Information Processing in Japan Vol. 9, 1969

A Numerical Experiment on the Fairing Free-Form Curves

YOSHIMI OKADA*

Abstract

An algorithm for fairing free-form curves of the body-lofting is presented using the simulation of the technique commonly employed in the process of the manual fairing with the batten. With the help of a smoothing procedure devised for this kind of problem it is shown how one may mathematically proceed from the given ordered set of points to the faired curve. The obtained curve is intended to satisfy the same continuity conditions as if it were faired by a skilled hand.

1. *Introduction*

For many years, the drawing of the automobile body lines has been done using the batten and the weight very satisfactorily for interpolating between points. These works are exceedingly laborious, since they entail a large amount of trial and error iterations in order to assure that the curve is "fair". Therefore, the mathematical fairing is required as the use of numerical control machinery increases.

For the arbitrarily placed points in a plane or in space obtained by some methods, the problem in mathematical fairing is focused on the following two points.

(1) The given points must be adjusted, since they include errors in making model, in measuring, or in reading a scaled drawing.

(2) The curve passing through these points must be mathematically described and must be smooth.

In this paper, the author describes an algorithm, as a simulation of the iteration process of manual fairing, using the idea [2] for fairing by Dr. Hosaka.

2. *Fairing Method*

2.1 *Fundamental Equation*

In Fig. 1, let $\boldsymbol{P}_i$, $i=1, \cdots, n$, be n distinct position vectors in space, and let $\boldsymbol{Y}_i$, $i=1, \cdots, n-1$, and $\boldsymbol{X}_i$, $i=1, \cdots, n$, be tangent vectors.

This paper first appeared in Japanese in Joho Shori (the Journal of the Information Processing Society of Japan) Vol. 10, No. 3 (1969), pp. 143-149.

* Data Center, Toyo Kogyo Co., Ltd. Hiroshima.

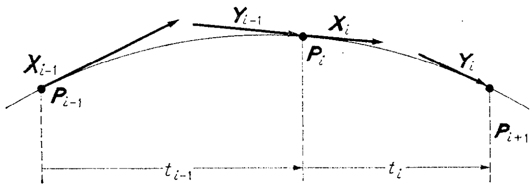


Fig. 1.

The objective curve passing through these points is composed of $n-1$ curve segments. Using Ferguson's expression [1], the ith curve segment is represented by the following form;

$$\boldsymbol{P}_i(t_i)=\sum_{j=0}^{3}\boldsymbol{R}_{i,j}\cdot t_i^{\,j},\quad 0\leqq t_i\leqq 1 \tag{1}$$

such that $\boldsymbol{P}_i(0)=\boldsymbol{P}_i$, $\boldsymbol{P}_i(1)=\boldsymbol{P}_{i+1}$, $\dot{\boldsymbol{P}}_i(0)=\boldsymbol{X}_i$, $\dot{\boldsymbol{P}}_i(1)=\boldsymbol{Y}_i$ ($\dot{}$ denotes d/dt).

After these preparations, the objective curve is assumed to be an elastic curve in order to simulate the actual operation with the batten and the weight. The following way is based on Dr. Hosaka's idea [2]. The "fair" curve is obtained by minimizing the strain energy stored in it, under the restricted conditions.

As is generally known, the strain energy by bending moment, under small deflections, is given as follows;

$$U=\frac{E\cdot I}{2}\int_0^l \kappa^2\cdot ds \tag{2}$$

where the flexural rigidity $E\cdot I$ is constant, l is the length of beam, κ is the curvature, and s is the length of arc.

κ in Eq. (1) is approximately obtained from the following relations;

$$\dot{\boldsymbol{P}}=\boldsymbol{P}'\cdot\frac{ds}{dt},\quad \ddot{\boldsymbol{P}}=\boldsymbol{P}''\cdot\left(\frac{ds}{dt}\right)^2+\boldsymbol{P}'\cdot\frac{d^2s}{dt^2}, \tag{3}$$

$$\kappa=|\boldsymbol{P}''|\fallingdotseq|\ddot{\boldsymbol{P}}|\Big/\left(\frac{ds}{dt}\right)^2\quad\left({}'\text{ denotes}\frac{d}{ds}\right). \tag{4}$$

Whence the scalar quantity V_i, which is directly proportional to the strain energy stored in the successive curve segments $\widehat{\boldsymbol{P}_{i-1}\boldsymbol{P}_{i+1}}$, is obtained;

$$\begin{aligned}V_i&=\frac{1}{C_{i-1}^3}\int_0^1|\ddot{\boldsymbol{P}}_{i-1}(t_{i-1})|^2\cdot dt_{i-1}+\frac{1}{C_i^3}\int_0^1|\ddot{\boldsymbol{P}}_i(t_i)|^2\cdot dt_i\\&=\frac{1}{C_{i-1}^3}\{(\boldsymbol{R}_{i-1,2}\boldsymbol{R}_{i-1,2})+3(\boldsymbol{R}_{i-1,2}\boldsymbol{R}_{i-1,3})+3(\boldsymbol{R}_{i-1,3}\boldsymbol{R}_{i-1,3})\}\\&\quad+\frac{1}{C_i^3}\{(\boldsymbol{R}_{i,2}\boldsymbol{R}_{i,2})+3(\boldsymbol{R}_{i,2}\boldsymbol{R}_{i,3})+3(\boldsymbol{R}_{i,3}\boldsymbol{R}_{i,3})\}\end{aligned} \tag{5}$$

where,

$$\boldsymbol{R}_{i,2}=3(\boldsymbol{P}_{i+1}-\boldsymbol{P}_i)-2\boldsymbol{X}_i-\boldsymbol{Y}_i=3(\boldsymbol{P}_{i+1}-\boldsymbol{P}_i)-2\boldsymbol{X}_i-k_{i+1}\cdot\boldsymbol{X}_{i+1} \tag{6}$$

$$\boldsymbol{R}_{i,3}=2(\boldsymbol{P}_i-\boldsymbol{P}_{i+1})+\boldsymbol{X}_i+\boldsymbol{Y}_i=2(\boldsymbol{P}_i-\boldsymbol{P}_{i+1})+\boldsymbol{X}_i+k_{i+1}\cdot\boldsymbol{X}_{i+1} \tag{7}$$

$$C_i=|\boldsymbol{C}_i|=|\boldsymbol{P}_{i+1}-\boldsymbol{P}_i| \tag{8}$$

$$k_i = C_{i-1}/C_i, \quad s_i = C_i \cdot t_i. \tag{9}$$

Assume that k_i is constant and use the notations $\boldsymbol{X}_i = (X_i^1, X_i^2, X_i^3)$ and $\boldsymbol{P}_i = (P_i^1, P_i^2, P_i^3)$. Then, the tangent vector and the position vector, which reduce the quantity V_i, are obtained from the following equations respectively.

$$\left(\boldsymbol{i}\frac{\partial}{\partial X_i^1} + \boldsymbol{j}\frac{\partial}{\partial X_i^2} + \boldsymbol{k}\frac{\partial}{\partial X_i^3}\right) V_i = \frac{4k_i}{C_{i-1}^3}(\boldsymbol{R}_{i-1,2} + 3\boldsymbol{R}_{i-1,3}) - \frac{4}{C_i^3}\boldsymbol{R}_{i,2} = 0, \tag{10}$$

$$\left(\boldsymbol{i}\frac{\partial}{\partial P_i^1} + \boldsymbol{j}\frac{\partial}{\partial P_i^2} + \boldsymbol{k}\frac{\partial}{\partial P_i^3}\right) V_i = \frac{4}{C_{i-1}^3}(-3\boldsymbol{R}_{i-1,3}) + \frac{4}{C_i^3}(3\boldsymbol{R}_{i,3}) = 0. \tag{11}$$

Eq. (10) and Eq. (11) are represented respectively by the recurring relations as follows;

$$\boldsymbol{X}_{i-1} + 2k_i(1+k_i)\cdot\boldsymbol{X}_i + k_i^2\cdot k_{i+1}\cdot\boldsymbol{X}_{i+1} = 3(\boldsymbol{C}_{i-1} + k_1^2\cdot\boldsymbol{C}_i), \quad i = 2, \cdots, n-1 \tag{12}$$

$$-2\boldsymbol{P}_{i-1} + 2(1+k_i^3)\boldsymbol{P}_i - 2k_i^3\cdot\boldsymbol{P}_{i+1} = \boldsymbol{X}_{i-1} + k_i(1-k_i^2)\boldsymbol{X}_i - k_i^3\cdot k_{i+1}\cdot\boldsymbol{X}_{i+1},$$
$$i = 2, \cdots, n-1. \tag{13}$$

2.2 An Algorithm

The following is a description of an algorithm in outline form.

Step 1: Set the counter m to zero, and read in n points and the following auxiliary informations to restrict the curve form;

(1) to set the coordinates of points to be adjusted,

(2) to set the ranges to be adjusted,

(3) to set the points to be fixed,

(4) to set the directions of the tangent vectors, if they are known or specified.

Step 2: Compute the tangent vectors $\boldsymbol{X}_1$ and $\boldsymbol{X}_n$ at both ends to solve the Eq. (12).

The author's method is as follows.

(1) The "unit vectors" are determined from the circular arc passing through the three points at the end.

(2) The "magnitude" is determined by the unit vectors in (1) and two parametric quadratics in the two curve segments, making the second derivative vectors equal at the middle point.

Step 3: Compute the vectors $(\boldsymbol{X}_2, \cdots, \boldsymbol{X}_{n-1})$ using the Eq. (12) and the current vectors $(\boldsymbol{P}_1, \cdots \boldsymbol{P}_n, \boldsymbol{X}_1, \boldsymbol{X}_n)$.

If an auxiliary information (4) in Step 1 is set at the point i, divide a set of points into two partitions $(\boldsymbol{P}_1, \cdots, \boldsymbol{P}_i)$, $(\boldsymbol{P}_{i+1}, \cdots, \boldsymbol{P}_n)$, determine the magnitude of $\boldsymbol{X}_i$ using (2) in Step 2, and solve the Eq. (12) for each partition.

When many auxiliary informations (4) in Step 1 are given, the same procecure is repeated.

Step 4: Step the counter m up by one and calculate the curvatures on the projected planes respectively on account of the space curve.

Step 5: Divide a set of points into several partitions according to the auxiliary information (3) in Step 1.
In each partition, adjust the components of the position vectors, using the Eq. (13), the results in Step 3 and the auxiliary information (1), (2) in Step 1, to ensure the fair curve, and replace the position vectors ($\boldsymbol{P}_2, \cdots, \boldsymbol{P}_{n-1}$) in order to prepare for the next iteration.

Step 6: If $m=1$, go back to Step 2.
If $m>1$, decide whether the curve is "fair" or not.
The "fairness" of the curve is decided when the sign of the curvatures at points, $i=1, \cdots, n$, are the same to that of the last iteration.
If the curve is fair, go to Step 7. If the curve is not fair, go back to Step 2.

Step 7: Compute the coefficient vectors $\boldsymbol{R}_{i,j}$, $i=1, \cdots, n-1$, $j=0, \cdots, 3$.

3. *Experiment*

A program for experiment was made according to the algorithm described in the above section. In this program, the input points are restricted under 200. It is fairly easy to solve Eq. (12) or Eq. (13), because the absolute values of the diagonal elements in the coefficient matrix are greater than others.

Then, following experiments were made:

1. to draw a fair curve by using the batten and the weight,
2. to add some errors, which are approximately 10 times more than usual, to the points on this curve,
3. to regenerate a fair curve with computer under several restrictions.

The following is a typical example for three dimentional fairing.

Fig. 2 shows curve forms composed of three views, where the fixed points (marked $\times$) and curve names are shown in only the side view. These curves

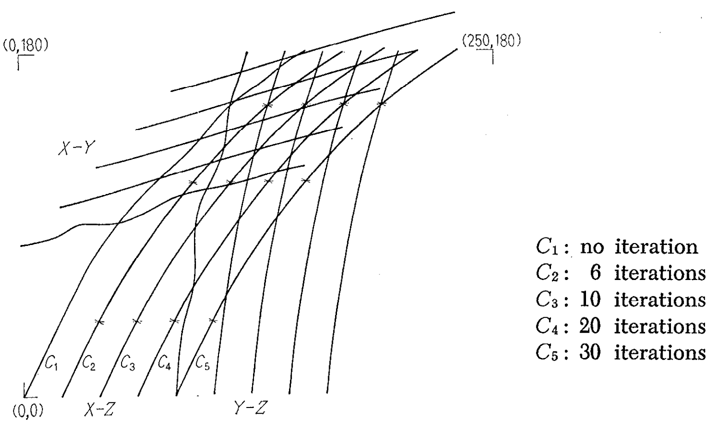


Fig. 2.

Table 1

ITERATION (6)

	DATA-X	DATA-Y	DATA-Z	FAIR-X	FAIR-Y	FAIR-Z	MOVED-X	MOVED-Y	MOVED-Z
1	0.0	100.00000	0.0	0.0	100.00000	0.0	0.0	0.0	0.0
2	10.00000	102.00000	21.00000	10.23811	102.14383	20.89567	0.23811	0.14383	-0.10433
3	20.00000	104.80000	40.00000	20.00000	104.80000	40.00000	0.0	0.0	0.0
4	30.00000	110.00000	60.00000	31.25575	107.85837	59.78788	1.25575	-2.14163	-0.21212
5	50.00000	111.00000	90.00000	51.10435	113.42693	88.59508	1.10435	2.42693	-1.40492
6	70.00000	119.00000	111.50000	70.00000	119.00000	111.50000	0.0	0.0	0.0
7	90.00000	127.00000	130.00000	88.86891	124.48463	132.18835	-1.13109	-2.51537	2.18835
8	110.00000	130.30000	151.60000	110.00000	130.30000	151.60000	0.0	0.0	0.0
9	130.00000	135.30000	166.00000	129.57525	135.46245	166.52465	-0.42475	0.16245	0.52465
10	140.00000	138.00000	174.00000	140.15232	137.97043	173.78894	0.15232	-0.02957	-0.21106
11	150.00000	140.00000	180.00000	150.00000	140.00000	180.00000	0.0	0.0	0.0

Table 2

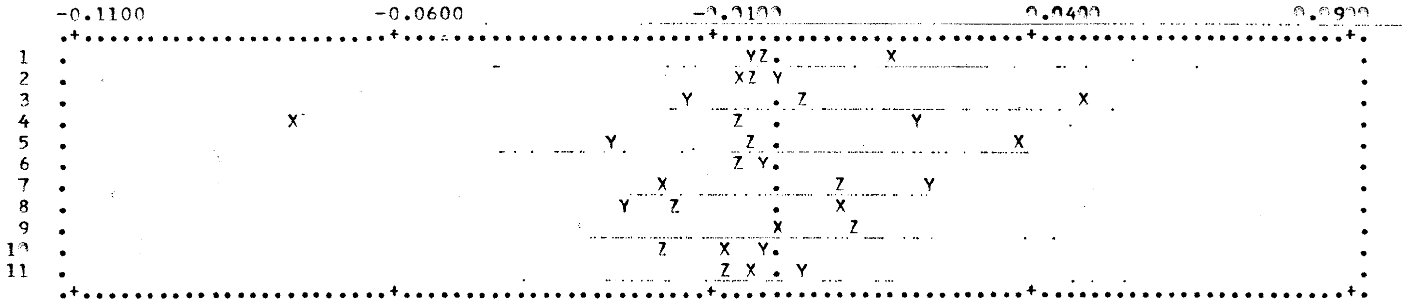
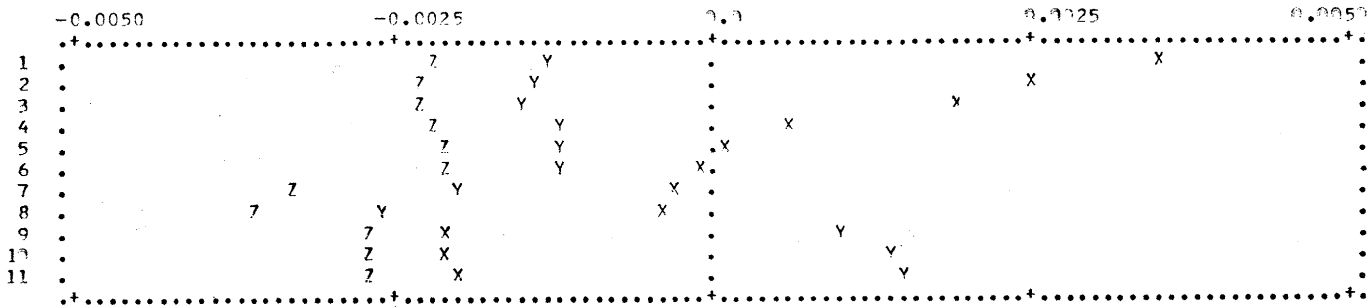


Table 3



consist of 11 points and four curves (C_2-C_5) are given the following auxiliary information.

1. All the components are to be adjusted,
2. Each range to be adjusted is not restricted,
3. The 3rd, 6th and 8th points are to be fixed.

Table 1 shows the result of points adjustment on C_2, and Table 2 and Table 3 show the change of curvature in the case of C_1 and C_5.

From this experiment, it can be seen that the change of curvature is continuous over the length of the curve by way of iterative calculation within the computer. As seen in Fig. 2, C_2 which is decided to be fair by the program is nearly restored to the original shape.

4. *Conclusion*

Generally errors included in the points are not so large as in this experiment, and so this method may be applied to the actual work. A problem encountered on applying this method is that the human intervention is required a little in order to prepare auxiliary information. For example, it is important to decide which points to fix for the curve whose curvature changes radically.

Therefore, the author considers it effective to utilize the graphic display unit.

5. *Acknowledgment*

The author is grateful to Mr. Y. Mikami for helpful suggestions and observations and a critical reading of the manuscript.

References

[1] Ferguson, J., Multivariable Curve Interpolation, *Journal of the Association for Computing Machinary, 11,* 2 (April 1964), 221–228.

[2] Hosaka, M., Computer Aided Design, *J. S. M. E. Japan, 71,* 590 (May 1968), 402–411.