

# A Study of Optimal Spatial Partition Size and Field of View in Massively Multiplayer Online Game Server

Youngsik Kim\*

*\*Department of Game and Multimedia Engineering,  
Korea Polytechnic University, Republic of Korea.*

*Orcid ID: 0000-0003-2842-4190*

## Abstract

In a Massively Multiplayer Online (MMO) game, the packet data between the server and the client must be minimized. Spatial partitioning and field of view processing are indispensable to reduce the packet data. This paper analyzes the size of the spatial partition and the range of the field of view. Experiments were conducted on the optimal size of the spatial partition and field of view in the 2D MMO game server with dummy clients. The best performance was achieved when the field-of-view range was 1.5 times the size of the spatial partition. In addition, the size of the space partition also has a large effect on the performance because the parallelism affects the number of object retrieval.

**Keywords:** Massively Multiplayer Online (MMO), Game Server, Spatial Partition, Field of View

## INTRODUCTION

An online game is a game in which two or more participants compete and compete in order to achieve the purpose of the game on the communication network. When the number of concurrent participants reaches from thousands to tens of thousands, it is commonly called Massively Multiplayer Online (MMO) Game. Especially when the genre of the game is Role-Playing, it is called Massively Multiplayer Online Game (MMORPG).

As mobile games using smart phones are more popular than online games using PCs, many mobile game companies are still developing many games. There are three functions of the game server. First, it is participant management. Log in, log out, save status information, and notify other participants. Second, dynamic state information management. The game server should manage status information such as the player's movement, level, items, and party settings related to the game progress. Third, space management and server communication. Server communication, space management, and message filtering [1]. As a communication structure of a game server, a state update message is a means for transmitting an event generated by a participant to other participants. Events that occur in the game space, such as attacking opponents or acquiring items, are delivered to clients that need to

synchronize via the server in the form of packets [1].

The server-client communication structure for effectively communicating status messages can be summarized as shown in Table 1 [2].

**Table 1:** Communication Structure between MMO Game Server and Clients [2]

Types	Number of Players	Descriptions
peer-to-peer	Max. 16	Client performs message forwarding, Communication burden is minimal, not scalable
client-server	16 ~ 200	Server performs message relay function, Message filtering on the server Depending on the number of participants, The server down is connected to the whole system down
map-based distributed server	More than 200	It divides virtual space into regions and deals with one region for each server.

The MMORPG genre has become more popular on mobile as well as on the PC, and the performance improvement of MMO servers has become even more important. In a large online game, it is important to reduce the number of packets exchanged as much as possible because it sends and receives packets to many clients. Spatial partitioning and field of view processing are indispensable factors in reducing this packet. A number of performance improvements are achieved by dramatically reducing the number of clients that need to send packets through space division and by reducing the amount of packets that need to be sent once again by users who are visible to the user through vision processing. However, optimization is not done simply by spatial partitioning and field of view processing. The size of the spatial partition and how the range of field of view is affected greatly affects the performance of the MMO server.

In [3], a mathematical method was proposed to calculate the allowable critical angle efficiently for the direction which is one of the main character state update variables in the dead reckoning implementation for the MMOFPS game. In [4], a grid-based spatial partitioning technique was proposed for static and dynamic objects in MMORPG games. In [5], a spatial partitioning technique was proposed for faster rendering improvement by improving the Quad-Tree technique, which is one of the existing space division methods, so that more efficient space division can be performed on static / dynamic objects that do not have height values in MMORPG . In [6], the performance of the clustering algorithm based on spatial partitioning is analyzed according to the size of various partitions through time complexity and experimental methods. The results show that, in most 3D environments, good performance can be obtained when the size of the partition is 0.5 power of the number of agents. In [7], a distributed MMORPG game server using 2Layer-Cell method was proposed.

This paper tries to find out the relationship between the size of the space division and the field of view and how the two have the optimal size. This paper is composed as follows. Section 2 will look more at spatial partitioning, especially coordinate hashing, which is often used in MMO servers. Section 3 describes the field of view processing. Section 4 investigates the optimization of the size of the spatial partition and the field of view. Finally, Section 5 concludes this paper.

### SPATIAL PARTITIONING TECHNIQUES FOR MMO GAME SERVER

The game space is divided into Zoned world and Seamless world according to the way in which the divided spaces are connected [2]. The Zoned world method divides the entire game space into independent zones and moves through each portal between the zones. On the other hand, the seamless world method divides the entire game space into several regions and allocates charge servers to each region and manages it without moving through the portal. MMORPG generally uses a map-based distributed server architecture shown in Table 1, which is a structure that efficiently handles the data and message traffic generated by the game without being limited by the size of the game space. Multi-resolution spatial partitioning is a method of managing the space of online games in top-down direction [8].

One world is divided into several zones, and each zone is managed by the responsible server. The boundary between zone and zone is information that only the system knows, and participants are not conscious. The zone is again divided into cells, which are the minimum space units, and the area of interest (AOI) is defined as the set of cells and neighbor cells to which the participant belongs. AOI, which is the area of interest of a participant, may be defined based on a cell or

regardless of a cell. The area of interest is effective to filter unnecessary message traffic by limiting the propagation of an event generated by a means for setting a surrounding area that a participant expects to be interested within an area of interest. Typical AOI configurations are Circle-type AOI [9] and Interleaved-squaring AOI [10].

Quad-Tree, Oct-Tree, and Coordinate Hashing are some of the spatial partitioning of map-based distributed server structure. Quad-Tree and Oct-Tree are good for preventing unnecessary memory waste. However, in order to obtain information of a specific cell, it is not suitable since it is necessary to determine which cell belongs to which cell among the separated virtual cells, and then there is a cost to access and a cost to maintain various cells. So MMOs use coordinate hashing, which is commonly used. It has multiple zones in a single World, which is managed by each responsible server. In this zone, again divided into cells of a certain size, divided and managed. The cell maintains location information and attribute information. More specifically, a cell with a size of 20m x 20m is created virtually. Each cell becomes the head node of the linked list of users in the zone. Whenever the position of the player changes, the index of the cell to which the cell belongs is recalculated. If the index is changed, the cell is removed from the user linked list of the cell indicated by the old index, and the cell is added to the user linked list of the cell indicated by the new index do. Figure 1 shows an example of the data structure of the cell in the map-based distributed game server.

In Figure 1, m\_cellX and m\_cellZ represent the index of the corresponding cell and have lock variables for synchronization between the object list of the player and the NPC and the thread. In this data structure and manner, the cells corresponding to the coordinates of the users at all times and their respective linked lists are maintained. It also helps in the field of vision processing by dividing the space. The search speed is much faster because the user is searched only in the cell to which he belongs or the cell around itself.

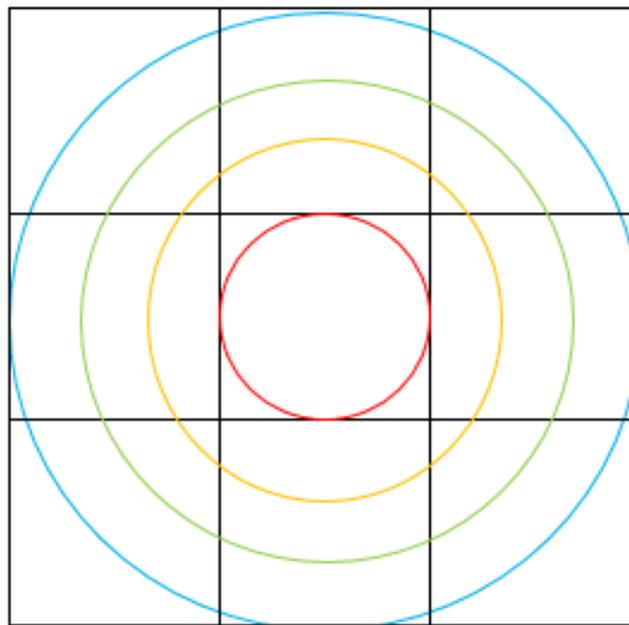
Cell Data Structure	
Cell	
{	
USHORT	m_cellX;
USHORT	m_cellZ
ViewObjectList	m_playerListInSector;
ViewObjectList	m_npcListInSector;
Lock	m_npc_list_lock;
Lock	m_player_list_lock;
};	

**Figure 1:** An Example of Cell Data Structure.

## FIELD OF VIEW PROCESSING

In MMORPG, the field of view is almost always treated as a circle. Assuming that the world has been partitioned by coordinate hashing, the number of cells can be varied depending on the view range or the size of the space division. If the size of the cell is large, it is necessary to search for objects out of sight, and the parallelism is reduced. Conversely, if the size of a cell is small, many cells must be searched.

Figure 2 shows the number of cells to be searched according to the field of view. In the second small circle, there are nine cells



**Figure 2:** Size of Field of View and Spatial Partitioning.

across the circle. Therefore, when one user searches for objects around the user will search for 9 cells. Also, whenever a player (user) moves or another player moves, he or she must notify the surrounding players of the list of objects in the cell and the packets corresponding to the updates.

Figure 3 shows the algorithm for the visual field processing when the player moves.

Field of View Processing Algorithm
<ol style="list-style-type: none"> <li>① After moving, determine which cell is the current cell and loop for the near cell that matches the size of the field of view.</li> <li>② Create a new object list to hold the objects in the view.</li> <li>③ For every cell that is being searched, loop through the list of objects managed by that cell to determine whether or not it is within the player's field of view. If it is within sight, put it in the New Object List.</li> <li>④ The objects in the New Object List are compared with the object list (Old Object List) that the player already had.</li> <li>⑤ If the object is not in the Old Object List, add this object, send the object add packet to me, access the relative object list, and if it is in the relative object list, send the mobile packet to the opponent. Sends a packet.</li> <li>⑥ If I have the object in the Old Object List, I send the mobile packet to me. If I am in the Relative Old View List, the correspondent also sends the mobile packet. If not, it adds me to the Relative Old View List and sends the object addition packet.</li> <li>⑦ If it is not in New Object List for all objects in Old View List, put it in Remove List.</li> <li>⑧ Remove the Old View List for each object while looping on the Remove List. If the object disappears, send the object removal packet to itself, remove the old object list if it is included in the other's old object list, and send the removal packet.</li> </ol>

**Figure 3:** Field of View Processing Algorithm for Player Movement.

## PERFORMANCE EVALUATION

The optimization experiment for the size of the segmentation and the field of view determines the degree of optimization by connecting the player as much as possible and randomly moving the number of simultaneous connections. The experiment will be a total of two experiments. In the first experiment, we will try to find out how the range of the field of view should be optimized by changing the size of the spatial segmentation and varying the field of view. The second experiment is an experiment in which the spatial resolution is varied and the field of view is applied as an optimal method through the conclusions derived from the first experiment.

In the second experiment, we will find the optimized space partition size. The client to be used for the experiment uses the dummy client created by using the window API. Run the 2D MMO server and run the dummy client to connect up to 10,000 clients from the client. In the Update loop, the client requests the server randomly every directional packet from each client connected to the server every second. The server receives this and processes the visual processing, spatial processing, synchronization, and sends the result to the client. On the client, it cuts to 100m x 100m and color is painted for each space. If there is no client, the color gradually becomes blue as more white.

The performance of the computer used in experiments is processor: Intel (R) Core (TM) i7-6700HQ CPU @ 2.60GHz, memory: 8GB, 64bit operating system, graphics card: Geforce Nvidia 960m.

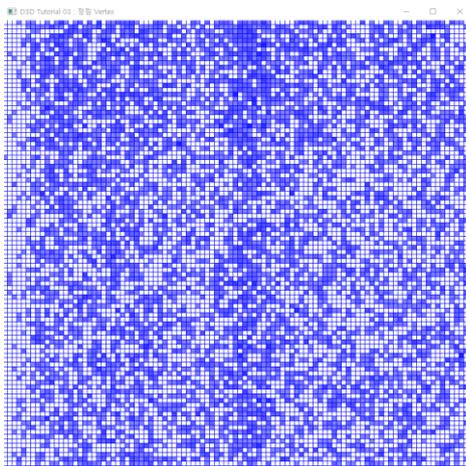


Figure 4: Screen Shot for Dummy Clients.

### Variable Field of View with Fixed Spatial Partitioning Size

When the size of the character is set as 1m, the experiment I fixed the spatial partition size to 100m and experimented with increasing the field of view. As you can see in Figure 5, when the radius was 75 m, the number of concurrent players was the greatest. The radius 75m is the second circle in the case of Figure 2. The size of the second circle is 1.5 times the size of

the spatial partition, and the performance is the best. And as the field of view grows larger, the result is that the number of concurrent players increases. This result is due to the fact that the number of objects to be searched increases as the field of view becomes wider, and the cost of searching and the number of packets transmitted accordingly increases.

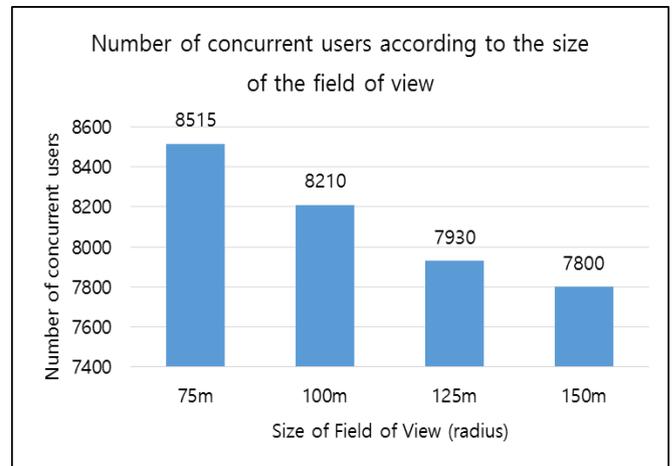


Figure 5: Number of concurrent users according to the size of the field of view.

### Variable Spatial Partitioning Size

Based on the optimal values from the first experiment, this paper performed two experiments. As shown in Figure 6, the second experimental result showed the highest optimization at 500. This is the best part when the visible part of the screen is 800m x 800m, as shown in the first experiment, when viewing 1.5 times the size of the spatial partition. In other words, 500m x 1.5 is 750m, which is similar to the size of the map inside the screen, and the best results are obtained. As the number of objects to be searched increases, the performance becomes lower.

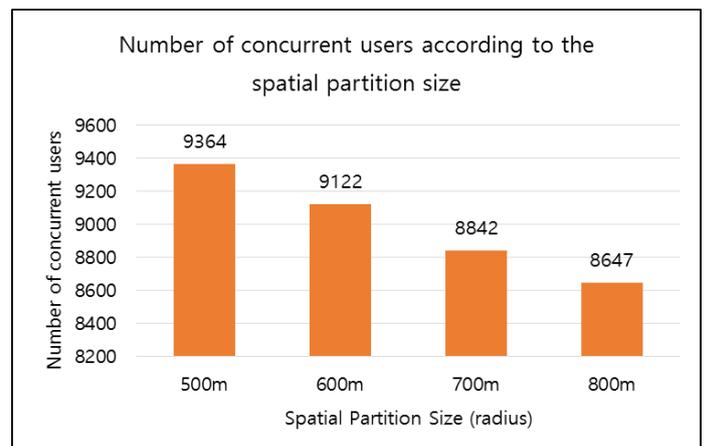


Figure 6: Number of concurrent users according to the spatial partition size.

## CONCLUSION

This paper analyzes the optimal size of the space and the field of view in the 2D MMO game server with dummy clients through experiments. It is important to determine the size of the field because the field of view is too large to increase the number of objects to be searched, and too small to be parallel. Experimental results show that the best performance is obtained when the field of view is set to 1.5 times the size of the space division. In addition, the size of the space partition also has a large effect on the performance because the parallelism affects the number of object retrieval. This paper has studied optimization based on 2D game, but the future study needs to examine whether it applies to 3D game with more variables than 2D. If not, what variable should be considered, it is necessary to study whether to set the size of the spatial partition.

## ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. 2016-0-00204, Development of mobile GPU hardware for photo-realistic real time virtual reality).

## REFERENCES

- [1] Seokjong Yu, 2005, "Distributed Game Server and Spatial Partitioning Techniques", Communications of the Korean Institute of Information Scientists and Engineers 23(6), pp.29-35.
- [2] Gwang-Ho Yang, Gwang-Hyeon Sim, Il-Gyu Park, and Jong-Sung Kim, 2001, "Technology Trend of Online Game Server", Electronic Telecommunications Trend Analysis, 16 (4), 14-22.
- [3] Lim, Jong-Min, Dong-Woo Lee, and Youngsik Kim, 2014, "An Efficient Method to Update Character Moving Directions for Massively Multi-player Online FPS Games." Journal of Korea Game Society 14 (5), pp.35-42.
- [4] Young-Mahan Kim, Kil-Sang Yoo, Je-Hyeok Yeon, Won-Hyung Lee, 2004, "Grid Spatial Segmentation in MMORPG," the Internet Society of Korea, 2004 Spring Conference Vol.5 No.1, pp.289-292.
- [5] Young-Mahan Kim, Kil-Sang Yoo, Je-Hyeok Yeon, Won-Hyung Lee, 2003, "A Grid space partition method using leaf nodes of QuadTree on MMORPG", Journal of The Korean Society for Computer Game, (3), pp.12-18.
- [6] Jae Moon Lee, Sung Dong Kim, 2010, "A Grid space partition method using leaf nodes of QuadTree on MMORPG", Journal of The Korean Society for Computer Game, (20), pp.111-118.
- [7] Su-Min Jang, Jae-Soo Yoo, 2007, "An Efficient MMORPG Distributed Game Server", Journal of The Korea Contents Association 7(1), pp.32-39.
- [8] B. Ng, A. Si, R. W.H. Lau, F. W.B. Li, 2002, "A Multi-Server Architecture for Distributed Virtual Walkthrough", Proceedings of the ACM symposium on Virtual Reality Software and Technology, pp.163-170.
- [9] C. M. Greenhalgh, and S. D. Benford, 1995, "MASSIVE: A Distributed Virtual Reality System Incorporating Spatial Trading", Pro-ceedings of 15th International Conference on Distributed Computing Systems, Los Alamitos CA, ACM Press, pp. 27-34.
- [10] J. Huang, Y. Du, C. Wang, 2003, "Design of the Server cluster to support avatar migration", Proceedings of the IEEE Virtual Reality.