

**Department of Electrical and Electronic  
Engineering  
University of Hong Kong**



**Computer Engineering Final Year Project**

**Visualization of Objects under Relativistic Effects**

**Project Report**

*Author:* Edward Hung  
*University Number:* 9500448  
*Stream:* CE3  
*Date:* 14 March 1998  
*Reference:* final.doc  
*Issue:* 1.0

**Distributions:** Dr. W. Wang (CS)  
Prof. Francis Y.L. Chin (CS)  
Edward Hung

# TABLE OF CONTENTS

Acknowledgment	5
List of Figures	5
List of Tables	6
List of Symbols	7
<b>1. SUMMARY</b>	<b>9</b>
<b>2. INTRODUCTION</b>	<b>10</b>
2.1 Scope of the project	10
2.2 Purpose of the project	11
2.3 Organization of the report	12
2.4 Brief survey of existing work	13
<b>3. PROJECT MANAGEMENT</b>	<b>14</b>
3.1 Summary of progress (1/9/1997 - 27/3/1998)	14
3.2 Problems encountered during the project	15
3.3 Meetings	16
3.4 Internet	16
<b>4. THEORETICAL BACKGROUND</b>	<b>17</b>
4.1 Introduction to the special theory of relativity	17
4.2 Introduction to the relativistic effects	18
4.3 Basic formulae and equations	19
<b>5. HYPERBOLIC TRANSFORMATION AND POLYGON SHADING</b>	<b>23</b>
5.1 Introduction	23
5.2 Design and derivation	24
5.2.1 Data structure	24

5.2.2 Original view	24
5.2.3 Transformation for apparent position	25
5.2.4 Fast transformation for apparent position	28
5.2.5 Spectral shift and aberration	30
5.2.6 Representation of light of different wavelengths in RGB values	31
5.2.7 Intensity shift	32
5.2.8 User interface	33
5.3 Implementation	34
5.3.1 Product and platform	34
5.3.2 Transformation for apparent position	34
5.3.3 Features	35
5.3.4 Sample results	37
5.4 Design and implementation of 3D object editing program using graphical user interface	38
5.4.1 Introduction	38
5.4.2 Design	39
5.4.3 Implementation	41
<b>6. FOUR-DIMENSIONAL RAY-TRACING</b>	<b>42</b>
6.1 Introduction	42
6.2 The working principle	43
6.3 Planes	44
6.3.1 Equations of a plane under Lorentz transformation	44
6.3.2 Intersection with a moving plane	47
6.4 Quadric surfaces	48
6.4.1 Quadric surfaces under Lorentz transformation	48
6.4.2 Intersection with a moving quadric surface	51
6.5 Relativistic optics	52
6.5.1 Transformation of rays from light sources	52
6.5.2 Transformation of incident, reflected and refracted rays in vacuum	53

6.5.3 Transformation of incident, reflected and refracted rays in a moving medium	55
6.5.4 Relativistic Doppler's effect	56
6.6 Color model	57
6.7 Implementation	60
6.7.1 Product and platform	60
6.7.2 Features	60
6.7.3 Sample results	61
6.8 Modification on the object editing programs	62
<b>7. DISCUSSION OF SAMPLE RESULTS</b>	<b>64</b>
<b>8. CONCLUSION</b>	<b>66</b>
<b>9. APPENDICES</b>	<b>67</b>
9.1 Table of chromaticity coordinates and color-matching functions for RGB system	67
9.2 Graph of color-matching functions for RGB system	69
9.3 Explanation of some terms in color science	69
9.4 Progress	70
<b>10. REFERENCE</b>	<b>71</b>

## Acknowledgment

I wish to give my sincere thanks to Dr. Wenping Wang, my supervisor, who has given me valuable advice and helps.

Besides, I would like to thank Prof. Francis Y.L. Chin, the second examiner, who has given his kind attention and interest to my project.

I would also like to thank Mr. B. Chan who has given his valuable opinions and comments on my algorithms.

## List of Figures

Figure 1	Rotations of a vector to so that it is parallel to the x-axis	25
Figure 2	FitzGeraid-Lorentz contraction of a position vector	29
Figure 3	Original plane in object's frame and real plane in observer's frame	45
Figure 4	FitzGeraid-Lorentz contraction of a position vector	46
Figure 5	A moving plane	47
Figure 6	Rotations of a quadric surface	49
Figure 7	A moving light source	52
Figure 8	Graph of color-matching functions for RGB system	69
		Color Plate Page
Figure C1	Ordinary view of a cube	1
Figure C2	Real View of a cube	1
Figure C3	Ordinary view of a sphere	1
Figure C4	Real view of a sphere	1
Figure C5	Editing a cube in Object Edit Program	1
Figure C6	Editing a sphere in Object Edit Program	2
Figure C7	Setting of the wavelength of color and velocity of the object in Object Edit Program	2
Figure C8	Editing in Object Maindata Preview Program	2
Figure C9	Ordinary view of a cube at time=0	3
Figure C10	Real view of a cube at time=0 without spectral shift effect on light source	3
Figure C11	Real view of a cube at time=0	3
Figure C12	Ordinary view of a cube at time=14	4
Figure C13	Real view of a cube at time=14 without spectral shift effect on light source	4

Figure C14 Real view of a cube at time=14	4
Figure C15 Ordinary view of a cube at time=5	5
Figure C16 Real view of a cube at time=5 without spectral shift effect on light source	5
Figure C17 Real view of a cube at time=5	5
Figure C18 Ordinary view of a cube at time=2	6
Figure C19 Real view of a cube at time=2 without spectral shift effect on light source	6
Figure C20 Real view of a cube at time=2	6
Figure C21 Real view of geometric appearance of a cube at time=2; $v=0.98c$ without spectral shift	7
Figure C22 Real view of geometric appearance of a cube at time=5; $v=0.98c$ without spectral shift	7
Figure C23 to 26 Real view of a long plane with $v=0.6c$ without spectral shift effect on the light source	8
Figure C27 to C30 Real view of a long plane with $v=0.6c$	9
Figure C31 Editing a cube in Object Edit Program 2	10
Figure C32 One of the windows for adjusting the color attributes of a surface of the object in Object Edit Program 2	10
Figure C33 Adjusting the direction and magnitude of the object in Object Edit Program 2	10
Figure C34 Adjusting the other attributes of a surface of the object in Object Edit Program 2	11
Figure C35 Editing in Object Maindata Preview Program 2	11
Figure C36 Adjusting of the color of the light source in Object Maindata Preview Program 2	11

## List of Tables

Table 1 Telescope aperture required for different distance	11
Table 2 Summary of progress	14
Table 3 Action of clicking mouse in window 1 of Object Edit program	39
Table 4 Action of clicking middle mouse button in window 1 of Object Edit program	40
Table 5 Chromaticity coordinates and color-matching functions for RGB system	67
Table 6 Explanation of some terms in color science	69
Table 7 Progress	70

## List of Symbols

$ \mathbf{a} $	magnitude of vector $\mathbf{a}$
$\mathbf{a} \cdot \mathbf{b}$	dot product of vector $\mathbf{a}$ and vector $\mathbf{b}$
$\mathbf{a} \times \mathbf{b}$	cross product of vector $\mathbf{a}$ and vector $\mathbf{b}$
$\mathbf{a}, \mathbf{a}, \mathbf{a}', \mathbf{a}'$	$\mathbf{a}$ and $\mathbf{a}$ is in the object's frame; $\mathbf{a}'$ and $\mathbf{a}'$ is in the observer's frame
$\beta$	speed of the object (divided by light speed)
$\gamma$	$\sqrt{1 - \beta^2}$
$\theta_a$	angle between the directions of the ray $\mathbf{a}$ and the velocity $\mathbf{v}$
$\nu$	frequency of a light ray (wave)
$A, B, C, D$	coefficients and constant in a equation of a plane, i.e. $Ax + By + Cz + D = 0$
$b$	$b$ value of color in RGB system
$c$	light speed in vacuum
$g$	$g$ value of color in RGB system
$\mathbf{I}$	incident ray
$\mathbf{i}$	unit vector along x-axis
$\mathbf{j}$	unit vector along y-axis
$\mathbf{k}$	unit vector along z-axis
$k_d$	diffusion-reflection coefficient
$k_s$	specular-reflection coefficients
$k_t$	transmission coefficients
$n$	refractive index of the medium
$\mathbf{n}$	normal vector of a plane
$\mathbf{p}, \mathbf{p}_i$	position vector of a vertex or vertex $i$
$\mathbf{p}_a$	vector perpendicular to velocity vector $\mathbf{v}$ and coplanar with $\mathbf{v}$ and $\mathbf{a}$
$r$	$r$ value of color in RGB system
$\mathbf{R}$	reflected ray
$R_{-a}$	rotational matrix of clockwise rotation along a-axis
$R_a$	rotational matrix of anti-clockwise rotation along a-axis
$S_i$	shadow coefficients of object $I$
$t$	time

<b>T</b>	refracted ray
<b><math>\mathbf{u}_a</math></b>	unit vector of vector <b>a</b> , i.e. $\mathbf{a} /  \mathbf{a} $
<b>v</b>	velocity of the object (divided by light speed)
<b>x</b>	x coordinate
<b>y</b>	y coordinate
<b>z</b>	z coordinate



# 1. SUMMARY

This project ("Visualization of Objects under Relativistic Effect"), as it is named, aimed at applying computer graphics to visualize the relativistic effects on objects which are moving rapidly relative to the observer. Special theory of relativity is considered. Two approaches were used, which are "hyperbolic transformation and polygon shading" and "four-dimensional ray-tracing". The first one is efficient, using transformation and existing OpenGL Graphics library. The second one, which is a slow one, based on the conventional ray-tracing and appropriate formulae, however, can give realistic pictures. The formulae and algorithms used are clearly derived and explained. This report states how to solve the problems met in order to achieve the goal.

## 2. INTRODUCTION

This project is an individual final year project for the Bachelor degree of Engineering (Computer Engineering) of the University of Hong Kong, proposed by the project student, Edward Hung. The degree is jointly offered by the Department of Electrical and Electronic Engineering and the Department of Computer Science (CS). The project supervisor is Dr. Wenping Wang (CS) and the second examiner is Prof. Francis Y.L. Chin (CS). The project started in mid-September 1997 and is expected to be completed in March of 1997.

### 2.1 Scope of the project

This project is named as "Visualization of Objects under Relativistic Effects" (VORE) which consists of two parts: study and implementation.

The study part includes:

- study of physics of theories of relativity (the Special Theory of Relativity and the General Theory of Relativity, both by A. Einstein)
- study of physics related to visualization of objects under relativistic effects
- study of programming principles of computer graphics
- study of implementation of a computer graphics software using the OpenGL software library and OpenGL Utility Toolkit (GLUT).

The implementation part includes the design and implementation of programs which:

- allow the user to build and edit three-dimensional object models and define their attributes,
- show the user the original (ordinary) view the observer can see without considering the relativistic effects,
- show the user the actual (real) view the observer can see with considering relativistic effects. Two different approaches have been implemented, which are

"hyperbolic transformation and polygon shading" and "four-dimensional ray-tracing".

The relativistic effects implemented include only those of the Special Theory of Relativity due to the limitation of time and resource.

## 2.2 Purpose of the project

The objective of this project is to study the application of computer graphics to scientific visualization of objects under relativistic effects and to implement a real software application with this functionality.

There are several reasons to support the project:

- Visual appearance of moving objects with speed close to the light speed is counter-intuitive. Visualization of objects under relativistic effects lets relativists visualizing relativistic effects of the results of numerical calculation. Until now, not much sophisticated computer simulations have been available. [1][16]
- A more astrophysical minded person might think that it would be better to build telescopes powerful enough to actually look at an object such as an X-ray pulsar in nature. The following table gives the necessary aperture of an ideal telescope (one whose resolution is limited by diffraction) needed to resolve a ten-kilometer object at various typical distances. It is obvious that a telescope cannot be realized for objects that are outside our own solar system.

Distance		Telescope aperture
20 000 km	Australia	1 mm
400 000 km	Earth -- Moon	20 mm
80 million km	Earth -- Mars	4 m
4 billion km	Earth -- Neptune	200 m
$4 \times 10^{13}$ km = 4 light years	Nearest star besides Sun	2 000 km
400 light years	Cosmic neighborhood	200 000 km
12 000 light years	X-ray pulsar Her X-1	6 million km

Table 1 - Telescope apertures required for different distance

(Note: 1 light year = distance that light travel in one year, which is approximately equal to:

$300\,000\text{ km} \times 3600 \times 24 \times 365.25 = 9467280000000\text{ km} = 9.46728 \times 10^{12}\text{ km}$  )

Also the current technology does not allow us to take a closer look by traveling to objects outside own solar system, and this is likely to remain the same for at least several thousand years to come. Thus, the computer is the only method. [1]

- The software can be used for the purpose of physics education. [16]
- The software can contribute to the popularization of the abstract concepts in pure science.

## 2.3 Organization of the report

This document is the final report of the final year project "Visualization of Objects under Relativistic Effects". The report will first summarize the management of the project. Then the theoretical background knowledge, the derivation and the use of the equations and formulae will be given. Afterwards, the design of the algorithms and the implementation of the two approaches will be described. Finally, the results obtained will be discussed. All materials in this report were the results of the work of the project unless stated specially.

## 2.4 Brief survey of existing work

Many books about the theories of relativity are available. However, books and papers about the visualization of objects under relativistic effects are much fewer. Those found include [1][2][4][6][10][16]. Not much published work on the topic of computer simulation of visualization of objects under relativistic effects can be found. Those found include [1][16].

Similar works have been done by Hans-Peter Nollert, Ute Kraus, and Hanns Ruder of Institute for Astronomy and Astrophysics, University of Tubingen, Germany. They have worked on the approach of hyperbolic transformation and polygon shading for the special relativity and the approach of four-dimensional ray-tracing for the general

relativity [1]. Their first approach is similar to the first approach used in this final year project and this will be discussed later in this final report.

Besides, Andrew Howard, Les Kitchen and Sandy Dance of Department of Computer Science, University of Melbourne, Australia have also worked on the approach of modifying an existing ray-tracing package to make relativistic corrections for the special relativity [16]. The limitation of their product is that all objects (except the observer) must have the same velocity and there is no spectral shift because their color model was not modified. In this final year project, there is no limitation like that.

This final year project only deals with the special relativity. The extension of that to cover the general relativity is valuable, which is also much more difficult and time-consuming. For example, full general relativistic simulation takes seven hours using a parallelized ray-tracing program on three HP workstations - extrapolated to a PC, a 386/33 MHz would need about two months to do the same [1].

### 3. PROJECT MANAGEMENT

#### 3.1 Summary of progress (1/9/1997 - 27/3/1998)

Date	Work or event
15/9/97- 10/10/97	Project planning and project plan writing; Project proposal writing
1/9/97- 13/10/97	Study of theories of relativity
1/10/97- 27/11/97	Study of OpenGL and GLUT programming; creating of project homepage ( <a href="http://www.cs.hku.hk/~ehung/vore">http://www.cs.hku.hk/~ehung/vore</a> )
1/11/97- 27/11/97	Design of algorithm of the approach of hyperbolic transformation and polygon shading
29/11/97- 18/12/97	Implementation of the program (vore.c) using the approach of hyperbolic transformation and polygon shading; Study of color science and models
19/12/97	Demonstration of the Program (vore.c) to the supervisor and Requirement from the supervisor about the GUI for 3D object editing.
20/12/97- 2/1/98	Design and implementation of 3D object editing programs using GUI (oe.c, op.c)
27/12/97- 11/1/98	Writing of progress report
15/1/98	Mid-term Review with the second examiner; Demonstration of the programs (vore.c, oe.c, op.c) and discussion of the project.
3/1/98- 2/2/98	Study of relativistic optics and Design of the algorithm of the approach of Four Dimensional Ray Tracing
3/2/98- 2/3/98	Implementation of the program (rt.c) using the approach of Four Dimensional Ray Tracing; Modification of 3D object editing programs using GUI (oe2.c, op2.c)
14/3/98	Demonstration of the programs (rt.c, oe2.c, op2.c) to the supervisor

27/2/98- 15/3/98	Writing of final report and technical paper
16/3/98	Submission of final report and technical paper
27/3/98	Presentation

Table 2 - Summary of progress

The details of the progress can be found in Appendix 9.4.

## 3.2 Problems encountered during the project

### 1. Representation of light of different wavelengths in RGB values:

Study and collecting of data were done from reference books [11][12][13]. Data of RGB values of lights within the visible spectrum was found in [13]. Color-matching functions in **1931 CIE (Commission Internationale de l'Eclairage) colorimetric System (R, G, B)** are used, which include negative values. [13] The problem of representation using non-negative values of R, G, B was solved. The approach is discussed in the later part of this report.

### 2. Color model used in four-dimensional ray-tracing

In ray-tracing, the r, g, b values are directly used and the diffusion-reflection coefficients ( $k_d$ ), specular-reflection coefficients ( $k_s$ ), transmission coefficients ( $k_t$ ) and shadow coefficients ( $S_i$ ), etc. are all based on the RGB values. However, the wavelengths or spectra of light rays are required in calculating the spectral shifts. It is the problem how to relate both of them. To solve that, assumptions were made and the approach is discussed in the later part of this report.

### 3. Difficulty in verifying the results obtained in four-dimensional ray-tracing

The geometric shape of the result obtained from the four-dimensional ray-tracing can be compared easily with that obtained by using the approach of hyperbolic transformation and polygon shading. However, the directions of incident, reflected, refracted rays, etc. are not easy to verified. Thus a lot of time is used on testing in running the programs and calculations on paper.

### 4. Very heavy workload from other courses:

Assignments and tests from other courses, especially the Computer Graphics course, which, however, helps the study and practice of programming in OpenGL and GLUT.

## 3.3 Meetings

Several meetings were held between the project supervisor and the project student.

## 3.4 Internet

A homepage for the project is being built under the student's personal homepage directory (<http://www.cs.hku.hk/~ehung/vore>). HTML versions of project documents: project plan and proposal were placed there. Further searching of information in Internet was done.



## 4. THEORETICAL BACKGROUND

### 4.1 Introduction to the special theory of relativity [2]

The Michelson-Morley experiment was designed to measure the speed with which the earth moves through the medium of transmission of light. The null result of this experiment, combined with the results of other experiments, indicates the necessity of a major modification in the Newtonian notions of speed and time.

Furthermore, the galilean transformation law adequately describes the relation between the coordinates and times of an event relative to two different inertial systems *only* in those circumstances involving relative speeds much smaller than light speed ( $c$ ).

Thus, A. Einstein introduced two postulates:

1. The dynamic postulate of special relativity: Principle of Relativity:  
*All inertial reference systems are completely equivalent as regards our description of all laws of nature.*
2. The kinematic postulate of special relativity: The invariance of the speed of light:  
*The speed of light in a vacuum has the same value relative to all inertial reference frames and is independent of the relative velocity of the light source and the observer.*

There exist transformation equations relating the coordinates ( $x, y, z$ ) and  $t$  relative to an inertial observer  $S$  of an event and the coordinates ( $x', y', z'$ ) and  $t'$  relative to another inertial observer  $S'$  of the same event. The form of these transformation equation is restricted by the kinematic postulate of special relativity, and the explicit equations for the transformation depend on the relation between  $S$  and  $S'$ . For the

simplest nontrivial case,  $x' = \frac{x - Vt}{\sqrt{1 - (V^2 / c^2)}}$ ,  $y' = y$ ,  $z' = z$ ,  $t' = \frac{t - (V / c^2)x}{\sqrt{1 - (V^2 / c^2)}}$ . ... (1)

The formula for Lorentz contraction can be derived from the Lorentz transformation equations by finding the simultaneous positions relative to an inertial observer of the end points of a measuring rod moving relative to that observer. Furthermore, the Lorentz transformation equations show that there is no inconsistency in the fact that each of two inertial observers in relative motion observes the time dilation and Lorentz contraction in the clocks and rods of the other.

The instantaneous location of a rapidly moving extended objects is determined by the simultaneous positions of all points on the object; it differs from that of a stationary objects because of the effects of Lorentz contraction. The instantaneous visual appearance of a rapidly moving extended object is determined by the light that reaches the eye of the viewer simultaneously. This appearance differs from that of a stationary object because of the effects of the finite speed of light and the Lorentz contraction.

## 4.2 Introduction to the relativistic effects

According to Einstein's Special Theory of Relativity introduced in 1905, when an object is moving relatively to an observer with a speed closing to (but slower than) light speed, the object will be measured to contract in its direction of motion. It is a real effect on the *physical measurement*, i.e. the length of the object decreases if simultaneous measurements are made at the back and the front of the object. It is called as Lorentz contraction.

However, Einstein himself does not seem to have realized the difference between *measuring* an object in a moving frame of reference, and *looking* at it. Gamov actually gave a wrong description of what the world around us would look like if the speed of light were much slower than it really is, in his book "Mr Tompkins in wonderland".

In fact, we also have to consider the effect due to the finite speed of light. The actual image that the observer can see is distorted from the shape of the object (after Lorentz contraction). It is because light rays from different points of the objects will take different time to reach the observer's eyes (here the effect due to the separation of two eyes is neglected). Thus all the rays (from different points of the object) that reach the

observer's eyes simultaneously were not emitted from the objects simultaneously. Therefore what the observer sees will be a distorted object.

Apart from the distortion of the shape (geometric appearance), there is also a spectral shift (relativistic Doppler's effect). Since the speed of light in a vacuum has the same value relative to all inertial reference frames and is independent of the relative velocity of the light source and the observer, so the frequency (as well as wavelength) will be different when the object is approaching or leaving from the observer. It is just like that when the object is coming to us, it seems to be compressing the light waves when they leave it, resulting as the blue shift effect when we receive the light. When the object is leaving from us, it seems to be stretching the light waves when they leave it, resulting as the red shift effect when we receive the light.

The above effects (geometric distortion and spectral shift) are the relativistic effects on a fast moving object according to the special theory of relativity. There are two approaches to implement the effects. One is hyperbolic transformation and polygon shading, described in Section 5. The other is four-dimensional ray-tracing, described in Section 6.

### 4.3 Basic formulae and equations

It is difficult to find out the basic formulae and equations necessary, which are described below, no matter from books, papers, or journals. However, it is more difficult to decide the conditions, environments and usage of them. It is even more difficult to derive the advanced formulae and equations (from those basic ones) required in the implementations in order to satisfy the general case. For example, all the formulae assume that the velocity of the object (or frame) is moving in the direction of positive x-axis. Thus, in order to allow the objects to move in any direction, advanced formulae have to be obtained from the combinations and derivation from the basic equations. The derivations and use of them will be described in later sections of this report.

For the formulae and equations below, let the inertial reference frame (observer) S' be moving with respect to another inertial reference frame S with velocity  $v$ . The coordinates  $(x, y, z)$  and  $t$  are relative to S of an event and the coordinates  $(x', y', z')$  and  $t'$  relative to S' of the same event. Here we can consider that we ourselves are in the frame S. Most of the formulae and equations can be found in [2][3][7][8][9][10] and many other reference books, except specially stated.

**Lorentz Transformation** (derivation can be found in many reference books)

$$x' = \frac{x - vt}{\sqrt{1 - (v^2 / c^2)}}, y' = y, z' = z, t' = \frac{t - (v / c^2)x}{\sqrt{1 - (v^2 / c^2)}}. \quad \dots(2)$$

Let  $\mathbf{b} = v / c$  and change the subject of the equation, then,

$$x = \frac{x' + \mathbf{b}ct'}{\sqrt{1 - \mathbf{b}^2}}, y = y', z = z', t = \frac{t' + \mathbf{b}x' / c}{\sqrt{1 - \mathbf{b}^2}}. \quad \dots(3)$$

**Time dilatation**

$$\Delta t = \frac{\Delta t'}{\sqrt{1 - \mathbf{b}^2}} \quad \dots(4)$$

**FitzGerald-Lorentz contraction**

$$\Delta x = \Delta x' \sqrt{1 - \mathbf{b}^2} \quad \dots(5)$$

**Doppler Effect (Spectral Shift)**

$$\mathbf{n} = \sqrt{\frac{1 - \mathbf{b}}{1 + \mathbf{b}}} \mathbf{n}' \text{ for a receding object (S')} \quad \dots(6)$$

$$\mathbf{n} = \sqrt{\frac{1 + \mathbf{b}}{1 - \mathbf{b}}} \mathbf{n}' \text{ for an approaching object (S')} \quad \dots(7)$$

where  $\mathbf{n}$  is the frequency of light the observer in S measures;  $\mathbf{n}'$  is the frequency of light the observer in S' measures.

**Aberration** [5][7][10]

$$\mathbf{n} = \frac{1 + \mathbf{b} \cos \mathbf{q}'}{\sqrt{1 - \mathbf{b}^2}} \mathbf{n}' \quad \dots(8) \text{ (Doppler Effect in general case)}$$

$$\cos \mathbf{q} = \frac{\mathbf{b} + \cos \mathbf{q}'}{1 + \mathbf{b} \cos \mathbf{q}'}, \quad \dots(9)$$

$$\sin \mathbf{q} = \frac{\sqrt{1 - \mathbf{b}^2}}{1 + \mathbf{b} \cos \mathbf{q}'} \sin \mathbf{q}'; \quad \dots(10)$$

$$\text{i.e. } \tan \frac{\mathbf{q}}{2} = \frac{\sin \mathbf{q}}{1 + \cos \mathbf{q}} = \sqrt{\frac{1 - \mathbf{b}}{1 + \mathbf{b}}} \tan \frac{\mathbf{q}'}{2} \quad \dots(11)$$

where  $\mathbf{n}$  is the frequency of light traveling when the observer in  $S$  measures;  $\mathbf{n}'$  is the frequency of light the observer in  $S'$  measures;  $\mathbf{q}$  is the angle of the light arriving when the observer in  $S$  measures;  $\mathbf{q}'$  is the angle of the light arriving when the observer in  $S'$  measures.

**Transformation law for velocities** [7]:

$$u_x = \frac{u_x' + v}{1 + \mathbf{b} u_x' / c}, u_y = \frac{u_y' \sqrt{1 - \mathbf{b}^2}}{1 + \mathbf{b} u_x' / c}, u_z = \frac{u_z' \sqrt{1 - \mathbf{b}^2}}{1 + \mathbf{b} u_x' / c} \quad \dots(12)$$

where  $u_x, u_y, u_z$  are components of velocity of an object measured in  $S$ ;  $u_x', u_y', u_z'$  are components of velocity of an object measured in  $S'$ .

**Intensity shift** [1]:

$$\frac{I_n}{\mathbf{n}^3} = \text{constant} \quad \dots(13)$$

i.e. an invariant scalar along the path of any photon where  $I_n$  is the intensity and  $\mathbf{n}$  is the frequency of that photon.

**Transformation for apparent position** [6][10][1]:

The observer is at rest at the origin a coordinate system  $S$ . The system  $S'$  is moving in the  $x$  direction with a velocity  $\mathbf{b}$  in terms of  $c$ . The  $x$  axes are in the same line and when  $t = t' = 0$  the two origins  $0$  and  $0'$  are coincident. For  $0'$  to appear at  $0$ , the incident of observation (i.e. photography) is at  $t = 0$  and the light from  $0'$  will have been emitted at  $t = 0$ . The light from any other point  $P'$  ( $x', y', z'$ )

in  $S'$  which appears at  $P(x, y, z)$  in  $S$  must have been emitted at  $t = - (x^2 + y^2 + z^2)^{1/2} / c$  in order to be observed simultaneously with the light from  $O'$ .

The Lorentz transformations give

$$\mathbf{x}' = \mathbf{g} (\mathbf{x} - \mathbf{b} ct) = \mathbf{g} [\mathbf{x} + \mathbf{b} (x^2 + y^2 + z^2)^{1/2}], \mathbf{y}' = \mathbf{y}, \mathbf{z}' = \mathbf{z}, \text{ where as usual } \mathbf{g}^{-2} = 1 - \mathbf{b}^2. \quad \dots(14)$$

By expressing  $x, y, z$  in terms of  $x', y', z'$  the positions of points in  $S'$  can be found as they appear to the observer in  $S$ , viz.

$$\mathbf{x} = \mathbf{g} [\mathbf{x}' - \mathbf{b} (x'^2 + y'^2 + z'^2)^{1/2}], \mathbf{y} = \mathbf{y}', \mathbf{z} = \mathbf{z}'. \quad \dots(15)$$

Alternately it can be noted that the event of observation of light from the  $OO'$  coincidence is  $(0, 0, 0, 0)$  in  $S$  or  $(0, 0, 0, 0)$  in  $S'$ . Light from a point  $(x', y', z')$  on the object which arrives simultaneously with the ray from  $OO'$  must have started at  $t' = - (x'^2 + y'^2 + z'^2)^{1/2} / c$ .

Then the above expression for  $x$  follows directly from the Lorentz transformation  $x = \mathbf{g} (x' + vt')$ .

The apparent shape of a  $y'$ - $z'$  plane is one branch of a two-sheeted hyperboloid of revolution as noted by Yngstrom. The shape is obtained by setting  $x' =$  a constant or, say,  $x' = A$ . Then the above relation gives

$$(\mathbf{x} - \mathbf{g} A)^2 / \mathbf{g}^2 - \mathbf{b}^2 A^2 - \mathbf{y}^2 / A^2 - \mathbf{z}^2 / A^2 = 1 \quad \dots(16),$$

which, it can be seen, is the equation of an hyperboloid. Hence, straight lines perpendicular to the  $x'$  axis appear hyperbolic in shape.

### Velocity of Light in a moving medium [17]

Let the inertial reference frame (observer)  $S'$  be moving with respect to another inertial reference frame  $S$  with velocity  $v$ , the medium be at rest with respect to  $S'$ ,  $n$  be the refractive index of the medium,  $b'$  be the velocity of the light in the medium relative to  $S'$  and  $b$  be the velocity of the light in the medium relative to  $S$ , then:

$$b' = \frac{b + v}{1 + v / nc} \quad \dots(17)$$

## 5. HYPERBOLIC TRANSFORMATION AND POLYGON SHADING

### 5.1 Introduction [1]

There are two approaches to implement the relativistic effects (geometric distortion and spectral shift). One is four-dimensional ray-tracing. The other is hyperbolic transformation and polygon shading, described in this section..

This approach is simpler and more efficient than the four-dimensional ray-tracing, which will be described in Section 6. Objects are described as a collection of *luminous* polygons, excluding exterior light sources. Efficient algorithms are available for projecting these polygon onto the screen, determining obstructions from views by intervening polygons, and finding their shade for the resulting picture.

Using the Lorentz transformation and the traveling time for the photons coming from different points of the object, we can then compute the points in space from where the photons making up the picture have been emitted. Formulae (15), (16) are the results.

Given the description of an objects in its rest frame, all we have to do is to transform the positions of its defining points according to (15). The transformed object can then be displayed by conventional rendering method. Formulae (2), (8), (9), (10), (11), (13), (15) will be involved.

However, this approach has some limitations. It has to be assumed that all the objects in the scene are self-luminous. All effects of objects being illuminated by other parts of the scene is neglected. Shadows and light that is reflected by one object and illuminates another is neglected. Besides, reflection, refraction, transparency, etc., also cannot be handled.

In Section 5, the usage of notations (e.g.  $x$ ,  $x'$ ,  $x''$ , etc.) is different from other parts of the report. In other parts,  $x'$  is for the observer's frame while  $x$  is for the object's frame

if it is not specified. In this section,  $x'$  and  $x$  are not fixed to be used in that. Their meanings will be indicated appropriately.

## 5.2 Design and derivation

### 5.2.1 Data structure

Data of objects moving in certain velocities (not necessary to be the same) are loaded from a main data file selected by the user, data files of each object, and a data file storing RGB values of wavelengths in the visible spectrum. Data of each object include its velocity, coordinates of each vertex, vertices that each (triangular) plane consists of, wavelength of the color of each plane (or the coordinates of the center and radius of a sphere if it is a sphere). Besides, the current time, the coordinates and the velocity of the viewpoint are also included.

Each object and its data and related things exist as an instance of a class (object). Each object has its all data and its functions for transformation and drawing.

### 5.2.2 Original view

To draw an object in the original view (without considering the relativistic effects), use its *draw\_ordinary()* function. If the object consists of planes, then the planes will be drawn using the vertices composing them. If the object is a sphere, then a regular icosahedron (which is a Platonic solid composed of twenty faces that span twelve vertices, each face of which is an equilateral triangle) is used, each of whose triangular faces will be subdivided into four smaller triangles recursively and normalized, to form an approximation of a sphere for arbitrary accuracy ([14] p 83-89).



### 5.2.3 Transformation for apparent position

To draw an object in actual view (with considering the relativistic effects), use its *init\_transform()* function. If the object consists of planes, then all vertices will be transformed using the formula (15):

$$x = \mathbf{g} [x' - \mathbf{b}(x'^2 + y'^2 + z'^2)^{1/2}], y = y', z = z';$$

where  $x', y', z'$  are the coordinates of a point to be transformed,  $x, y, z$  are the coordinates of a point after transformation and as usual  $\mathbf{g}^{-2} = 1 - \mathbf{b}^2$ .

However, the only formula found applicable is for the case that the object is moving in the positive  $x$ -direction. Therefore transforms have to be performed on all the vertices of the object so that they are now moving in  $x$ -direction.

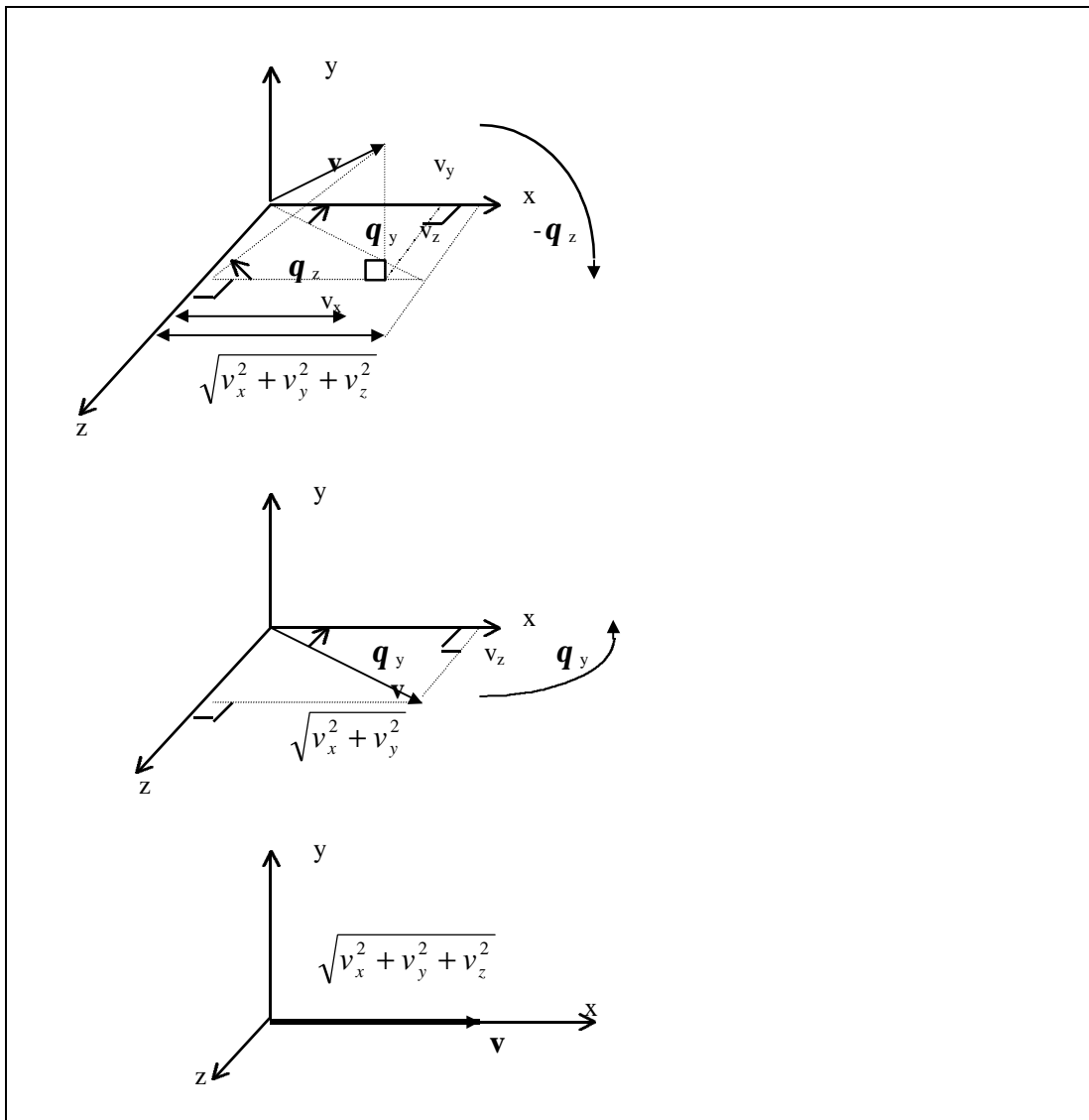


Figure 1 - Rotations of a vector to so that it is parallel to the x-axis

Let the velocity of the object be a vector  $\mathbf{v}$ , which has 3 components  $v_x, v_y, v_z$  parallel to the x-axis, y-axis and z-axis respectively. First, a rotation of angle  $-\mathbf{q}_z$  along the z axis is performed to bring the velocity vector to lie on the x-z plane. Then a rotation of angle  $\mathbf{q}_y$  along the y axis is performed to bring the velocity vector to lie on the x-axis.

The rotations can be represented as the follow matrices  $R_{-z}$  and  $R_y$ .

$$R_{-z} = \begin{bmatrix} \cos \mathbf{q}_z & \sin \mathbf{q}_z & 0 & 0 \\ -\sin \mathbf{q}_z & \cos \mathbf{q}_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ where } \tan \mathbf{q}_z = v_y/v_x \quad \dots(18)$$

$$R_y = \begin{bmatrix} \cos \mathbf{q}_y & 0 & \sin \mathbf{q}_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \mathbf{q}_y & 0 & \cos \mathbf{q}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ where } \tan \mathbf{q}_y = \frac{v_z}{\sqrt{v_x^2 + v_y^2}} \quad \dots(19)$$

The new velocity vector  $\mathbf{v}' = R_y R_{-z} \mathbf{v}$ , i.e.

$$\begin{bmatrix} v_x' \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \mathbf{q}_y & 0 & \sin \mathbf{q}_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \mathbf{q}_y & 0 & \cos \mathbf{q}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \mathbf{q}_z & \sin \mathbf{q}_z & 0 & 0 \\ -\sin \mathbf{q}_z & \cos \mathbf{q}_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix}$$

$$\text{i.e. } \begin{bmatrix} v_x' \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \mathbf{q}_y \cos \mathbf{q}_z & \cos \mathbf{q}_y \sin \mathbf{q}_z & \sin \mathbf{q}_y & 0 \\ -\sin \mathbf{q}_z & \cos \mathbf{q}_z & 0 & 0 \\ -\sin \mathbf{q}_y \cos \mathbf{q}_z & -\sin \mathbf{q}_y \sin \mathbf{q}_z & \cos \mathbf{q}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix}$$

$$\text{where } v_x' = \sqrt{v_x^2 + v_y^2 + v_z^2} \quad \dots(20)$$

Then, all the vertices of the object (or the vertices of the triangular planes of a regular icosahedron inscribing the sphere, if the object is a sphere) will be transformed similar to the velocity vector.

The new position vector of a vertex  $\mathbf{p}' = R_y R_{-z} \mathbf{p}$ , where  $\mathbf{p}$  is the original position vector, i.e.

$$\begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \mathbf{q}_y & 0 & \sin \mathbf{q}_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \mathbf{q}_y & 0 & \cos \mathbf{q}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \mathbf{q}_z & \sin \mathbf{q}_z & 0 & 0 \\ -\sin \mathbf{q}_z & \cos \mathbf{q}_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

$$\text{i.e.} \begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \mathbf{q}_y \cos \mathbf{q}_z & \cos \mathbf{q}_y \sin \mathbf{q}_z & \sin \mathbf{q}_y & 0 \\ -\sin \mathbf{q}_z & \cos \mathbf{q}_z & 0 & 0 \\ -\sin \mathbf{q}_y \cos \mathbf{q}_z & -\sin \mathbf{q}_y \sin \mathbf{q}_z & \cos \mathbf{q}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad \dots(21)$$

After the above rotations, then the object is now moving in the x-direction, and so the transformation  $p_x'' = \mathbf{g} [p_x' - \mathbf{b} (p_x'^2 + p_y'^2 + p_z'^2)^{1/2}]$ ,  $p_y'' = p_y'$ ,  $p_z'' = p_z'$  (where  $p_x'$ ,  $p_y'$ ,  $p_z'$  are the coordinates of a point to be transformed,  $p_x''$ ,  $p_y''$ ,  $p_z''$  are the coordinates of a point after transformation) ... (22) can be applied to every vertex.

However, as the object is moving with speed  $\mathbf{b}$ , so the position of the vertex should varies with time. Since now we have obtained the position of the vertex  $(p_x', p_y', p_z')$  at time zero in the object's frame, while the velocity  $\mathbf{v}$  of the object is measured in the observer's frame, so velocity of the object measured in the object's frame is  $\mathbf{g} \mathbf{v}$  where  $\mathbf{g}^{-2} = 1 - \mathbf{b}^2$ . Thus the original position vector becomes  $\mathbf{p} + \mathbf{g} t \mathbf{v}$  ... (23)

and the original  $p_x'$  becomes  $p_x' + \mathbf{g} t \mathbf{b}$ . ... (24)

Then we can use the new  $\mathbf{p}'$  to obtain  $\mathbf{p}''$  using the formulae (22).

Then the transformed vertices have to be rotated back by the matrix  $R_z R_y$ , where

$$R_y = \begin{bmatrix} \cos \mathbf{q}_y & 0 & -\sin \mathbf{q}_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin \mathbf{q}_y & 0 & \cos \mathbf{q}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \dots(25)$$

$$R_z = \begin{bmatrix} \cos \mathbf{q}_z & -\sin \mathbf{q}_z & 0 & 0 \\ \sin \mathbf{q}_z & \cos \mathbf{q}_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \dots(26)$$

$$\begin{bmatrix} p'''_x \\ p'''_y \\ p'''_z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \mathbf{q}_z & -\sin \mathbf{q}_z & 0 & 0 \\ \sin \mathbf{q}_z & \cos \mathbf{q}_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \mathbf{q}_y & 0 & -\sin \mathbf{q}_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin \mathbf{q}_y & 0 & \cos \mathbf{q}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p''_x \\ p''_y \\ p''_z \\ 1 \end{bmatrix}$$

$$\text{i.e.} \begin{bmatrix} p'''_x \\ p'''_y \\ p'''_z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \mathbf{q}_y \cos \mathbf{q}_z & -\sin \mathbf{q}_z & -\sin \mathbf{q}_y \cos \mathbf{q}_z & 0 \\ \cos \mathbf{q}_y \sin \mathbf{q}_z & \cos \mathbf{q}_z & -\sin \mathbf{q}_y \sin \mathbf{q}_z & 0 \\ \sin \mathbf{q}_y & 0 & \cos \mathbf{q}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p''_x \\ p''_y \\ p''_z \\ 1 \end{bmatrix} \quad \dots(27)$$

Actually,  $R_z$  is the inverse as well as transpose of  $R_z$  and  $R_y$  is the inverse as well as transpose of  $R_y$ . Obviously,  $R_z R_y$  is the inverse as well as transpose of  $R_y R_z$ .

After all the transformation of all vertices described as above, each (triangular) plane will be subdivided into four smaller triangles recursively, normalized, to form an approximation of the actual curved surface (a plane will be transformed into a curved surface) for arbitrary accuracy. In order to normalize the smaller triangles, the vertices on the original plane are transformed, and then used as the vertices of the small triangles. The recursive subdivisions are repeated until a certain level of depth is obtained or the difference between the small triangle just formed and the corresponding part of the actual curved plane is smaller than a threshold value. If the object consists of a sphere, then the triangular planes of a regular icosahedron inscribing the sphere will be processed similar to the above process to render the object in actual view after transformation.

The above contribute to the apparent shift of the position and the distortion in the shape of an object under relativistic effects.

## 5.2.4 Fast transformation for apparent position

The algorithm described in section 5.2.3 was used in the implementation of `vore.c` (visualization program using the approach of hyperbolic transformation and polygon shading). However, additional four rotations (i.e. tens of multiplications and additions)

are required for the position or velocity vector so that the velocity vector is now along the positive x-direction and the formula (15) can be applied. In order to have better performance, it is necessary to generalize the formula. Under thorough consideration and understanding of the formula, it is finally generalized as the following formula:

$$\mathbf{p}' = \mathbf{p} - \mathbf{v} |\mathbf{p}| \quad \dots(28)$$

where  $\mathbf{p}$  is the position vector at a particular time  $t$  (obtained by formula (23)),  $\mathbf{v}$  is the velocity vector,  $|\mathbf{p}|$  is the magnitude of  $\mathbf{p}$ . This formula generalizes the part  $[\mathbf{p}' - \mathbf{b}(\mathbf{p}_x'^2 + \mathbf{p}_y'^2 + \mathbf{p}_z'^2)^{1/2}]$  in formula (15) for the effect of finite speed of light.

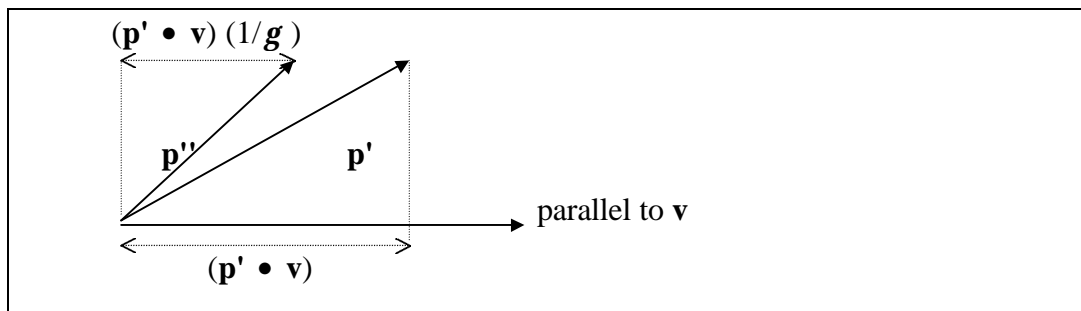


Figure 2 - FitzGeraid-Lorentz contraction of a position vector

Then,

$$\begin{aligned} \mathbf{p}'' &= \mathbf{p}' - (\mathbf{p}' \cdot \mathbf{v}) \mathbf{v} + (\mathbf{p}' \cdot \mathbf{v}) (1/g) \mathbf{v} \\ &= \mathbf{p}' + (\mathbf{p}' \cdot \mathbf{v})(1/g - 1) \mathbf{v} \quad \dots(29) \end{aligned}$$

where  $\mathbf{p}''$  is the apparent position vector,  $g^{-2} = 1 - \mathbf{b}^2$ . This formula generalizes the Lorentz transformation part in formula (15) which is originally for the object moving in positive x-direction only.

Now, these simple formulae can replace all the operations in Section 5.2.3. The number of operations are greatly reduced and the performance is much improved.

The algorithm described here was used in the implementation of the preview in `rt.c` (visualization program using the approach of four-dimensional ray-tracing, but the preview is using the approach of hyperbolic transformation and polygon shading).

## 5.2.5 Spectral shift and aberration

About the light received by the observer from the object, in this approach of hyperbolic transformation and polygon shading, it is necessary to assume that all objects are luminous, non reflective and non-transparent.

The following describes how to obtain the spectral shift of the colors of surfaces of an object.

The formulae used are as the following:

$$\mathbf{n} = \frac{1 + \mathbf{b} \cos \mathbf{q}'}{\sqrt{1 - \mathbf{b}^2}} \mathbf{n}', \quad \cos \mathbf{q} = \frac{\mathbf{b} + \cos \mathbf{q}'}{1 + \mathbf{b} \cos \mathbf{q}'} \quad (\text{ formulae (8), (9) (Doppler Effect in$$

general case)) where  $\mathbf{n}$  is the frequency of light that the observer measures;  $\mathbf{n}'$  is the frequency of light when it is leaving from the object (i.e. the frequency of the light if the object is static);  $\mathbf{q}$  is the angle of the light arriving when the observer measures as the object is moving;  $\mathbf{q}'$  is the angle of the light arriving when the observer measures if the object is static.

Using the above formulae, the actual wavelength of light that the observer receives can be calculated. For a particular vertex,  $\cos \mathbf{q}$  can be obtained by dividing the dot product of the velocity vector and the position vector of that vertex by the product of magnitudes of the velocity vector and the position vector of that vertex

$$\text{i.e. } \cos \mathbf{q} = \frac{\mathbf{p} \cdot \mathbf{v}}{|\mathbf{p}| |\mathbf{v}|} \quad \dots(30).$$

Moreover, as the color of a surface does not necessarily consist of light of one and only one frequency, however, it could be a combination of lights of some frequencies. It is usually the case found in everyday life and matters in the universe. Thus, the spectrum of the color of a surface is considered to be the combination of an impulse at a certain frequency and a flat spectrum of the range over the whole visible and invisible spectrum. As a result, even when the impulse is shifted to the range of invisible spectrum, the surface still has a dark gray color with a minimum but non-zero intensity.

## 5.2.6 Representation of light of different wavelengths in RGB values

Study and collecting of data were done from reference books [11][12][13]. Not all wavelengths within the visible spectrum can be represented by RGB (red, green, blue) model [11]. Using RGB model, the hue of some spectral lights can be matched, but not the saturation. For some lights, in order to match the saturation, one component (R, G or B) need to be subtracted from the combination of the other two. Obviously, this kind of subtraction cannot be done [12]. Data of RGB values of lights within the visible spectrum was found in [13]. Color-matching functions in **1931 CIE (Commission Internationale de l'Eclairage) colorimetric System (R, G, B)** are used. The system employs monochromatic stimuli at 700.0 nm, 546.1 nm, 435.8 nm as primaries (R, G, B). The color-matching functions include negative values (see Appendix 9.1, 9.2), which is inconvenient when tristimulus values and chromaticity coordinates of a given stimulus are to be evaluated by the less sophisticated computational methods from the spectral energy distribution of the stimulus and the standard color-matching functions. Reasons such as these led the CIE to adopt a transformation of the trichromatic system based on the primaries R, G, B to one based on new primaries X, Y, Z (some terms related to color science are explained in Appendix 9.3) [13]. In order to represent any color using non-negative values of color-matching functions, the following formulae were derived and used in the implementation:

The following shows the derivation:

Represent RGB values of a color as a point (r, g, b). in RGB space

Without loss of generality, for some given point (R, G, B) where R, G and B are some values, let  $R \geq 0, G \geq 0, B < 0$ .

In RGB space, on the plane  $r+b+g = R+G+B = C$  (constant) ... (31)

the hue is the same on the line connecting (R, G, B) and (C/3, C/3, C/3) (which is the white/grey/black color), but the saturation decreases when the point is closer to (C/3, C/3, C/3).

$$\text{Equation of the line: } \frac{r-R}{\frac{C}{3}-R} = \frac{g-G}{\frac{C}{3}-G} = \frac{b-B}{\frac{C}{3}-B} \quad \dots(32)$$

Now, it is required to find the point  $(R', G', 0)$  which is on the line above. Substitute it into the equation, the following are obtained:

$$R' = \frac{(R-B)(R+G+B)}{G+R-2B}, G' = \frac{(G-B)(R+G+B)}{G+R-2B}, B'=0 \quad \dots(33)$$

Since points on a line passing through the origin have the same color but different brightness,  $R'$  and  $G'$  are then normalized so that the points  $(R', G', 0)$  and  $(R, G, B)$  are equidistant from the origin. However, it can be ensured that neither of normalized  $R'$  nor  $G'$  are greater than 1 because from the Color-matching functions in **1931 CIE (Commission Internationale de l'Eclairage) Colorimetric System (R, G, B)** (found in [13]), the distances of all  $(r, g, b)$  from the origin are less than 1.

### 5.2.7 Intensity shift

Finally, for the intensity shift (formula (13)), the ratio of the intensity of the approaching to that of the receding object can be very great (which is 64) for a flat spectrum (so that the observed frequencies are fixed) even the relativistic speed is only 60% of light speed. This will make it impossible to display the intensity change realistically without losing almost all the resolution in brightness that a computer screen offers. The situation becomes worse for a higher speed. Therefore, this effect is not considered. Otherwise, it will be too dark to display for a receding object. [1]

However, the intensity of the light received is inversely proportional to the square root of the distance between the observer and the object where the light is emitted (in the case of a point light source for simplicity). The attenuation factor =  $1 / d^2$  where  $d$  is the distance. Nevertheless, in order to make it more likely as in the real life, so the attenuation factor is made to be  $1 / (1 + Cd^2)$  where  $C$  is a constant defined [11].



## 5.2.8 User interface

About the user interface, only a mouse is needed. A pop-up menu (opened by clicking the right mouse button) is also provided. The user is allowed to change the direction of view, change the time, play animation, etc. Three views are provided, which are ordinary (original) view (where relativistic effects are not considered), real view (which uses perspective projection, considers relativistic effects and is the view that an observer will really see) and real view using parallel projection.

## 5.3 Implementation

### 5.3.1 Product and platform

The source file of the completed program is "vore.c". The program was written in C++ with the OpenGL (Open Graphics Library) system which is a software interface to graphics hardware [14], and GLUT (OpenGL Utility Toolkit) which is a programming interface for writing window system independent OpenGL programs [15]. The hardware being used are SGI machines, including O2, Indy and Indigo. Besides, the preview of the four-dimensional ray-tracing program (rt.c) was also implemented using this approach.

### 5.3.2 Transformation for apparent position

In the actual implementation of the algorithm described in the section 5.2.3, after the data of an object have been loaded from the data file, passed to the object instance created, the  $q_z$ ,  $q_y$ ,  $\cos q_z$ ,  $\sin q_z$ ,  $\cos q_y$ ,  $\sin q_y$  are then calculated in the beginning (constructor of the object) so that the first set of rotations ( $R_y R_z$ , necessary for each vertex) can be performed more efficiently using:

$$\begin{aligned} p'_x &= \cos q_y \cos q_z p_x + \cos q_y \sin q_z p_y + \sin q_y p_z; \\ p'_y &= -\sin q_z p_x + \cos q_z p_y; \\ p'_z &= -\sin q_y \cos q_z p_x - \sin q_y \sin q_z p_y + \cos q_y p_z; \quad \dots (34) \end{aligned}$$

The second set of rotations ( $R_z R_y$ , also necessary for each vertex after transformation for apparent position) is performed using OpenGL subroutines:

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glRotatef (thetaz, 0.0, 0.0, 1.0);
glRotatef (-thetay, 0.0, 1.0, 0.0);
```

where thetaz is  $q_z$ , thetay is  $q_y$ .

The OpenGL subroutine glRotatef is used in the second set of rotations but not in the first set of rotations because:

1. after the first set of rotations, the transformation for apparent position is different for every vertex, thus no matrix can be applied generally like rotations using `glRotatef` or other simple transformations (e.g. translation or scaling), so the first set of rotations has to use the formulae above directly;
2. before the second set of rotations is applied, the image expected (the object after transformations) has already been drawn, so the rotations can be applied using OpenGL subroutine (which performs on the modeling, drawing) simply.

On the other hand, the preview of the four-dimensional ray-tracing program (`rt.c`) was implemented using the algorithm described in Section 5.2.4, which are much more efficient.

### 5.3.3 Features

#### **All relativistic effects were implemented**

All relativistic effects (possible to be implemented), including the change in apparent positions, distortion of shapes, spectral shift, etc., were all implemented except intensity shift, whose reason was stated in Section 5.2.7.

#### **File handling**

Data of objects to be shown are not coded in the program but are inputted from data files so that different objects with different parameters can be processed. The user is allowed to select a data file to load the scene wanted.

#### **Three views**

Three views are provided, which are **ordinary (original) view** (where relativistic effects are not considered), **real view** (which uses perspective projection, considers relativistic effects and is the view which an observer will really see) and **real view using parallel projection** (which also considers relativistic effects, but shows how the shaded polygon actually looks like in the 3D coordinates system, which is in fact

neither the view any observer can see nor how the actual objects are due to Lorentz transformation).

In fact, Lorentz transformation is the one and only one actual transformation about position physically, but what seen is not simply only that; what is seen optically can be obtained by the formulae of transformation of apparent position (equation (15) in proposal, and also mentioned in Section 2.2.3 of this report).

### **Deactivation of view windows**

One or more view windows can be deactivated or activated so that if more windows are deactivated, then the processing can be faster.

### **Toggle of the option of spectral shift**

Spectral shift for both windows for real view using perspective projection and real view using parallel projection can be toggled to be used or not. If the option of spectral shift is off, then the objects in the real view will be shown in their original colors and can be recognized more easily because in most of the time the real colors of the objects are dark gray due to the spectral shift. Moreover, the geometric shape of the objects in real view can be seen more easily. Besides, the option of spectral shift being off can also help the processing a bit faster.

### **Toggling of background color**

The background color of the view windows can be toggled between gray and black so that the user can find out the objects in real view (which are usually dark gray most of the time) more easily against the gray background.

### **Changing of direction of view.**

The user is allowed to change the direction of view easily at any time by clicking the mouse button or selecting from pop-up menu.

### **Changing of size of view plane.**

The user is allowed to change the size of the view plane easily at any time by dragging the border of the window.

## Changing of time

The user is allowed to change the time in the world being simulated, i.e. the time can be set so that how the objects look like at a particular time can be seen.

## Animation

The user is allowed to play an animation with the data loaded on one or more view windows. With testing in O2 machines, the number of frames per second in the animation can be as high as 10 provided that the total number of polygons in all the original objects is less than 20 and the precision of the pictures is just acceptable or higher and only the real view window is active. If the total number of polygons in all the original objects increases or the precision of the pictures increases, then the number of frames per second in the animation will decrease.

In general, the factors affecting the number of frames per second in the animation are the total number of polygons in all the original objects, the precision of the pictures, the speeds of the objects, and the distances of the objects from the view point, the number of view windows being active.

### 5.3.4 Sample results

Some sample results of the ordinary view and the real view of a cube (Figure C1, Figure C2) and a sphere (Figure C3, Figure C4) moving from the left to the right under high speed are shown. Discussion on the sample results are in Section 7.

## 5.4 Design and implementation of 3D object editing program using graphical user interface

### 5.4.1 Introduction

As a request from the supervisor during the demonstration of the program (vore.c) using the approach of hyperbolic transformation and polygon shading, it was decided to design and implement a GUI (Graphical User Interface) for editing objects. As a result, the user does not need to modify the data file directly and can see how the objects look like while editing them. It gives the user a more convenient intuitive and easier way to edit objects that are to be loaded by the program vore.c.

There are two programs, Object Edit program and Object Maindata (Scene) Preview program, whose source codes are "oe.c" and "op.c" respectively.

**Object Edit program** is used for editing an object used by the vore.c, i.e. Visualization of Objects under Relativistic Effects Software (VORES)). This program lets the user edit an object in an intuitive way where the user can move the vertex, create or delete a vertex or a triangular plane, move the whole object, change the color of a selected plane, create a sphere and set its position and radius, set the velocity of the object, load or save the object from or to a file, etc.

**Object Maindata Preview program** is used for editing an object used by the vore.c, i.e. Visualization of Objects under Relativistic Effects Software (VORES)). This program lets the user edit a maindata file and preview the ordinary view of the objects described in the maindata file. The user can load and save a maindata file, load objects into the maindata file, play the animation, rotate the view, etc.

## 5.4.2 Design

### 5.4.2.1 Design of Object Edit program

The data structure is similar to the one in *vore.c*. Data of an object are loaded from a data file of the object, and a data file storing RGB values of wavelengths in the visible spectrum. Data of an object include its velocity, coordinates of each vertex, vertices that each (triangular) plane consists of, wavelength of the color of each plane (or the coordinates of the center and radius of a sphere if it is a sphere).

In window 1 (Figure C5, C6) each vertex will have its projection on each projection plane (xy-plane, yz-plane, xz-plane) respectively. Each plane will have its wireframe projected on each projection plane (xy-plane, yz-plane, xz-plane) respectively.

The interface was designed to allow the user to do all the editing with a mouse only and no keyboard is needed. A pop-up menu (opened by clicking the right mouse button) is also provided

In the window 1, where the object is shown:

After clicking the left mouse button...

What is clicked on	Action
vertex, which is not one of the vertices of the selected plane (if any)	delete the vertex and all the planes involved; deselect all selected vertices (if any)
vertex on a plane which is selected	delete the plane
vertex on projection	move the vertex (dragging the mouse while the button is still being pressed)
projected wireframe	move the plane without affecting the third dimension (dragging the mouse while the button is still being pressed)
nothing	if no plane was selected, rotate the view (dragging the mouse while the button is still being pressed), otherwise rotate the whole object about its centroid

Table 3 - Action of clicking left mouse button in window 1 of Object Edit program

After clicking the middle mouse button...

What is clicked on	Action
vertex, and vertex on projection	select a vertex; if two vertices have been selected then: if no plane exists, create a new plane and select it; else select the plane consisting of the three vertices; if a plane is selected then: deselect the plane and select the vertex clicked
plane	select the plane and the three vertices of the plane as well
projected wireframe	move the whole object without affecting the third dimension (dragging the mouse while the button is still being pressed)
nothing	deselect vertices selected (if any), create a new vertex at origin

Table 4 - Action of clicking middle mouse button in window 1 of Object Edit program

The pop-up menu (opened by clicking right mouse button) allows the user to reset the projection (view), load or save object, clear the object, toggle to create a sphere.

In window 2 (Figure C7), the user can set the wavelength of the color of any plane, the direction and magnitude of velocity of the object.

#### 5.4.2.2 Design of Object Maindata Preview program

The data structure is similar to the one in `vore.c`. Data of objects moving in certain velocities (not necessary to be the same) are loaded from a main data file selected by the user, data files of each object, and data file storing RGB values of wavelengths in the visible spectrum. Data of each object include its velocity, coordinates of each vertex, vertices that each (triangular) plane consists of, wavelength of the color of each plane (or the coordinates of the center and radius of a sphere if it is a sphere). Besides, the current time, the coordinates and the velocity of the viewpoint are also included.



In the window (Figure C8), each vertex will have its projection on each projection plane (xy-plane, yz-plane, xz-plane) respectively. Each plane will have its wireframe projected on each projection plane (xy-plane, yz-plane, xz-plane) respectively.

The interface was designed to allow the user to do all the editing with a mouse only and no keyboard is needed. The user can click the left mouse button and drag to rotate the view. The pop-up menu (opened by clicking right mouse button) allows the user to reset the projection (view), load or save object maindata file, load one or more objects from data files, clear all the objects, play animation.

### 5.4.3 Implementation

#### **Product**

The source files of the completed programs (Object Edit program and Object Maindata Preview program) are "oe.c" and "op.c" respectively.

#### **Platform**

Same as "vore.c", the programs "oe.c" and "op.c" were written in C++ with the OpenGL (Open Graphics Library) system which is a software interface to graphics hardware [14], and GLUT (OpenGL Utility Toolkit) which is a programming interface for writing window system independent OpenGL programs [15]. The hardware being used is SGI machines.

#### **Compatibility of data files**

The data files used by vore.c, oe.c, and op.c are completely compatible so that all data files produced by oe.c and op.c can be used immediately by vore.c i.e. objects edited by oe.c and op.c can be used by vore.c immediately to show the view under relativistic effects.

#### **Features**

All features described in the design were implemented.

## 6. FOUR-DIMENSIONAL RAY-TRACING

### 6.1 Introduction

Section 5 describes the approach of hyperbolic transformation and polygon shading. However, this approach has some limitations. It has to be assumed that all the objects in the scene are self-luminous. All effects of objects being illuminated by other parts of the scene is neglected. Shadows and light that is reflected by one object and illuminates another are neglected. Besides, refraction, transparency, etc., also cannot be handled.

If images are actually obtained by ray-tracing, then the conceptual difference to "conventional" ray-tracing in Newtonian space is not dramatic. Photons still travel in straight lines, but since they now travel at a finite velocity, we have to keep track of time while tracing the rays and looking at their intersections with objects in the scene, i.e. the ray-tracing now is a "four-dimensional" one. Although complete ray-tracing can give us a realistic image with all relevant effects included, but it is very expensive in terms of computational resources. However, this approach can correctly treat shadows reflection, refraction, etc. [1]

In this approach, the objects under ray tracing can be moving in a speed comparative with the speed of light, thus the following are necessary to be considered:

1. the intersection of moving objects and tracing rays;
2. relativistic optics: e.g. direction of incident, reflected and refracted rays in the object's frame and observer's frame, direction of tracing rays from moving objects to moving light source in the object's frame and observer's frame;
3. Doppler's effect / spectral shift (wavelength of light in the object's frame and observer's frame);
4. color model suitable in this ray-tracing.

Formulae (2), (3), (5), (8), (9), (10), (11), (12), (13), (17), (18), (19), (23), (25), (26), (30) will be involved and some other advanced formulae and equations have to be derived, which are described in later sections.

Besides, since many properties/attributes of the object are needed by the ray tracing, so a new data structure and a file format for storing the data of objects have to be designed. As a result, some parts of the Object Edit program and Object Maindata Preview program (whose source codes are "oe.c" and "op.c") were modified.

In the notations (e.g.  $x$ ,  $x'$ ) in Section 6,  $x'$  is for the observer's frame while  $x$  is for the object's frame if it is not specified (except  $\mathbf{v}$ ,  $\mathbf{b}$ ,  $\mathbf{g}$  which is universally defined as relating to the velocity of the object relative to the observer).

## 6.2 The working principle

The main principle of the four-dimensional ray-tracing for special theory of relativity is the same as the conventional ray-tracing. However, there are some modifications.

1. The tracing ray leaves from the view point and tries to intersect with a moving object. Calculations are based on the observer frame (refer to Section 6.3 and 6.4). The ray is traveling in the negative time direction/axis. The time (which is in the past) that the intersection occurs is noted.
2. The vectors of the incident ray, refracted ray, refracted ray and the ray from the light source in the observer's frame are obtained as done in the conventional ray-tracing.
3. The vectors of the incident ray, refracted ray, refracted ray and the ray from the light source in the object's frame are obtained from those in the observer frame according to relativistic optics (refer to Section 6.5).
4. The pixel color due to diffuse and regular reflection is obtained as in the conventional ray-tracing (but using the vectors in observer's frame). However, the colors of the lights of the light sources are modified by Doppler's effect / spectral shift (refer to Section 6.5).
5. The pixel color due to recursive reflection and refraction is obtained as in the conventional ray-tracing (but using the vectors in observer's frame). However, the vectors of the refracted and refracted rays in the observer's frame (refer to Section

- 6.5) and the time of intersection are used as the input to the recursion and the color obtained is modified by Doppler's effect / spectral shift (refer to Section 6.5).
6. The total color obtained is modified by Doppler's effect / spectral shift and returned (refer to Section 6.5).

Section 6.3 describes the Lorentz transformation for a plane and the calculation of intersection.

Section 6.4 talks about the Lorentz transformation for a quadric surface and the calculation of intersection.

Section 6.5 is about the relativistic optics which deal with the transformation of the laws of reflection and refraction and Doppler's effect.

Section 6.6 says about the color model used in this ray-tracing.

## 6.3 Planes

### 6.3.1 Equations of a plane under Lorentz transformation

The unit vectors along the positive x, y, z directions are  $\mathbf{i}$ ,  $\mathbf{j}$ ,  $\mathbf{k}$  respectively.

Here the original plane is defined as  $Ax + By + Cz + D = 0$  consisting of 3 vertices  $p_i = (x_i, y_i, z_i)$  where  $i = 0, 1, 2$ , moving in velocity  $\mathbf{v}$ , with a unit normal vector  $\mathbf{u}_n$ .

$$\text{velocity vector, } \mathbf{v} = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k} \quad \dots(35)$$

$$\text{speed, } \mathbf{b} = |\mathbf{v}| \quad \dots(36)$$

$$\text{unit velocity vector, } \mathbf{u}_v = \mathbf{v} / |\mathbf{v}| \quad \dots(37)$$

$$\mathbf{g}^{-2} = 1 - \mathbf{b}^2. \quad \dots(38)$$

$$\text{unit normal vector of the plane } Ax + By + Cz + D = 0 \text{ is } \mathbf{u}_n = \frac{A\mathbf{i} + B\mathbf{j} + C\mathbf{k}}{\sqrt{A^2 + B^2 + C^2}} \dots(39)$$

The derivation below will use normalized plane so that  $A^2 + B^2 + C^2 = 1$

$$\text{i.e. } \mathbf{u}_n = A\mathbf{i} + B\mathbf{j} + C\mathbf{k}. \quad \dots (40)$$

Now it is necessary to obtain the normalized plane (in the observer's frame)  $A'x + B'y + C'z + D' = 0$  where  $A'^2 + B'^2 + C'^2 = 1$ .

There are two approaches to find the A', B', and C'. One is to find out the unit normal vector of the plane in the observer's frame from the unit normal vector of the plane in the object's frame directly. The other is to find out the positions of the three vertices of the plane in the observer's frame, then compute the normal vector of the plane consisting of them.

The following describes the first approach first.

According to the special theory of relativity, the Lorentz transformation will make the object frame contract in the direction of the object's velocity direction.

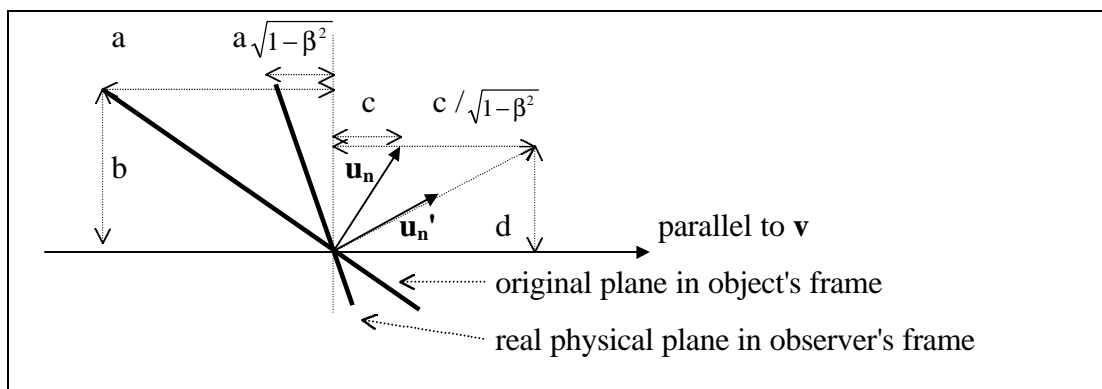


Figure 3 - Original plane in object's frame and real plane in observer's frame

In the object's frame, the original plane has a unit normal vector  $\mathbf{u}_n$ .

By Lorentz transformation, in the observer's frame, the real physical plane is a contracted one (a result of Lorentz contraction), compared with the original plane in the object's frame. The corresponding unit normal vector is  $\mathbf{u}_n'$ .

The contraction of the plane is in the direction of the velocity  $\mathbf{v}$ , along the plane (where the origin is on) perpendicular to  $\mathbf{v}$ . The factor of contraction is  $\sqrt{1 - \beta^2}$  (from (5)).

If the unit vector  $\mathbf{u}_n'$  is extended so that it has the same vector component of the vector perpendicular to  $\mathbf{v}$  as  $\mathbf{u}_n$ , then its vector component parallel to  $\mathbf{v}$  of  $\mathbf{u}_n'$  can be shown to be  $1 / \sqrt{1 - \beta^2}$  longer than that of  $\mathbf{u}_n$ .

For the original plane, the product of the slope of the plane and the slope of the normal vector is -1, i.e.  $-\frac{a}{b} \times \frac{c}{d} = -1$  ... (41)

For the real physical plane, the product of the slope of the plane and the slope of the normal vector must be also -1,

$$\text{i.e. } -\frac{a\sqrt{1-\beta^2}}{b} \times \frac{c/\sqrt{1-\beta^2}}{d} = -\frac{a}{b} \times \frac{c}{d} = -1 \dots(42), \text{ which is also satisfied.}$$

As a result,  $\mathbf{u}_n'$  can be written in terms of  $\mathbf{u}_n$ , unit velocity vector  $\mathbf{u}_v$  and  $\beta$ .

$$\begin{aligned} \mathbf{u}_n' &= \mathbf{u}_n - (\mathbf{u}_n \cdot \mathbf{u}_v) \mathbf{u}_v + (\mathbf{u}_n \cdot \mathbf{u}_v) \mathbf{u}_v / \sqrt{1-\beta^2} \\ &= \mathbf{u}_n + (\mathbf{u}_n \cdot \mathbf{u}_v) \mathbf{u}_v (1 / \sqrt{1-\beta^2} - 1) \\ &= \mathbf{u}_n + (\mathbf{u}_n \cdot \mathbf{u}_v) \mathbf{u}_v (\gamma - 1) \end{aligned} \quad \dots(43)$$

$$\mathbf{u}_n' = A\mathbf{i} + B\mathbf{j} + C\mathbf{k} \quad \dots(44)$$

$$\text{i.e. } A' = \mathbf{u}_n' \cdot \mathbf{i}, \quad B' = \mathbf{u}_n' \cdot \mathbf{j}, \quad C' = \mathbf{u}_n' \cdot \mathbf{k} \quad \dots(45)$$

Then the following describes the second approach, which is simpler and used in the implementation.

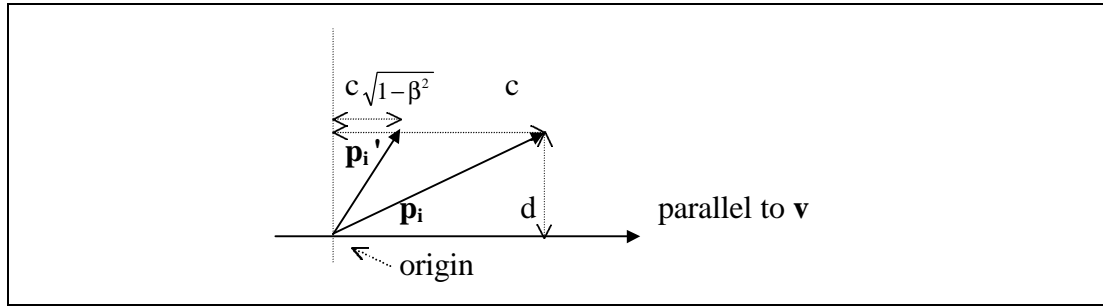


Figure 4 - FitzGeraid-Lorentz contraction of a position vector

By Lorentz transformation, in the observer's frame, the position of each vertex  $\mathbf{p}_i$  ( $i = 0, 1, 2$ ) is shifted in the direction of the velocity  $\mathbf{v}$  to become  $\mathbf{p}_i'$ . The distance of the vertex from the plane (where the origin is on) perpendicular to  $\mathbf{v}$  is contracted. The factor of contraction is  $\sqrt{1-\beta^2}$  (from (5)). Similar to (43),

$$\begin{aligned} \mathbf{p}_i' &= \mathbf{p}_i - (\mathbf{p}_i \cdot \mathbf{u}_v) \mathbf{u}_v + (\mathbf{p}_i \cdot \mathbf{u}_v) \mathbf{u}_v \sqrt{1-\beta^2} \\ &= \mathbf{p}_i + (\mathbf{p}_i \cdot \mathbf{u}_v) \mathbf{u}_v (\sqrt{1-\beta^2} - 1) \\ &= \mathbf{p}_i + (\mathbf{p}_i \cdot \mathbf{u}_v) \mathbf{u}_v (1 / \gamma - 1) \quad \text{where } i = 0, 1, 2 \end{aligned} \quad \dots(46)$$

Then, with the definition of normal vector in the object's frame

$$\mathbf{n} = (\mathbf{p}_1 - \mathbf{p}_0) \times (\mathbf{p}_2 - \mathbf{p}_0) \quad \dots(47) \quad (\text{where } \times \text{ is cross product})$$

the normal vector of the plane in the observer's frame is

$$\mathbf{n}' = (\mathbf{p}_1' - \mathbf{p}_0') \times (\mathbf{p}_2' - \mathbf{p}_0') \quad \dots(48)$$

Then  $\mathbf{n}'$  is normalized to obtain  $\mathbf{u}_n'$  and  $A'$ ,  $B'$  and  $C'$  can be obtained as (45).

Now, A', B' and C' are found, to find D', we can use the distance between the two planes and the origin.

distance of the plane in the object's frame from the origin

$$= d = \mathbf{p}_i \cdot \mathbf{u}_n = -D \quad \text{for any } i \text{ from } 0, 1, 2 \dots (49)$$

distance of the plane in the observer's frame from the origin

$$= d' = \mathbf{p}_i' \cdot \mathbf{u}_n' = -D' \quad \text{for any } i \text{ from } 0, 1, 2 \dots (50)$$

Now, we have obtained the normalized plane in the observer's frame,  $A'x + B'y + C'z + D' = 0$  which is used for calculation of intersection in ray-tracing

### 6.3.2 Intersection with a moving plane

The normalized plane in the observer frame moving with velocity  $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ , is  $A'x + B'y + C'z + D' = 0$  where  $A'^2 + B'^2 + C'^2 = 1$ .

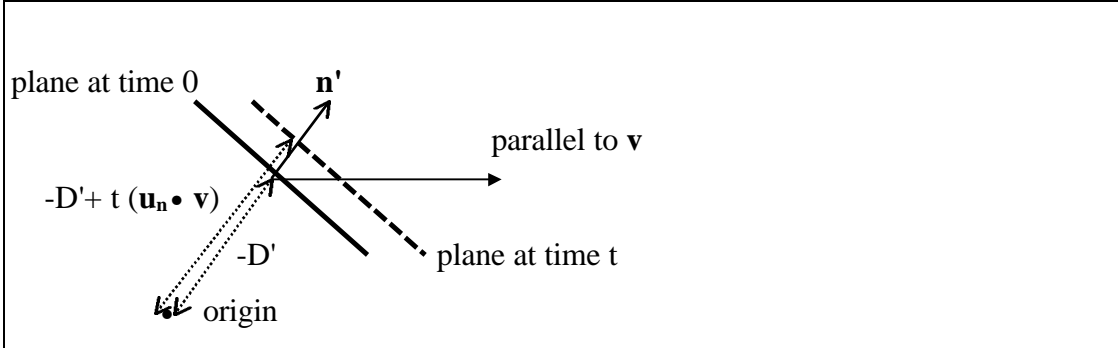


Figure 5 - A moving plane

At time 0, the distance of the plane from the origin =  $-D'$

At time t, the distance of the plane from the origin

$$= -D' + t (\mathbf{u}_n' \cdot \mathbf{v}) = -D' + t (A'v_x + B'v_y + C'v_z) \quad \dots (51)$$

Thus, substitute the above into the original plane equation:

$$A'x + B'y + C'z + D' - t (\mathbf{u}_n' \cdot \mathbf{v}) = 0$$

$$A'x + B'y + C'z + D' - t (A'v_x + B'v_y + C'v_z) = 0$$

$$A'(x - tv_x) + B'(y - tv_y) + C'(z - tv_z) + D' = 0 \dots (52)$$

Since the ray is traveling in the negative time axis, so the equation (52) becomes

$A'(x + tv_x) + B'(y + tv_y) + C'(z + tv_z) + D' = 0$  where the time  $t$  increases while the time in the real world is decreasing ... (53)

Assume each point  $(x, y, z)$  along the tracing ray from a point  $(x_p, y_p, z_p)$  with the direction  $(\Delta x, \Delta y, \Delta z)$  where  $\Delta x^2 + \Delta y^2 + \Delta z^2 = 1$ . For a tracing ray traveling inside a medium with a speed of  $|\mathbf{T}|$ , then  $\Delta x^2 + \Delta y^2 + \Delta z^2 = |\mathbf{T}|^2$ .

Substitute  $x = x_p + t\Delta x$ ,  $y = y_p + t\Delta y$ ,  $z = z_p + t\Delta z$ , the equation (53) becomes:

$$A'(x_p + tv_x + t\Delta x) + B'(y_p + tv_y + t\Delta y) + C'(z_p + tv_z + t\Delta z) + D' = 0 \quad \dots(54)$$

$$t = -\frac{A'x_p + B'y_p + C'z_p + D'}{A'(\Delta x + v_x) + B'(\Delta y + v_y) + C'(\Delta z + v_z)} \quad \dots(55)$$

## 6.4 Quadric surfaces

### 6.4.1 Quadric surfaces under Lorentz transformation

The general form of a quadric surface is

$$f(x, y, z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k = 0 \quad \dots(56)$$

or

$$P^T Q P = 0, \text{ where } Q = \begin{bmatrix} a & d & f & g \\ d & b & e & h \\ f & e & c & j \\ g & h & j & k \end{bmatrix} \text{ and } P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \dots(57)$$

can represent spheres, ellipsoids, cylinder, cone, parabolic and hyperbolic surfaces.

Given a transformation matrix  $M$ , the quadric surface equation can be changed to:

$$Q' = (M^{-1})^T Q M^{-1} \quad \dots(58)$$

so  $P^T Q' P = 0$  is the transformed quadric surface. [11]



Now we let the quadric surface in the object's frame is represented by (57). We require to obtain the quadric surface in the observer's frame.

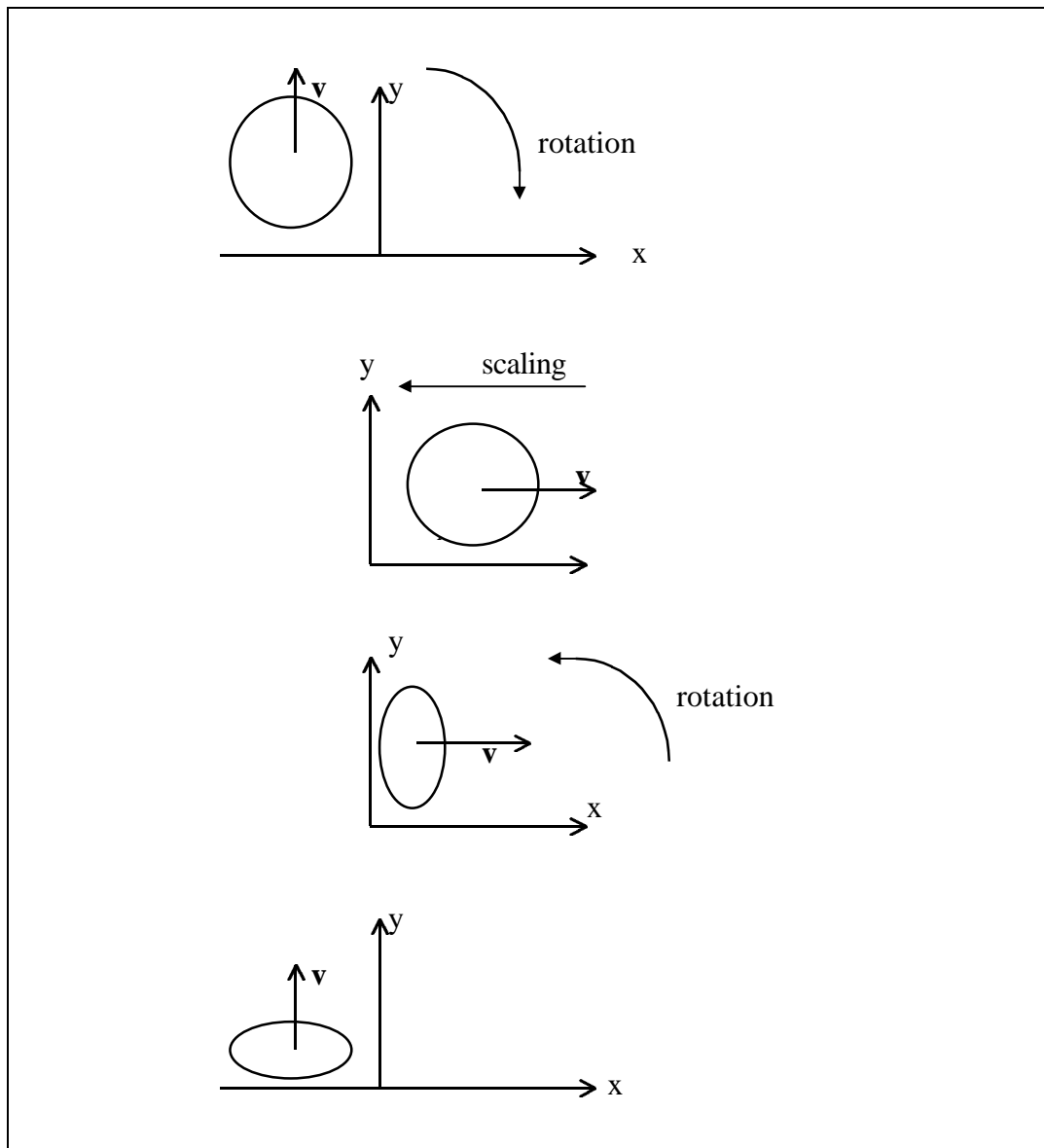


Figure 6 - Rotations of a quadric surface

To perform Lorentz transformation on the quadric surface, we need to rotate the surface so that the velocity vector now is parallel to the positive  $x$  direction, then perform scaling with a factor of  $\sqrt{1-\beta^2}$  in the  $x$ -direction, then rotate the scaled quadric surface back so that the velocity vector is in its original direction again. The transformations are similar to Section 5.2.3. Detailed description of the rotation transformations can be referred to there.

Let the velocity of the object be a vector  $\mathbf{v}$ , which has 3 components  $v_x$ ,  $v_y$ ,  $v_z$  parallel to the  $x$ -axis,  $y$ -axis and  $z$ -axis respectively. First, a rotation of angle  $-\mathbf{q}_z$  along the  $z$

axis is performed to bring the velocity vector to lie on the x-z plane. Then a rotation of angle  $\mathbf{q}_y$  along the y axis is performed to bring the velocity vector to lie on the x-axis.

The rotations can be represented as the matrix  $R_y R_z$  where:

$$\text{From (18): } R_z = \begin{bmatrix} \cos \mathbf{q}_z & \sin \mathbf{q}_z & 0 & 0 \\ -\sin \mathbf{q}_z & \cos \mathbf{q}_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ where } \tan \mathbf{q}_z = v_y/v_x$$

$$\text{From (19): } R_y = \begin{bmatrix} \cos \mathbf{q}_y & 0 & \sin \mathbf{q}_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \mathbf{q}_y & 0 & \cos \mathbf{q}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ where } \tan \mathbf{q}_y = \frac{v_z}{\sqrt{v_x^2 + v_y^2}}$$

Then we perform a scaling transformation S.

$$S = \begin{bmatrix} \sqrt{1-b^2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \dots(59)$$

$$\text{The inverse of S is } S^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \sqrt{1-b^2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \dots(60)$$

It is obvious that the transpose of  $S^{-1}$  is also  $S^{-1}$ .

Then the scaled quadric surface has to be rotated back by the matrix  $R_z R_y$ , where

$$\text{From (25): } R_y = \begin{bmatrix} \cos \mathbf{q}_y & 0 & -\sin \mathbf{q}_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin \mathbf{q}_y & 0 & \cos \mathbf{q}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{From (26): } R_z = \begin{bmatrix} \cos \mathbf{q}_z & -\sin \mathbf{q}_z & 0 & 0 \\ \sin \mathbf{q}_z & \cos \mathbf{q}_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It is obvious that the inverse of  $R_y$  is  $R_y$  and the inverse of  $R_z$  is  $R_z$ . Besides, the transpose of  $R_y$  is  $R_y$  and the transpose of  $R_z$  is  $R_z$ .

$$\text{The total transformation } M = R_z R_y S R_y R_z \quad \dots(61)$$

The inverse of  $M$  is  $M^{-1} = R_z R_y S^{-1} R_y R_z$  ... (62)

The transpose of inverse of  $M$  is  $(M^{-1})^T = R_z R_y S^{-1} R_y R_z$  ... (63)

i.e.  $(M^{-1})^T = M^{-1}$

From (58),  $Q' = (M^{-1})^T Q M^{-1} = M^{-1} Q M^{-1}$

i.e.  $Q' = R_z R_y S^{-1} R_y R_z Q R_z R_y S^{-1} R_y R_z$  ... (64)

so the quadric surface in the observer's frame is  $P^T Q' P = 0$  ... (65)

or  $f(x, y, z) = a'x^2 + b'y^2 + c'z^2 + 2d'xy + 2e'yz + 2f'xz + 2g'x + 2h'y + 2j'z + k' = 0$   
... (66)

## 6.4.2 Intersection with a moving quadric surface

For the quadric surface  $P^T Q' P = 0$  moving with velocity  $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ , similarly to the calculation of intersection with a moving plane in Section 6.3.2, we can find out the intersection with a moving quadric surface by the following:

From (66), replace  $x$  by  $(x - tv_x)$ ,  $y$  by  $(y - tv_y)$ , and  $z$  by  $(z - tv_z)$ , then it becomes the equation of a moving quadric surface:

$f(x, y, z) = a'(x - tv_x)^2 + b'(y - tv_y)^2 + c'(z - tv_z)^2 + 2d'(x - tv_x)(y - tv_y) + 2e'(y - tv_y)(z - tv_z) + 2f'(x - tv_x)(z - tv_z) + 2g'(x - tv_x) + 2h'(y - tv_y) + 2j'(z - tv_z) + k' = 0$  ... (67)

Since the ray is traveling in the negative time axis, so the equation (67) becomes

$f(x, y, z) = a'(x + tv_x)^2 + b'(y + tv_y)^2 + c'(z + tv_z)^2 + 2d'(x + tv_x)(y + tv_y) + 2e'(y + tv_y)(z + tv_z) + 2f'(x + tv_x)(z + tv_z) + 2g'(x + tv_x) + 2h'(y + tv_y) + 2j'(z + tv_z) + k' = 0$   
... (68)

Assume each point  $(x, y, z)$  along the tracing ray from a point  $(x_p, y_p, z_p)$  with the direction  $(\Delta x, \Delta y, \Delta z)$  where  $\Delta x^2 + \Delta y^2 + \Delta z^2 = 1$ . For a tracing ray traveling inside a medium with a speed of  $|\mathbf{T}|$ , then  $\Delta x^2 + \Delta y^2 + \Delta z^2 = |\mathbf{T}|^2$ .

Substitute  $x = x_p + t\Delta x$ ,  $y = y_p + t\Delta y$ ,  $z = z_p + t\Delta z$ , the equation (68) becomes:

$a'(x_p + tv_x + t\Delta x)^2 + b'(y_p + tv_y + t\Delta y)^2 + c'(z_p + tv_z + t\Delta z)^2 + 2d'(x_p + tv_x + t\Delta x)(y_p + tv_y + t\Delta y) + 2e'(y_p + tv_y + t\Delta y)(z_p + tv_z + t\Delta z) + 2f'(x_p + tv_x + t\Delta x)(z_p + tv_z + t\Delta z) + 2g'(x_p + tv_x + t\Delta x) + 2h'(y_p + tv_y + t\Delta y) + 2j'(z_p + tv_z + t\Delta z) + k' = 0$  ... (69)

which is a quadratic equation of  $t$ , thus the intersections can be solved.

## 6.5 Relativistic optics

The directions of incident rays, reflected rays and refracted rays and rays from light sources in the observer's frame are different from those in the object's frame. Besides, the Doppler's effect has to be considered. Equations (8), (9), (10), (11), (12), (17) in Section 4.3 are used for the transformation.

### 6.5.1 Transformation of rays from light sources

For a static light source, it is easy to find the direction of the ray from the light source to the intersection point (with the tracing ray) of the object in the observer's frame ( $\mathbf{S}'$ ). However, for a moving light source, since while the tracing ray is traveling, the light source is also moving, so it is not simple to find that.

Let  $\mathbf{S}_c'$  be the vector from the intersection point to the light source at the time of intersection,  $\mathbf{v}_{ls}$  be the velocity vector of the light source. Let the ray leaving from the light source to the intersection point at the intersection time be emitted at time  $t$  ago. Let  $\mathbf{S}'$  be the vector from the intersection point to the light source at the time of intersection minus time  $t$  (i.e. the time of emission of that ray).

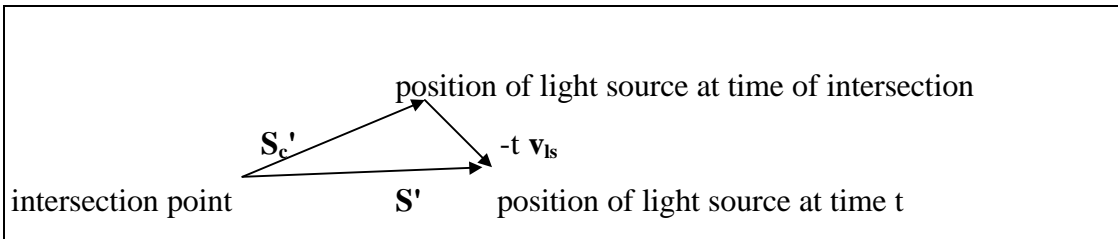


Figure 7 - A moving light source

$$|\mathbf{S}_c' - t \mathbf{v}_{ls}| = |\mathbf{S}_t'| = t \quad \dots(70)$$

$$(\mathbf{S}_c' - t \mathbf{v}_{ls}) \cdot (\mathbf{S}_c' - t \mathbf{v}_{ls}) = t^2$$

$$(|\mathbf{v}_{ls}|^2 - 1) t^2 - 2 \mathbf{S}_c' \cdot \mathbf{v}_{ls} + |\mathbf{S}_c'|^2 = 0 \quad \dots(71)$$

It is a quadratic equation. The discriminant  $= 4(\mathbf{S}_c' \cdot \mathbf{v}_{ls})^2 - 4 |\mathbf{S}_c'|^2 (|\mathbf{v}_{ls}|^2 - 1) > 0$  (because  $|\mathbf{v}_{ls}|^2 < 1$ )  $\dots(72)$

The smaller positive  $t$  should be used. Now  $\mathbf{S}'$  can be obtained by:

$$\mathbf{S}' = \mathbf{S}_c' - t \mathbf{v}_{ls} \quad \dots (73)$$

After getting  $\mathbf{S}'$ , which is in the observer's frame, we have to obtain  $\mathbf{S}$  or  $\mathbf{u}_S$ , which is in the object's frame. We can first find the angle  $\theta_S'$ , which is the angle between  $\mathbf{S}'$  and the velocity vector of the object. Then, we have to obtain the angle  $\theta_S$ , which is the angle between  $\mathbf{S}$  in the object's frame and the velocity vector of the object by formula (9):

$$\text{i.e. } \cos \theta_S = \frac{\mathbf{b} + \cos \theta_S'}{1 + \mathbf{b} \cos \theta_S'}$$

Since  $\mathbf{S}$ ,  $\mathbf{S}'$ ,  $\mathbf{v}$  are coplanar, so we can obtain  $\mathbf{u}_S$ , from  $\mathbf{u}_{S'}$  and  $\mathbf{u}_v$  ( $\mathbf{u}_S$  is the unit vector of  $\mathbf{S}$ ,  $\mathbf{u}_{S'}$  is the unit vector of  $\mathbf{S}'$ ,  $\mathbf{u}_v$  is the unit vector of  $\mathbf{v}$ ):

We first find the vector  $\mathbf{p}_S$  which is perpendicular to  $\mathbf{v}$  but coplanar with  $\mathbf{S}'$  and  $\mathbf{v}$ .

$$\mathbf{p}_S = (\mathbf{v} \times \mathbf{S}') \times \mathbf{v} \quad \dots(74)$$

The unit vector perpendicular to  $\mathbf{v}$  but coplanar with  $\mathbf{S}'$  as well as  $\mathbf{S}$  is  $\mathbf{u}_{PS} = \mathbf{p}_S / |\mathbf{p}_S|$   
 $\dots(75)$

With  $\theta_S$  and  $\mathbf{u}_{PS}$ , we can obtain  $\mathbf{u}_S$ ,

$$\mathbf{u}_S = \mathbf{u}_v \cos \theta_S + \mathbf{u}_{PS} \sin \theta_S \quad \dots(76)$$

## 6.5.2 Transformation of incident, reflected and refracted rays in vacuum

In the four-dimensional ray-tracing, after the intersection is found (in the observer's frame), the direction of the incident tracing ray in the observer's frame ( $\mathbf{I}'$ ) can be found easily (like the conventional ray-tracing). Then, however, from now on, we have to use the incident tracing ray in the object's frame ( $\mathbf{I}$ ) to calculate the reflected tracing ray ( $\mathbf{R}$ ) and refracted tracing ray ( $\mathbf{T}$ ) later because we have to use the normal ( $\mathbf{u}_n$ ) and the laws of reflection and refraction in the object frame.

This section deals with rays in vacuum, so the speed of rays is light speed in vacuum, which, as a convention, we use 1. Thus the magnitudes of rays in the observer's frame or the object's frame are also 1.

To get the incident tracing ray in the object frame ( $\mathbf{I}$ ), we can first find the angle  $\theta_i'$ , which is the angle between the incident tracing ray in the observer's frame ( $\mathbf{I}'$ ) and the velocity vector of the object. Then, we have to obtain the angle  $\theta_i$ , which is the angle between the incident tracing ray in the object's frame ( $\mathbf{I}$ ) and the velocity vector of the

object. We cannot use the formula (9) directly because it is for rays entering the object, however, the incident tracing ray is in fact leaving the object as the time goes in the positive direction. Thus, we have to consider that the time is now going to the negative direction and the object now is going to the opposite direction (use  $-\beta$  instead of  $\beta$ ) and the incident tracing ray is now entering the object, then we can use formula (9). As a result, giving the following equation:

$$\cos \theta_I = \frac{\cos \theta_I' - \mathbf{b}}{1 - \mathbf{b} \cos \theta_I'} \quad \dots(77)$$

Since  $\mathbf{I}$ ,  $\mathbf{I}'$ ,  $\mathbf{v}$  are coplanar, so we can obtain  $\mathbf{u}_I$ , from  $\mathbf{u}_{I'}$  and  $\mathbf{u}_v$  ( $\mathbf{u}_I$  is the unit vector of  $\mathbf{I}$ ,  $\mathbf{u}_{I'}$  is the unit vector of  $\mathbf{I}'$ ,  $\mathbf{u}_v$  is the unit vector of  $\mathbf{v}$ , to be precise, we use the rays' unit vectors instead of themselves although they are the same):

We first find the vector  $\mathbf{p}_I$  which is perpendicular to  $\mathbf{v}$  but coplanar with  $\mathbf{I}'$  and  $\mathbf{v}$ .

$$\mathbf{p}_I = (\mathbf{v} \times \mathbf{I}') \times \mathbf{v} \quad \dots(78)$$

The unit vector perpendicular to  $\mathbf{v}$  but coplanar with  $\mathbf{I}'$  as well as  $\mathbf{I}$  is  $\mathbf{u}_{PI} = \mathbf{p}_I / |\mathbf{p}_I|$

$$\dots(79)$$

With  $\theta_I$  and  $\mathbf{u}_{PI}$ , we can obtain  $\mathbf{u}_I$ ,

$$\mathbf{I} = \mathbf{u}_I = \mathbf{u}_v \cos \theta_I + \mathbf{u}_{PI} \sin \theta_I \dots(80)$$

Then now we can get the reflected ray  $\mathbf{R}$  by law of reflection and refracted ray  $\mathbf{T}$  by law of refraction in the object's frame. The magnitude of  $\mathbf{R}$  and  $\mathbf{T}$  are the reciprocals of the refractive indices of the media they are in respectively.

For recursive ray-tracing using the refracted tracing ray and refracted tracing ray, we have to get those in the observer's frame (i.e.  $\mathbf{R}'$ ,  $\mathbf{T}'$ ) because the calculations of intersections of the tracing rays with objects are performed in the observer's frame.

For the reflected ray or refracted ray in vacuum, change the subject of formula (9),

$$\cos \theta' = \frac{\cos \theta - \mathbf{b}}{1 - \mathbf{b} \cos \theta} \quad \dots(81)$$

Similar as the way above to obtain  $\mathbf{I}$ ,  $\mathbf{R}'$  or  $\mathbf{T}'$  can be obtained.

$$\mathbf{R}' = \mathbf{u}_{R'} = \mathbf{u}_v \cos \theta_{R'} + \mathbf{u}_{PR} \sin \theta_{R'} \quad \dots(82)$$

$$\mathbf{T}' = \mathbf{u}_{T'} = \mathbf{u}_v \cos \theta_{T'} + \mathbf{u}_{PT} \sin \theta_{T'} \quad \dots(83)$$

### 6.5.3 Transformation of incident, reflected and refracted rays in a moving medium

The above are only applicable to the rays in vacuum. In a medium of refractive index  $n$ , the magnitude of the ray in the object's frame is  $1/n$ .

Formula (17) is useful to find the speed of light in the object's frame in a medium moving in the direction parallel to the direction of light.

To generalize it, we combine formulae (17) and (12) (transformation of law for velocities).

For incident tracing ray  $\mathbf{I}'$  (in the observer's frame), the magnitude of  $\mathbf{I}'$  (speed of ray) can be known from the procedure who calls this recursively. The magnitude of component of  $\mathbf{I}'$  perpendicular to  $\mathbf{v}$  and coplanar with  $\mathbf{I}'$  and  $\mathbf{v}$  is  $\sqrt{|\mathbf{I}'|^2 - (\mathbf{I}' \cdot \mathbf{v})^2}$ . By (12), we can obtain  $\mathbf{I}$  in the object's frame as:

$$\mathbf{I} = \frac{1}{1 + \mathbf{I}' \cdot \mathbf{v}} ((\mathbf{I}' \cdot \mathbf{v} + |\mathbf{v}|)\mathbf{u}_v + \sqrt{1 - |\mathbf{v}|^2} \sqrt{|\mathbf{I}'|^2 - (\mathbf{I}' \cdot \mathbf{v})^2} \mathbf{u}_{pI}) \quad \dots(84)$$

where  $\mathbf{u}_{pI}$  is found as in Section 6.5.2.

For reflected tracing ray  $\mathbf{R}$  (or refracted tracing ray  $\mathbf{T}$ ) in the object's frame, its magnitude (speed of ray) should be  $1/n$  if  $n$  is the refractive index of the medium where it is. The magnitude of component perpendicular to  $\mathbf{v}$  and coplanar with  $\mathbf{v}$  and  $\mathbf{R}$  (or  $\mathbf{T}$ ) is  $\sqrt{1/n^2 - (\mathbf{R} \cdot \mathbf{v})^2}$  or  $\sqrt{1/n^2 - (\mathbf{T} \cdot \mathbf{v})^2}$ . By (12), we can find  $\mathbf{R}'$  or  $\mathbf{T}'$  in the observer's frame as:

$$\mathbf{R}' = \frac{1}{1 - \mathbf{R} \cdot \mathbf{v}} ((\mathbf{R} \cdot \mathbf{v} - |\mathbf{v}|)\mathbf{u}_v + \sqrt{1 - |\mathbf{v}|^2} \sqrt{1/n^2 - (\mathbf{R} \cdot \mathbf{v})^2} \mathbf{u}_{pR}) \quad \dots(85)$$

where  $\mathbf{u}_{pR}$  is found as in Section 6.5.2.

$$\mathbf{T}' = \frac{1}{1 - \mathbf{T} \cdot \mathbf{v}} ((\mathbf{T} \cdot \mathbf{v} - |\mathbf{v}|)\mathbf{u}_v + \sqrt{1 - |\mathbf{v}|^2} \sqrt{1/n^2 - (\mathbf{T} \cdot \mathbf{v})^2} \mathbf{u}_{pT}) \quad \dots(86)$$

where  $\mathbf{u}_{pT}$  is found as in Section 6.5.2.

## 6.5.4 Relativistic Doppler's effect

After transformation of the rays between the object's frame and the observer's frame, it also requires the shifting of the frequency of the light rays.

$$\text{From (8): } \quad \mathbf{n} = \frac{1 + \mathbf{b} \cos \mathbf{q}'}{\sqrt{1 - \mathbf{b}^2}} \mathbf{n}' \quad (\text{Doppler Effect in general case})$$

where  $\mathbf{n}$  is the frequency of light traveling when the observer in frame S measures;  $\mathbf{n}'$  is the frequency of light the observer in frame S' measures;  $\mathbf{q}$  is the angle of the light arriving when the observer in S measures;  $\mathbf{q}'$  is the angle of the light arriving when the observer in S' measures, i.e. it is for the transformation from frame S' to frame S.

$\cos \mathbf{q}'$  can be obtained by dividing the dot product of the velocity vector ( $\mathbf{v}$ ) and the ray vector (say,  $\mathbf{r}'$ ) in the observer's frame by the product of magnitudes of the velocity

$$\text{vector and that ray vector i.e. } \cos \theta' = \frac{\mathbf{r}' \cdot \mathbf{v}}{|\mathbf{r}'| |\mathbf{v}|} \quad \dots(87).$$

Thus, given the frequency of the light in the observer's frame ( $\gamma'$ ), we can use (8) to get the frequency of that light in the object's frame ( $\gamma$ ). On the other hand, given the frequency of the light in the object's frame ( $\gamma$ ), we can use (88) below to get the frequency of that light in the observer's frame ( $\gamma'$ ). Formula (88) is obtained by interchanging the frames S and S' in (8).

$$\mathbf{v}' = \frac{1 + \beta \cos \theta}{\sqrt{1 - \beta^2}} \mathbf{v} \quad \dots(88)$$

$\cos \mathbf{q}$  can be obtained by dividing the dot product of the velocity vector ( $\mathbf{v}$ ) and the ray vector (say,  $\mathbf{r}$ ) in the object's frame by the product of magnitudes of the velocity vector

$$\text{and that ray vector i.e. } \cos \theta = \frac{\mathbf{r} \cdot \mathbf{v}}{|\mathbf{r}| |\mathbf{v}|} \quad \dots(89).$$



## 6.6 Color model

In this four-dimensional ray-tracing, the  $r$ ,  $g$ ,  $b$  values are directly used and the diffusion-reflection coefficients ( $k_d$ ), specular-reflection coefficients ( $k_s$ ), transmission coefficients ( $k_t$ ) and shadow coefficients ( $S_i$ ), etc. are all based on the RGB values. However, the wavelengths or spectra of light rays are required in calculations in the spectral shifts (Doppler's effect). It is the problem how to relate both of them. To solve that, assumptions have to be made.

Here, it is assumed that the all colors, including the color of every pixel, every point on the object or the ray, are the combinations with different weights of the lights (called as three standard lights) of three wavelengths which are 605nm (red), 540nm (green), 445nm (blue), which are called as the three standard wavelengths (their corresponding frequencies are called as standard frequencies). The raw data of  $r$ ,  $g$ ,  $b$  values of each wavelength after the adjustment as described in Section 5.2.6 are called as intermediate data. The three wavelengths are chosen because their intermediate data give the largest  $r$ ,  $g$ ,  $b$  value (which are called as the standard  $r$ ,  $g$ ,  $b$  values) respectively. The ratios of  $r$ ,  $g$ ,  $b$  values of intermediate data of each wavelength to the standard  $r$ ,  $g$ ,  $b$  values will be used to represent the  $r$ ,  $g$ ,  $b$  values (called as normalized data) used in the ray-tracing. As a result, the  $r$  value of 605nm (red) wavelength is 1, the  $g$  value of 540nm (green) wavelength is 1, the  $b$  value of 445nm (blue) wavelength is 1. Given a set of  $r$ ,  $g$ ,  $b$  values (e.g. a reflected tracing ray), then the color represented is the combination of the three standard lights with weights of  $r$ ,  $g$ ,  $b$  respectively.

With the Doppler's effect described in Section 6.5.4, the three standard lights are shifted in wavelength (or frequency). Then the new  $r$ ,  $g$ ,  $b$  values will be obtained by the values resulted from the combination of the original  $r$ ,  $g$ ,  $b$  values and the standard data of the three new wavelengths after shifting. The details are given below.

To make things simpler, it was decided that:

1. the colors of the rays from the light sources, reflected and refracted rays are the combinations of the three standard lights in the observer's frame;
2. the actual amount the above rays contribute to the color of the intersection point are calculated with the coefficients (diffusion-reflection coefficients ( $k_d$ ), specular-

- reflection coefficients ( $k_s$ ), transmission coefficients ( $k_t$ ) and shadow coefficients ( $S_i$ ) of the object in the object's frame;
- the final color of that intersection point (i.e. the color of the incident tracing ray) will be represented as the combination of the three standard lights in the observer's frame.

The following gives the details.

For the rays from light sources  $S'$ , reflected rays  $R'$  and refracted rays  $T'$ , which are in fact going to the object as the time passes in the positive direction, the color ( $r'$ ,  $g'$ ,  $b'$ ) is given in the observer's frame. Thus, the three standard lights can be shifted in frequency using formula (8)

$$\text{From (8): } \quad \mathbf{n} = \frac{1 + \mathbf{b} \cos \mathbf{q}'}{\sqrt{1 - \mathbf{b}^2}} \mathbf{n}' \quad (\text{Doppler Effect in general case})$$

Let  $\nu_r'$ ,  $\nu_g'$ ,  $\nu_b'$  be the frequencies of the three standard light in the observer's frame, i.e. the standard frequencies (  $c/605$  GHz,  $c/540$  GHz,  $c/445$  GHz) (or let  $\lambda_r'$ ,  $\lambda_g'$ ,  $\lambda_b'$  be the wavelengths of the three standard light in the observer's frame, i.e. the standard wavelengths ( 605nm, 540nm, 445nm)). Then  $\nu_r$ ,  $\nu_g$ ,  $\nu_b$  (also  $\lambda_r$ ,  $\lambda_g$ ,  $\lambda_b$ ) can be obtained from (8).  $r_{\lambda_r}$ ,  $g_{\lambda_r}$ ,  $b_{\lambda_r}$ , which are the normalized  $r$ ,  $g$ ,  $b$  values of the wavelength  $\lambda_r$ , can be obtained, so for  $\lambda_g$  and  $\lambda_b$ . (Note that  $r_{\lambda_r}' = g_{\lambda_g}' = b_{\lambda_b}' = 1$ , others are zero.)

Now the color ( $r$ ,  $g$ ,  $b$ ) in the object's frame can be obtained by the following:

$$\begin{aligned} r &= r_{\lambda_r} r' + r_{\lambda_g} g' + r_{\lambda_b} b' \\ g &= g_{\lambda_r} r' + g_{\lambda_g} g' + g_{\lambda_b} b' \\ b &= b_{\lambda_r} r' + b_{\lambda_g} g' + b_{\lambda_b} b' \end{aligned} \quad \dots(90)$$

The ( $r$ ,  $g$ ,  $b$ ) can be used to multiplied with the corresponding coefficients of the object and the result can be added to the final color.

For incident ray  $I$ , which is in fact leaving from the object as the time passes in the positive direction, the color ( $r$ ,  $g$ ,  $b$ ) is given in the object's frame. Thus, the three standard lights can be shifted in frequency using formula (88)

From (88): 
$$v' = \frac{1 + \beta \cos \theta}{\sqrt{1 - \beta^2}} v \quad (\text{Doppler Effect in general case})$$

Let  $v_r, v_g, v_b$  be the frequencies of the three standard light in the object's frame, i.e. the standard frequencies (  $c/605$  GHz,  $c/540$  GHz,  $c/445$  GHz) (or let  $\lambda_r, \lambda_g, \lambda_b$  be the wavelengths of the three standard light in the observer's frame, i.e. the standard wavelengths (  $605\text{nm}, 540\text{nm}, 445\text{nm}$ )). Then  $v_r', v_g', v_b'$  (also  $\lambda_r', \lambda_g', \lambda_b'$ ) can be obtained from (87).  $r_{\lambda_r'}, g_{\lambda_r'}, b_{\lambda_r'}$ , which are the normalized  $r', g', b'$  values of the wavelength  $\lambda_r'$ , can be obtained, so for  $\lambda_g'$  and  $\lambda_b'$ . (Note that  $r_{\lambda_r} = g_{\lambda_g} = b_{\lambda_b} = 1$ , others are zero.)

Now the color ( $r', g', b'$ ) in the object's frame can be obtained by the following:

$$\begin{aligned} r' &= r_{\lambda_r'} r + r_{\lambda_g'} g + r_{\lambda_b'} b \\ g' &= g_{\lambda_r'} r + g_{\lambda_g'} g + g_{\lambda_b'} b \\ b' &= b_{\lambda_r'} r + b_{\lambda_g'} g + b_{\lambda_b'} b \end{aligned} \quad \dots(91)$$

The ( $r', g', b'$ ), which is the final color of the incident ray in the observer's frame from the intersection point, can be returned to the previous subroutine which calls it (produces this incident tracing ray).

## 6.7 Implementation

### 6.7.1 Product and platform

The source file of the completed program is "rt.c". The program was written in C++ with the OpenGL (Open Graphics Library) system which is a software interface to graphics hardware [14], and GLUT (OpenGL Utility Toolkit) which is a programming interface for writing window system independent OpenGL programs [15]. The hardware being used are SGI machines, including O2, Indy and Indigo. Besides, the preview for the real view was implemented using the approach described in Section 5.2.4.

### 6.7.2 Features

#### **Most relativistic effects were implemented**

Most of the relativistic effects (mentioned in Section 6) were implemented except the quadric surfaces, different media and moving light sources due to the lack of time.

#### **File handling**

Data of objects to be shown are not coded in the program but are inputted from data files so that different objects with different parameters can be processed. The user is allowed to select a data file to load the scene wanted.

#### **Two ray-tracing views and their previews**

Two ray-tracing views are provided, which are **ordinary (original) view** (where relativistic effects are not considered), **real view** (which uses perspective projection, considers relativistic effects and is the view which an observer will really see).

Their corresponding previews are also provided, which use the technique of OpenGL polygon shading. The preview for the **real view** is very similar to the program (vore.c) implemented mentioned in Section 5.2.4.

**Toggle of the option of spectral shift**

Spectral shift for the ray-tracing real view can be toggled to be used or not. If the option of spectral shift is off, then the objects in the real view will be shown in their original colors and can be recognized more easily because in most of the time the real colors of the objects are black due to the spectral shift. Moreover, the geometric shape of the objects in real view can be seen more easily.

**Changing of direction of view.**

The user is allowed to change the direction of view easily at any time.

**Changing of size of view plane.**

The user is allowed to change the size of the view plane easily at any time by dragging the border of the window.

**Changing of time**

The user is allowed to change the time in the world being simulated, i.e. the time can be set so that how the objects look like at a particular time can be seen.

### 6.7.3 Sample results

Some sample results (Figure C9 to Figure C30) are given in the report. The figures were generated in O2 machines, saved as bitmaps, and printed by a color ink jet printer using a PC. The quality of the figures printed degrades, compared with those shown on the screen. The generation of an image of real view of a cube takes about 1.5 minute.

Figure C9 to Figure C20 show the ordinary view, the real view (without spectral shift effect on the light sources) and the real view (with spectral shift effect on the light sources) of a cube moving to the positive x-axis (the right hand side) at a speed of 58% of light speed with time = 0, 14, 5, 2 sec. The observer is at the origin. The angle between the negative z-axis (left 0) and the image of the object is represented as left X where X is the degree of the angle measured anti-clockwise.

Figure C21, C22 show the large distortion in the geometric shape of a cube moving at speed of 98% of light speed.

Figure C23 to C26 show the spectral shift of white color long plane moving at speed of 60% of light speed with a white light source (without spectral shift effect on the light sources).

Figure C27 to C30 show the spectral shift of white color long plane moving at speed of 60% of light speed with a white light source (with spectral shift effect on the light sources).

Discussion on the sample results are in Section 7.

## 6.8 Modification on the object editing programs

Since many properties/attributes of the object are needed by the ray tracing, so a new data structure and a file format for storing the data of objects have to be designed. As a result, some parts of the Object Edit program and Object Maindata Preview program (whose source codes are "oe.c" and "op.c") were modified. The resultant programs are the Object Edit program 2 and Object Maindata (Scene) Preview program 2 (whose source codes are "oe2.c" and "op2.c"). They were written in C++ with the OpenGL (Open Graphics Library) system which is a software interface to graphics hardware [14], and GLUT (OpenGL Utility Toolkit) which is a programming interface for writing window system independent OpenGL programs [15]. The hardware being used is SGI machines. The programs give the user a more convenient intuitive and easier way to edit objects that are to be loaded by the program rt.c.

**Object Edit program 2** is used for editing an object used by the rt.c. This program lets the user edit an object in an intuitive way where the user can move the vertex, create or delete a vertex or a triangular plane, move the whole object, set the coefficients of the surfaces, refractive index, create a sphere and set its position and radius, set the velocity of the object, load or save the object from or to a file, etc.

The data structure is similar to the one in rt.c. Data of an object are loaded from a data file of the object, and a data file storing RGB values of wavelengths in the visible spectrum. Data of an object include its velocity, coordinates of each vertex, vertices

that each (triangular) plane consists of, coefficients of each plane (or the coordinates of the center and radius of a sphere if it is a sphere).

The window 1 (figure C31) is the same as the window 1 of *oe.c* (figure C5) (please refer to Section 5.4.2.1), in the appearance and functions. However, *oe2.c* has four palette windows (figure C32) which are used to adjust the diffusion-reflection coefficients ( $k_d$ ), specular-reflection coefficients ( $k_s$ ), transmission coefficients ( $k_t$ ) and shadow coefficients ( $S_i$ ). It also has a window to set the direction and the magnitude of the velocity of the whole object (figure C33) and a window for the user to set the refractive index and the specular reflection exponent of the surface (or each plane) (figure C34).

**Object Maindata Preview program 2** is used for editing an object used by the *rt.c*. This program lets the user edit a maindata file, set up the light sources and preview the ordinary view of the objects described in the maindata file. The user can load and save a maindata file, load objects into the maindata file, play the animation, rotate the view, etc.

The data structure is similar to the one in *rt.c*. Data of objects moving in certain velocities (not necessary to be the same) are loaded from a main data file selected by the user, data files of each object, and data file storing RGB values of wavelengths in the visible spectrum.

The window 1 (figure C35) is the same as the window 1 of *op.c* (figure C8) (please refer to Section 5.4.2.2), in the appearance and functions. However, *op2.c* allows the user to set up multiple light sources. The user can do things to the light sources like what they can do in *oe2.c* except creating a plane. After a light source is selected, the user can set its color by the palette window (figure C36). If no light source is selected, the user can set the ambient light by the same palette window.

## 7. DISCUSSION OF SAMPLE RESULTS

Figure C1 to figure C4 are the sample results of the approach of hyperbolic transformation and polygon shading.

Figure C1 shows a cube in the ordinary view and figure C2 shows the same cube in the real view. The cube is distorted in shape and color. As the cube is moving to the right, it can be seen that the color of its one face changes from blue (left) to red (right). It is because the left part is approaching us and so there is a "blue shift" while the right part is leaving from us and so there is a "red shift". Apart from the color, the vertical edges are found to bend so that the middle part is to the right more and the upper part and the lower part are to the left more. This is because the light ray leaving the upper part (or lower part) at a particular time and the light ray leaving the middle part at the time a bit later arrive at the observer at the same time.

Figure C3 shows a sphere with different colors on its surface in the ordinary view and figure C4 shows the same sphere in the real view. The sphere is not distorted in shape, which was proved in some past papers [10]. However, there is still changes in the color. The explanation is the same as in the example of a cube above.

Figure C9 to figure C30 are some sample results using the approach of four-dimensional ray-tracing.

Figure C9 to figure C20 show the ordinary view, the real view (without spectral shift effect on the light sources) and the real view (with spectral shift effect on the light sources) of a cube moving to the positive x-axis (the right hand side) at a speed of 58% of light speed with time = 0, 14, 5, 2 sec. The observer is at the origin. The angle between the negative z-axis (left 0) and the image of the object is represented as left X where X is the degree of the angle measured anti-clockwise.

Figure C9 to C11 show the cube at time zero sec. It is obvious that when the cube is just passing us in the ordinary view, the observer will see that the cube is still far away and coming to the observer (so it is extremely blue shifted).

Figure C12 to C14 show the color of the cube shifted to become red at time 14 sec. It is shifted to red because it is not leaving from the observer.

Figure C15 to C17 show the almost zero spectral shift effect on the color of the cube at time 5 sec. It is interesting that when there is almost zero spectral shift effect, the



cube is about "left 20" degrees to the observer, i.e., it is still coming to the observer. The explanation is that it is the result of the combination of the effect of aberration and relativistic Doppler's effect.

Figure C18 to C20 show the color of the cube shifted to become blue at time 2 sec as it is coming to the observer.

Figure C21, C22 show the large distortion in the geometric shape of a cube moving at speed of 98% of light speed. The explanation is the same as above explaining figure 2. Besides, it can be noted that the face of the cube can be seen which ordinarily can't be seen. It is known as "rotational effect", which is due to the limited speed of light, similar to the distortion of the shape. [4][6][10]

Figure C23 to C26 show the spectral shift of white color long plane moving at speed of 60% of light speed with a white light source (without spectral shift effect on the light sources). Generally, it can be seen that there is the change from blue-shift to red-shift as from left to right.

Figure C27 to C30 show the spectral shift of white color long plane moving at speed of 60% of light speed with a white light source (with spectral shift effect on the light sources). Generally, it can also be seen that there is the change from blue-shift to red-shift as from left to right. However, only the part around the middle (left zero degree, just passing by the observer) has greater intensity. This is because the light source is about right behind the observer and if there is spectral shift effect on the light source, then for the left part (coming to the observer) or the right part (leaving from the observer) the light rays coming to them are also shifted to blue or red, and probably shifted to the invisible spectrum. As a result those parts seem to be dark or even black.

The sample results match with theoretical results well.

## 8. CONCLUSION

This final year project deals with the application of technique of computer graphics to the implementation for the visualization of objects under relativistic effects according to the special theory of relativity. Two approaches were used, which are "hyperbolic transformation and polygon shading" and "four-dimensional ray-tracing". Since only very basic formulae were available, some more general ones were needed. The necessary formulae and algorithms were derived and implemented. The results match well with expected theoretical results.

The extension of this project is to cover the general theory of relativity, which is valuable but also much more difficult and time-consuming. Efforts have been made by some researchers. In special relativity, light rays can still be considered as straight lines in flat space-time. This is different in general relativity, and therefore the paths of all photons reaching an observer will have to be integrated explicitly, by using the geodesic equation:  $\ddot{x}^m + \Gamma_{nl}^m \dot{x}^n \dot{x}^l = 0$  ... (92) The resulting image is therefore determined not only by the object itself, but also by the space-time surrounding this object. [1]

## 9. APPENDICES

### 9.1 Table of chromaticity coordinates and color-matching functions for RGB system [13]

Table 5 - Chromaticity coordinates and color-matching functions for RGB system

Table 5 - Chromaticity coordinates and color-matching functions for RGB system  
(continued)

## 9.2 Graph of color-matching functions for RGB system

[13]

Figure 8 - Graph of color-matching functions for RGB system

## 9.3 Explanation of some terms in color science [13]

<b>Terms</b>	<b>Explanation</b>
Hue	the attribute of a color perception denoted by blue, green, yellow, red, purple, and so on
Saturation	the attribute of a color perception determining the degree of its difference from the achromatic color perception most resembling it
Primary colors	the colors of three reference lights by whose additive mixture nearly all other colors may be produced
Tristimulus values	tristimulus values of a color are the amounts of the three reference lights (matching stimuli, primary colors) required to give by additive mixture a match with the color considered
Color-matching functions	the tristimulus values, with respect to three given primary colors, of monochromatic lights of equal radiant energy, regarded as functions of the wavelength
Chromaticity coordinates	chromaticity coordinates of a color are the ratios of each tristimulus value of the color to their sum

Table 6 - Explanation of some terms in color science

## 9.4 Progress (1/9/1997 - 27/3/1998)

Date	Work or event
15/9/1997- 10/10/1997	Project planning and project plan writing; Project proposal writing
13/10/1997	Submission of project plan and project proposal
1/9/1997- 13/10/1997	Study of theories of relativity
13/10/1997	Meeting with the supervisor and Submission of project plan (plan1.doc, plan1.mpp) and project proposal (proposal.doc)
1/10/1997- 27/11/1997	Study of OpenGL and GLUT programming; creating of project homepage ( <a href="http://www.cs.hku.hk/~ehung/vore">http://www.cs.hku.hk/~ehung/vore</a> )
31/10/1997	Meeting with the supervisor and Feedback from the supervisor about the project plan and project proposal
1/11/1997- 27/11/1997	Design of algorithm of the approach of hyperbolic transformation and polygon shading
28/11/1997	Meeting with the supervisor, Discussion of progress and design of the approach of hyperbolic transformation and polygon shading, Confirmation of date of demonstration
29/11/1997 - 18/12/1997	Implementation of the program (vore.c) using the approach of hyperbolic transformation and polygon shading; Study of color science and models
19/12/1997	Demonstration of the Program (vore.c) to the supervisor and Requirement from the supervisor about the GUI for 3D object editing.
20/12/1997 -2/1/1998	Design and implementation of 3D object editing programs using GUI (oe.c, op.c)
27/12/1997 -11/1/1998	Writing of progress report
12/1/1998	Submission of progress report to the supervisor; Meeting with the supervisor and discussion of the Mid-term Review
14/1/1998	Submission of progress report to the second examiner

15/1/1998	Mid-term Review with the second examiner; Demonstration of the programs (vore.c, oe.c, op.c) and discussion of the project.
3/1/1998 - 2/2/1998	Study of relativistic optics and Design of the algorithm of the approach of Four Dimensional Ray Tracing
3/2/1998 - 2/3/1998	Implementation of the program (rt.c) using the approach of Four Dimensional Ray Tracing; Modification of 3D object editing programs using GUI (oe2.c, op2.c)
12/3/1998	Demonstration of the programs (rt.c, oe2.c, op2.c) to the supervisor
27/2/1998 - 15/3/1998	Writing of final report and technical paper
16/3/1998	Submission of final report and technical paper
27/3/1998	Presentation

Table 7 - Progress

## 10. REFERENCES

Most reference relevant to this project and referred in this report are from the books or papers listed as below.

The seventh to the tenth items are in Chinese (or translated into Chinese) while the others are in English. Item 16 was found in Internet.

1. Nollert H. P., Kraus U., Ruder H., "Visualization of Objects via Four-Dimensional Ray-Tracing", Institute for Astronomy and Astrophysics, University of Tubingen, Germany. This article is from Hekl F. W., Puntigam R. A., Ruder H., "Relativity and Scientific Computing". Springer, 1996, p. 314 - 329. In page 2,3 of the same book, 2 pictures related can be found.
2. Skinner R., "Relativity for Scientists and Engineers", Dover Publications, New York, 1982, p. 1-72, 146.
3. Einstein A., "Relativity - The Special and General Theories", 1952.
4. Mook, Vargish, "Inside Relativity", Princeton University Press, 1987, p. 117-136
5. Rindler, "Essential Relativity", p. 54-60
6. Scott G. D., Viner M. R., "The Geometrical Appearance of Large Objects Moving at Relativistic Speed", American Journal of Physics 33, 1965, pp. 534-536.

7. S. D. J. (1978): Wu D. Y., "Theoretical Physics (Vol. 4) - Theories of Relativity", 1978.
8. C. Y. C., L. S. K. (1990): Chin C. Y., Kwong L. S., "Relativity & Gravitation", Asia Books Company, 1990.
9. Y. J. W. [(1996): Wang Y. J., "Space, Time and Gravitation", 1996, Newton.
10. U. A. A. (1984): A. E. E. I. S. I. U. 1/2 x x A. E. E. E. E.
11. Foley, van Dam, Feiner, Hughes, "Computer Graphics: Principles and Practice", Addison Wesley, 2/e in C, 1996, p.574-585, 726
12. Hall R., "Illumination and Color in Computer Generated Imagery", 2/e, Springer-Verlag, New York, 1981, p.47-51.
13. Wyszeccki G., Stiles, W. S., "Color Science: Concepts and Methods, Quantitative Data and Formulae", 2/e, John Wiley & Sons, New York, 1982, p. 230-234, 263-274.
14. Woo M., Neider J., Davis T., "OpenGL Programming Guide", 2/e, Addison Wesley.
15. Kilgard M. J., "The OpenGL Utility Toolkit (GLUT) Programming Interface", API Version 3, Silicon Graphics, Nov 13, 1996.
16. Howard A., Kitchen L., Dance S., "Relativistic ray-tracing: simulating the visual appearance of rapidly moving objects", 13 August 1995. (<http://munkora.cs.mu.oz.au/~andrbh/raytrace/raytrace.html>)
17. Ditchburn R. W., "Lights", Academic Press, London, 1976, p. 402-421.