Pergamon

Comput. & Graphics, Vol. 18, No. 5, pp. 691–694, 1994
Copyright © 1994 Elsevier Science Ltd
Printed in Great Britain
0097-8493/94 $6.00 + .00

0097-8493(94)00080-8

Short Technical Notes/Tutorials/Systems

ON THE CORRECT DETERMINATION OF ROTATIONAL ANGLES FOR TWISTED-PROFILED SWEEP OBJECTS

MARKO MARHL
Faculty of Education, University of Maribor, Koroska cesta 160, 62000 Maribor, Slovenia,
E-mail: marko.marhl@uni-mb.si

and

VAROL AKMAN
Department of Computer Engineering and Information Science, Faculty of Engineering, Bilkent
University, Bilkent, 06533 Ankara, Turkey

Abstract—In the context of the paper of V. Akman and A. Arslan, "Sweeping with All Graphical Ingredients in a Topological Picturebook" [*Computers & Graphics*, Vol. 16(3), pp. 273–281, 1992], a new construction of the rotational matrix is presented. This fixes a bug discovered by the first author.

1. INTRODUCTION

In the article [1], a general matrix for representing twisted-profiled sweep objects is presented. The matrix has the following form:

$$[S_{w_{tp}}] = [R_{tw}][S_p][R_x][R_y][R_z][T_{xyz}], \qquad (1)$$

where $[R_{tw}]$ represents the rotation of a contour curve C that is centered at the origin, $[S_p]$ represents the scaling of C, $[R] = [R_x][R_y][R_z]$ represents the rotation of C in 3-D with respect to the tangent vector of a trajectory curve T, and $[T_{xyz}]$ represents the translation of C to each point of T.

Sweep objects are then defined as follows:

$$[\textit{Sweep surface}] = [C][S_{w_{tp}}]. \qquad (2)$$

Our purpose in this paper is not to repeat this useful idea for the construction and therefore the representation of sweep objects. It is known that it solves the well-known problem of normal determination at the "Frenet frame"[2–7] and the "Offset"[8] constructions. We want just to supplement this work by introducing a technical improvement. During subsequent research on this topic by the first author it was found that the determination of the rotational angles is not correctly defined and therefore the rotational matrix $[R_x][R_y][R_z]$ is in general incorrect. The present technical note aims to fix this problem and is essentially a slightly revised version of a manuscript written by the first author. It should be remarked that in the original implementation (**Tb**) in which the incorrect formulation was used[1], this problem did not make itself evident. This was first thought to be surprising. However, later we realized that **Tb** generally avoids matrix algebra (in order to be more efficient) and the non-matrix formulation used by the program was able to bypass this bug.

2. CONSTRUCTION OF THE NEW MATRIX

We have decided to construct a well-defined rotation based only on two successive rotations of the contour curve. We were inspired for such a step from the work of J. P. Fillmore[9]: "The connection between three-by-three orthogonal matrices and the rotations of space that they describe is quite misleading when trying to describe a rotation of space by angles of rotation about the three coordinate axes." In his article, the equivalent rotation about only one axis is presented. But this is not appropriate for our problem because the corresponding axis and the angle of rotation are not so easily obtained. On the other hand, it is true that in many cases the rotation has to be described by three successive rotations about the three coordinate axes, *e.g.*, practical applications in robotics[10]. But as the problem of decomposing any 3-D orthogonal matrices into primitive rotations represents no problem[11], we have decided to work out the "two-axes" rotation. Our solution is understandable (hence, the possibility of misunderstanding is minimized[9]) and is, without loss of generality[11], also relatively easy to define.

It can be proved by an example that the original definition of the rotational angles in [1] is incorrect.

Let the contour curve C be the 2-D curve in the x-y plane as it is defined in [1] (centered at the origin). Then the unit normal vector of the contour plane (x-y plane) is $\mathbf{n} = [0, 0, 1, 1]$.

The trajectory T is the 3-D curve as it is defined in [1]. We are interested only in one point of the trajectory (denoted by index j) because for all other points the procedure is the same. Let us take the tangent vector on trajectory T at that point for our example $\mathbf{t}_j = [T_{x_{j+1}} - T_{x_{j-1}}, T_{y_{j+1}} - T_{y_{j-1}}, T_{z_{j+1}} - T_{z_{j-1}}, 1] = [1, 1, 1, 1]$.

Now the relation between the contour curve and the trajectory has to be determined. This is made with the constraint[1]: "The twisted-profiled contour curve C must be rotated in 3-D with respect to the tangent vector at each point of trajectory curve T. That is, the plane of C must be made perpendicular to the tangent vector of T at each point." It means that after the rotation of the contour curve, the condition $\mathbf{n}[R_x][R_y][R_z] = k\mathbf{t}_j$ must be satisfied, where $k \in \mathbf{R}$, and the both vectors are expressed in the non-homogeneous coordinates.

For our vectors, of course, does not hold. We have:

$$[0, 0, 1, 1][R_x][R_y][R_z] = \left[\frac{1}{2}\left(1 + \frac{1}{\sqrt{2}}\right), \frac{1}{2}\left(1 - \frac{1}{\sqrt{2}}\right), \frac{1}{2}, 1\right]$$

or equivalently:

$$\left[\frac{1}{2}\left(1 + \frac{1}{\sqrt{2}}\right), \frac{1}{2}\left(1 - \frac{1}{\sqrt{2}}\right), \frac{1}{2}\right] \neq k[1, 1, 1],$$

where

$$[R_x][R_y][R_z] = \begin{bmatrix} c\beta_j c\gamma_j & c\beta_j s\gamma_j & -s\beta_j & 0 \\ c\gamma_j s\alpha_j s\beta_j - c\alpha_j s\gamma_j & c\alpha_j c\gamma_j + s\alpha_j s\beta_j s\gamma_j & c\beta_j s\alpha_j & 0 \\ c\alpha_j c\gamma_j s\beta_j + s\alpha_j s\gamma_j & c\alpha_j s\beta_j s\gamma_j - c\gamma_j s\alpha_j & c\alpha_j c\beta_j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{and} \begin{cases} s\alpha_j = \dfrac{1}{\sqrt{2}}, & c\alpha_j = -\dfrac{1}{\sqrt{2}} \\ s\beta_j = \dfrac{1}{\sqrt{2}}, & c\beta_j = -\dfrac{1}{\sqrt{2}} . \\ s\gamma_j = \dfrac{1}{\sqrt{2}}, & c\gamma_j = -\dfrac{1}{\sqrt{2}} \end{cases}$$

We therefore propose a matrix of rotation containing only the rotations about the y and z axes:

$$[R_{NEW}] = [R_y][R_z] = \begin{bmatrix} c\beta_j c\gamma_j & c\beta_j s\gamma_j & -s\beta_j & 0 \\ -s\gamma_j & c\gamma_j & 0 & 0 \\ c\gamma_j s\beta_j & s\beta_j s\gamma_j & c\beta_j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (3)$$

where

$$s\beta_j = \sin(\beta_j), \quad s\gamma_j = \sin(\gamma_j),$$
$$c\beta_j = \cos(\beta_j), \quad c\gamma_j = \cos(\gamma_j),$$

and the two angles of rotation β_j and γ_j about the y-axis and the z-axis, respectively, are determined by the following equations:

$$\beta_j = \arccos\left(\frac{T_{z_{j+1}} - T_{z_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2 + (T_{z_{j+1}} - T_{z_{j-1}})^2}}\right), \quad \beta_j \in [0, \pi] \qquad (4)$$

and

$$\gamma_j = \begin{cases} \gamma_j^*, & \text{if} \quad (T_{y_{j+1}} - T_{y_{j-1}}) \geq 0 \\ 2\pi - \gamma_j^*, & \text{if} \quad (T_{y_{j+1}} - T_{y_{j-1}}) < 0 \end{cases}, \qquad \gamma_j \in [0, 2\pi], \quad (5)$$

where

$$\gamma_j^* = \arccos\left(\frac{T_{x_{j+1}} - T_{x_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2}}\right), \qquad \gamma_j^* \in [0, \pi].$$

It should be noticed that with the definition of the trajectory curve as a depth modulation curve[1] in practice it cannot happen that $T_{x_{j+1}} - T_{x_{j-1}} = T_{y_{j+1}} - T_{y_{j-1}} = 0$ but to be more general, we solve this problem of undefined rotational angle γ_j by taking $\gamma_j = \gamma_{j-1}$. It is also important to note that in this case the choice of the angle value has no influence on the mutual orthogonal position of the contour plane and the tangent vector of the trajectory at that point. It determines only the rotation of the contour plane about the axis determined by the tangent vector (about the z-axis).

Let us now see how the new rotation works for another, more "exotic" vector:
Let the vector be: $\mathbf{t}_j = [1, -3, -7, 1]$.
With this vector our rotational matrix has the form:

$$[R_{NEW}] = \begin{bmatrix} \dfrac{-7}{\sqrt{590}} & \dfrac{21}{\sqrt{590}} & -\sqrt{\dfrac{10}{59}} & 0 \\ \dfrac{3}{\sqrt{10}} & \dfrac{1}{\sqrt{10}} & 0 & 0 \\ \dfrac{1}{\sqrt{59}} & \dfrac{-3}{\sqrt{59}} & \dfrac{-7}{\sqrt{59}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

It is now easy to see that the required condition of mutually orthogonal contour plane and the tangent vector of the trajectory curve holds. Namely:

$$[0, 0, 1, 1][R_{NEW}] = \left[\frac{1}{\sqrt{59}}, \frac{-3}{\sqrt{59}}, \frac{-7}{\sqrt{59}}, 1\right]$$

and hence:

$$\left[\frac{1}{\sqrt{59}}, \frac{-3}{\sqrt{59}}, \frac{-7}{\sqrt{59}}\right] = k[1, -3, -7], \text{ with } k = \frac{1}{\sqrt{59}}.$$

As the normal vector on the contour plane is a normalized vector and as the rotational matrix is the orthogonal matrix, it is obvious, that the rotated vector is also a normalized vector. If the tangent vector of the trajectory is not normalized, as it is generally the case (*e.g.*, our example), then k represents the coefficient of normalization of the tangent vector.

For the general case, we offer a proof of correctness of the rotational matrix defined by Eq. (3).

For an arbitrary tangent vector $\mathbf{t}_j = [T_{x_{j+1}} - T_{x_{j-1}}, T_{y_{j+1}} - T_{y_{j-1}}, T_{z_{j+1}} - T_{z_{j-1}}, 1]$ it must be true that $\mathbf{n}[R_{NEW}] = k\mathbf{t}_j$, where $k = \frac{1}{\|\mathbf{t}_j\|}$ and the vectors are expressed in the non-homogeneous coordinates.

Since

$$\mathbf{n}[R_{NEW}] = [0, 0, 1, 1]\begin{bmatrix} c\beta_j c\gamma_j & c\beta_j s\gamma_j & -s\beta_j & 0 \\ -s\gamma_j & c\gamma_j & 0 & 0 \\ c\gamma_j s\beta_j & s\beta_j s\gamma_j & c\beta_j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [c\gamma_j s\beta_j, s\beta_j s\gamma_j, c\beta_j, 1]$$

it must be the case that

$$[c\gamma_j s\beta_j, s\beta_j s\gamma_j, c\beta_j] = \frac{1}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2 + (T_{z_{j+1}} - T_{z_{j-1}})^2}} \times [T_{x_{j+1}} - T_{x_{j-1}}, T_{y_{j+1}} - T_{y_{j-1}}, T_{z_{j+1}} - T_{z_{j-1}}]. \quad (6)$$

We prove this separately for all three components:

First, we prove that

$$c\gamma_j s\beta_j = \frac{T_{x_{j+1}} - T_{x_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2 + (T_{z_{j+1}} - T_{z_{j-1}})^2}}.$$

Since

$$c\gamma_j = \cos\left(\arccos\left(\frac{T_{x_{j+1}} - T_{x_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2}}\right)\right)$$

$$c\gamma_j = \frac{T_{x_{j+1}} - T_{x_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2}} \quad (7)$$

and

$$s\beta_j = \sin\left(\arccos\left(\frac{T_{z_{j+1}} - T_{z_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2 + (T_{z_{j+1}} - T_{z_{j-1}})^2}}\right)\right)$$

$$s\beta_j = \frac{\sqrt{\left(\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2 + (T_{z_{j+1}} - T_{z_{j-1}})^2}\right)^2 - (T_{z_{j+1}} - T_{z_{j-1}})^2}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2 + (T_{z_{j+1}} - T_{z_{j-1}})^2}}$$

$$s\beta_j = \frac{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2 + (T_{z_{j+1}} - T_{z_{j-1}})^2}} \quad (8)$$

it is true that

$$c\gamma_j s\beta_j = \frac{T_{x_{j+1}} - T_{x_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + T_{y_{j+1}} - T_{y_{j-1}})^2 + (T_{z_{j+1}} - T_{z_{j-1}})^2}}.$$

Second, we prove that

$$s\beta_j s\gamma_j = \frac{T_{y_{j+1}} - T_{y_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2 + (T_{z_{j+1}} - T_{z_{j-1}})^2}}.$$

Since $s\beta_j$ is determined by Eq. (8) and

$$s\gamma_j = \sin\left(\arccos\left(\frac{T_{x_{j+1}} - T_{x_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2}}\right)\right)$$

$$s\gamma_j = \frac{\sqrt{\left(\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2}\right)^2 - (T_{x_{j+1}} - T_{x_{j-1}})^2}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2}}$$

$$s\gamma_j = \frac{T_{y_{j+1}} - T_{y_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2}} \tag{9}$$

it is true that $s\beta_j s\gamma_j = \dfrac{T_{y_{j+1}} - T_{y_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2 + (T_{z_{j+1}} - T_{z_{j-1}})^2}}$.

Third, we must prove that

$$c\beta_j = \frac{T_{z_{j+1}} - T_{z_{j-1}}}{\sqrt{(T_{x_{j+1}} - T_{x_{j-1}})^2 + (T_{y_{j+1}} - T_{y_{j-1}})^2 + (T_{z_{j+1}} - T_{z_{j-1}})^2}},$$

which is straightforward to obtain from Eq. (4).

To conclude, we specify the new general matrix for representing a twisted-profiled sweep object:

$$[S_{w_{tp_{NEW}}}] = \begin{bmatrix} s_j \cos\theta_j c\beta_j c\gamma_j - s_j \sin\theta_j s\gamma_j & s_j \cos\theta_j c\beta_j s\gamma_j + s_j \sin\theta_j c\gamma_j & -s_j \cos\theta_j s\beta_j & 0 \\ -s_j \sin\theta_j c\beta_j c\gamma_j - s_j \cos\theta_j s\gamma_j & -s_j \sin\theta_j c\beta_j s\gamma_j + s_j \cos\theta_j c\gamma_j & s_j \sin\theta_j s\beta_j & 0 \\ s_j s\beta_j c\gamma_j & s_j s\beta_j s\gamma_j & s_j c\beta_j & 0 \\ a & b & c & 1 \end{bmatrix}. \tag{10}$$

Sweep objects constructed in this way are now defined as follows:

$$[\mathit{Sweep\ surface}] = [C][S_{w_{tp_{NEW}}}]. \tag{11}$$

Acknowledgements—The first author wishes to thank his postgraduate supervisor Prof. Nikola Guid for fruitful suggestions and discussions. Both authors are grateful to Prof. José L. Encarnação for his role in the preparation of this note.

REFERENCES

1. V. Akman and A. Arslan, Sweeping with all graphical ingredients in a topological picturebook. *Comp. & Graph.* **16** (3), 273-281 (1992).
2. F. Klok, Two moving coordinate frames for sweeping along a 3-D trajectory. *Comp. Aided Geom. Design* **3**, 217-229 (1986).
3. W. F. Bronsvoort and F. Klok, Ray tracing generalized cylinders. *ACM Trans. Graph.* **4** (4), 291-303 (1985).
4. W. F. Bronsvoort, P. R. van Nieuwenhuizen, and F. H. Post, Display of profiled sweep objects. *The Vis. Comp.* **5**, 147-157 (1989).
5. H. Guggenheimer, Computing frames along a trajectory. *Comp. Aided Geom. Design*, **6**, 77-78 (1989).
6. W. F. Bronsvoort and J. J. Waarts, A method for converting the surface of a generalized cylinder into a B-spline surface. *Comp. & Graph.* **16** (2), 175-178 (1992).
7. W. F. Bronsvoort, A surface-scanning algorithm for displaying generalized cylinders. *The Vis. Comp.* **8**, 162-170 (1992).
8. S. Coquillart, A control-point-based sweeping technique. *IEEE Comp. Graph. & Appl.* **7** (11), 36-45 (1987).
9. J. P. Fillmore, A note on rotation matrices. *IEEE Comp. Graph. & Appl.* **4** (2), 30-33 (1984).
10. B. Sabata and J. K. Aggarwal, Estimation of motion from a pair of range images: A review. *CVGIP: Image Understanding*, **54**(3), 309-324 (1991).
11. T. Herter and K. Lott, Algorithms for decomposing 3-D orthogonal matrices into primitive rotations. *Comp. & Graph.* **17** (5), 517-527 (1993).