

An Advanced Celestial Map Rendering Platform with 3D Graphics Support

Young Chun Kwon¹ and Nakhoon Baek^{2,*}

¹*Samsung Electronics, Suwon, Korea*

²*School of Computer Science and Engineering,
Kyungpook National University, Daegu 702-701, Korea
kown10@gmail.com, oceancru@gmail.com*

Abstract

For amateur astronomers, it is not easy to find a specific celestial object in the night sky. The celestial calculations are also complex and hard to be correctly achieved. However, there are very few numbers of supporting tools for them. We represent a 3D celestial map rendering platform which reproduces the virtual night sky at the specific time and location. With the OpenGL-based full 3D-graphics features, our platform brings a sense of spatial realism, with user-friendly interfaces. For the more accurate celestial map rendering, we referred to a set of more than 10,000 stars from the famous celestial lists such as the Messier list, the Yale BSC list, and others. For each celestial object, our system provides three-dimensional views and additional two-dimensional simulations for more details. Since it is developed on typical PCs, our system can provide accurate celestial maps in an interactive manner, without any special equipment. Our system is helpful for amateur astronomers and ordinary persons to find the specific celestial object in the real night sky.

Keywords: *celestial map; night sky; 3D simulator; 2D simulation*

1. Introduction

There are various kinds of *celestial objects* including stars, planets, satellites, and constellations in the night sky. Each celestial object has its own location and appearance, according to its original moving path, the rotation and the revolution of the earth, and the observation point and time. Solar planets and moons change their locations and appearances relatively frequently, while far-away fixed stars usually keep their locations and appearances.

Typical astronomical catalogues show the positions of stars, or more precisely, their *right ascensions* and *declinations* [1]. However, it is not easy for ordinary persons to get the real locations of specific stars from this astronomical information. Complex and heavy computations are required to get the exact star locations even with high-priced special devices, in the ever-changing night sky.

In this paper, we represent a 3D graphics application program to easily tracking the location of celestial objects, as shown in Figure 1. Though specifying a few intuitive choices, users can get the exact virtual night sky on PC screens. Our system was developed on the ordinary PCs with the OpenGL 3D graphics library [2-3]. It finally shows realistic celestial maps much similar to the real night skies.

Among various computer programs for supporting astronomical observation, *Google Sky* may be one of the most popular ones [4]. As shown in Figure 2, *Google Sky* uses the real astronomical photographs from the Hubble space telescope, and makes them

* Corresponding author: Nakhoon Baek, oceancru@gmail.com

directly to be their celestial maps. Thus, the users of Google Sky can find various celestial objects explicitly on the real photographs.



Figure 1. Our Celestial Map Rendering System with 3D Graphics Support

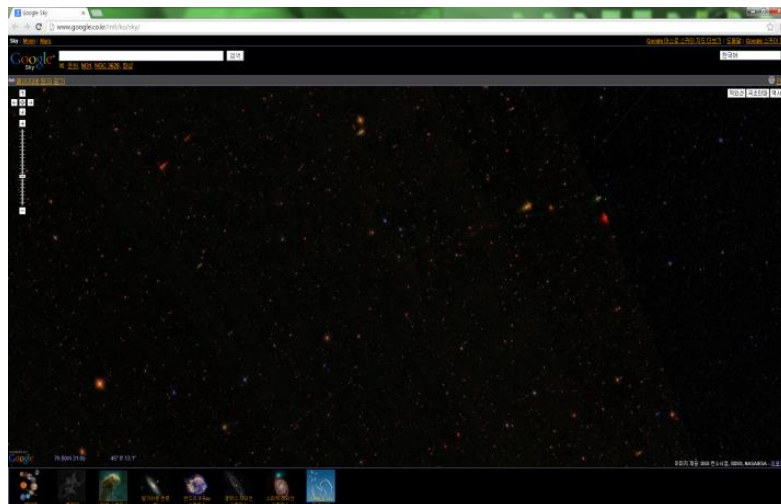


Figure 2. A Snapshot from Google Sky[

With their intuitive and clear user interfaces, Google Sky receives honorable mentions that they reproduce the night sky on the PC screens. In contrast, the system fundamentally supports two-dimensional data, since they are based on the photographs. Due to distortions on the photographs and other processing errors, their results are not so accurate to be used for regular astronomical observations. Additionally, they always require internet connections, while most vista points for the astronomical observations are usually far away from the internet connections.

Astronomical telescope manufacturers know well that most amateur astronomers have difficulties in using their astronomical telescopes [5]. Thus, most manufacturers provide various observation supporting devices, as shown in Figure 3. These devices are still too expensive to be popular. Most of them only support two-dimensional simple configurations.



Figure 3. An Observation Supporting Device for Vixen Astronomical Telescopes



Figure 4. A Snapshot from the Stellarium Program

Stellarium [6] is one of the most popular celestial map rendering programs for ordinary users, which is also the goal of our work, as shown in Figure 4. This program shows night skies in the PC screens, with modern 3D graphics technology. This program provides a set of required information, for regular celestial operations. However, it lacks the detailed description of diurnal motions for various celestial objects.

Our work provides similar results with the popular celestial map rendering programs, including Stellarium. Additionally, our result shows the detailed diurnal motions in two-dimensional and even three-dimensional graphics output. Although we have a set of celestial map rendering programs, they do not provide their implementation details. We will present the design and implementation details of our work, and it is the first literature on this topic, at least to the best of our knowledge.

2. Design and Analysis

Our system aimed to produce the virtual night sky at the user-specified location and time. Additionally, according to the lapse of time, it simulates the exact motions of the celestial objects. Figure 5 shows the overall architecture of our system.

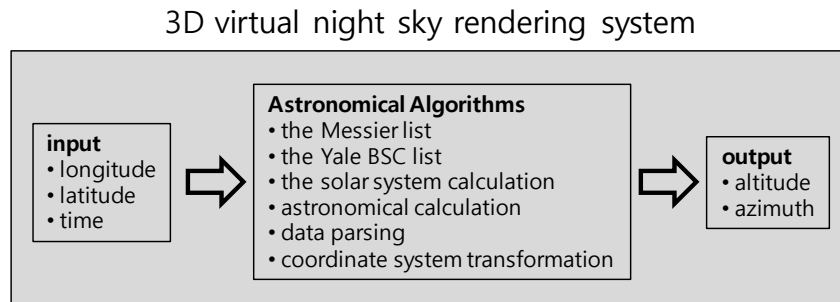


Figure 5. The Overall Architecture of Our System

Our system uses our own algorithms to get the altitude and azimuth of each celestial object, from the location and time settings. After a mass of these astronomical calculations, we render all the stars, planets, and constellations on the screen at the precise locations.

Although each celestial object has its own right ascensions and declinations, we need a set of algorithms to get the relative location on the screen, applying the rotation and the revolution of the earth. Another set of algorithms are required to calculate the diurnal motions of solar planets, since their motions are somewhat different from the outside fixed stars [7-8].

For the precise rendering of the night sky, we used exact information on each astronomical object from the astronomical databases. There are more scholarly databases, such as *the list of Messier objects* (or the Messier list) [9] and *the Yale Bright Star Catalogue* (or the Yale BSC list) [10], from various astronomical societies.

We finally used the Messier list, the Yale BSC list, and the solar system, to render the virtual night sky. Our system covers more than 10,000 astronomical objects, and can be extended to additional databases.

Figure 6 shows the skeleton of our internal algorithm architecture for the Yale BSC and the list of Messier objects display. As shown here, the altitudes and azimuths of celestial objects are calculated from the current location and time settings, based on the Yale BSC and the list of Messier objects.

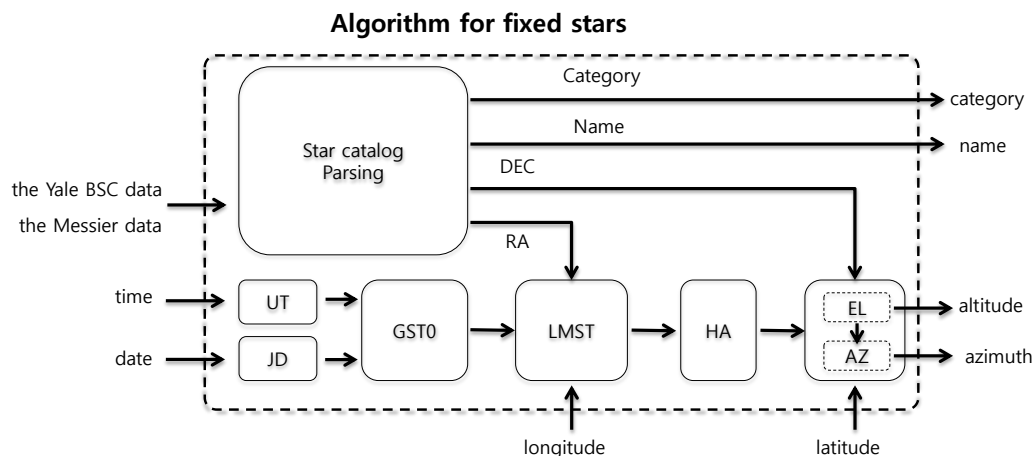


Figure 6. The Overall Algorithm Flow Design for the Yale BSC and the List of Messier Objects

The Yale BSC list and the list of Messier objects are two of the most famous celestial object lists in astronomical observation. In these lists, we can find its name, category, right ascension, declination, and other various celestial values, for each celestial object.

Through building up a large database from these lists, we can easily extract the required celestial values, as shown in Figure 6.

2.1. Calculations for the Messier List and the Yale BSC List

In the overall flow design, our algorithm starts from the right ascension and declination values found in the celestial lists. Providing the current date and time, our system calculates the corresponding *Julian Date* (JD) and the *Universal Time* (UT). Then, it also calculates the *Greenwich Sidereal Time* (GST0). The *right ascension* (RA) value of the celestial object is converted to the *azimuth* and *altitude* at the given position, with its *longitude* and *latitude* values. Through connecting pre-specified celestial objects with line segments, we can display the corresponding constellations. The more details of the celestial object-related calculations will be shown in the followings.

In most astronomical almanacs, they give us the *right ascension* and *declination* values of each astronomical object. Using these values, we need to calculate its *altitude* and *azimuth* at the given location. Given a radius value, we get the three-dimensional (x, y, z) coordinates from the altitude and azimuth values.

In astronomy and celestial navigation, the *hour angle* (HA) is one of the coordinates used in the equatorial coordinate system to give the direction of a point on the celestial sphere. The hour angle of a point is the angle between two planes: one containing the Earth's axis and the zenith (the meridian plane), and the other containing the Earth's axis and the given point (the hour circle passing through the point).

The *local hour angle* (LHA) of an object in the observer's sky is

$$\theta_{LHA} = \theta_{LST} - \alpha,$$

or

$$\alpha = \theta_{LST} - \theta_{LHA},$$

Where θ_{LHA} is the local hour angle of the object, θ_{LST} is the angle for the local sidereal time, and α is the object's right ascension (RA). These angles can be measured in time (24 hours to a circle) or in degrees (360 degrees to a circle).

Based on the local hour angle θ_{LHA} and the right ascension α , the altitude h and the azimuth A can be calculated as follows:

$$\sinh = \sin \phi \sin \delta + \cos \phi \cos \delta \cos HA,$$

and

$$\tan A = \frac{-\cos \delta \sin HA}{\sin \delta \cos \phi - \cos \delta \sin \phi \cos HA},$$

Where ϕ is the *geodetic latitude*, and δ is the *declination*, at the specified location.

For most programming languages, the above equation can be rewritten with atan2 (or arc tangent) function, as follows:

$$A = \tan^{-1} \frac{-\cos \delta \sin HA}{\sin \delta \cos \phi - \cos \delta \sin \phi \cos HA} \\ = \text{atan2}(-\cos \delta \sin HA, \sin \delta \cos \phi - \cos \delta \sin \phi \cos HA)$$

Most celestial objects in the Messier list and the Yale BSC list are sufficiently far away, except solar objects. Thus, those objects show very small changes in their right ascension and declination values. Thus, assuming that the distances from the earth to those objects are sufficiently large, their (x, y, z) coordinates on the celestial sphere can be calculated from their azimuth and the altitude calculation results.

2.2 Calculations for Solar Objects

The celestial objects in our solar system show relatively large changes in their right ascension and declination values, in contrast to the faraway stars listed in the Yale BSC list and the Messier list. Additionally, each planet has its own revolution speed and revolution orbit, and thus, each planet requires its own detailed calculations to trace it. In the case of satellites or moons, we need a couple of calculations: the moon revolves the earth, while the earth itself revolves the sun. Thus, the orbit calculations are much complex in comparison to the faraway stars.

Our overall calculations are performed in three stages. At the first stage, the current location of a planet is calculated with respect to its own orbit plane. And, at the second stage, the calculated locations are projected to *the plane of the ecliptic*, to get their *heliocentric longitude* and *heliocentric latitude*. At the final stage, the center of the coordinate system is transformed from the sun to the earth, and we finally get the *celestial longitude* and *celestial latitude*. The celestial longitude and latitude are also known as the *ecliptic longitude and latitude*, respectively.

From the ecliptic longitude and latitude, we get the ecliptic altitude and azimuth. These values are finally transformed to get the final altitude and azimuth at the current location, based on the values of the right ascension, declination, and current time, as shown in Figure 7.

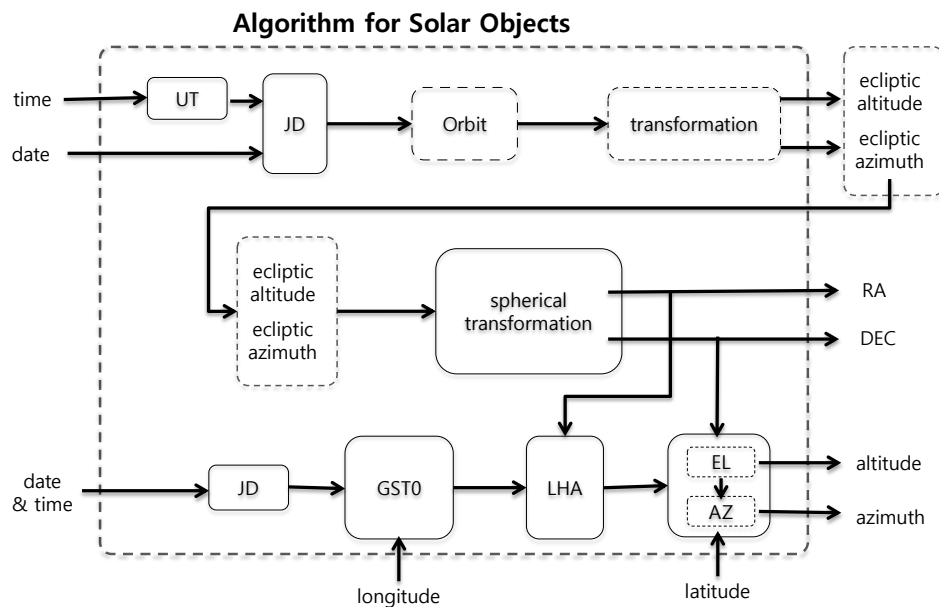


Figure 7. Algorithmic Flow for Solar Objects

In celestial mechanics, *mean anomaly* means a parameter relating position and time for a body moving in an orbit. It is based on the fact that equal areas are swept in equal intervals of time by a line joining the focus and the orbiting body. The mean anomaly increases uniformly from 0 to 2π radians during each orbit. However, it is not an angle. Due to Kepler's second law [11], the mean anomaly is proportional to the area swept by the focus-to-body line since the last *periapsis*.

The *mean anomaly* is usually denoted by the letter M , with respect to the time t , and is given by the formula:

$$M = nt = t \sqrt{\frac{G(M^* + m)}{a^3}},$$

Where n is the mean motion, a is the length of the orbit's semi-major axis, M^* and m are the orbiting masses, and G is the gravitational constant. The mean anomaly is the time since the last periapsis multiplied by the mean motion, and the *mean motion* is 2π divided by the duration of a full orbit.

In another way, the mean anomaly M for a planet can be calculated as:

$$M = \frac{360}{365.242191} \times \frac{D}{T_p} + \varepsilon^{-\omega}$$

Where T_p is the tropical year, ε is the average longitude of the planet, and ω is the longitude at the perihelion. The *true anomaly* v of a planet can be calculated from the following equations:

$$v = M + e \sin M$$

Where e is the orbit eccentricity of the planet. The next step is getting the ecliptic longitude l as follows:

$$l = v + \omega,$$

or more precisely,

$$l = \frac{360}{365.242191} \times \frac{D}{T_p} + \frac{360}{\pi} \theta \sin\left(\frac{360}{365.242191} \times \frac{D}{T_p} + \varepsilon^{-\omega}\right) + \varepsilon$$

Also, we get the radius r from the following equation:

$$r = \frac{a(1 - e^2)}{1 + e \cos v}$$

The above calculations for the ecliptic longitude l and the radius r would be performed for both of the planet and the earth, independently. Letting the ecliptic longitude and the radius of the earth be L and R , respectively, the geocentric longitude λ and the geocentric latitude β are calculated separately for the inferior planets and superior planets.

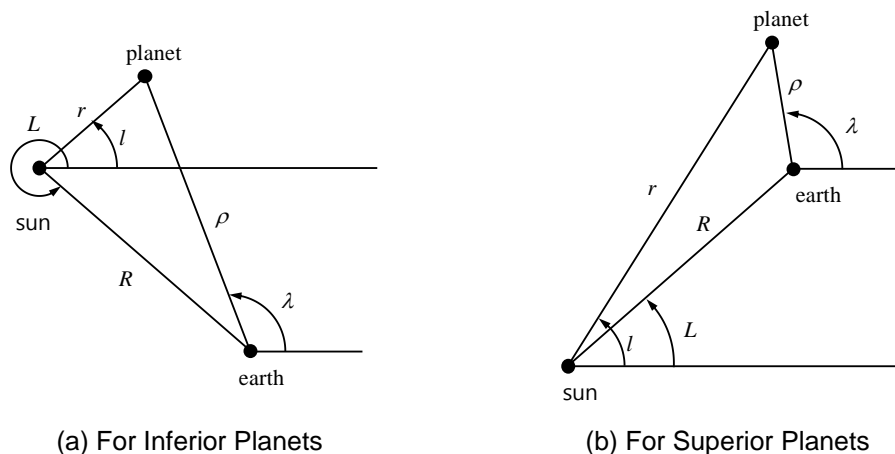


Figure 8. Calculations of the Geocentric Longitude: (A) For Inferior Planets, and (B) For Superior Planets

As shown in Figure 8, the geocentric longitude λ is calculated as:

$$\lambda = \tan^{-1}\left(\frac{R \sin(l-L)}{r-L \cos(l-L)}\right) + l'$$

For the inferior planets, and

$$\lambda = \pi + L + \tan^{-1}\left(\frac{r' \sin(L-l')}{R - r' \cos(L-l')}\right),$$

For the superior planets.

The geocentric latitude β is calculated as the following equation:

$$\beta = \tan^{-1}\left(\frac{r' \tan \psi \sin(\lambda - l')}{R \sin(l' - L)}\right),$$

Where the angle ψ is the angle between the current position and the projected position of the planet, as shown in Figure 9. Now, the geocentric longitude and latitude values for a planet are used to display it on the three-dimensional virtual night sky on the PC screen.

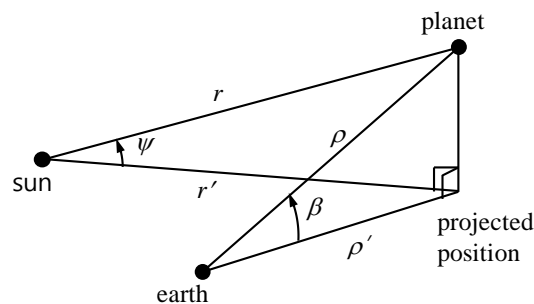


Figure 9. Calculation of the Geocentric Latitude

3. Results and Discussion

Figure 1 shows a snap shot of our celestial map rendering system. At the top of the screen, the system accepts the current time, the latitude and longitude of the current location, to render the current night sky at the current location. Also, user can specify other city names or other values of latitudes and longitudes, to render the fully-different night skies at the newly-specified location.

On the right side, a set of user interface buttons are provided to turning on and off the rendering of the whole sky, constellations, and planets. When a specific star or planet is selected, its detailed information is shown, based on the list of Messier objects, the Yale BSC, and other astronomical catalogues. We can drag the whole virtual night sky on the screen with the left mouse button. The middle wheel on the mouse can be used to zoom in and out for a specific celestial object.

The time display on the upper side shows the current local time, in our default configurations. When the user provides a specific time and/or the latitude and longitude of a specific location, he/she will get the corresponding virtual night sky just at that time, at that location. The radio boxes and check boxes on the right side of our user interface are selections on the rendering options. For example, user can choose only constellations or only planets to be displayed on the screen.

Our system calculates the new positions of the whole celestial objects on the screen, based on our algorithms explained in the previous sections. Since the earth rotates on its axis continuously, we need these calculations at least for every second.

Figure 10 shows snap shots from diurnal motion graphs, which is provided as an extra feature of our celestial map rendering system. When the user selects a specific star and some additional settings, this feature presents a set of motion graphs to precisely check the realistic diurnal motions.

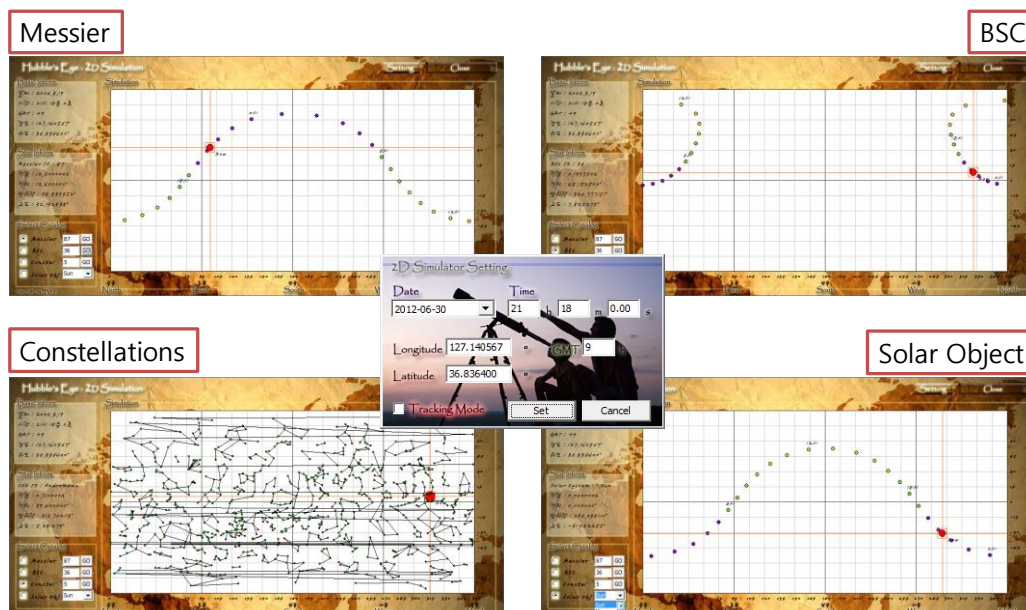


Figure 10. Snap Shots from Our Diurnal Motion Graphs

4. Conclusions

For amateur astronomers, the celestial calculations are complex and hard to be correctly achieved. In contrast, they have very few number of supporting tools at this time. We thus plan to provide computerized supporting tools for astronomers.

We developed a celestial map rendering system based on the precise astronomical information and modern 3D graphics features. Our system provides most of famous celestial catalogues, including the list of Messier objects, the Yale BSC list, planets, constellations, and all other celestial objects. Thus, our system finally displays more than 10.000 major celestial objects. For each of them, our system provides three-dimensional views and additional two-dimensional simulations for more details.

Our implementation successfully produces the virtual night sky from user settings. Some defects on the previous systems are also compensated. Conclusively, our system is helpful for amateur astronomers and ordinary persons to find the specific celestial object in the real night sky.

References

- [1] "Nautical Almanac Office, Explanatory Supplement to the Astronomical Ephemeris and the American Ephemeris and Nautical Almanac", H.M. Stationery Office, London, (1961).
- [2] M. Segal and K. Akeley, "The OpenGL Graphics System: A Specification", version 4.4, (2014).
- [3] J. Kessenich, "The OpenGL Shading Language", version 4.40, (2014).
- [4] Google Sky, <https://www.google.com/sky/>.
- [5] "Vixen astronomical telescopes", <http://www.vixenoptics.com/>.
- [6] "Stellarium", <http://www.stellarium.org/>.
- [7] D. A. Vallado, "Fundamentals of Astrodynamics and Application", Microcosm Press, ISBN 1-881883-12-4, (2001).
- [8] A. Blaauw, C. S. Gum, J. L. Pawsey and G. Westerhout, "The new IAU system of galactic coordinates", Monthly Notices of the Royal Astronomical Society, vol. 121, no. 2, (1960), pp. 123.
- [9] "List of Messier objects", <http://messier.seds.org/>.
- [10] "Yale Bright Star Catalogue", <http://tdc-www.harvard.edu/catalogs/bsc5.html>.
- [11] G. Holton and S. Brush, "Physics, the Human Adventure: From Copernicus to Einstein and Beyond", Piscataway, NJ: Rutgers University Press, (2009).

Authors



Young Chun Kwon, He was a master student in School of Computer Science and Engineering, Kyungpook National University. He is now a researcher in Samsung Electronics, Inc. His interests include computer graphics and parallel process.



Nakhoon Baek, He is currently a professor in the School of Computer Science and Engineering at Kyungpook National University, Korea. He received his B.A., M.S., and Ph.D. degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST) in 1990, 1992, and 1997, respectively. His research interests include graphics standards, graphics algorithms and real-time rendering. He is now also the Chief Engineer of Mobile Graphics Inc., Korea.