

# **SIX DEGREE OF FREEDOM ON ONE BUTTON – A 3D MOUSE INTERFACE**

***László Dudás***

PhD, Associate Professor  
*University of Miskolc*

## **ABSTRACT**

This paper presents a comfortable and easy to use mouse interface (named ZOOMBALL) for manipulating 3 dimensional virtual objects in the space. This interface applies one mouse button only for handling the six degrees of freedom: for rotating around three axes, for translating in two directions and for zooming. This maximum capability may be reached in full screen mode. In windowed mode the translations need the middle mouse button or the mouse wheels or a double click for changing between translation and other modes. The main advantages are in handling rotations and zooming in a new integrated, natural way. The suggested geometrical model and the realized C function proved to be very handy and powerful 3D space manipulation interface which can particularly be applied for CAD and animation programs.

*Key words:* 3D transformations, user interface, mouse.

## **THE APPLIED METHODS FOR TRANSFORMING 3D SPACE**

The keyboard methods are not in the scope of our interest now. There are special mouse constructions to handle more than two degrees of freedom like Shakin' Mouse [1.], Cubic Mouse [2.] and 6DOF input device [3.]. These positioning devices and the 3D trackball, 6D Spaceball and the 6D Polhemus are outside of our interests too. We would like to manipulate the 3D object by a general mouse because they have widely been spread. The known user interfaces of CAD programs associate the six degrees of freedom to selected controls or parts of the screen, or use mode changes with buttons on the mouse or visual buttons on the screen, or select the rotation axis or translation direction clicking a line in the 3D space, etc. Other applications use sliders for realization of movements. The most elegant methods apply dragging of objects with mouse. Sometimes the displayed objects are embedded in a spherical space and the user can rotate the space by dragging one of its point. This method is the modeling of the physical 3D trackball that has a freely rotate ball in its base. This leader technique has more inventors and slightly different realizations: The first reference mentions Evans et al. from 1981 [4.]. The most frequently cited is Chen's Virtual Sphere. Chen suggests his rotation interface with the next words:

“The Virtual Sphere is unusual in the sense that you seem to be able to squeeze an extra degree of freedom out of a 2-D input device. The action of rolling the Virtual Sphere lets you specify an arbitrary rotation axis in three-dimensional space, with

the advantage that you don't need to think about the rotation axis. You just roll, and the object turns in the expected way." [5.]

The third realization of virtual trackball was carried out by Ken Shoemake in 1992 and known as ARCBALL. [6.] He used a half-arc length rotation on the imaginary sphere to avoid hysteresis that featured the Virtual Sphere. For explanation of his model he used a more mathematically rigorous quaternion implementation with complex arithmetic.

In the mentioned best rotational models the translation and zooming was outside of interest of inventors. So zooming and translation need additional techniques.

## THE SET GOAL AND THE REALISATION

The goal was to develop a comfortable, natural 3D mouse interface. Among the targets was the minimum need for mouse manipulations, clicks or double clicks. Moreover I wanted to leave free the right mouse button for dropping down the popup menus of clicked objects. Extra visual objects for selecting or manipulating degrees of freedom were not desirable because they reduce the useful area of the display and disturb the view of drawn objects. Only natural workspace - mouse action associations were acceptable. The most natural object manipulation method is the seizing and moving. So the basis was the spherical space rotation. The question was: how to integrate the zooming and translations in a natural way?

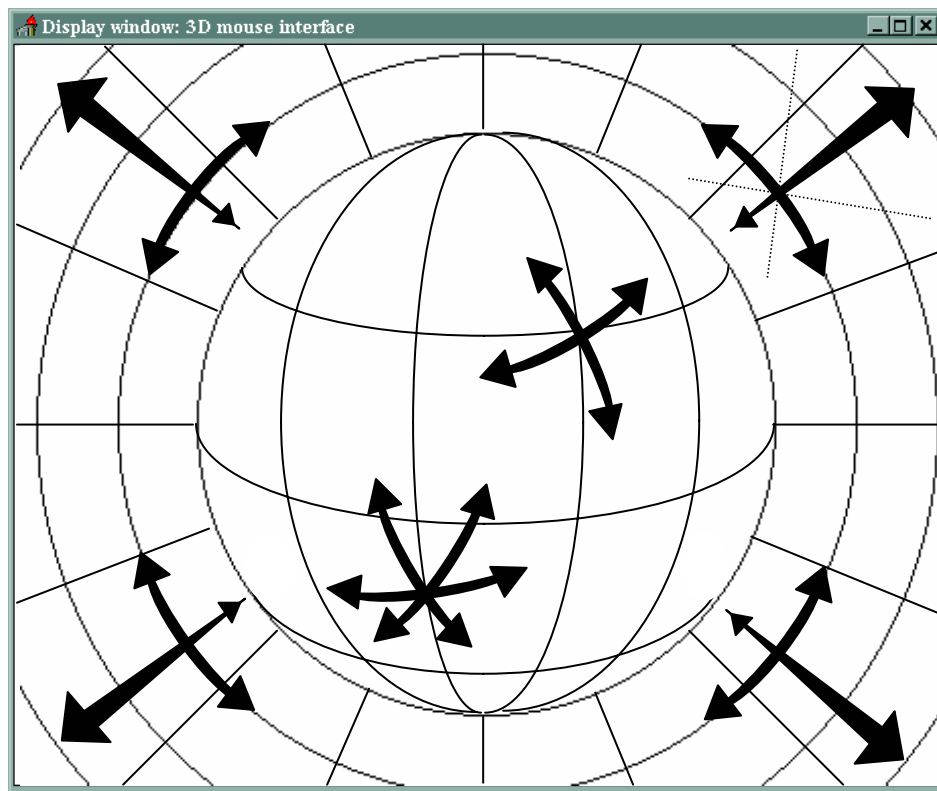


Fig. 1.  
The association of transformations and display workspace

Fig. 1. shows the suggested formation of the workspace association to rotations and zooming. Using full screen displaying this buildup is good for translation too without any plus modification or need for button clicking. In full screen mode an automatic translation occurs when the cursor knocks against the edge of the display area. This method is well known for virtual screens that bigger than the visible area of the physical screen. This effect has to be produced by the transformation software of course detecting the extremal points of the screen.

The figure shows the expedient association: most of the display area is connected to rotations. This central part corresponds to a half sphere and can handle the rotations around every axle. The working is similar to a track ball: the user can seize and move every point of the sphere and the displayed 3D objects in the screen will turn with the sphere. The axis of rotation is determined by the starting point and the direction of the elementary mouse step. The angle of rotation is given by the spherical angle between the projected starting and end points of the mouse step. Because of the lots of elementary small mouse steps the rotational movement will be very smooth, even. There is a continuous transition between spherical and plane area of the screen because the plane area is used for rotating around the axis of the sphere perpendicular to the plane. This technique was first applied in Virtual Sphere and ARCBALL.

The figure shows that the plane area has a second function in my model. This is the zooming. If the direction of the mouse movement is radial, the resulted transformation will be magnification or demagnification depending on the outbound or oncoming direction of the mouse movement. There is a possible algorithm of zooming to continue the demagnification inside the area of the sphere too if the starting point of the mouse movement was in the plane area. To avoid the simultaneous zooming and rotation in the plane area may be apply the angle range filter: if the direction of movement is closer to radial direction than tangential direction the zooming is working and if the direction is closer to tangential direction than the rotation is preferred. The dotted lines in the upper right corner of the figure show the angle regions for zooming or for rotation.

Figure 2. on the next page shows the technique of mapping the central area of the screen to the half sphere. The  $P0_s$ - $P1_s$  vector is an elementary mouse movement on the screen. The  $P0_s$  starting point is projected to the sphere. This  $P0$  point will be the contact point between the sphere and a tangential plane. Projecting the  $P1_s$  point to this plane the  $P1$  point will be determined. The  $P0$ - $P1$  vector will be the vector of the starting velocity. This method is different from methods applied in Virtual Sphere and ARCBALL models. The vector product of the  $P0$ - $P1$  vector and the  $C$ - $P0$  vector will give the axis of rotation. The angle of rotation is given by  $\varphi$  calculated between  $C$ - $P0$  and  $C$ - $P1_b$  vectors. The  $P1_b$  point is the projection of the  $P1_s$  point to the semi-sphere. This model applies different projected points for calculation of the axis and the angle of rotation. The model was programmed in C++ environment and the tests gave excellent moving. What can we tell about problems mentioned in [7] in connection to Virtual Sphere and ARCBALL. In this model the hysteresis may be similar to Chen's model because the two models are close to each other. The Virtual Sphere interprets the projection of mouse step vector as the direction of tangential vector of the middle point of the arc, this model

interprets as the speed direction vector of the starting point on the sphere. Smaller mouse steps results less difference between the motion of the two models.

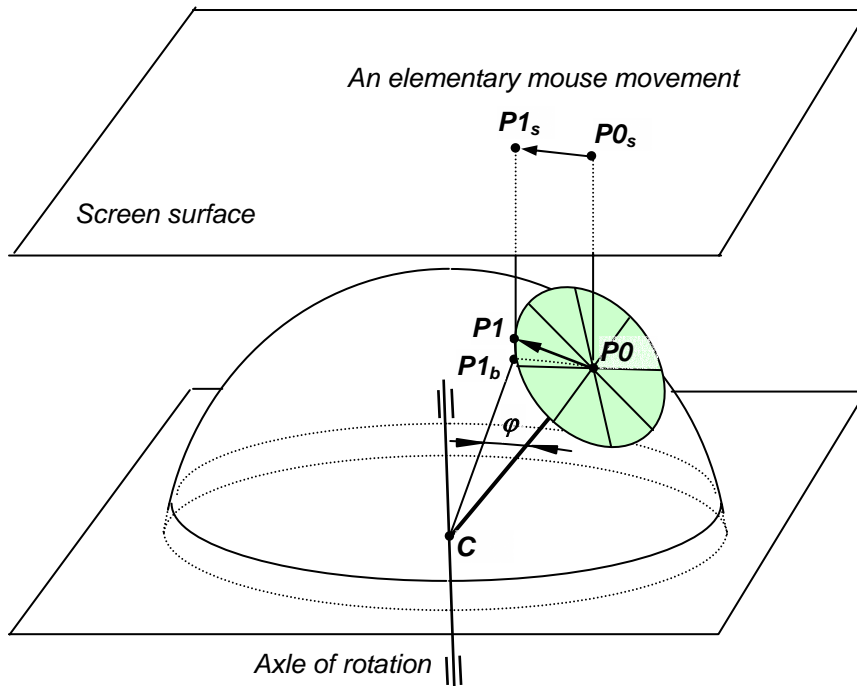


Fig.2.  
Interpretation of the axis and angle of rotation

## CALCULATING DATA FOR TRANSFORMATIONS

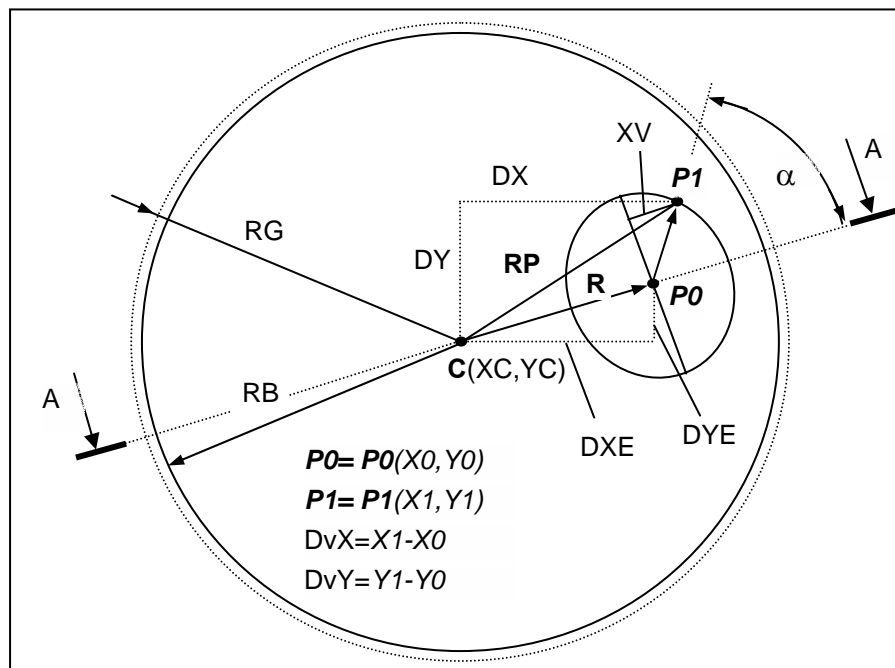


Fig. 3.  
Sizes are given on the plane of the rectangle

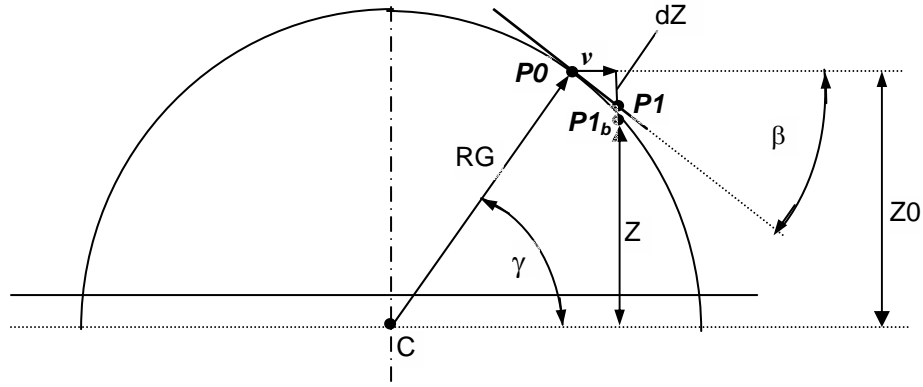


Fig.4.  
A-A section

The maximum rotation is close to 180 degree when the mouse goes across diagonally the sphere. The missing angle is about 16 degree and comes from the fact that the applied part of the sphere is a little smaller then a semi-sphere. This modification is applied to avoid overflow when projecting the points to the plane.

Expressions of calculation:

$$RB = (1-\varepsilon) \cdot RG, \dots \dots \dots 0 < \varepsilon < 0.02 \quad (1.)$$

$$DXE = X0 - XC; DYE = Y0 - YC; R^2 = DXE^2 + DYE^2; \quad (2.)$$

$$Z0 = \sqrt{RG^2 - R^2}; \quad (3.)$$

$$DX = X1 - XC; DY = Y1 - YC; RP^2 = DX^2 + DY^2; \quad (4.)$$

$$Z = \sqrt{RG^2 - RP^2}; \quad (5.)$$

$$\text{Angle\_of\_rot} = \arccos( (DXE \cdot DX + DYE \cdot DY + Z0 \cdot Z) / (RG \cdot RG) ); \quad (6.)$$

$$\text{ReX} = DXE / R; \text{ReY} = DYE / R; \text{XV} = \text{ReX} \cdot DvX + \text{ReY} \cdot DvY; \quad (7.)$$

$$\beta = \arcsin(R / RG); dZ = XV \cdot \tan(\beta); \quad (8.)$$

The components of the rotation-axis vector **ra**: (9.)

$$\mathbf{ra}_x = -DYE \cdot dZ - DvY \cdot Z0; \mathbf{ra}_y = DvX \cdot Z0 + DXE \cdot dZ; \mathbf{ra}_z = DXE \cdot DvY - DvX \cdot DYE;$$

The rotation has to be carried out around the **ra** vector by Angle\_of\_rot degree.

The rotation around Z axis perpendicular to the plane:

$$YV = \sqrt{DvX^2 + DvY^2 - XV^2}; \quad (10.)$$

The decision between rotation or zooming:

If ( $\text{abs}(XV) > \text{abs}(YV)$ ) then Zoom\_based\_on\_XV  
else Rotation.

The rotation angle can be calculated between the C-P0 vector and the C-P1 vector.

## CONCLUSION

In the paper was proposed a comfortable 3D mouse interface which uses only one mouse button for handling six degrees of freedom in full screen mode or can handle the x,y,z rotations and the zooming using dragging with one mouse button and needs an extra button or key for changing between the rotation and integrated zoom or the translation operations.

## ACKNOWLEDGEMENTS

This work was made within the scope of N°1221/1997. project of Higher Educational Research and Development Competition supported by the Hungarian Ministry of Culture and Education.

This work was moreover supported by the National Scientific Research Foundation, Grant N° T 025542.

## REFERENCES

- [1] Balakrishnan, R., Baudel, T., Kurtenbach, G., Fitzmaurice, G.: **The Rockin' Mouse: Integral 3D Manipulation on a Plane** [http://reality.sgi.com/gordo\\_tor/papers/rockmouse/rbl.htm](http://reality.sgi.com/gordo_tor/papers/rockmouse/rbl.htm)
- [2] Froehlich, B., Plate, J.: **The Cubic Mouse, A New Device for Three-Dimensional Input**. Proceedings of CHI. 2000. ACM
- [3] Hinckley, K., Tullio, P., Pausch, R., Profitt, D., Kassel, N.: **Usability Analysis of 3D Rotation Techniques**. Proceedings of UIST'97. 1997. pp.1-10.
- [4] Chen, M., Mountford, S.J., Sellen, A.: **A Study on Intractive 3-D Rotation Using 2-D Control Devices**. Proceedings of SIGGRAPH'88. 1988. ACM. pp. 121-129.
- [5] Chen, M.: **3-D Rotation Using a 2-D input device**. June 93.  
[http://www.machtech.com/articles/develop/issue\\_14/Chen\\_final.html](http://www.machtech.com/articles/develop/issue_14/Chen_final.html)
- [6] Shoemake, K.: **ARCBALL: a user interface for specifying three dimensional orientation using a mouse**. Proceedings of Graphics Interface'92. 1992. pp.151-156.
- [7] Hinckley, K.: **Usability Analysis of 3D Rotation Techniques** Chapter 6. 1996.  
<http://www.research.microsoft.com/Users/kenh/thesis/rotation.htm>