$ . The ratio of  $X_i$  to  $V_i$  is termed **visual ratio** and denoted as  $r = \frac{X}{V}$ .

A system, if underestimating the ideal visible distance ( $r < 1$ ), retrieves visibly relevant objects later than the ideal system, thus leading to frequent object poppings. An overestimating system ( $r > 1$ ), on the other hand, retrieves objects earlier than the ideal, meaning that it may retrieve visibly irrelevant objects more often.

The expected number of retrieved items by the object model ( $Ret_o$ ) is computed as the sum of binomial indicator functions of individual objects, counting the number of objects whose approximated visible region covers a user – *i.e.*,  $Y_i \leq X_i$ . For fair comparisons, we transform all  $X$  values to their corresponding ideal value  $X^{opt5}$ , which is invariant of different visual ratios.

$$\begin{aligned} Ret_o &= \sum_{i=1}^n I_{X_i}(Y_i) = \sum_{i=1}^n I_{rX_i^{opt}}(Y_i) \\ &= n \int_{X^{opt}} f_X(x) F_Y(rx) dx \end{aligned} \quad (15)$$

<sup>5</sup>Actually, it is equal to  $V$ . For clarity purpose, it is expressed as a function of  $X$ .

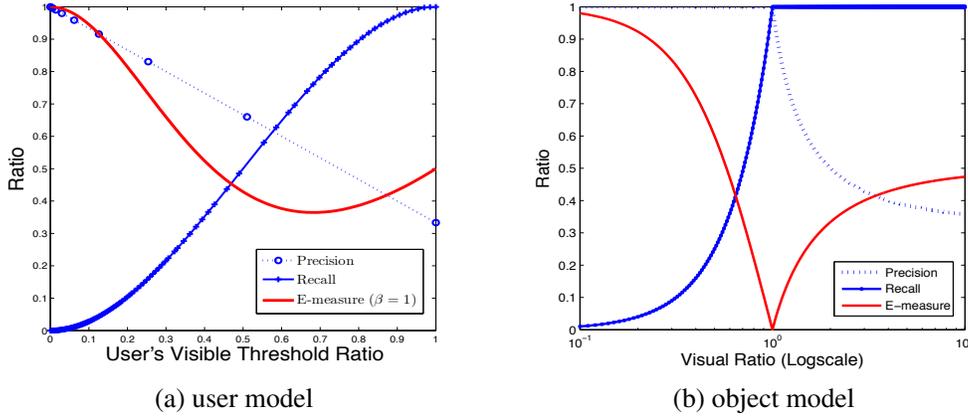


Figure 4: Precision, Recall, and E-measure trends of (a) user model as a function of user's visible threshold ratio and of (b) object model as a function of object's visual ratio under uniform distributions of  $X$  and  $Y$ , where users and objects are all stationary.

The expected number of relevant items ( $Rel_o$ ) is the number of objects whose ideal visible distance is greater than or equal to the Euclidean distance to a user ( $Y_i \leq X_i^{opt}$ ). Since the visibly relevant items are identical regardless of different visibility algorithms, they will be the same as for the user model (shown in Eq. 13).

$$Rel_o = Rel_u \quad (16)$$

The expected number of retrieved and relevant items ( $Ret \cap Rel_o$ ) is the cardinality of the intersection set of  $Ret_o$  and  $Rel_o$ . When  $r < 1$ , it is equivalent to  $Ret_o$ . Otherwise, it is  $Rel_o$ .

$$Ret \cap Rel_o = \min(Ret_o, Rel_o) \quad (17)$$

Fig. 4(b) depicts the performance trend of the object model when uniform distributions of  $X$  and  $Y$  are applied. If the visible distance is configured correctly ( $r = 1$ ), the object model remains in an optimal condition. If the visible distance is overestimated, irrelevant objects will be unnecessarily retrieved (lower  $P$ ), while all relevant objects retrieved ( $R = 1$ ); and if underestimated, all retrieved objects will be visibly relevant ( $P = 1$ ), but some visibly relevant objects will be missed (lower  $R$ ). The figure also shows that slight overestimation or underestimation of the object model outperform the user model by a sizeable margin. Hence, we conclude that the slightly approximated object model is still preferable to the user model.

A concern may be that since the object model uses a fixed visual ratio regardless of users' heterogeneous visual strength, it can not satisfy all different visual desires. For some users, it would return the optimal results, while for others it would return underestimated or overestimated results. In simulated virtual environments however, every user is expected to possess the same sight vision since visual details are limited by the display resolution. It is doubtful whether allowing different visual strengths per user is practically meaningful in such environments.

### 3.4. Analysis of Hybrid AOI Model

Here we assume that every object has a visible distance ( $X$ ) and a user has a visible threshold  $z$ , while the ideal visible distance between the object and the user is  $X^{opt}$ .

The expected number of retrieved items by the hybrid model ( $Ret_h$ ) is the sum of all binomial indicator functions of individual objects,  $I_X(Y)$ :

$$\begin{aligned} Ret_h &= \sum_{i=1}^n I_{X_i+z}(Y_i) = n Pr(Y \leq rX^{opt} + z) \\ &= n \int_{X^{opt}} f_X(x) F_Y(rx + z) dx \end{aligned} \quad (18)$$

The expected number of relevant items of the hybrid model ( $Rel_h$ ) is the same as that of the object model.

$$Rel_h = Rel_o = Rel_u \quad (19)$$

And the expected number of relevant and retrieved items ( $Ret \cap Rel_h$ ) is modelled as

$$Ret \cap Rel_h = n Pr(Y \leq rX^{opt} + z, Y \leq X^{opt}). \quad (20)$$

Eq. 20 is subdivided into two cases when  $r < 1$  and  $r \geq 1$ . For the former, there exists a crossing point  $x_p \in X^{opt}$  of two lines  $x$  and  $rx + z$ , i.e.,  $x_p = \frac{z}{1-r}$ . If  $x_p$  exceeds the range of the sample space  $D_X$ ,  $Ret \cap Rel_h$  will be  $Rel_h$ , since  $rx + z$  is greater than  $x$ . Otherwise,  $Pr(Y \leq rX^{opt} + z, Y \leq X^{opt})$  will be  $Pr(Y \leq X^{opt})$  when  $X^{opt} \leq x_p$  and  $Pr(Y \leq rX^{opt} + z)$  when  $X^{opt} > x_p$ . For the latter, the result becomes  $Rel_h$ , since  $rx + z > x$ .

As a result, the final form of  $Ret \cap Rel_h$  is summarized as:

$$\begin{cases} n \cdot \left( \int_0^{\frac{z}{1-r}} f_X(x) F_Y(x) dx + \int_{\frac{z}{1-r}}^{D_X} f_X(x) F_Y(rx + z) dx \right) & : r < 1, z \leq D_X(1-r) \\ Rel_h & : \text{otherwise} \end{cases} \quad (21)$$

Fig. 5 depicts how the visual ratio and user's visible threshold are co-related in terms of the determination quality of the hybrid model. First, if the visible scope of every object is overestimated, any increase in the user's visible threshold will simply worsen the quality. Second, if the visual ratio is underestimated (i.e.,  $r = \frac{1}{2}, \frac{1}{4}$  in the figure), there exists an optimal user's visible threshold that achieves the local minima of the quality. Fig. 6 shows the trend of the local minima and its optimal user threshold value for each visual ratio. In the figure,

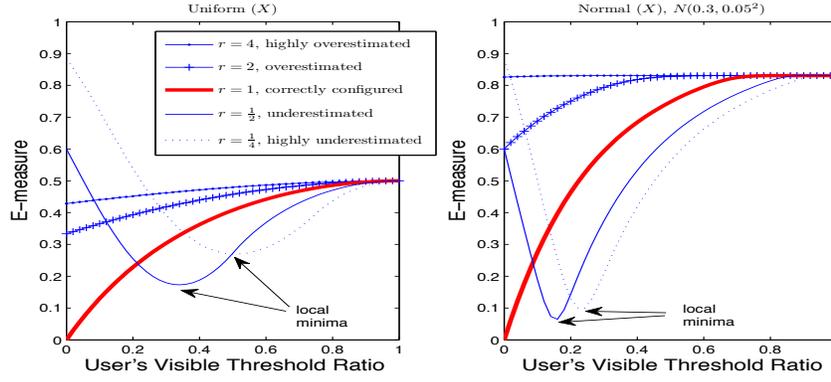


Figure 5: Under uniform distribution of  $Y$  and uniform (left) and normal (right) distributions of  $X^{opt}$ , the E-measure values of the hybrid model are plotted as a function of user’s visible threshold with five different cases of object’s visual ratio: two under-estimations ( $r = \frac{1}{4}, \frac{1}{2}$ ), two over-estimations ( $r = 2, 4$ ), and one correct estimation ( $r = 1$ ).

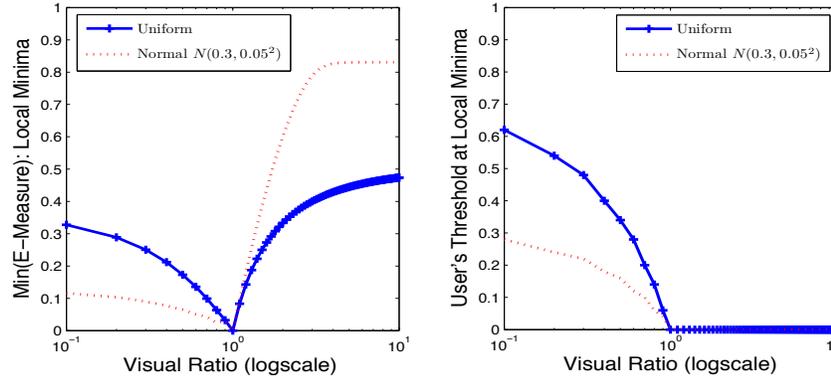


Figure 6: The local minima (left) of the E-measure values of individual visual ratios, obtained by varying user’s visible threshold, and their corresponding visible threshold points (right) are plotted as a function of object’s visual ratio.

as the underestimated visual ratio increases, the optimal visible threshold at the local minima monotonically decreases. If the object’s visual scope is overestimated, the optimal threshold value is then always zero. Therefore, a hybrid model-based system should be carefully designed to use the optimal pair of visual ratio and user’s threshold. Nevertheless, the best system configuration is, as we suggest, to use the object model.

#### 4. A Moving User in a Stationary Environment

In the previous section, the user model was unfairly treated, since it couldn’t utilize local resources. If a user is allowed to move in a space, the user model, especially, may take advantage of using two aggressive strategies, **caching** and **prefetching**, while the object model has no benefit. The caching method allows the user to reuse objects that were retrieved during previous rounds. The prefetching mechanism recognizes objects that will be retrieved during the next rounds and fetches them early.

On the other hand, if objects are allowed to move, such caching and prefetching may be limited in how they can improve the retrieval quality of the user model, since cached or prefetched items are not guaranteed to be usable at a future time. Thus, the best operation scenario for the user model

is when a user moves while objects are stationary. This implies that the analysis of a moving user in a stationary environment will depict the upper limit of the performance by the user model. In this section, we analyze the following scenario, quantifying how much the user model benefits from caching and prefetching.

**Scenario:** A user moves from  $O_1$  to  $O_2$  in a space, where all static objects were evenly distributed within a circle of radius  $D_Y$ . Let the length of  $O_1O_2$  be  $v$ .

##### 4.1. Performance Metrics

In dynamically moving virtual environments, the traditional definitions of Precision and Recall are inappropriate for quantifying the performance characteristics of the visibility algorithms, since some objects may contribute to the calculation of the performance metrics unnecessarily. For example, if the number of retrieved objects is associated with network delivery cost, the objects, delivered and served during previous rounds, and still being served at the current time, do not contribute to the current network cost.

To exclude such cases, we modify the precision definition that only quantifies the accuracy of “recently (or newly) retrieved objects”. Accordingly, the recall metric is redefined to

evaluate the comprehensiveness of “all retrieved objects” because of the following reasons. First, a client machine may keep all retrieved objects in its cheap local storage medium for future use. Second, the cost of local visibility determination on the previously downloaded objects is much cheaper than the transmission cost.

$$P = \frac{\text{relevant items among \textbf{recently retrieved items}}}{\text{recently retrieved items}}$$

$$R = \frac{\text{relevant items among \textbf{all retrieved items}}}{\text{relevant items}}$$

Before analyzing these performance metrics, we introduce several useful lemmas and definitions. Imagine two arbitrary circles  $C_1$  and  $C_2$  of radii  $r_1$  and  $r_2$  whose centers are separated by a distance  $v$ . Without loss of generality, their centers, say  $O_1$  and  $O_2$ , are assumed to be located on a horizontal axis with  $O_1 < O_2$ . We also assume that, for an arbitrary point  $P$ , the line lengths to the centers  $\overline{O_1P}$  and  $\overline{O_2P}$  are denoted by  $Y^1$  and  $Y^2$ , respectively.

If the two circles overlap, the arc length of  $C_1$  inside  $C_2$  – i.e., the size of the set of points on  $C_1$  that satisfy  $Y^1 = r_1, Y^2 \leq r_2$  – will be given by Lemma 1. The proof of the Lemma 1 is straightforward and omitted here. The length of the arc of  $C_2$ , on which a point satisfies  $Y^2 = r_2, Y^1 \geq r_1$ , is also derived in a similar manner:  $2r_2 \cdot \arccos\left(\frac{r_1^2 - r_2^2 - v^2}{2vr_2}\right)$ .

**Lemma 1.** *Let two overlapping circles  $C_1$  and  $C_2$  of radii  $r_1$  and  $r_2$  and centered at  $(0, 0)$  and at  $(v, 0)$  intersect in a region shaped like a lens. The length of the arc of  $C_1$  in the lens is given by  $2r_1 \cdot \arccos\left(\frac{v^2 + r_1^2 - r_2^2}{2vr_1}\right)$ .*

We extend Lemma 1 to a more general case, where two arbitrary circles are given. The circles may overlap, cover, or be disjoint. To simplify such spatial relations, we introduce two probability functions  $\ell_{r_1}^{r_2}(v)$  and  $\mathfrak{R}_{r_2}^{r_1}(v)$ , which are formally defined in Lemma 2 (see details in Appendix A).

**Lemma 2.** *Let two arbitrary circles  $C_1$  and  $C_2$  with radii  $r_1$  and  $r_2$  be separated by a distance  $v$  and  $C_1$  be located at the left side of  $C_2$ . The probability that any point on a given  $C_1$  is inside  $C_2$  is given by  $\ell_{r_1}^{r_2}(v)$ , where*

$$\ell_{r_1}^{r_2}(v) = \begin{cases} 1 & : r_1 \leq r_2 - v \\ 0 & : r_1 \geq v + r_2, r_1 \leq v - r_2 \\ \arccos\left(\frac{r_1^2 + v^2 - r_2^2}{2r_1v}\right)/\pi & : \text{otherwise} \end{cases} \quad (22)$$

*The probability that any point on a given  $C_2$  is outside  $C_1$  is given by  $\mathfrak{R}_{r_2}^{r_1}(v)$ , where*

$$\mathfrak{R}_{r_2}^{r_1}(v) = \begin{cases} 1 & : r_1 \leq |v - r_2| \\ 0 & : r_1 \geq v + r_2 \\ \arccos\left(\frac{r_1^2 - r_2^2 - v^2}{2r_2v}\right)/\pi & : \text{otherwise} \end{cases} \quad (23)$$

$\ell_{r_1}^{r_2}(v)$  symbolizes the probability of a point on a given  $C_1$  that satisfies  $Y^2 \leq r_2$  – that is,  $Pr(Y^2 \leq r_2 | Y^1 = r_1, |\overline{O_1O_2}| = v)$ . Similarly,  $\mathfrak{R}_{r_2}^{r_1}(v)$  abstracts the probability of a point on a given  $C_2$  that satisfies  $Y^1 \geq r_1$ . Its probabilistic model is then  $Pr(Y^1 \geq r_1 | Y^2 = r_2, |\overline{O_1O_2}| = v)$ .

In the remainder of this section, we use a notational convention  $Y^i$  that denotes the Euclidean distance to an object of a user at time  $t_i$ .

#### 4.2. Analysis of User AOI Model

We start with the analysis of a simple scenario of the user model, where it does not utilize any user-side resources. Later, we explore more complex usage scenarios, where the user uses caching and prefetching strategies. To distinguish different sets of objects at a discrete time-slot, we use a superscript  $i$  on the set which is associated with time  $t_i$ . For example,  $Ret^i$  denotes the number of retrieved objects at time  $t_i$ .

##### 4.2.1. A simple client

In this scenario, a server, running a user-centric visibility determination model, computes objects who are within the visible threshold radius  $z$  of a client, and sends them to the client. At the client’s side, the received objects are rendered and then discarded after the rendering. At time  $t_1$ , the user was at  $O_1$ , and then moved to  $O_2$  at time  $t_2$ , where  $|\overline{O_1O_2}| = v$ .

A newly transmitted stationary object, which is fixed at  $P$ , was invisible at  $t_1$  and becomes visible at  $t_2$ . Assuming that  $Y^1$  and  $Y^2$  are the distance of  $P$  to  $O_1$  and  $O_2$ , respectively, then the probability that an object is newly retrieved by the user model is modelled as  $Pr(Y^1 > z, Y^2 \leq z)$  and can be expanded by  $Y^2$ , using the  $\mathfrak{R}$  function.

$$\int_0^z f_{Y^2}(Y^2 = y) \cdot f_{Y^1}(Y^1 > z | Y^2 = y) dy = \int_0^z f_{Y^2}(y) \mathfrak{R}_y^z(v) dy.$$

It can be similarly expanded by  $Y^1$ , i.e.,  $\int_z^{D_y} f_{Y^1}(y) \ell_y^z(v) dy$ .

The expected number of relevant items among newly retrieved items is given by  $n \cdot Pr(Y^2 \leq z, Y^1 > z, Y^2 < X)$ , which is then expanded as a function of  $Y^2$ :

$$n \cdot \int_0^z f_{Y^2}(Y^2 = y) \cdot Pr(Y^1 > z, X \geq Y^2 | Y^2 = y) dy.$$

For a given value  $y \in Y^2$ , the two events,  $X \geq y$  and  $Y^1 > z$ , are independent. Thus, their joint probability function is expressed as the dot product of  $Pr(X \geq y | Y^2 = y)$  and  $Pr(Y^1 > z | Y^2 = y)$ . As a result, the above equation is rewritten as

$$n \int_0^z f_{Y^2}(y) \mathfrak{R}_y^z(v) \overline{F}_X(y) dy.$$

Finally, the Precision metric of the user model is as follows:

$$\frac{(Ret^1 - Ret^0) \cap Rel^1}{Ret^1 - Ret^0} = \frac{\int_0^z f_{Y^2}(y) \mathfrak{R}_y^z(v) \overline{F}_X(y) dy}{\int_0^z f_{Y^2}(y) \mathfrak{R}_y^z(v) dy}.$$

The Recall metric of this scenario equals to that of a stationary user at  $t_2$ , since the simple client does not cache any retrieved objects after the rendering.

Fig. 7 reveals that, under uniform distributions of  $X$  and  $Y$ , the precision metric of a moving user by the user model deteriorates linearly as the user’s visible range increases and is always worse than that of a stationary user. It also depicts that

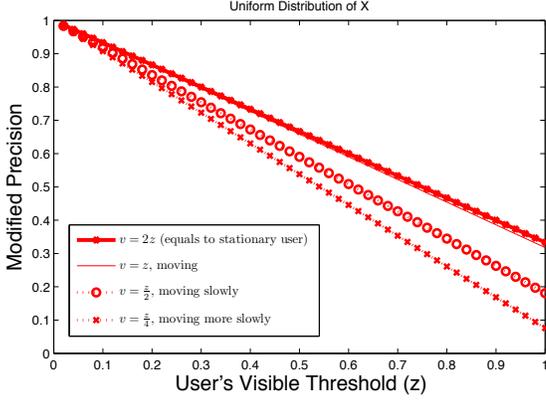


Figure 7: Distribution of modified Precision metric by user model with a simple strategy. A moving speed of the user ( $v$ ) is configured by the multiple of the user's visible threshold ratio ( $z'$ ). If  $v = 2z'$ , the modified  $P$  of a moving user will be the same as the original  $P$  of a stationary user, because there exist no overlapping objects when the user moves.

a slower user tends to retrieve visibly irrelevant objects more. Such tendency also appears with a normal distribution of  $X$ .

From the analysis, we observe that a moving user with a simple prefetching strategy experiences worse visual experience than the stationary user, indicating that the accuracy for a static user is the upper bound of the performance for a moving user.

#### 4.2.2. Single lookahead and caching

Consider a slightly advanced operation scenario illustrated in Fig. 8(a). At time  $t_0$ , a user with a visible threshold  $z$  was located somewhere in 2D space. Then he moved to the current position at time  $t_1$  and shall move to a new location at time  $t_2$ . Without loss of generality, the user locations at time  $t_0, t_1$ , and  $t_2$  are assumed to be  $O_0(-v, 0)$ ,  $O_1(0, 0)$ , and  $O_2(v, 0)$ , respectively.

A client using a single lookahead and caching policy keeps all collected items that were retrieved during a previous timeslot (*single caching*) and fetches new items that will have been identified by a next timeslot in advance (*single lookahead*). In other words, at  $t_1$ , the user has the previous search results ( $Ret^0 \cup Ret^1$ ) that were obtained at  $t_0$  and receives new search result  $Ret^2$ . If he moves to  $O_2$  at  $t_2$ ,  $Ret^0$  will be discarded from the user's local storage and  $Ret^3$  will be fetched. Therefore, all the retrieved items at  $t_1$  are  $Ret^0 \cup Ret^1 \cup Ret^2$ , while visibly relevant items at  $t_1$  are  $Rel^1$ . Finally, the modified Precision and Recall measures by the single lookahead and caching policy are expressed as

$$P = \frac{(Ret^2 - (Ret^1 \cup Ret^0)) \cap Rel^1}{Ret^2 - (Ret^1 \cup Ret^0)}, \quad R = \frac{Rel^1 \cap (Ret^0 \cup Ret^1 \cup Ret^2)}{Rel^1}.$$

Newly retrieved items are among the prefetched objects that have not been retrieved at  $t_0$  and  $t_1$ . Their probabilistic form is then  $n \cdot Pr(Y^2 \leq z, Y^1 > z, Y^0 > z)$ , where  $Y^0, Y^1$ , and  $Y^2$  are  $O_0P, O_1P$ , and  $O_2P$  for a given object location  $P$ , respectively.  $Y^2 \leq z, Y^1 > z, Y^0 > z$  are then reduced to  $Y^2 \leq z, Y^1 > z$ . The expected number of newly retrieved objects is described by the

following integral form:

$$\begin{aligned} Ret^2 - (Ret^1 \cup Ret^0) &= n Pr(Y^2 \leq z, Y^1 > z) \\ &= n \int_{Y^1} Pr(Y^1 = y, Y^2 \leq z) dy \\ &= n \int_z^{v+z} f_{Y^1}(y) \ell_y^z(v) dy \end{aligned} \quad (24)$$

The expected number of relevant items at  $t_1$  ( $Rel^1$ ) is the same as that of the user model in Eq. 13. It can also be equally expressed as an integral of  $Y^1$ . The probability  $Pr(X \geq Y^1)$  is expanded by the total probability theorem. For a given object distant at  $y \in Y^1$ , the probability that its visible distance is greater than or equal to the actual distance is  $Pr(X \geq y)$ , which by definition is reduced to  $1 - F_X(y)$  or simply  $\bar{F}_X(y)$ . Since  $X$  and  $Y$  are independent, their joint density function is the product of two individual density functions.

$$Rel^1 = n \int_{Y^1} f_{Y^1}(y) \bar{F}_X(y) dy$$

Since  $\bar{F}_X(y)$  is zero where  $y \geq D_X$ , the above equation is finalized as

$$Rel^1 = n \int_0^{D_X} f_{Y^1}(y) \bar{F}_X(y) dy. \quad (25)$$

The number of relevant items among newly retrieved objects is the cardinality of  $(Ret^2 - (Ret^0 \cup Ret^1)) \cap Rel^1$ . Its probabilistic form is written as an integral of  $Y^1$ :

$$n \int_z^{D_Y} f_{Y^1}(y) \cdot Pr(Y^2 \leq z, X \geq Y^1 | Y^1 = y) dy$$

Although two events,  $Y^2 \leq z$  and  $X \geq Y^1$ , are dependent on  $Y^1$ , they are independent of a chosen value  $y \in Y^1$ . Thus, the above expression is rewritten as the product of  $Pr(Y^2 \leq z | Y^1 = y)$  and  $Pr(X \geq y)$ . Using Lemma 2, we obtain the following result:

$$(Ret^2 - (Ret^1 \cup Ret^0)) \cap Rel^1 = n \int_z^{D_X} f_{Y^1}(y) \ell_y^z(v) \bar{F}_X(y) dy \quad (26)$$

The total number of retrieved items ( $Ret^0 \cup Ret^1 \cup Ret^2$ ) is divided into three non-overlapping sets:  $Ret^1, Ret^0 - Ret^1$ , and  $Ret^2 - Ret^1$ . Due to the symmetric nature of object locations in a uniform distribution of  $Y$ ,  $Ret^0 - Ret^1$  and  $Ret^2 - Ret^1$  are probabilistically identical. Thus, the total number of retrieved items is the sum of  $Ret^1$  and  $2 \cdot (Ret^2 - Ret^1)$ . As a result, relevant items among all retrieved items are rewritten as a union of two disjoint sets.

$$Rel^1 \cap (Ret^0 \cup Ret^1 \cup Ret^2) = (Rel^1 \cap Ret^1) + 2(Rel^1 \cap (Ret^2 - Ret^1))$$

The first term,  $Rel^1 \cap Ret^1$ , is the same as that of a stationary user.

$$Rel^1 \cap Ret^1 = n \cdot Pr(X \geq Y^1, Y^1 \leq z) = n \int_0^z f_{Y^1}(y) \bar{F}_X(y) dy \quad (27)$$

The second term,  $2 \cdot (Rel^1 \cap (Ret^2 - Ret^1))$ , is twice the number of relevant items among newly retrieved items. Its integral form is easily derived from Eq. 26:

$$2 \cdot (Rel^1 \cap (Ret^2 - Ret^1)) = 2n \int_z^{D_X} f_{Y^1}(y) \ell_y^z(v) \bar{F}_X(y) dy \quad (28)$$

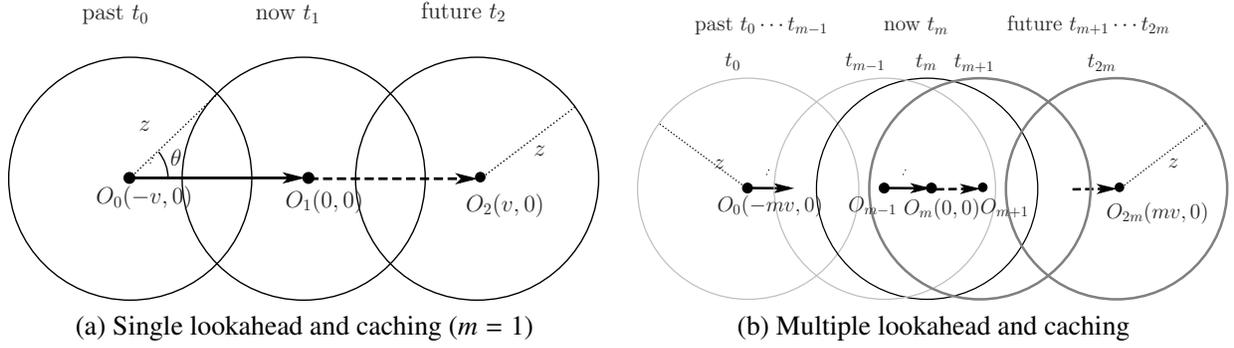


Figure 8: A user's moving scenario for (a) single and (b) multiple caching and prefetching policy. Every time interval  $t_{i+1} - t_i$ , a user moves from  $O_i$  to  $O_{i+1}$ , while caching data collected past  $t_0$  to  $t_{m-1}$  and prefetching next  $t_{m+1}$  to  $t_{2m}$ .

Combining Eqs. 27 and 28, we obtain the expected number of relevant items among all retrieved items as follows:

$$\begin{aligned} & Rel^1 \cap (Ret^0 \cup Ret^1 \cup Ret^2) \\ = & n \left( \int_0^z f_Y(y) \bar{F}_X(y) dy + 2 \int_z^{D_X} f_Y(y) \ell_y^z(v) \bar{F}_X(y) dy \right) \quad (29) \end{aligned}$$

#### Summary

For a given user threshold  $z$ , the modified precision measure of the user model with single lookahead and caching is obtained from Eqs. 24 and 26.

$$P = \frac{\int_z^{D_X} f_Y(y) \ell_y^z(v) \bar{F}_X(y) dy}{\int_z^{v+z} f_Y(y) \ell_y^z(v) dy} \quad (30)$$

The modified recall measure is similarly obtained from Eqs. 25 and 29.

$$R = \frac{\left( \int_0^z f_Y(y) \bar{F}_X(y) dy + 2 \int_z^{D_X} f_Y(y) \ell_y^z(v) \bar{F}_X(y) dy \right)}{\int_0^{D_X} f_Y(y) \bar{F}_X(y) dy} \quad (31)$$

#### 4.2.3. Multiple lookahead and caching

We further investigate a more aggressive caching and prefetching scheme: multiple lookahead and caching. It allows a user machine to prefetch objects much farther in time than the single look-ahead strategy and use all cached items that have been stored so far.

A usage example is visualized in Fig. 8(b): there exist  $(2m + 1)$  non-concentric congruent circles of radius  $z$ , each of which represents a user's visible region at each time, and they are equally spaced along the x-axis by  $v$ . At time  $t_{m-1}$ , a user at  $O_{m-1}$  has stored objects that had been recognized during the last  $m$  timeslots and the future  $m$  timeslots - i.e.,  $Ret^{-1} \cup \dots \cup Ret^{2m-1}$ . As the user moves to  $O_m$  by  $t_m$ , it drops  $Ret^{-1}$  and prefetches new objects  $Ret^{2m} - Ret^{2m-1}$ .

The newly prefetched objects by the multiple lookahead scheme, however, are less visibly relevant than those by the single scheme, since they are located much farther away than the user's current location, leading to a much lower Precision value. On the other hand, the Recall metric by the multiple scheme is higher than by the single scheme, since all items,

$Ret^0 \cup \dots \cup Ret^{2m}$ , contain more visibly relevant objects than singly cached items. Eqs. 32 and 33, assuming all consecutive circles overlap with each other, show the approximation of the precision and recall metrics by the multiple lookahead and caching method<sup>6</sup>.

$$\begin{aligned} P &= \frac{\int_{\sqrt{z^2+pq}}^{p+z} g(p) \Delta - \int_{\sqrt{z^2+pq}}^{q+z} g(q) \Delta}{\int_{\sqrt{z^2+pq}}^{p+z} f_Y(y) g(p) dy - \int_{\sqrt{z^2+pq}}^{q+z} f_Y(y) g(q) dy} \quad (32) \\ R &\approx \frac{\int_0^z \Delta + \int_z^{\sqrt{p^2+z^2}} \arccos \left( \sqrt{1 - \left(\frac{z}{y}\right)^2} \right) \Delta + \int_{\sqrt{p^2+z^2}}^{p+z} g(p) \Delta}{\int_0^{D_X} \Delta} \quad (33) \end{aligned}$$

where  $p = mv$ ,  $q = (m-1)v$ ,  $z \geq \frac{p-q}{2}$ ,  $g(o) = \arccos \frac{y^2+o^2-y^2}{2yo}$ , and  $\Delta$  represents  $f_Y(y) \bar{F}_X(y)$ .

The derivation of the equations, although it is non-trivial to present proofs, is omitted here, since it can be obtained straightforwardly.

#### 4.3. Analysis of Object AOI Model

In Section 3, we concluded that the object model provides the optimal solution for a stationary user in a static environment, as long as every user possesses the same sight vision and the visible distance of an object is set to the ideal distance.

The object model, however, has one crucial practical issue: the total number of visibly relevant objects fetched by this model tends to be much larger than that by the user model. For example, the expected number of retrieved items by the object model is  $\frac{nD_X^2}{3D_Y^2}$  if  $X$  and  $Y$  are uniformly distributed. It implies that, if  $D_X$  equals  $D_Y$ , the object model should retrieve "one third" of all objects in a space.

If a user moves in a static environment, the object model may utilize the local storage space of the user client. It means that the user doesn't have to retrieve all visibly relevant items at once, because some of the items were retrieved during previous rounds. Since the object model is guaranteed to retrieve only visibly relevant objects, we focus only on analyzing the expected number of newly retrieved objects. We use the same moving scenario presented in Section 4.2.1.

<sup>6</sup>Detailed proofs are omitted due to space limitation.

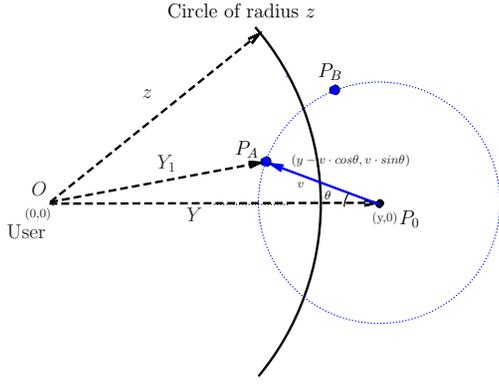


Figure 9: A simple object's moving scenario. The object with moving speed  $v$  at  $P_0(y, 0)$  may move to either inside user's visible range ( $P_A$ ) or outside ( $P_B$ ).

A newly recognized object is one that was invisible when a user was on  $O_1$  and becomes visible after the user moves to  $O_2$ . Thus, the probabilistic expression that an object is newly visible is expressed as  $Pr(Y^1 > X, Y^2 \leq X)$ , which can easily be expanded as double integral of  $X$  and  $Y^1$ .

$$\int_X f_X(x) \int_x^{D_Y} f_{Y^1}(y) Pr(Y^2 \leq x | Y^1 = y, |O_1 O_2| = v) dy dx$$

After substituting  $Pr(Y^2 \leq x | Y^1 = y, |O_1 O_2| = v)$  with  $\ell_y^x(v)$ , we obtain the expected number of newly retrieved objects as follows:

$$\begin{aligned} Ret_o^1 - Ret_o^0 &= n \cdot Pr(Y^1 > X, Y^2 \leq X) \\ &= n \int_0^{D_X} \int_x^{D_Y} f_X(x) f_{Y^1}(y) \ell_y^x(v) dy dx. \end{aligned}$$

## 5. A Stationary User in a Moving Environment

In this section, we present the performance model of the visibility algorithms under the following scenario:

**Scenario:** A user stands at a fixed position  $O$ , while every object with the same moving speed  $v$  moves randomly from one place to another in a circle of radius  $D_Y$ . Let the object in the circle be originally placed at  $P_1$  and be relocated to  $P_2$  after a given time interval, where the length of  $\overline{P_1 P_2}$  is  $v$ .

The performance metrics for this analysis reuse the same definitions for the analyses of a moving user in a stationary environment.

### 5.1. Analysis of User AOI Model

As all objects in the sample space move randomly, any aggressive caching and prefetching strategy cannot be applied in this model. It may even hurt the system performance severely. For example, the location of a cached object needs to be updated regularly, which requires additional state update cost between a server and a client. Since such update cost is very expensive, we assume a minimal prefetching and caching policy, which updates previously retrieved objects only once and disposes them after a single reuse.

A user whose visible threshold is given by  $z$  is centered at  $O$ . As exemplified in Fig. 9, an object that was initially positioned

outside the user's visual scope (say,  $P_0(y, 0)$ ) moves to either inside the user's visible range ( $P_A$ ) or outside ( $P_B$ ). If entering into the user's region, it will be newly recognized by the user model. Assuming  $Y$  and  $Y^1$  are the distances of the object at  $P_0$  and  $P_A$  (or  $P_B$ ) to the user, respectively, the probability that the object is newly identified by the user model is expressed as  $Pr(Y^1 \leq z, Y > z)$ , which is then expanded as an integral of all possible  $Y$  positions, which are located outside the user's scope. The portion of the arc length of the circle centered at  $P_1$  within the user's scope can be derived from Lemma 2, where  $r_1$ ,  $r_2$  and  $v$ , respectively, correspond to  $v$ ,  $z$ , and  $y$ . As a result, the probability is induced to  $\int_z^{y+z} f_Y(y) \ell_v^z(y) dy$  and the expected number of newly retrieved objects is computed as

$$n \cdot Pr(Y > z, Y^1 \leq z) = n \int_z^{y+z} f_Y(y) \ell_v^z(y) dy. \quad (34)$$

The probability that a newly retrieved object is also visibly relevant after its movement is expressed as  $Pr(Y > z, Y^1 \leq z, X \geq Y^1)$ . It is also expanded as the double integral of  $X$  and  $Y$ . If  $X$  is less than  $z$ , the probabilistic condition  $Y^1 \leq X, Y^1 \leq z$  will be dominated by  $Y^1 \leq X$ . Otherwise, it would be  $Y^1 \leq z$ . Thus, the expected number of newly retrieved and visibly relevant objects is

$$\begin{aligned} n &\cdot Pr(X \leq z) \cdot Pr(Y > z, Y^1 \leq X) \\ &+ n \cdot Pr(X > z) \cdot Pr(Y > z, Y^1 \leq z) \\ &= n \int_0^z \int_z^{z+v} f_X(x) f_Y(y) \ell_v^x(y) dy dx \\ &+ n \int_z^{D_X} \int_z^{z+v} f_X(x) f_Y(y) \ell_v^z(y) dy dx \end{aligned}$$

The relevant items when objects move to  $P_1$  are computed from  $Ret^1$ , whose probabilistic model was given in Eq. 13.

All retrieved items among relevant items are the sum of the relevant items that are either retrieved or discarded after objects' movement :  $n \cdot Pr(Y^1 \leq X, Y^1 \leq z) + n \cdot Pr(Y^1 \leq X, Y^1 > z, Y \leq z)$ . The left term (retrieved) is computed from Eq. 14. The right term (discarded) is the expected number of objects which were retrieved at the current position and then become invisible at the next position. Although no longer in service, the discarded objects can still be available from the user's local storage, thus being good candidates that can be reused for the time being. Their visible distance needs to be larger than  $z$ . Otherwise, they could not be visibly relevant, because  $Y^1 > z, z \geq X \equiv Y^1 > X$ . Therefore, the expected number of visibly relevant but discarded objects after the movement is computed as follows:

$$\begin{aligned} n \cdot Pr(Y^1 \leq X, Y^1 > z, Y \leq z) \\ &= n \int_z^{D_X} f_X(x) Pr(Y \leq z, z < Y^1 \leq X | X > z) dx \\ &= n \int_z^{D_X} f_X(x) \int_0^z f_Y(y) \{ \mathfrak{R}_v^z(y) - \mathfrak{R}_v^x(y) \} dy dx \end{aligned}$$

Using the above results, the modified  $P$  and  $R$  metrics of the user model, where only objects move, are obtained as follows:

$$P = \frac{\int_0^z \int_z^{z+v} \ell_v^x(y) \Delta + \int_z^{D_X} \int_z^{z+v} \ell_v^z(y) \Delta}{\int_z^{y+z} f_Y(y) \ell_v^z(y) dy}$$

$$R = \frac{\int_0^z f_X(x)F_Y(x)dx + \int_z^{D_x} f_X(x)F_Y(z)dx + \square}{\int_0^{D_x} f_X(x)F_Y(x)dx},$$

where  $\Delta = \int_0^z f_X(x)f_Y(y)dydx$ ,  $\square = \int_z^{D_x} f_X(x)\int_0^z f_Y(y)\{\mathfrak{R}_v^z(y) - \mathfrak{R}_v^x(y)\}dydx$ .

## 5.2. Analysis of Object AOI Model

Newly retrieved objects by the object model are modelled as  $Pr(Y > X, Y^1 \leq X)$ . Expanded as an integral of  $Y$ , we have  $\int_x^z f_X(x) \cdot Pr(Y > x, Y^1 \leq x)$ . For a given  $x$ ,  $Pr(Y > x, Y^1 \leq x)$  can easily be induced from Eq. 34 by substituting  $z$  with  $x$ . Finally, the expected number of retrieved objects is obtained as follows:

$$\int_0^{D_x} f_X(x) \int_x^{x+v} f_Y(y)\ell_v^x(y)dydx$$

So far, we have presented the analytical models of the performance metrics of individual AOI filtering approaches with different settings. Such modelling results are analyzed for accurately predicting the expected performance behaviors for different configuration settings by varying input parameters - *i.e.*, the relative distance between a user and its target object, the intrinsic visible strength of an object, and the moving speed of a moving user or object. In the following section, we report on the detailed performance characteristics of the individual filtering models.

## 6. Evaluation

To examine whether our analysis model predicts the performance of the visibility models accurately, we compare prediction results computed from our analytical models with simulation results. During the simulations, we varied the number of populated objects that are uniformly distributed, ranging from ten thousand to one million and collected the performance results of individual visibility models. We also present the analytical results and the simulation results for different distributions of  $X$  (a uniform and a normal distribution). For the normal distribution, we used the normalized mean value of 0.3 and the standard deviation of 0.05. It means that the average visible distance of an object occupies 30 % of the side length of the unit space. Overall, all the simulation results matched their corresponding prediction results accurately and consistently, which validates the correctness of our analytical models. To emphasize the variability of the simulation results, we only show the simulation results with the medium-sized object population (*i.e.*, 100K).

### 6.1. A Stationary User in a Stationary Environment

Fig. 10(a) shows the E-measure distributions of the user model. This figure reveals two typical performance trends of the user model. First, there exists a local minima at non-zero visible threshold of the user. It appears around (normal distribution) or after (uniform distribution) the average visible threshold. Second, the given normal distribution shows a better performance than the given uniform distribution. Moreover, its local minima is comparable to the optimal condition.

Fig. 10(b) depicts the E-measure distributions of the object/hybrid model, where the visible distance of an object is configured as its ideal visible distance. Since the object model always achieves the optimality of the determination quality, we draw the results of the hybrid model by varying the user's visible threshold. The figure validates that the ideal visible distance of the object is the most crucial factor in the performance of the hybrid model, while deferring the use of user's visible threshold. If the visible distance is configured longer than the ideal, the figure reveals that there exists the most suitable user threshold that achieves the local minima of the E-measure metrics, which significantly improves the performance.

In summary, the object model achieves optimality only when user's visible strength is homogeneous, the visible distance of an object is heterogeneous, and the ideal visible distance is derivable. The user model tends to be inferior to the object model, but it can achieve comparable performance under certain conditions. If the hybrid model is applied, the system designer should estimate how accurately the visible distance is approximated to its ideal. If determined overestimated or correctly, the system should not consider the user's visible threshold. Otherwise, it should use the threshold that matches the local minima of the E-measure metrics.

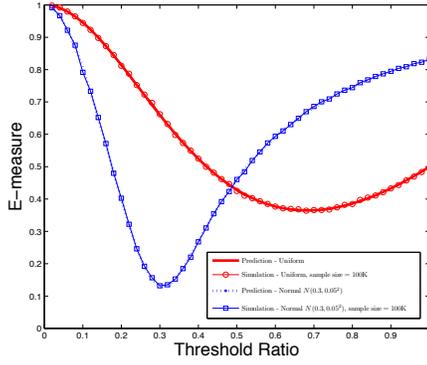
### 6.2. A Moving User in a Stationary Environment

Fig. 11 depicts the E-measure values obtained by two variants of the user model. It shows similar performance tendencies that we observed in the case where every user and object are stationary: there exists a local minima of the performance and it is typically worse than that by the user model in the stationary case, while its optimal user's thresholds tend to be placed ahead of those of the user model in the stationary case. These observations indicate that newly retrieved objects by the user model are less visibly relevant, thus hurting the Precision significantly. The figure also reveals that the single lookahead and caching scheme is generally superior to the multiple scheme at the individual local minima.

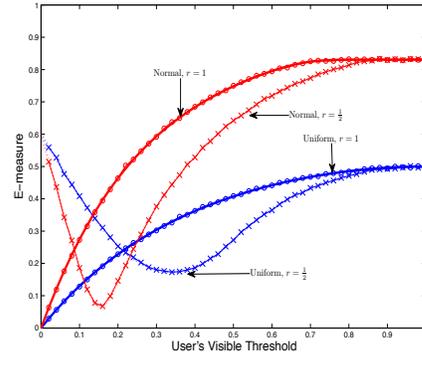
As stated earlier, the object model has a crucial flaw in the stationary case, requiring to retrieve an improbable number of visibly relevant objects at once. In Fig. 12 however, we observed that the object model newly retrieves a much smaller number of visibly relevant objects, which is very practically meaningful, although it does not surpass the user model all the time. In reality, the moving speed of objects will be much smaller than plotted in the figure, since the visible distance extends much farther than conventional distance ranges and the speed is normalized by the maximum visible distance. Thus, if objects move at most 1 - 2 % of the unit space per time unit, the size of the actually retrieved objects will be much lower than 2 % of the whole object population size. Additionally, the figure also demonstrates that the newly retrieved objects by the user model are independent of the distribution of  $X$ , matching the prediction of Eq. 24.

### 6.3. A Stationary User in a Moving Environment

Interestingly, the performance metrics (Fig. 13 for the E-measure and Fig. 14 for the size of newly retrieved objects) of

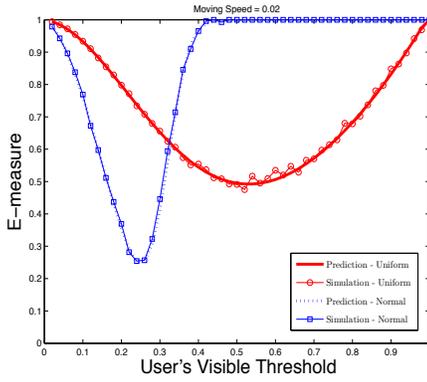


(a) User model

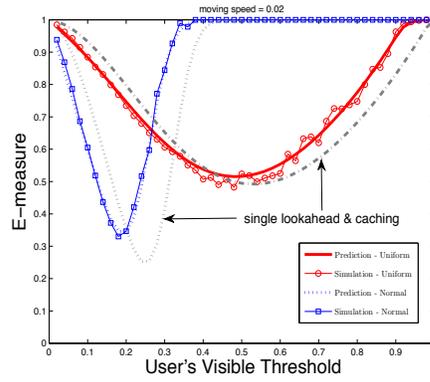


(b) Hybrid model

Figure 10: Analytical and simulation results of E-measure values of (a) user and (b) hybrid model. The curves are plotted as a function of normalized user's visible threshold in a uniform and normal distribution  $N(0.3, 0.05^2)$  of  $X$ .



(a) single lookahead ( $m = 1$ )



(b) multiple lookahead ( $m = 5$ )

Figure 11: Analytical and simulation results of the E-measure values of the user model with different lookahead and caching policies. They are plotted as a function of user's visible threshold for uniform and normal distribution of  $X$ . During the simulation runs, the user moved 2 % of the given unit space.

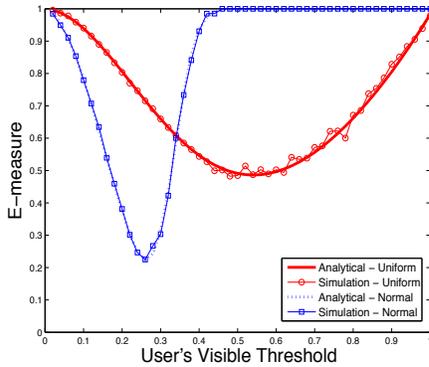


Figure 13: The E-measure metrics of user model for a stationary user with moving objects that are assumed to move 2 % of the unit space.

both models follow the same tendencies depicted in Figs. 11 and 12. It is especially surprising that a minimal prefetching and caching scheme for a stationary user with moving objects is mimicking very similar results that the complicated prefetching and caching scheme of the user model for a moving user achieved.

## 7. Discussion and Conclusions

In this article, we classify existing visibility determination algorithms into three models : user, object, and hybrid model. The user model assumes that only users have their own unique visual strength (we interchangeably call it as “user's visible threshold”); the object model assumes that every user possesses equal sight vision, while every object has a unique visible characteristic stemming from its spatial attributes such as size; and the hybrid model assumes every object and user have separate visible criteria. Using the visual acuity model to derive the ideal visible distance both from the user's visual strength and from the object size, we formally define their visible condition and derive the analytical model of their performance metrics (Precision, Recall, and E-measure) in three distinct cases: (1) a stationary user in a stationary environment, (2) a moving user in a stationary environment, and (3) a stationary user in a moving environment. To embrace the dynamics of the last two cases, we propose modified definitions of conventional precision and recall metrics.

In case of a stationary user with stationary objects, the object model achieves optimal performance, in which all visibly relevant objects are retrieved and all retrieved objects are visibly

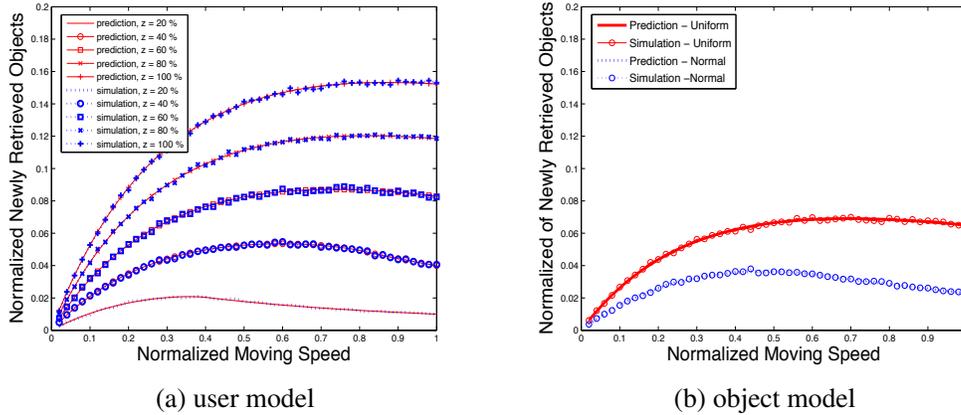


Figure 12: Analytical and simulation results of the number of newly retrieved objects by (a) the user model with single scheme and (b) the object model are plotted as a function of user’s moving speed ( $v$ ).

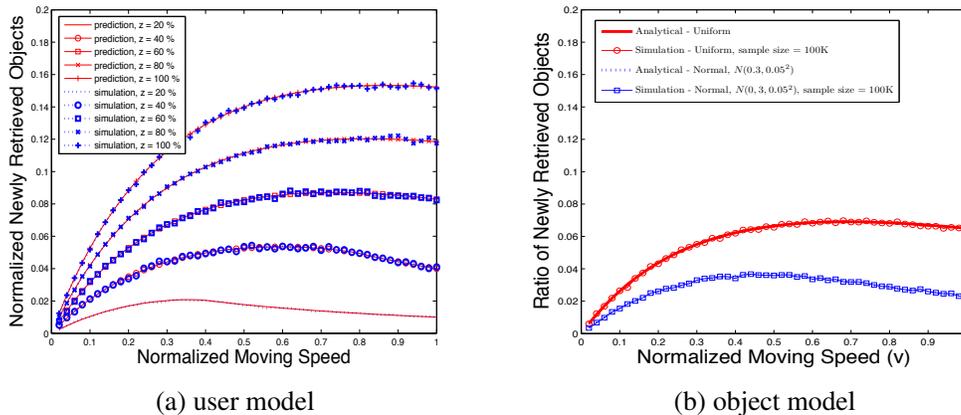


Figure 14: The normalized size of newly retrieved objects by both models for a stationary user with moving objects that are assumed to have a normalized moving speed  $v$ . For clarity purposes, the user model plots the results under uniform distribution of  $X$  and  $Y$ , while the object model under two distributions of  $X$  and uniform distribution of  $Y$ .

relevant. The determination quality of the user model depends on user’s visible threshold and there exists a local minima at a specific visible threshold value. But the local minima is still far from the optimal point. The hybrid model suggests that (1) if the visible distance of an object is over-estimated, the use of the user’s visible threshold should not be used to avoid performance degradation and (2) if underestimated, the user’s visible threshold should be assigned to the value that satisfies the local minima of the performance. In case of a moving user with stationary objects, the user model has a worse performance than that in the case of the stationary user, mainly due to the modified definition of the performance metrics. Among several intelligent policies for the user model, a single lookahead and caching scheme has a slightly better local minima than others. In terms of the number of newly retrieved objects, the object model requires fewer objects than that in the case of the stationary user, while it is comparable to that of the user model. Lastly, in case of a stationary user with moving objects, the user and the object models showed similar performance trends observed in the case of the moving user.

So far, we have described how much different choices of visibility algorithms impact the system performance in terms

of search and retrieval quality and retrieval size. We also provided various decision policies that fit well for individual visibility algorithms. Although all analyses presented in this study were focused on the visibility determination of client/server based networked virtual environments, our arguments and conclusions can be immediately applied to in many practical areas, where spatial attributes are necessary and can be transformed into quantifiable values. Please note that our conclusions may not be applicable for some attributes that can not be easily transformed to a value such as *interest*, *attraction*. If a system adopts these qualitative attributes and intentionally associates them with a distance, we expect our analytical conclusions to still reveal intrinsic performance properties of the system and to suggest the optimal configuration scenario.

## References

- [1] T. Akenine-Moller, T. Moller, E. Haines, Real-Time Rendering, A. K. Peters, Ltd., Natick, MA, USA, 2002.
- [2] D. Cohen-Or, Y. L. Chrysanthou, C. T. Silva, D. Durand, A survey of visibility for walkthrough applications, IEEE Transactions on Visualization and Computer Graphics 09 (3) (2003) 412–431.
- [3] H. Abrams, K. Watsen, M. Zyda, Three-tiered interest management for large-scale virtual environments, in: VRST ’98: Proceedings of the ACM

symposium on Virtual reality software and technology, ACM Press, New York, NY, USA, 1998, pp. 125–129.

- [4] J.-S. Boulanger, J. Kienzle, C. Verbrugge, Comparing interest management algorithms for massively multiplayer games, in: Proceedings of the 5th ACM workshop on Network and System Support for Games (NETGAMES) '06, 2006.
- [5] S. Benford, L. Fahlén, A spatial model of interaction in large virtual environments, in: ECSCW'93: Proceedings of the third conference on European Conference on Computer-Supported Cooperative Work, Kluwer Academic Publishers, Norwell, MA, USA, 1993, pp. 109–124.
- [6] E. Teler, D. Lischinski, Streaming of complex 3d scenes for remote walkthroughs, *Comput. Graph. Forum* 20 (3).
- [7] J. Chim, R. W. Lau, H. V. Leong, A. Si, Cyberwalk: a web-based distributed virtual walkthrough environment, *IEEE Transactions on Multimedia* 5 (4) (2003) 503–515.
- [8] B. Seo, R. Zimmermann, Edge indexing in a grid for highly dynamic virtual environments, in: MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia, 2006, pp. 402–411.
- [9] T. Ohshima, H. Yamamoto, H. Tamura, Gaze-directed adaptive rendering for interacting with virtual space, *Virtual Reality Annual International Symposium (VRAIS 96) 00* (1996) 103.
- [10] J. D. Pfautz, Depth perception in computer graphics, Ph.D. thesis, Trinity College, University of Cambridge (2000).
- [11] N. R. Bartlett, J. L. Brown, Y. Hsia, C. G. Mueller, L. A. Riggs, *Vision and Visual Perception*, John Wiley & Sons, Inc., 1965.
- [12] C. J. Van Rijsbergen, *Information Retrieval*, 2nd edition, Dept. of Computer Science, University of Glasgow, 1979.

## Appendix A. Lemma 2

**Proof** To derive  $\ell_{r_1}^{r_2}(v)$ , we assume  $C_1$  and  $C_2$  be centered at  $O_1(0, 0)$  and at  $O_2(v, 0)$ . The spatial relations between  $C_1$  and  $C_2$  are classified in two cases:  $O_2$  is inside  $C_1$  and  $O_2$  is outside  $C_1$ .

**Case (1):  $O_2$  is inside  $C_1$  - i.e.,  $v < r_1$**

- If  $r_1 < r_2 - v$ , every point in  $C_1$  will be spatially covered by  $C_2$ . Thus, it returns one.
- If  $r_2 - v \leq r_1 \leq r_2 + v$ , the two circles will intersect. The ratio of the arc length of  $C_1$  intersecting with  $C_2$  to its circumference length is derived from the lemma 1.
- Otherwise ( $r_1 > r_2 + v$ ), there would exist no intersection point between the two circles.

**Case (2):  $O_2$  is outside  $C_1$**

- If  $r_1 < v - r_2$ , the two circles will be disjoint. In this case, there exist no points on  $C_1$  that also lie inside  $C_2$ .
- If  $d - r_2 \leq r_1 \leq v + r_2$ , the two circles intersect.
- Otherwise ( $r_1 > v + r_2$ ),  $C_1$  would cover  $C_2$ .

The derivation of  $\mathfrak{R}_{r_2}^{r_1}(v)$  is also similarly proved, assuming  $C_1$  and  $C_2$  be centered at  $O_1(-v, 0)$  and at  $O_2(0, 0)$ . The arc length of  $C_2$  outside  $C_1$ , if existing, is computed from  $2\pi\theta$ , where an intersection point on  $C_2$  that is expressed as  $(r_2 \cos \theta, r_2 \sin \theta)$  also lies on  $C_1$ , thus, satisfying  $(r_2 \cos \theta + d)^2 + (r_2 \sin \theta)^2 = r_1^2$ . Especially, if two circles are disjoint when  $r_1 < v - r_2$ , every point on  $C_2$  is located outside  $C_1$  as well. In such situation, the function  $\mathfrak{R}_{r_2}^{r_1}(v)$  returns one.

From the above cases, the lemma is correct. ■