

透視3次元スケッチを利用した平面对称自由曲線生成

Perspective 3D Sketch for Creating 3D Plane-Symmetric Curves

錫亨¹⁾, 木島竜吾²⁾

Seok-Hyung BAE and Ryugo KIJIMA

1) 岐阜大学 バーチャルシステムラボラトリー

(〒501-1193 岐阜県岐阜市柳戸1-1, bae@vsl.gifu-u.ac.jp)

2) 岐阜大学 工学部

(〒501-1193 1193 岐阜県岐阜市柳戸1-1, kijima@info.gifu-u.ac.jp)

Abstract : The 3D evaluation of design shapes is an essential step in product styling. However, in the concept design stage, it is not appropriate to make physical mockups or 3D CAD models because the product style is rapidly changing. Therefore, this paper presents a method of directly constructing 3D plane-symmetric freeform curves for the early 3D evaluation by introducing the perspective 3D sketch, which is extended from the digital design sketch. The 2D curve drawn by the designer in an arbitrary perspective view, is simultaneously converted to a 3D space curve without an 3D ambiguity problem except in only a few special cases to be specified.

Key Words: *perspective 3D sketch, digital design sketch, 3D plane-symmetric curves*

1. Introduction

As the customer's respect to products is being transformed from functionality to design, the importance of *product styling* is highly increasing in the whole product-development process. For evaluating the 3D shape of products before actual production, it has been common to make physical models of clay, wood, etc. [10] However, physical models require a lot of money and time. Thus, recently, the effort to substitute physical models with computer models, has been explored.

So far, this kind of endeavor has focused on only final (or semi-final) stage design alternatives. Designers in the early phase of the design-development stage, where most of design concept is fixed, still uses *raster-type* 2D graphics S/Ws. It causes serious problems as follows: 1) the difficulty of the early *3D evaluation*, and 2) *no digital connectivity* with downstream processes.

In this paper, as our first attempt to develop a *direct* 3D freeform-shape creation S/W that allows the both of the early 3D evaluation and digital continuity, a method of constructing *3D plane-symmetric freeform curves* with a *sketch interface* is given (many products are plane symmetric including automobiles). The freeform curves symmetric to the center plane drawn by the designer are simultaneously converted to 3D space curves without caus-

ing 3D ambiguity (a few special viewing-situations in which the method does not work will be specified later).

The organization of this paper is as follows: related work to the direct creation of computer models will be given in Section 2. In Section 3, so-called a *perspective 3D sketch* scheme is suggested. In Section 4, the overview of a method converting 2D curves drawn in a perspective view to 3D curves is given, and then the actual calculation procedure of finding 3D points from 2D points is presented with special case treatments in Section 5. The software implementation is given in Section 6 followed by discussions and conclusions in Section 7.

2. Related Work

We categorized research on *direct* 3D computer-model creations as following three based on the user interface: 1) *sketch-based* methods, 2) *suggestive* methods, and 3) *VR-interface* methods. Sketch-based methods have been mostly applied to creating CSG-like models composed of simple primitives [5][9][11]: the most interest was focused on the *primitive recognition* from a rough scribble, and the *topology reconstruction*. There were little studies about directly creating freeform-shapes: one is Teddy [7] for *rounded* freeform models, which, however, is not appropriate for product styling. A suggestive interface is

to forecast possible subsequent operations to be executed by users, and to gives action alternatives [3][8]. It shows great possibility for creating simple polygon models, but is limited for freeform-shapes. Recently, much research with 3D interfaces (VR interfaces) has been done. However, VR techniques for the direct creation of freeform shapes looks immature in a practical point of view (heavy equipment, precision problem, etc.).

There are two representative studies about 3D space-curve creation: Cohen, et al. [2] suggested a method by sketching a curve and its shadow curve on the floor in a perspective view. Grossman, et al. [6] proposed a reverse way where a depth surface is constructed first, and the curve drawn on orthographic plane is projected on it.

3. Perspective 3D Sketch

The sketch interface proposed in this paper, *perspective 3D sketch*, is straightforwardly extended from the intuitive 2D sketch interface, so called the *digital design sketch* [1], which enables the designer to freely create accurate curves intended by allowing repetitive scribbling (scribbles already drawn are spread and grayed out as the number of them increases). Shown in Fig. 1 is the flowchart of the perspective 3D sketch scheme.

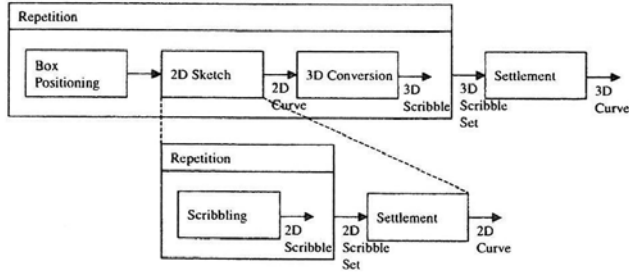


Fig. 1: Flowchart of perspective 3D sketch

The designer can choose an arbitrary perspective view by rotating the *see-through* box provided as a *reference unit* for estimating the dimension of 3D curves to be projected with it. Then he/she draws a curve, $\tilde{c}(t)$, on 2D image plane (Fig. 2(a)) based on *design perspective* [4]. By repetition of perspective scribbling, a set of 3D curves, $\{c_i(t)\}$, are generated (Fig. 2(b)), and at last, a final curve, $\bar{c}(t)$, is settled (Fig. 2(c)).

4. Conversion of 2D Curve to 3D

The *pinhole-camera model* widely used in computer graphics and machine vision is composed of the *optical center*, e , and the *retinal* (or image) plane, Π_R . Between an arbitrary 3D point, p , and its projection point, \tilde{p} ,

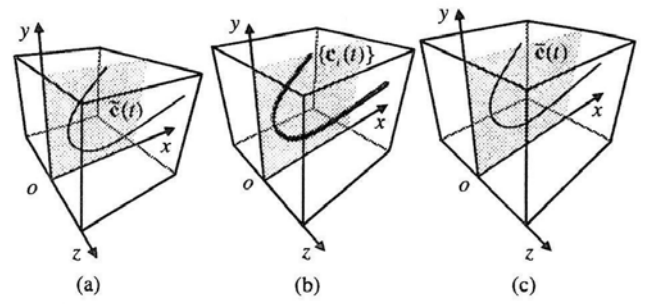


Fig. 2: Illustrative example of perspective 3D sketch

there is a *morphism* as:

$$\tilde{p}^w = H p^w \quad (1)$$

where H is the *perspective projection matrix* ($\text{rank}(H) = 3$), p^w and \tilde{p}^w are the *homogeneous coordinates* of p and \tilde{p} , respectively.

In general, the inverse problem of perspective projection—finding p from \tilde{p} —is an *under-determined* problem, which has an infinite numbers of solutions. In this study, the *3D ambiguity* is resolved by imposing the *plane-symmetry* condition. Fig. 3 shows our system configuration where q is the intersection point of the *see-through* box and the optical ray passing through \tilde{p} and p .

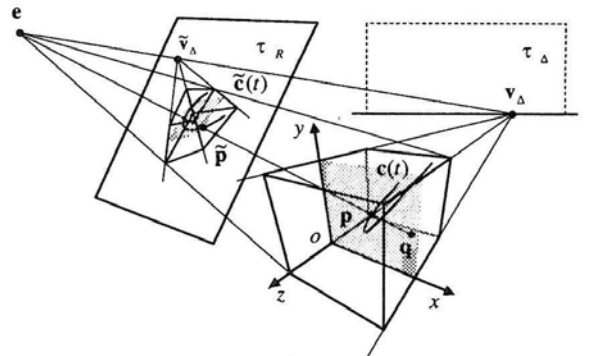


Fig. 3: System configuration for inverse perspective projection

The overall procedure of converting a 2D curve drawn by the designer to a 3D plane-symmetric freeform curve is as follows: 1) polygonizing a 2D parametric-curve, 2) matching a 2D-point pair, 3) finding a 3D-point pair on *see-through* box, 4) finding a 3D-point pair looked for, and 5) creating a resulting 3D parametric-curve.

4.1 Parametric-curve Polygonization

Given a parametric 2D curve, $\tilde{c}(t)$, is sampled, and 2D point set, $\{\tilde{p}_i(\tilde{x}_i, \tilde{y}_i)\}$, is obtained by the parametric-curve polygonization algorithm.

4.2 Matching 2D-point Pair

Using the 3-point perspective [4], a 2D-point pair that is expected to be symmetric in 3D space, can be acquired. First, the vanishing point, \tilde{v}_∞ , is found by extending the edges of the see-through box parallel to z-axis, and given point, $\tilde{p}_1 \equiv \tilde{p}_i$, is mapped to its corresponding point, \tilde{p}_2 , by calculating the intersection point between a line passing through \tilde{p}_1 and \tilde{v}_∞ , and $\tilde{c}(t)$ (see Fig. 4).

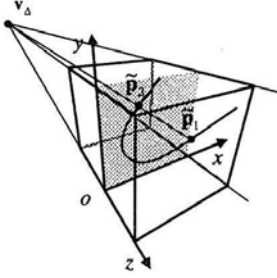


Fig. 4: 2D-point pair matching

4.3 Finding 3D-point Pair on See-Through Box

Instead of directly finding 3D-space points, (p_1, p_2) (as mentioned earlier, it is impossible to calculate them without the 3D ambiguity), their projection points on see-through box, (q_1, q_2) are calculated from 2D points, $(\tilde{p}_1, \tilde{p}_2)$, as follows (see Fig. 5(a) and refer our viewing system configuration given in Fig. 3) : For each 2D point, 1) finding *two* faces its optical ray pass through, 2) choosing the face between them, on which its 2D image point is closer to the center for more accurate calculation, 3) calculating two parameters of an *affine combination* in the 2D space, (μ, η) , using two corresponding vanishing points, and 4) obtaining the 3D projection point on the see-through box using affine combination in the 3D space with the parameters previously calculated (see Fig. 5).

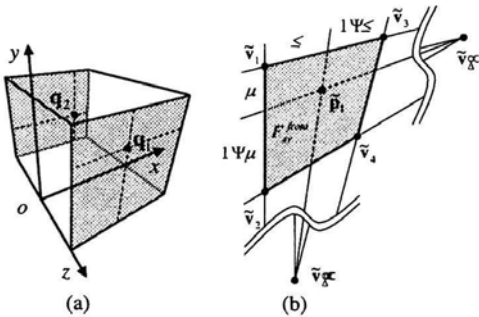


Fig. 5: Finding 3D-point pair on see-through box

4.4 Finding 3D-point Pair in Space

Now, we can define two optical rays, $(l_1(t), l_2(s))$, from the optical center, e , using (q_1, q_2) as follows:

$$l_1(t) = e + t(q_1 - e), \quad (2)$$

$$l_2(s) = e + s(q_2 - e). \quad (3)$$

The 3D points, $p_1 = l_1(t^*) = (p_{1x}, p_{1y}, p_{1z})$ and $p_2 = l_2(s^*) = (p_{2x}, p_{2y}, p_{2z})$, must satisfy the center-plane symmetry condition. That is, $p_{1x} = p_{2x}$ & $p_{1y} = p_{2y}$ & $p_{1z} = -p_{2z}$. Thus, a solution-parameter pair, (t^*, s^*) , can be obtained by solving a system of three linear equations as follows (for more details, see Section 5):

$$t(q_{1x} - e_x) = s(q_{2x} - e_x), \quad (4)$$

$$t(q_{1y} - e_y) = s(q_{2y} - e_y), \quad (5)$$

$$e_z + t(q_{1z} - e_z) = -e_z - s(q_{2z} - e_z). \quad (6)$$

4.5 3D Parametric-curve Creation

As the final step of converting a 2D curve to a 3D space curve, the approximation of $\{p_i(x_i, y_i, z_i)\}$ to a parametric curve such as B-spline (or Bezier) curve, $c(t)$, is performed by applying standard curve fitting methods while keeping a plane symmetry.

5. Solving System of Linear Equations

As given in (4)-(6), our inverse projection of a 2D-point pair to 3D points under the plane-symmetry condition is an *over-determined* problem having *two* unknowns and *three* equations. Because of the pinhole-camera model assumption ($t, s > 0$), the solution space is $(t^*, s^*) \in (0, \infty) \times (0, \infty)$. For most cases, (t^*, s^*) can be simply determined by minimizing the sum of least-squares errors, $f = (NF - G)^T(NF - G)$, where $NF = G$ is the matrix form of (4)-(6).

There exist special viewing conditions in which the proposed method does not work. First of all, let us consider the cases that one of three equations vanishes as shown in Fig. 6. The case that only (4) vanishes, that is, $q_{1x} - e_x \approx 0$ & $q_{2x} - e_x \approx 0$, $\{e, p_1, p_2, q_1, q_2\}$ are all on the same plane parallel to the yz-plane (see Fig. 6(a)). In the case, it is possible to calculate (t^*, s^*) with remaining two equations. Similarly, if (5) vanishes, the solution exists (Fig. 6(b)). However, when (6) vanishes, that is, $q_{1z} - e_z \approx 0$ & $q_{2z} - e_z \approx 0$ & $e_z \approx 0$, $\{e, p_1, p_2, q_1, q_2\}$ are all on the xy-plane because $p_{1x} \approx p_{2x}$ & $p_{1y} \approx p_{2y}$ & $p_{1z} \approx -p_{2z}$ or $p_1 \approx p_2$ & $q_1 \approx q_2$. Thus, there are an infinite numbers of solutions (Fig. 6(c)) (it is so called the *impossible-to-solve* case). On the other hand, the cases having only one equation, are all the impossible-to-solve cases (see Fig. 7). In other words, impossible-to-solve cases arise only when 1) the optical center is positioned on the center plane, or 2) the optical rays are parallel to the z-axis.

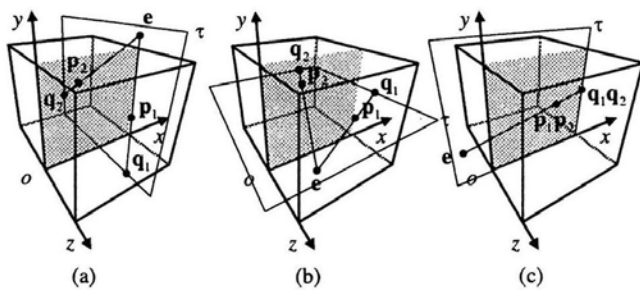


Fig. 6: Special cases only one equation vanishes

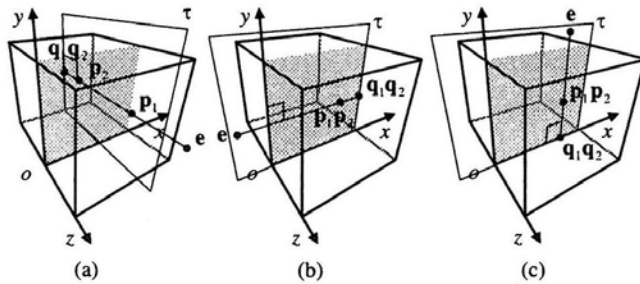


Fig. 7: Special cases only one equation remains

6. Implementation

A simple program for the proposed method is implemented as a *Java Applet* using JavaTM 2 Platform Standard Edition (J2SETM) and Java 3DTM API as shown in Fig. 8. For a pen-based user interface, WACOM IntuosTM2 tablet (9" × 12") is used.

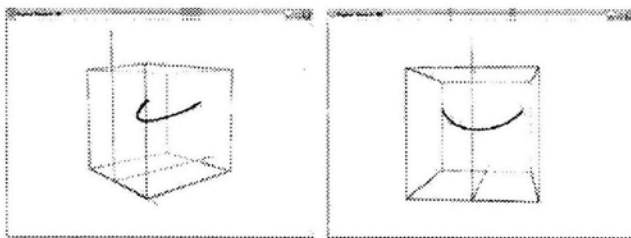


Fig. 8: Software implementation: repetitive 3D scribbling and settled freeform curve

7. Discussions and Conclusions

In this paper, we proposed a method of constructing 3D plane-symmetric curves using an intuitive sketch interface, which was inspired from a fact that a lot of industrial products have plane-symmetric forms. The designer can arbitrary choose a 3D perspective view, and draw a projected form of the curve intended. By repetition of the perspective 3D sketch (visual feedback and behavioral adaptation process), the designer can obtain the final form of a 3D space curve. Because of the assump-

tion of a plane-symmetry, the proposed method does not suffer the 3D ambiguity problem unlike a general inverse-projection problem (the exceptional viewing cases—only two—were addressed in Section 5).

One important point is that the proposed method is relies on the designer's accurate space sense based on design perspective. The assumption can be thought as reasonable remembering that projective geometry, which computer graphics is based on, was started from Renaissance painters' effort to correctly reproduce the perspective effects in images of the world that they were observing. Actually, many designers took a great interest in our program, and agreed it has much possibility to be developed as a powerful tool for design specialists.

References

- [1] Bae, S.-H., Kim, W.-S., Kwon, E.-S.: Digital Styling for Designers: Sketch Emulation in Computer Environment. LNCS 2669 (2003) 690-700
- [2] Cohen, J., Markosian, L., Zeleznik, R., Hughes, J., Barzel, R.: An Interface for Sketching 3D Curves. The Proceedings of SI3DG (1999)
- [3] Cypher, A.: Eager: Programming Repetitive Tasks by Example. The Proceeding of CHI (1991)
- [4] Doblin, J.: Perspective: A New System for Designers. Whitney Publications, NY (1956)
- [5] Eggi, L., Hsu, C.-Y., Bruderlin, B.D., Elber, G.: Inferring 3D Models from Freehand Sketches and Constraints. CAD 29(2) (1997) 101-122
- [6] Grossman, T., Balakrishnam, R., Kurtenbach, G., Fitzmaurice, G., Khan, A., Buxton, B.: Creating Principal 3D Curves with Digital Tape Drawing. In Proceedings of CHI (2002)
- [7] Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: Sketch Interface for a 3D Freeform Design. The Proceedings of SIGGRAPH (1999)
- [8] Igarashi, T., Hugh, J.F.: A Suggestive Interface for 3D Drawing. The Proceedings of UIST (2001)
- [9] Schweikardt, E., Gross, M. D.: Digital Clay: Deriving Digital Models from Freehand Sketches. The Proceedings of CHI (2002)
- [10] Yamada, Y.: Clay Modeling: Techniques for Giving Three-dimensional Form to Idea. Car Styling Extra Issues 93(1/2) (1993)
- [11] Zeleznik, R. C., Herdon, K. P., Hughes, J. F.: SKETCH: An Interface for Sketching 3D Scenes. The Proceedings of SIGGRAPH (1996)