

Ubiquitous Mathematical Graphic Viewer for Visually Impaired Students

Ryoji Fukuda
rfukuda@cc.oita-u.ac.jp
Faculty of Engineering,
Oita University
Japan

Akihiro Miura
am-40114@oct-net.ne.jp
Faculty of Engineering,
Oita University
Japan

Abstract: *In a science class, a teacher may provide some temporal graphical information, which cannot be understood by visually impaired students. We have assumed the availability of support staffs for visually impaired students, such as note-takers for aurally impaired students, and we have designed a graphic input system for them. In this system, drawn curves are recognized and corresponding curve types and parameters as well as compensation curves are displayed. This system also has learning functions to improve the evaluating functions for recognition and reference points for compensation.*

1. Introduction

Students often receive temporary graphical information in science classes. Teachers draw freehand graphs on the blackboard or they gesture to express the graphical information. It is extremely difficult for visually impaired students to acquire such information; they are unable to understand much of the taught content without tactile graphs or other alternative expressions. In a class specially designed for visually impaired students, the teacher prepares some tactile materials or other expressive methods. However, these methods may not be sufficient for brainstorming; the temporal expressions may be limited. On the other hand, aurally impaired students cannot acquire oral information. Notetaking is a standard service for mitigating this problem. Teachers require less effort to teach aurally impaired students when an efficient notetaker is available. Thus, we can easily evaluate the utility of support services for supplying graphical information to visually impaired students in a class. Our system is a graphical input system for this purpose. Some software tools for notetakers are available; an important advantage of these tools is that they provide electric data. The proposed system can be potentially enhanced in this regard.

In many cases, some elements of the supplied information are required only for temporal explanations, whereby tasks need to be performed quickly. Moreover, additional information besides tactile information of the graph shape is required. For example, in the case of a circle, the coordinates of the center and the radius are output in a text field. In the proposed system, these information elements are automatically generated for a single drawn curve at a time.

The target curve types are "Circle," "Straight line," "Parabola," and graphs of "Cubic Functions," "Exponential Functions," and "Logarithmic Functions". Cubic Functions are divided into 3 types according to the number of stationary points (2, 1, and 0). Theoretically, it may be quite difficult to distinguish between curves that are drawn by hand. However, each type of curve has a typical shape, and a user usually draws distinguishable curves. For a drawn curve, the proposed system calculates the evaluation value for each curve type (see Section 3), and the selected type attains the maximum value. For the selected curve, a compensation curve is obtained using least square error estimation.

Evaluation functions and compensation methods are defined using several parameters. We have to adjust these parameters because typical shapes of drawn curves vary from person to person. The training data are stored during the actual use of the system. The system selects a curve type and generates a compensation curve; the corresponding curve parameters are listed in a text box. Then, the user corrects the curve type and parameters if required. These results are stored when the user selects the input curve. For a curve type, the system searches for optimal parameters of evaluation and compensation after several data are stored for this curve type. To perform this search we use a simple generic algorithm.

2. Outline of the System

2.1 Objective of the system and the target curves.

The objective of the system is to supply graphical information during a science class for a visually impaired student. We assume that the system will be employed by a clear-sighted support staff. The support staff draws a curve using a mouse; then, the system selects the curve type and the corresponding compensation curve is displayed in a view field. In addition, several parameters are listed in a text field. After several corrections, the input curve displayed in a tactile display. The target curves and their corresponding parameters are listed in Table 1.

Table 2.1: Parameters

Curve type	Parameters
Circle	$Center_x, Center_y, Radius$
Straight line	$C_0, C_1 \quad (f(x) = C_0 + C_1x)$
Parabola	$C_0, C_1, C_2 \quad (f(x) = C_0 + C_1x + C_2x^2)$
Cubic fun. (0,1,2)	$C_0, C_1, C_2, C_3 \quad (f(x) = C_0 + C_1x + C_2x^2 + C_3x^3)$
Sin. curve	$C_0, C_1, C_2 \quad (f(x) = C_0 \sin(C_1x + C_2))$
Exp. fun.	$C_0, C_1 \quad (f(x) = C_0 \exp(C_1x))$
Log. fun.	$C_0, C_1 \quad (f(x) = C_0 \log(x + C_1))$

2.2 Layout

Figure 2.1 shows the layout of the system. Initially with, the x and y axes are drawn in the graphic field. Below the graphic field, there are four buttons: “Select,” “Redraw,” “Set,” and “Delete”. In the upper right area, there are icons for the input curve, we can input up to ten curves into the system. If a user clicks on a curve icon, the corresponding curve is selected and the curve name and parameters are listed in the text field. The next click enables the user to switch between the modes “Display” and “Not display.” The curve name and parameters in the text field are updated when the user clicks “Select” or some curve icon.

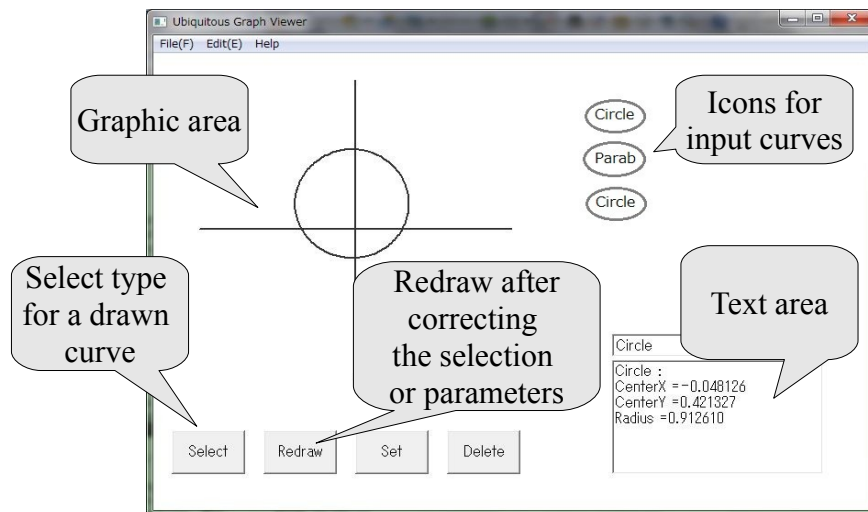


Figure 2.1 System Layout

2.3 Operation Flow

First, a user draw a curve in the graphic field. A curve is drawn only for an input task, and an old curve disappears when the user draw a new curve. When the user clicks “Select,” the system recognizes the curve type and selects a candidate. Simultaneously, the drawn curve is replaced by a compensation curve. The curve name appears in the combo box above the text field, where the parameters of the curve are displayed.

The curve name and parameters can be corrected, and when the user clicks “Redraw,” the compensation curve is updated. Then, the curve is input into the system by clicking “Set”.

The input curves are listed above the text field as small icons. By clicking on a displayed icon using the left button of a mouse, the corresponding curve is selected; then, the color of the icon changes and the parameters in the text field are updated. When the icon is clicked again, the mode changes from “Display” to “Not display” (or “Not display” to “Display”). In the “Display” mode, the corresponding curve is displayed in the graphic field, and the icon is blackened, in the “Not Display” mode, the curve disappears and the icon is grayed.

3. Curve Recognition

The recognition of the curve type is based on human intention. There are some typical shapes for each curve type, and these may take priority in terms of the local accuracy of function values. For example, a sine curve in $[0, \pi]$ may be sometimes recognized as a parabola.

3.1 Feature functions

For this recognition, we use the following feature functions. These functions describe the features of some curve types.

1. Distance from the start point to the end point.
2. Product of the direction angles at the start point and end point (direction angle is the angle between $(1,0)$ and the tangent vector.)
3. Difference between the direction angles at the start point and end point.
4. Number of stationary points.
5. Difference between the maximum direction angle and minimum direction angle (direction angles are in $[-\pi, \pi]$.)

6. Difference between two y -coordinates of the start point and end point.
7. Direction angle at the inflection point.
8. Number of the points above the x -axis.
9. Number of the points to the right of the y -axis.

3.2 Choquet integral with respect to Two-additive measure

We define an evaluation function for each curve type, and the recognition result is the curve type that attains the maximum value. The evaluation functions are defined as generalized Choquet integrals with respect to a two-additive measure.

Let A be a set of n points (here $n=9$) and μ be a fuzzy measure or general set function defined on the power set of A with $\mu(\emptyset)=0$. If μ is an additive measure, all the measure values are determined from $\mu(\{x\})$ ($x \in A$), that is, μ is defined by n values. However, a general set function is defined by 2^n values. Then, to reduce the number of parameters, we consider a two additive measure as the first generalization of non-additive measure analysis.

A two-additive measure is given by

$$\mu(A) = \sum_{x \in A} \mu_x + \sum_{\{x,y\} \subset A} \nu_{x,y},$$

using real constants $\mu_x, \nu_{x,y}$ ($x, y \in A$). Then two-additive measure on an n -point set is defined by $n+n(n-1)/2$ values. The generalized Choquet integral of a function $f(A \rightarrow [0,1])$ is given by

$$\int_A f d\mu = \sum_{x \in A} f(x)\mu_x + \sum_{\{x,y\} \subset A} f(x) \otimes f(y) \nu_{x,y},$$

where μ is a two additive measure determined by $\mu_x, \nu_{x,y}$ ($x, y \in A$) and \otimes is a Dombi t-norm defined by

$$x \otimes y = \frac{1}{((1/x-1)^\lambda + (1/y-1)^\lambda)^{1/\lambda}},$$

(here $\lambda=2.5$, (see [4],[5]).

A (generalized) Choquet integral is an extension of a weighted sum. For an element $x \in A$, $f(x)$ has a large effect on the integral value when $|\mu_x|$ is large, and for a pair $x, y \in A$, in the case where $\nu_{x,y}$ is large, the integral takes a large if both $f(x)$ and $f(y)$ take large values. Then, we are able to control the recognition abilities by using the two-additive measures.

There are nine evaluation functions, that is, we have to define nine two-additive measures for the recognition because all the feature functions are defined in relation to some graphical properties of certain curve types. Then we can manually define these two additive measures with several trial-and-error adjustments. The following table is a list of 10 times recognition results (for 4 curve types) by one test subject. He is different from the person who decided the two additive measures.

Curve Name	Rate	Errors	Curve Name	Rate	Errors
Circle	100%	No errors	Exp. fun.	70%	All Errors as poly. 3
Sin. curve	100%	No errors	Log. fun	100%	No errors

Table 3.1 Recognition Results

4. Compensation of Curves

After the system or a user determines the curve type, the curve is replaced by compensation curve as the corresponding curve type. We adopt least square error estimation with respect to 5 reference points for each drawn curve.

4.1 Reference Points

The drawn curve sometimes differs considerably from the ideal curve shape for the corresponding curve type. However, the user's intention must appear in some part of the drawn curve. We select five points on a drawn curve, that is, a sequence of two-dimensional points made by the system mouse events during mouse operation.

Let $\{\vec{p}_k\}_{k=1}^N$ be a drawn curve and $\{r_j\}_{j=1}^5$ be an increasing sequence in $[0,1]$. Then, the reference points are given by

$$\{\vec{p}_k\}_{k=1}^5, \quad k_j = [Nr_j + 0.5], \quad (j=1,2,\dots,5).$$

$\{r_j\}_{j=1}^5$ is a sequence of reference points' ratios. The starting values for the reference points ratios are equally spaced, that is,

$$\{r_j\}_{j=1}^5 = \left\{ \frac{j-1}{4} \right\}_{j=1}^5 = \{0.0, 0.25, 0.5, 0.75, 1.0\}.$$

4.2 Compensation for Polynomial Graphs

A straight line, a parabola, and the three types of cubic curves are graphs of polynomial functions. The compensation is based on linear regressions. Generally, let

$\{\vec{p}_k = (x_k, y_k)\}_{k=1}^N$ be a sequence of points and $f(x) = a_0 + \sum_{i=1}^n a_i x^i$ be a polynomial function. Then, we can obtain a_0, a_1, \dots, a_n that minimize

$$\sum_{k=1}^N (y_k - \vec{x}_k \cdot \vec{a} - a_0)^2, \quad \vec{x}_k = \begin{pmatrix} x \\ x^2 \\ \dots \\ x^n \end{pmatrix}, \quad \vec{a} = \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix}.$$

4.3 Compensation for Circles

The curve points are given by a vector-valued function $\vec{p}(\theta)$, and $\vec{p}(\theta) = (c_x, c_y) + (r \cos \theta, r \sin \theta)$.

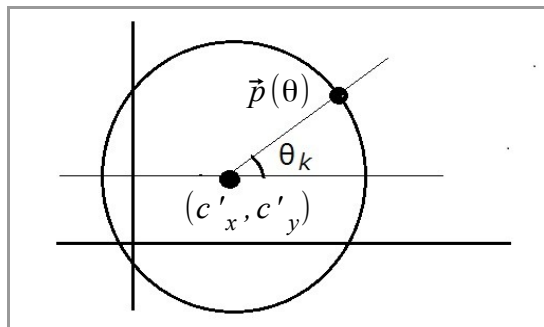


Figure 4.1 Circle

We define the tentative center point (c'_x, c'_y) by

$$(c'_x, c'_y) = \frac{1}{N} \sum_{k=1}^N (x_k, y_k)$$

and the angle $\{\theta_k\}$ by

$$\cos \theta_k = \frac{x_k - c'_x}{\sqrt{(x_k - c'_x)^2 + (y_k - c'_y)^2}},$$

$$\sin \theta_k = \frac{y_k - c'_y}{\sqrt{(x_k - c'_x)^2 + (y_k - c'_y)^2}}.$$

Using these angle parameters we estimate c_x, c_y and r to minimize

$$(x_k - c_x - r \cos(\theta_k))^2 + (y_k - c_y - r \sin(\theta_k))^2.$$

This can be done by an easy extension of linear regression.

4.4 Compensation for Sine Curve

We assume that the drawn sine curve intersects the x-axis more than two times. Our purpose is to find the optimal values of the parameters C_0, C_1 , and C_2 in the following function:

$$C_0 \sin(C_1 x + C_2).$$

Let u_0 be the first (minimum) x-intercept, u_1 be the second x-intercept and y_{\max} be the maximum value of $|y_k|$ while $x_k \in [u_0, u_1]$. Then the initial values of the above parameters are given by

$$C_0 = y_{\max}, \quad C_1 = \frac{(u_1 - u_0)}{2\pi}, \quad C_2 = -C_1 u_0.$$

Next using Newton Method in \mathbb{R}^3 we defined the parameters to minimize:

$$\sum_{i=1}^N (y_k - C_0 \sin(C_1 x_k + C_2))^2.$$

4.4 Compensation for Other Curves

For the graph of an exponential function or a logarithm function, the initial parameter values are given by the first two data of the curve. Then, using two-dimensional Newton method we obtain improved parameters.

5. Automatic Learning

There are some differences among individuals in the recognition of curve types and the compensation of curves. Therefore, we adjust the conditions or parameters for the recognition and compensation to customize the system for individuals.

5.1 History of Curves

The data for required automatic learning are stored during the actual use of the system. Once a user draws a curve and clicks "Select," then the system displays the curve type (recognition result) and the compensation curve. If these results are different from the user's intention, the curve type or the parameters are revised by the user. After that, the user inputs the curve. Simultaneously, the following information is stored in the system.

1. Point number and all point coordinates
2. The last curve type
3. All parameters for each curve type

5.2 Improvement of the Recognition

When an adequate amount of curve data (all coordinates) and the correct curve types are stored in the history data, we can repeat the same recognition process in different situation and obtain the recognition rate. Every evaluating function for curve types is determined by a two-additive measure. Hence, nine two-additive measures, that is, $9 \times (9 + 9 \times 8/2) = 324$ parameters, control the recognition, and we improve these parameters using a generic algorithm. In the present system, a gene is a 324-dimensional vector (a set of two-additive measures). We prepare L genes (the default value is 100). The procedure is as follows:

1. Set the starting values of the two-additive measures and copy them to all genes.
2. The starting values satisfy the condition that the sum of all absolute values of parameters is equal to 1 for every two-additive measure. We add i.i.d. $N(0, 0.1^2)$ random variable to all parameters of all the two-additive measures.
3. Each two-additive measure is divided by the corresponding sum of the absolute values of parameters.
4. Sort the genes by the recognition rate, and make 10 copies of the first to the L -th genes.
5. Iterate 2 ~ 4 several times

5.3 Improvement of the Compensation

We control the compensation curves by the dividing ratios $\{r_j\}_{j=1}^5$ defined in 4.1, and we also adopt the following generic algorithm. The genes are the dividing ratios (5 dimensional vectors). We prepare L genes (L is given in previous subsection)

1. All genes are initialized as the equal spaced dividing ratios.
2. We add i.i.d. $N(0, 0.1)$ to each element (ratio) in all genes.
3. For each gene $\{r_j\}_{j=1}^5$, sort the ratios in ascending order.
4. We obtain square errors using the reference points corresponding to each gene and the curve parameters in the history data.
5. Sort the genes by the square errors, and make ten copies of from the first gene to the L -th gene.
6. Iterate 2 ~ 5 several times.

6. Tactile Output

All input curves are targets for tactile output when they are selected. After some correction of the curve type and shape (parameters of the curve) for a drawn curve, a user inputs the curve. The curve is selected just after the input. After that the user can change the modes “Display” and “Not display” for each input curve. Figure 6.1 shows an example image of drawn curves and their tactile outputs. The tactile display used is Dot View (by KGS).

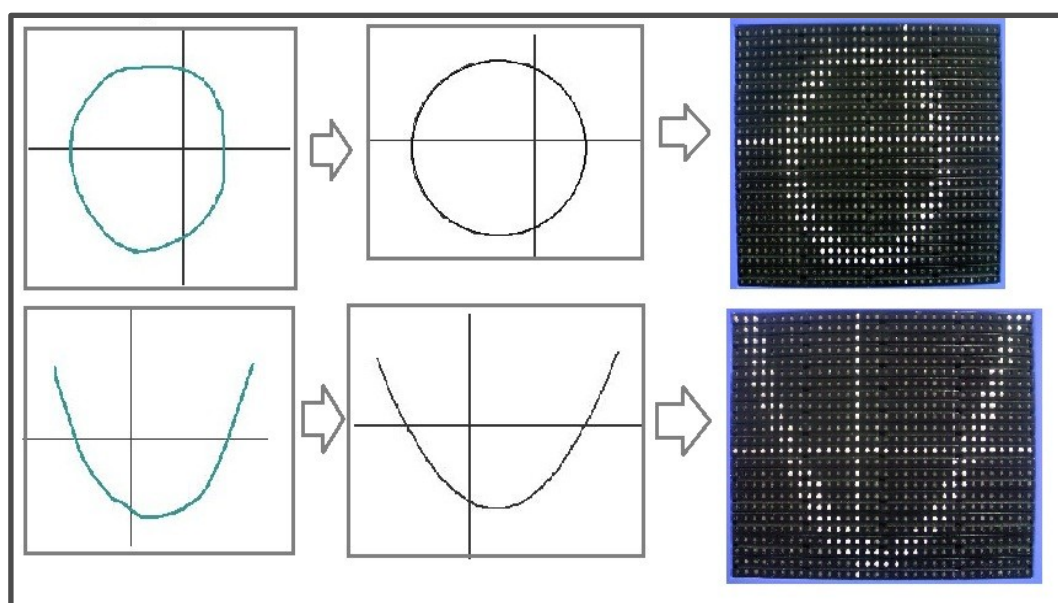


Figure 6.1 Tactile Output

7. Conclusion

We designed a curve input system that can aid visually impaired students in science classes. The curve type and corresponding parameters are automatically recognized, and the system learns how to perform the recognition and compensation according to the use of the system. The following are topics that could be addressed in future studies to improve our system.

1. Evaluation tests for the functions of the system in various situations
2. Improvement of the user interface
3. Increase of the target curve types and robust recognition system for changes of target curve types

After the above improvements, we will be able to proceed with the next step: accessibility and usability tests in actual use in scientific classes by visually impaired students.

References

- [1] Ota, H. (2006) Assistive Application Software of Notetaking for Students Who Are Deaf and Hard of Hearing, Journal of Shizuoka University of Welfare No.2
- [2] Kurita, S. IpTalk (software, <http://iptalk.hp.infoseek.co.jp/>)
- [3] Comparison of notetaking software, Web page, http://en.wikipedia.org/wiki/Comparison_of_notetaking_software
- [4] Fukuda, R. Nii, K. (2009), Handwriting Tool For Note Takers in Mathematical Classes, Proceedings of the 14th Asian Technology Conference in Mathematics
- [5] Dombi, J. A General Class of Fuzzy Operators, the DeMorgan Class of Fuzzy Operators and Fuzziness Measures Induced by Fuzzy Operators, Fuzzy Sets and Systems, 8, 149-163, 1982