

A Study on Moving Object Direction Control Technology in AR Video

Sungmo Jung, Yohwan So, Eunseok Lee and Seoksoo Kim*

Abstract— According to Gartner Report, Augmented Reality is a desirable next generation content technology that allows of inserting 3D virtual objects into the real-world environment seen through the camera. It maximizes interest and flow experience with its interactive feature. In other words, it is a technology that can reduce the difference between the virtual world and the real world. Thus, one of important objectives of Augmented Reality is to enable users to have enhanced flow experience toward the augmented contents. However, generally virtual objects do not have physical properties, which results in reduced flow experience in simulation on virtual objects. Therefore, his research suggests a technology that gives users control over directions of moving objects in AR video.

Keywords—Object Direction, Matrix Transformation, Rendering Pipeline, Augmented Reality, Software Engineering.

I. INTRODUCTION

As shown in Figure 1, Augmented Reality [1] is seen as a desirable next generation content technology according to 2010 Gartner Report [2].

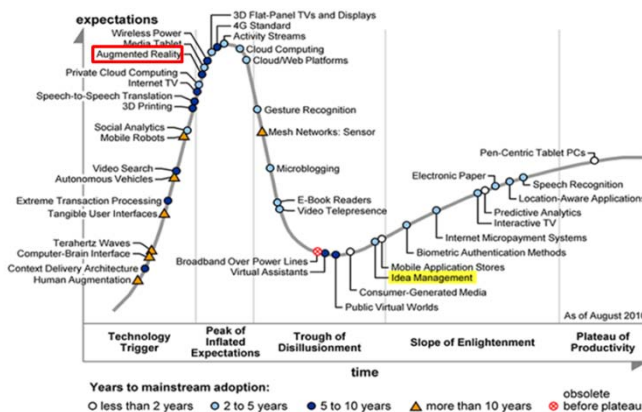


Fig. 1. emerging technologies hype cycle in 2010 Gartner report

Augmented Reality (AR) is a technology by which a 3D virtual object is inserted into the real-world environment seen through the camera. With the interactive feature of Augmented Reality technology, interest and flow experience of users can be maximized. In short, it enhances flow experience by creating an environment where the gap between the virtual world and the

This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2010

*Corresponding author : Seoksoo Kim (sskim0123@naver.com)

real world can be reduced.

In general, however, because Augmented Reality augments virtual objects that do not have physical elements including gravity, mass, speed, friction and force, it hinders such flow experience. Thus, in this research, a geometric model of augmented contents was used for kinetics-based simulation in order to resolve problems with unnatural motion and to suggest a method for providing users with visually natural motion.

For this research, a simulation system that allows the user to choose directions of a moving object was established.

II. RELATED RESEARCHES

A. Vector Operations

In this research, Vector Operations were performed to control the size and the direction of an object and unit vector was used in the rotation axis of the object. Basic 3D Vector Operations are as following:

i. Addition

$$p + q = (x_p, y_p, z_p) + (x_q, y_q, z_q) = (x_p + x_q, y_p + y_q, z_p + z_q) \quad (1)$$

ii. Scalar multiplication

$$c \otimes p = (cx_p, cy_p, cz_p) \quad (2)$$

iii. Dot product

$$p \cdot q = x_p x_q + y_p y_q + z_p z_q = |p||q| \cos \theta \quad (3)$$

iv. Cross product

$$p \times q = (p_y q_z - p_z q_y, p_z q_x - p_x q_z, p_x q_y - p_y q_x) \quad (4)$$

Subtraction and division are not defined as they are expressed as a combination, and for this research, dot products and cross products were used among the above basic arithmetic operations.

As shown in Figure 2, using cross product properties [3], vector values that cross at right angles with two vectors can be yielded. Cross product properties can be used only as functional call.

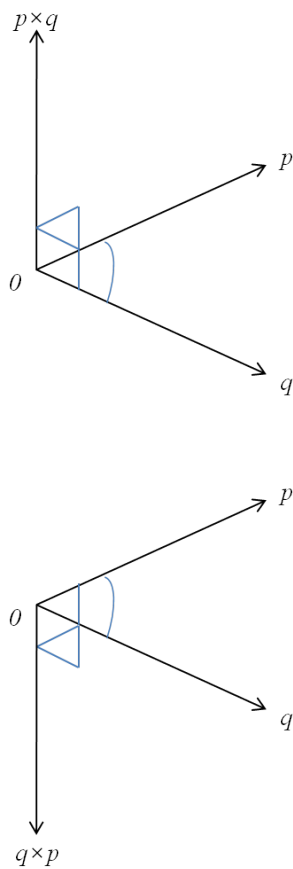


Fig. 2. cross product

In the 3D space, a face is created by 3 points or 3 vectors. With the cross product of two vectors, face-normal vector of the face can be calculated. As shown in the figure 2, vector $\vec{op} \vec{oq}$ creates a face, the face-normal vector of this face opq is $p \times q$. With values resulted from the dot product of this face-normal vector and the camera-vector, screen display shall be determined. In the cross product, it is important to note that the direction of vectors generated according to the order of operations is opposite[5].

The dot products are generally important values to determine if the angle between two vectors is over 90 degrees. As per the cos graph, cos has positive values only when it falls between 0 and 90 degrees and 270 and 360 degrees. Using this nature, the dot product of the eye-vector that is equivalent to the human eye and the face-normal vector that crosses at right angles with a face of an object can be calculated. The calculated values shall be used for backface-culling in which only positive values are displayed on the screen[6].

In particular, the dot product is used throughout the 3D graphic. It is most commonly used for coordinate transformation and illumination calculation. With only the dot product of the face-normal vector of a light source and that of a mesh surface, a plausible lighting can be created. Figure 3 briefly demonstrates the Lambertian Lighting Model[7].

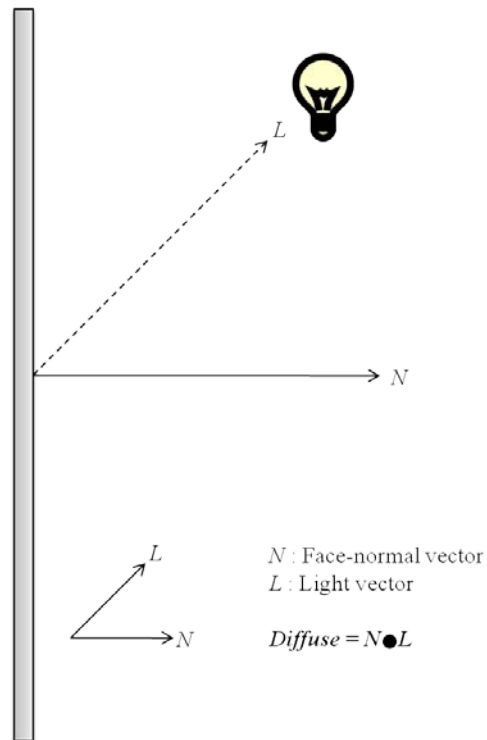


Fig. 3. dot product of lambertian lighting model

B. Quaternion

Quaternion [8] is the four-dimensional complex space vector and can be exhibited as (5) or (6).

$$q = \langle x, y, z, w \rangle = xi + yj + zk + w \tag{5}$$

$$q = s + v \quad (s = w, v = \langle x, y, z \rangle) \tag{6}$$

Quaternion has a close relationship with the rotation of a vector. The rotation at θ against the axis A can be exhibited as $q = \cos \frac{\theta}{2} + A \sin \frac{\theta}{2}$ in quaternion. To rotate the point P, qPq^{-1} shall be carried out.

In this research, the finally obtained quaternion was converted into the matrix shown below in order to use for rotation in the actual program.

$$R_q = \begin{bmatrix} 1 - 2y^2 & 2xy - 2wz & 2xz + 2wy & 0 \\ 2xy + 2wz & 1 - 2x^2 - 2z^2 & 2yz - 2wx & 0 \\ 2xz - 2wy & 2yz + 2wx & 1 - 2x^2 - 2y^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Furthermore, in this research, in order to enhance realizability, linear interpolation was used to obtain values from a linear equation considering two values of moving objects as points[10]. The linear equation $q(t) = (1 - t)q_1 + tq_2$ in Figure 4 is a straight line that connects between point q_1 and q_2 when t value changes from 0 to 1.

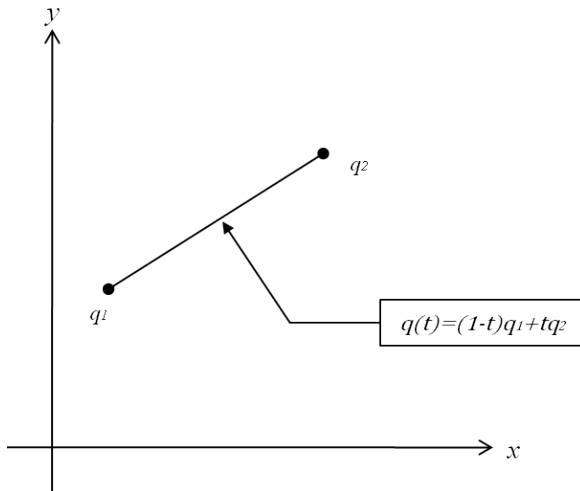


Fig. 4. 2D plane geometric graph of linear interpolation

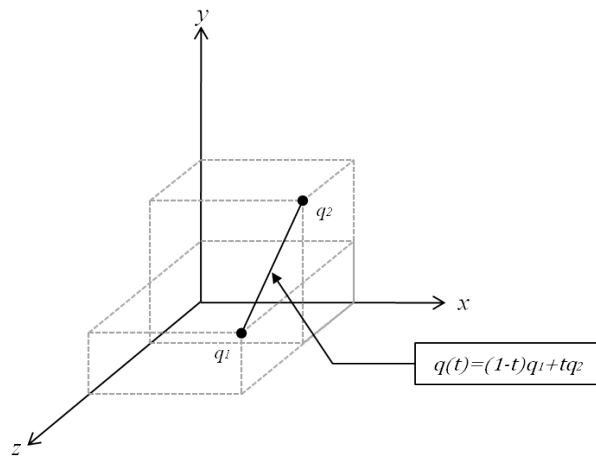


Fig. 5. 3D space geometric graph of linear interpolation

Quaternion space is a spherical space; therefore, using linear interpolation causes errors as following[11].

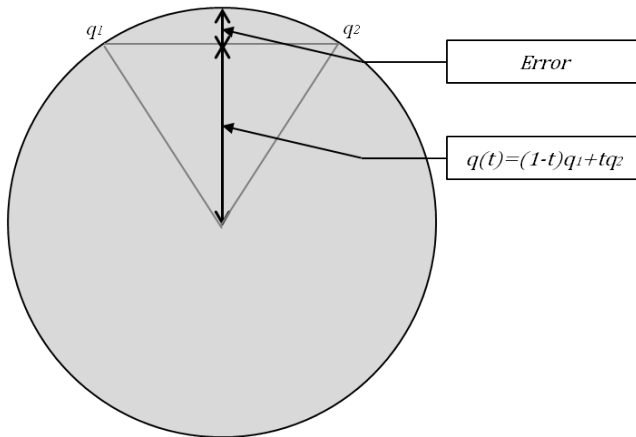


Fig. 6. error of linear interpolation

Such errors shall be corrected by normalization and a formula is drawn as following.

$$q(t) = \frac{(1-t)q_1 + tq_2}{\|(1-t)q_1 + tq_2\|} \tag{8}$$

However, due to the nature of linear interpolation, errors of speed between a line and a sphere occur. In this research, spherical linear interpolation was used to resolve such problem [12], and the result is as following.

$$q(t) = \frac{\sin \theta(1-t)}{\sin \theta} q_1 + \frac{\sin \theta t}{\sin \theta} q_2$$

(Then, $\theta = \cos^{-1}(q_1 \cdot q_2)$)

(9)

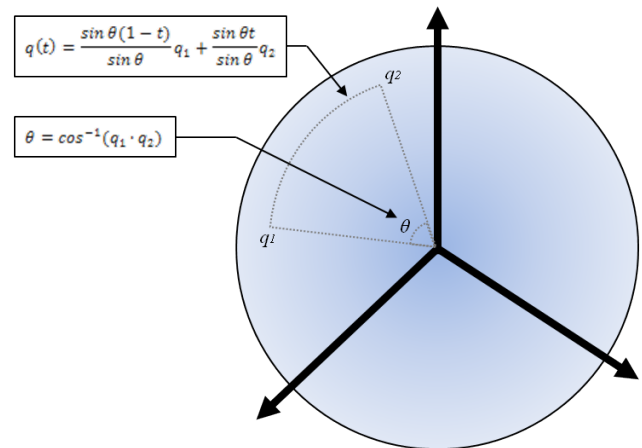


Fig. 7. quaternion space geometric figure of slerp

C. World transform matrix

World Transform Matrix [13] is a matrix that transforms a local coordinate system (or model coordinate system) into World Coordinates System which is a three-dimensional absolute coordinate system. This includes the transform matrix of movement, rotation and size. And even though there are crush transformers such as Twist or Shear, fundamentally a transform matrix extracted from 3D modeling tools and an animation matrix are combined into World transform matrix. The local coordinate system in Figure 8 is called a model coordinate system because it is a coordinate system used in modeling mesh [14].

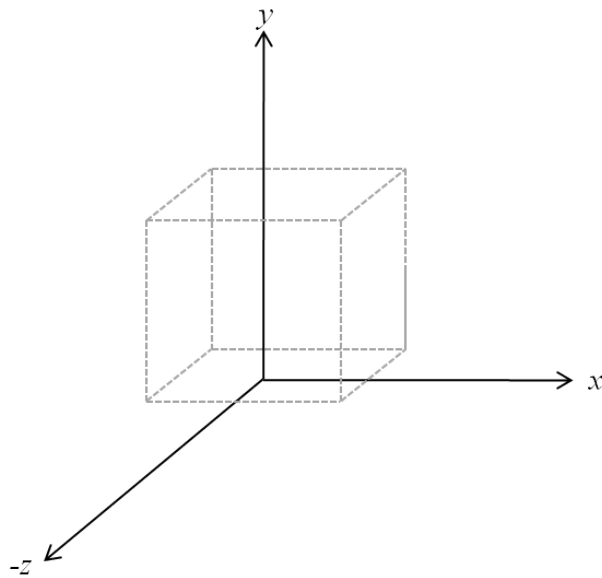


Fig. 8. local coordinate system

Applying the above local coordinate system to World transform matrix makes a coordinate system as shown in Figure 9.

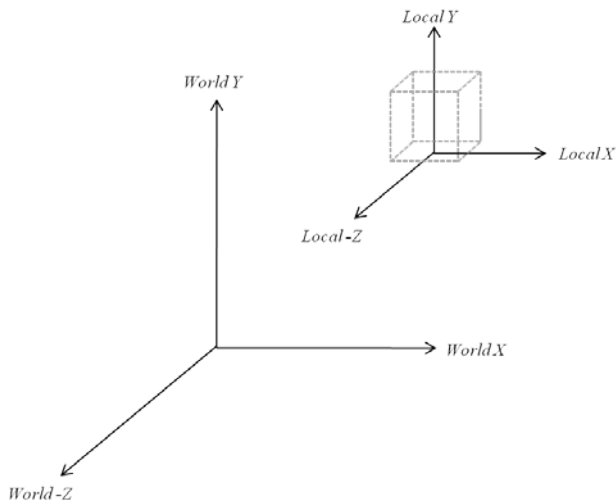


Fig. 9. world coordinate system

Because all mesh has its own unique local coordinate system, TM (Transform Matrix) is required when transforming it into World transform matrix. TM values shall be extracted with 3D modeling tools.

D. OpenGL

To understand movement of an object, it is necessary to understand the object itself first. An object has unique position, speed and acceleration. Speed and acceleration as well as position of the object are exhibited with vectors. A vector has its size and direction, and the size of the speed vector is the speed value.

OpenGL (Open Graphics Library) [15] is the graphics standard API specification for 2D and 3D graphics developed by Silicon Graphics Inc. in 1992 and supports a cross-language

and cross-platform programming. With over 250 different function calls, this API can be used to draw from simple primitives to complex 3D scenes. [16]

Functions that are most commonly used in this system are `glclear(GL_COLOR_BUFFER_BIT)` which empties the screen by clearing the color buffer, `glMatrixMode(GL_PROJECTION)` which uses 3D frustum matrix to express the visible area after initializing the projection matrix and `glLoadIdentity()` and `glFrustum()` that are used to transform the object exhibited in a comparative space to the camera into the projection space. Also, by using `glMatrixMode(GL_MODELVIEW)`, `glLoadIdentity()`, and `glTranslatef()`, this system initializes the model view matrix into the unit matrix and defines transform from the coordinate system relative to model to the camera space.

III. MOVING OBJECT DIRECTION CONTROL

A. Rendering Pipeline Configuration

Conceptual diagram of the Rendering Pipeline for the final outcome of this research is as following [17].

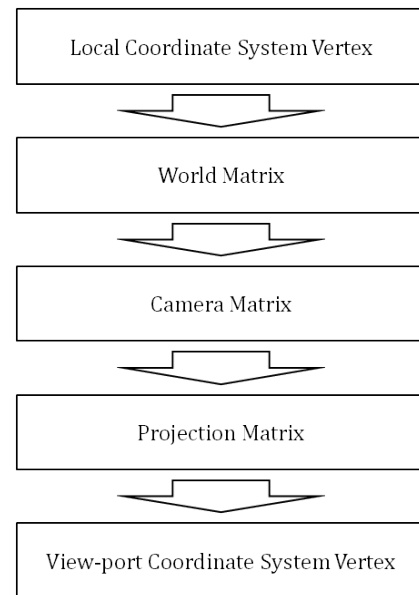


Fig. 10. rendering pipeline method

The above conceptual diagram demonstrates the Rendering Pipeline in the course of change in vertices through matrix. Figure 11 demonstrates the final display process in which the final pixel is rendered on the screen through the vertex shader step and the pixel shader step in the actual DirectX.

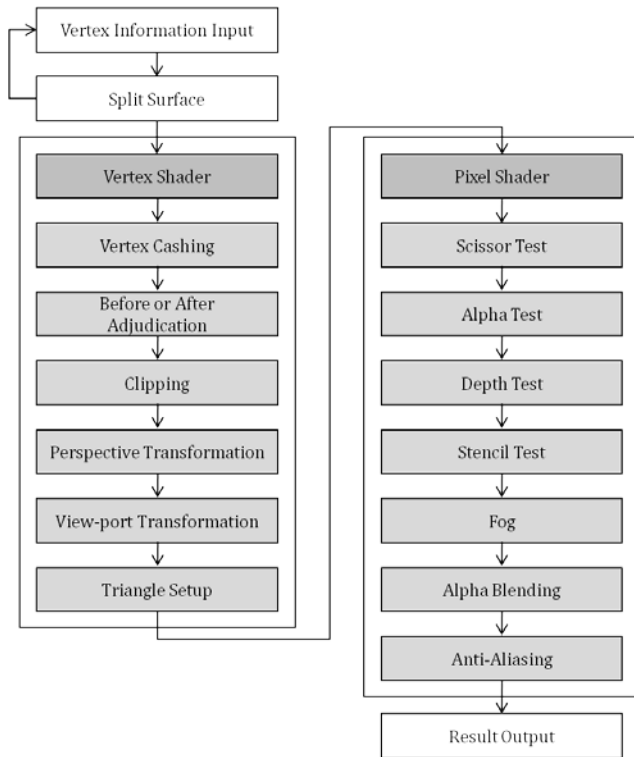


Fig. 11. result output rendering pipeline

The above figure demonstrates the process during which three-dimensional vertices is transformed into two-dimensional points after going through World, camera and projection matrix. At the point of time when projection is completed, viewpoint transform process is inserted, which is detailed in 3-B of this research.

B. View Coordinate Configuration

While World transform matrix is to transform the starting point of the local coordinate system into that of World, the function of the camera transform matrix is to create a new coordinate system based on the direction that the camera is toward with taking the viewpoint of the camera for the starting point. World transform matrix is transformed into this created coordinate system.

Camera transform matrix consists of movement matrix and coordinate system transform matrix. Movement matrix is a reverse direction matrix of Camera viewpoint coordinate system; therefore, it only requires switching signs.

If camera viewpoint coordinate is $c = (0, 0, 1)$, the object located at $(1, 1, 10)$ in the coordinate, direction vector of the camera, $n = (1, 1, 11)$, camera upward vector, $(0, 1, 0)$ and right side vector $v = (0, 1, 0) \times n$ can be displayed as following.

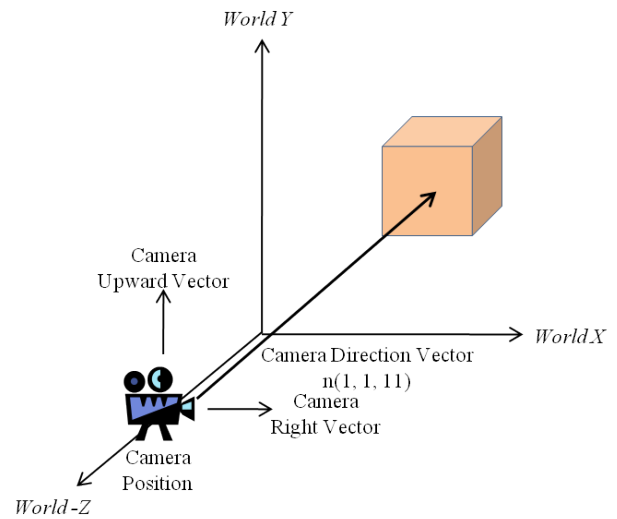


Fig. 12. before camera matrix applying

Conceptual diagram of View Coordinate Configuration is as following:

View coordinate system = World coordinate system \times Moving matrix \times Coordinate system transform matrix

$$[x', y', z', 1] = [z, y, z, 1] \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -c_x & -c_y & -c_z & 1 \end{bmatrix} \times \begin{bmatrix} v_x & u_x & n_x & 0 \\ v_y & u_y & n_y & 0 \\ v_z & u_z & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

If p =coordinate of the target, c =coordinate of the camera and up is upward vector, the axis of the new camera, v , u , and n in the formula mentioned above is as following:

$$n = \text{normalize}(p - c) \quad (11)$$

$$v = \text{normalize}(up \times n) \quad (12)$$

$$u = n \times v \quad (13)$$

The following is the final Camera transform matrix values with application of associative laws.

$$[x', y', z', 1] = [x, y, z, 1] \times \begin{bmatrix} v_x & u_x & n_x & 0 \\ v_y & u_y & n_y & 0 \\ v_z & u_z & n_z & 0 \\ -(c \cdot v) & -(c \cdot u) & -(c \cdot n) & 1 \end{bmatrix} \quad (14)$$

C. Coordinate System Transformation

In order to apply in this system, the coordinate system needs to be transformed into Perspective Projection Matrix. The purpose of perspective projection is to transform viewing frustum surrounded by radical plane and round plane into the

coordinate system of $(-1, -1, 0) \sim (1, 1, 1)$.

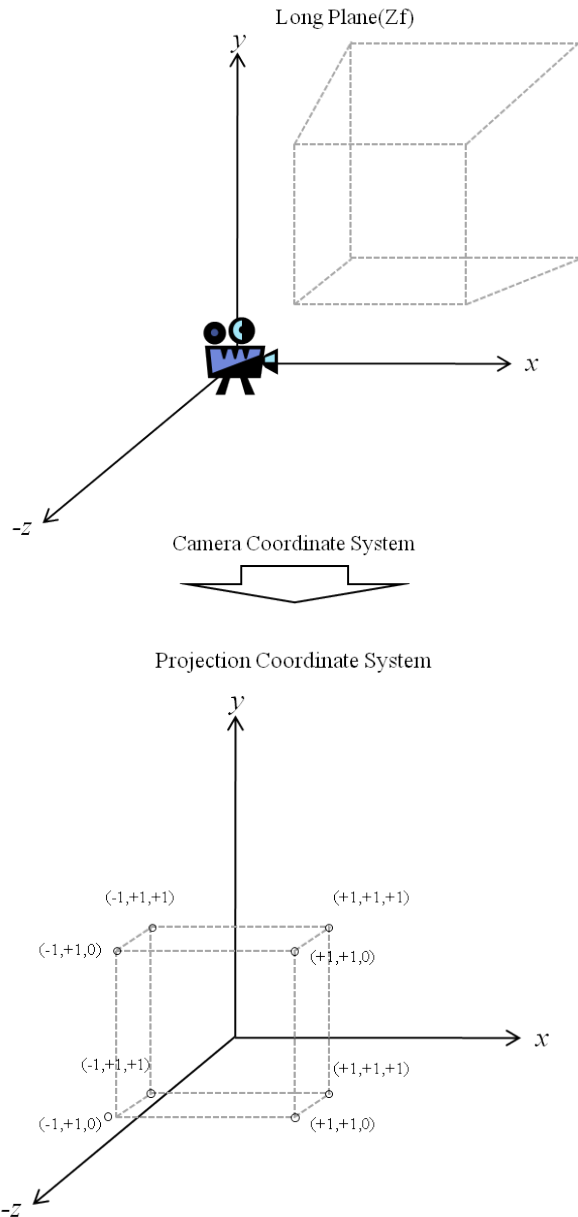


Fig. 13. transform camera coordinate system to projection coordinate system

In order to transform Camera coordinate system into Projection coordinate system, it is necessary to induce a random point of Camera coordinate system, $P = (p_x, p_y, p_z)$ to be transformed into $P' = (p'_x, p'_y, p'_z)$ of Project coordinate system. For this transformation, it is required to use a linear equation on the XZ plane as shown in the following figure.

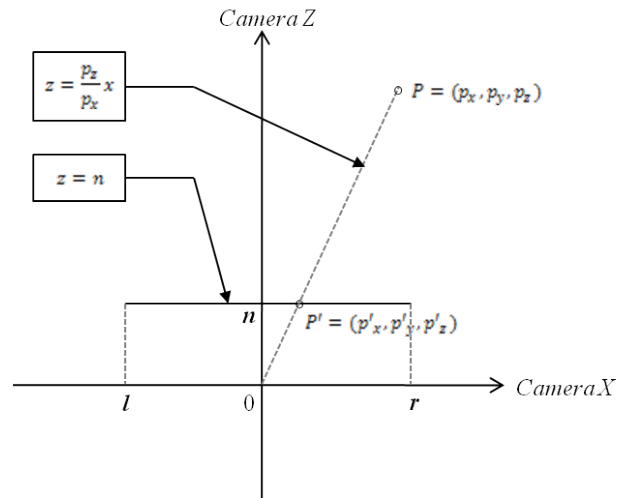


Fig. 14. the linear equation in the plane

Equation of points that are required for projection of the point within viewing frustum of Camera coordinate system, $P = (p_x, p_y, p_z)$ on the radical plane of Camera coordinate system can be obtained as following:

OP linear equation is

$$z = \frac{p_z}{p_x} x$$

and the linear equation on the radical XZ plane is

$$z = n$$

Therefore, the value of

$$x = \frac{np_x}{p_z}$$

can be calculated with these two equations.

Likewise, the value of

$$y = \frac{np_y}{p_z}$$

can be obtained as following.

Because the point of viewing frustum [18] satisfies $left \leq x \leq right, bottom \leq y \leq top$, a simple function can be created to transform this value into a value between $[-1, 1]$ as following.

$$x' = (x - l) \frac{2}{r - l} - 1 \text{ (Then, } l = \text{left, } r = \text{right)} \quad (15)$$

$$y' = (y - b) \frac{2}{t - b} - 1 \text{ (Then, } b = \text{bottom, } t = \text{top)} \quad (16)$$

By combining the above formulas, results can be calculated as following:

$$x' = \frac{2n}{r - l} \left(\frac{p_x}{p_z} \right) - \left(\frac{r + l}{r - l} \right) \quad (17)$$

$$y' = \frac{2n}{t-b} \left(\frac{p_y}{p_z} \right) - \left(\frac{t+b}{t-b} \right) \tag{18}$$

z is *near* $\leq z \leq$ *far*, but this value needs to be transformed into a value between [0, 1]. As the value of z is in denominator in the above formula, it is necessary to find term A and B that satisfy the following formula.

$$z' = \frac{A}{z} + B \tag{19}$$

Because *near* is 0, it shall be substituted with $z = n, z' = 0$ while *far* shall be substituted with $z = f, z' = 1$ as it is 1. It flows as following.

$$0 = \frac{A}{n} + B, 1 = \frac{A}{f} + B \text{ (Then, } n = \textit{near}, f = \textit{far}) \tag{20}$$

The above formula can be sorted as for A and B as following.

$$A = \frac{fn}{n-f} \tag{21}$$

$$B = \frac{f}{f-n} \tag{22}$$

z' shall be finally derived as following.

$$z' = \left(\frac{fn}{n-f} \right) \left(\frac{1}{p_z} \right) + \frac{f}{f-n} \tag{23}$$

The derived point $p' = (x', y', z')$ shall be transformed into the point of Homogenous coordinate system, $p'_4 = (x'p_z, y'p_z, z'p_z, p_z)$ as following:

$$x'p_z = \left(\frac{2n}{r-l} \right) p_x - \left(\frac{r+l}{r-l} \right) p_z \tag{24}$$

$$y'p_z = \left(\frac{2n}{t-b} \right) p_y - \left(\frac{t+b}{t-b} \right) p_y \tag{25}$$

$$z'p_z = \frac{-fn}{f-n} - \left(\frac{f}{f-n} \right) p_z \tag{26}$$

To finally apply in the system, the above formulas can be demonstrated as matrix as following.

$$\begin{bmatrix} \frac{2n}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2n}{t-b} & 0 & 0 \\ \frac{r+l}{r-l} & \frac{t+b}{t-b} & \frac{f}{f-n} & 1 \\ 0 & 0 & \frac{-fn}{f-n} & 0 \end{bmatrix} \tag{27}$$

Supposing $r-l = w, t-b = h, l = -r, b = -t$, the final perspective projection matrix is as following:

$$\begin{bmatrix} \frac{2n}{w} & 0 & 0 & 0 \\ 0 & \frac{2n}{h} & 0 & 0 \\ 0 & 0 & \frac{f}{f-n} & 1 \\ 0 & 0 & \frac{-fn}{f-n} & 0 \end{bmatrix} \tag{28}$$

D. Clipping and view-port matrix

Clipping matrix and view-port matrix are applied in the perspective transformation to transform clipping and view-port. View-port matrix carries out mapping $[-1, -1] \sim [1, 1]$ value to length and width of an actual window.

In this thesis, after 640x480 size display window was configured and values were substituted to matrix, the following formula was actually used as following.

$$M_{clip} = \begin{bmatrix} \frac{2}{c_w} & 0 & 0 & 0 \\ 0 & \frac{2}{c_h} & 0 & 0 \\ 0 & 0 & \frac{1}{z_{max} - z_{min}} & 0 \\ -1 - 2\frac{c_x}{c_w} & -1 - 2\frac{c_y}{c_w} & \frac{-z_{min}}{z_{max} - z_{min}} & 1 \end{bmatrix} \tag{29}$$

$$M_{vs} = \begin{bmatrix} \frac{dwWidth}{2} & 0 & 0 & 0 \\ 0 & -\frac{dwHeight}{2} & 0 & 0 \\ 0 & 0 & \frac{dwMaxZ - dwMinZ}{2} & 0 \\ dwX + \frac{dwWidth}{2} & dwY + \frac{dwHeight}{2} & dwMinz & 1 \end{bmatrix} \tag{30}$$

E. Library Configuration

OpenGL provides a primitive geometric structure with relatively simple shapes including box, sphere, plane, ray and cylinder. But there is limit of shapes that can be created only by combining such geometric structures. Therefore, complicated shapes of geometric structure were expressed through the triangular mesh method in this thesis. These created virtual objects move and react in the augmented environment with properties of rigid bodies in the same way as actual. However, it requires caution to ensure the value of the initial position axis z of the rigid body is not negative such as $r = (x, y, -z)$, because if it is negative, a collision with the surface occurs, which causes irregular simulation due to malfunction of the simulator.

IV. SIMULATION

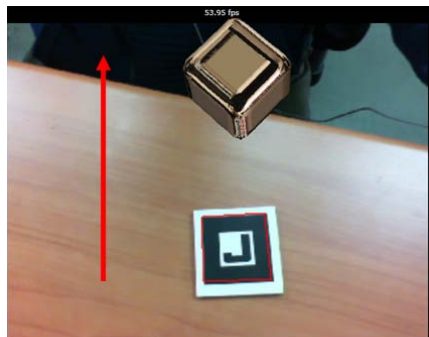
In this thesis, library was developed and the entire system was configured for physical simulation of Augmented Reality environment. To support this, different coordinate systems were accorded, transform matrix of marker was found, coordinate systems were re-configured, and designated coordinate axis was used to apply virtual objects.

In this system, a process in which transform matrix is divided into position vector, rotation matrix and scale vector was specially configured. Moreover, a routine was established in which natural motion effects were realized by synchronizing separated values with Rendering Pipeline in real-time and such effects could interact with the system through visual feedback. By controlling over directions and acceleration, it was configured to allow users to have flow experience in the Augmented Reality system.

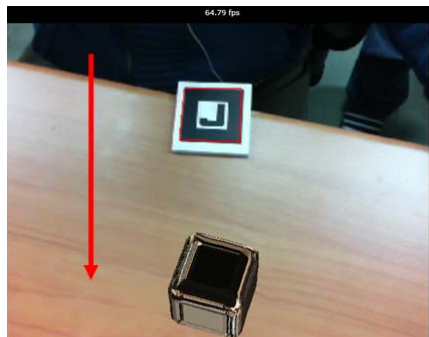
Firstly, it is the augmented object direction control screen. Augmented objects can change directions (toward top, bottom, left and right) in a stable speed as the user appoint.



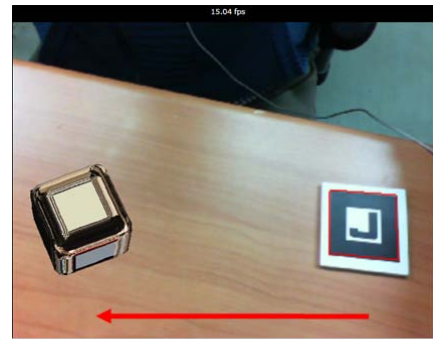
Augmented Object Screen (Hold)



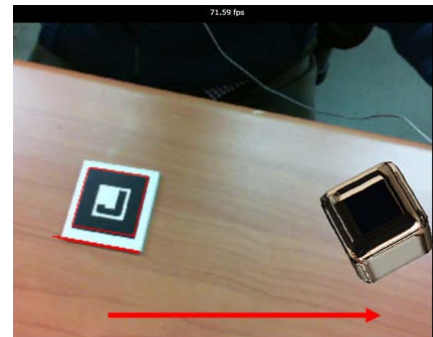
Move Top



Move Down



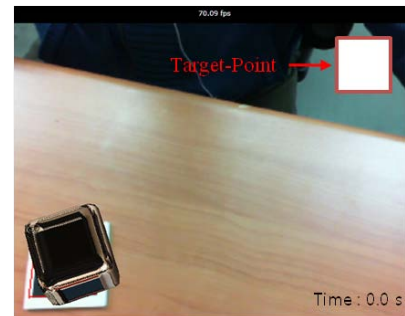
Move Left



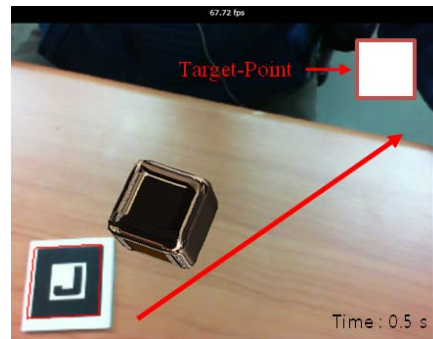
Move Right

Fig. 15. direction control

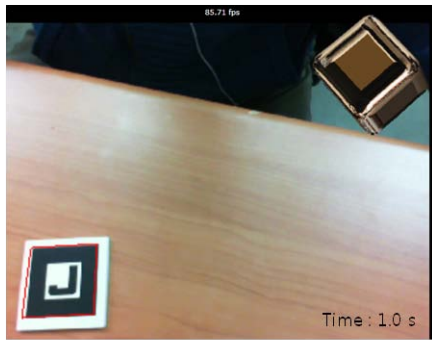
Lastly, it is the object acceleration control screen. It was captured twice while moving specific distance for one second. It can be verified that acceleration was applied by comparing distance moved for the first half second with that for one second.



Augmented Object Screen (Hold and Set the Target-point)



Move on the Target-point



Move Completed on the Target-point

Fig. 16. acceleration applied

V. CONCLUSION

According to Gartner Report, Augmented Reality is a desirable next generation content technology that allows of inserting 3D virtual objects into the real-world environment seen through the camera. It maximizes interest and flow experience with its interactive feature. In other words, it is a technology that can reduce the difference between the virtual world and the real world. Thus, one of important objectives of Augmented Reality is to enable users to have enhanced flow experience toward the augmented contents. However, generally virtual objects do not have physical properties, which results in reduced flow experience in simulation on virtual objects. Therefore, his research suggests a technology that gives users control over directions of moving objects in AR video.

Yet, because this thesis does not include research on refraction, shadows and physical collisions which can make objects look more real, there is limit to have perfect flow experience.

In the future, it is required to undertake researches on reflection, refraction, shadows and physical collisions in order to realize the Augmented Reality videos so that users can have enhanced flow experience.

REFERENCES

- [1] Azuma, R. and Bailiot, Y. and Behringer, R. and Feiner, S. and Julier, S. and MacIntyre, B., "Recent advances in augmented reality", *IEEE Computer Graphics and Applications*, Vol.21, No.6, 2001, pp.34-47
- [2] http://www.gartner.com/it/content/1395600/1395613/august_4_whats_hot_hype_2010_jfenn.pdf
- [3] Kauffman, L.H., "Map coloring and the vector cross product", *Journal of Combinatorial Theory, Series B*, Vol.48, No.2, 1990, pp.145-154
- [4] Lorensen, W.E. and Cline, H.E., "Marching cubes: A high resolution 3D surface construction algorithm", *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp.163-169
- [5] Xu, G. and Liu, GR, "Development of Irregular-Grid Finite Difference Method (IFDM) for Governing Equations in Strong Form", *WSEAS TRANSACTIONS ON MATHEMATICS*, Vol.5, No.10, 2006, pp.1117
- [6] El-Bakry, H.M. and Mastorakis, N., "A new approach for fast face detection", *WSEAS Transactions on Information*

- Science and Applications, Vol.3, No.9, 2006, pp.1725-1730
- [7] Ramamoorthi, R., "Analytic pca construction for theoretical analysis of lighting variability in images of a lambertian object", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.24, No.10, 2002, pp.1322-1333
- [8] Pletinckx, D., "Quaternion calculus as a basic tool in computer graphics", *The Visual Computer*, Vol.5, No.1, 1989, pp.2-13
- [9] Rossignac, J.R. and Borrell, P., "Multi-resolution 3D approximations for rendering complex scenes", IBM Research Division, TJ Watson Research Center, 1992
- [10] Cojocar, D. and Tanasie, R.T., "Graphic simulation for camera calibration in visual-servoing applications", *Proceedings of the 8th conference on Signal processing, computational geometry and artificial vision*, 2008, pp.150-155
- [11] Cao, W., "On the error of linear interpolation and the orientation, aspect ratio, and internal angles of a triangle", *SIAM journal on numerical analysis*, Vol.43, 2005, pp.19-40
- [12] Gautam, R.S. and Patkar, S.B., "Fitting convex surface to scattered data points with applications to medical imaging", *Wseas Transactions on Computers*, Vol.1, 2003, pp.50-56
- [13] Szeliski, R. and Shum, H.Y., "Creating full view panoramic image mosaics and environment maps", *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp.251-258
- [14] McMillan, L. and Bishop, G., "Plenoptic modeling: An image-based rendering system", *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp.39-46
- [15] <http://www.opengl.org/>
- [16] Kwon, K.Y. and Joh, J., "Development of a 3 D simulator for autonomous navigation of underwater robots", *WSEAS Transactions on Systems*, Vol.5, No.5, 2006, pp.1091-1099
- [17] Levoy, M., "Display of surfaces from volume data", *Computer Graphics and Applications*, IEEE, Vol.8, No.3, 2002, pp.29-37
- [18] Assarsson, U. and Moller, T., "Optimized view frustum culling algorithms for bounding boxes", *Journal of Graphics Tools*, Vol.5, No.1, 2000, pp.9-22

Sungmo Jung received a B.S. degree in Multimedia from Hannam University, Korea, 2008 and M.S. degree in Multimedia from Hannam University, Korea, 2010. And currently, on the Ph.D. course in Multimedia Engineering from Hannam University. His research interests include multimedia network system, ubiquitous system, medical information system, network security.

Yohwan So received a B.S. degree in Fine Art from Hongik University, Korea, and M.S. degree in Fine Art from Hongik University, Korea, and M.S degree in Engineering from New York Institute of Technology. And currently, he joined the faculty of Hannam University, Korea. His research interests include multimedia contents, 3D modeling, VFX.

Eunseok Lee received a B.S. degree in Fine Art from Daegu University, Korea, and M.S. degree in Fine Art from Sejong University, Korea, and ABD Doctor of Philosophy in Design form Sejong University, Korea. And currently, he

joined the faculty of Hannam University, Korea. His research interests include multimedia contents, Visual communication design.

Seoksoo Kim received a B.S. degree in Computer Engineering from Kyungnam University, Korea, 1989 and M.S. degree in Information Engineering from Sungkyun-kwan University, Korea, 1991 and Ph.D. degree in Information Engineering from Sungkyun-kwan University, Korea, 2002. In 2003, he joined the faculty of Hannam University, Korea, where he is currently a professor in the Department of Multimedia Engineering. His research interests include multimedia communication systems, distance learning, multimedia authoring, telemedicine, multimedia programming, computer networking, and information security. He is a member of KCA, KICS, KIMICS, KIPS, KMS, and DCS. He is editor-in-chief of IJMUE.