

Improvements and Evaluations of Tactile Graphical Viewer for the Visually Impaired

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:** Our tactile graphical viewer recognizes the type of a handwritten input curve and displays it on a tactile display. We propose new recognition method and change function for input curves. With this recognition method, we propose a procedure to add new curve types. Using the change function, we are able to reduce the time taken to create the required curves. This system has functions to provide some graphical information as a tactile images. We present some evaluations for receiving the information and understanding the underlying properties.*

1. Introduction

Tactile representation of the visual world has been investigated in numerous studies. Various types of tactile information are available for assisting the visually impaired. Some of them are expressed in three-dimensional or higher-dimensional forms [1],[2], which are highly precise and detailed. On the other hand, some of them are expressed using raster images [3],[4]. We can express only very simple figures by using raster images; however, such images can be easily stored as electronic data and can be made available for real-time communication by using tactile displays. We have previously presented a tactile graphic viewer for basic function graphs [5]. A user draws a curve using a mouse, and the system recognizes the curve type and estimates the corresponding parameters. After several corrections, the input curve is presented on a tactile display. In a situation wherein visually impaired students attend a science class where caregivers use our system to express graphical information on a blackboard, real-time communication would take priority over detailed expression.

The previous version of our system supports eight types of basic curves. We do not support all function types dealt with in high school mathematics; moreover, new curve type may be required. We will add new curve types in the future. We used a generalized Choquet integral with respect to a two-additive measure for the previous recognition engine. For this method, we have to develop a two-additive measure for each curve type, and we have no automatic rule for creating this measure. Moreover, we have to re-evaluate all two-additive measures to add a new curve type. In this study, we propose a new method for recognizing a curve type; the method uses a vector-valued Choquet integral. We describe a procedure, in which this method is used, to add a new curve type. Using the procedure, we add two curve types to our system.

During actual system operation, we need useful change functions for the shape or position of an input curve. The system is intended to provide tactile graphical objects for real-time communication. Then, the function must be quick and simple. A user inputs a curve using a mouse and makes changes by dragging the mouse. Each curve has several parameters, and all of them must be controlled by the change functions. When a user changes a curve, several reference points are displayed on the curve. There are several change functions, and we select the change function on the basis of drag point.

After the user inputs and arranges the curves, he/she sends them to a tactile display. The curves can be displayed in a mathematical class with some explanation. For example, the graph of $y=x$ is a straight line sloping from upper right to the lower left. We can easily relate this curve shape to the property “monotone increasing.” To understand this relation, the receiver must recognize the shape “slope upward to the right.” We have to check whether the receiver recognizes the corresponding graphical properties from the output of a tactile display. We prepare some graphical content and questions related to the content, check whether the graphical properties have been understood. We obtain the ratio of correct answers and the average time taken to give an answer and use them to evaluate understanding of the receiver.

2. Outline of the System

Our system is a tactile graphical viewer for supporting the visually impaired. A user draws curves using a mouse, and the system recognizes the curve types and corresponding parameters. After arranging the positions or shapes, the curves are sent to a tactile display. This section explains the functions of the system.

2.1 Hardware

Our system runs on a standard Windows PC with a keyboard and a mouse. We use a tactile display, Dot View (DV-1, KGS Corp.), for obtaining a tactile output. The display has 768 active pins in a 24×32 dot display array.



Figure 2.1 Dot View (DV-1)

2.2 Objective of the system

We consider a situation wherein visually impaired students attend a science class in which their caregivers use our system. During the class, a teacher draws freehand graphs on the blackboard or gestures to express graphical information. After that, the teacher explains some properties using these graphs. The caregivers draw the curve and the students receive these graphical objects using tactile display. If the drawn curves are too complex, they cannot be expressed on the tactile display, the students can not understand even if the contents can be expressed on the display. Then, the caregivers select a few of the curves if necessary. In consideration of this situation, the system must satisfy the following requirements.

1. Facilitate and rapid input of graphical objects
2. Provide simple edit functions to change the position and the shape of a curve.
3. Select function to output to the tactile display.

2.2 Basic functions

In this subsection, we explain the standard use of the system. Our target curve types are circles, straight lines, parabola curves, exponential and log curves, cubic curves, sine curves, and tangent curves. The shape of a cubic function with one stationary point differs significantly from that with two stationary points. We consider these curves to be of different types. The case with no stationary point is very rare to deal with; hence, we do not consider this case. Thus, we have nine curve types. We consider a new recognition method for adding new curve types; we will explain the details of this method in the next section.

The curve type and curve parameters are recognized. Then, the hand-drawn curve is replaced by a compensated one, and the corresponding parameters are listed in the text area. Click the “Set” button to complete the input of the curve. An elliptical icon appears to the left. We have two methods to change the position and shape of an input curve. While method involves changing the parameters directly in the text area, the other involves dragging a reference point on the selected curve.

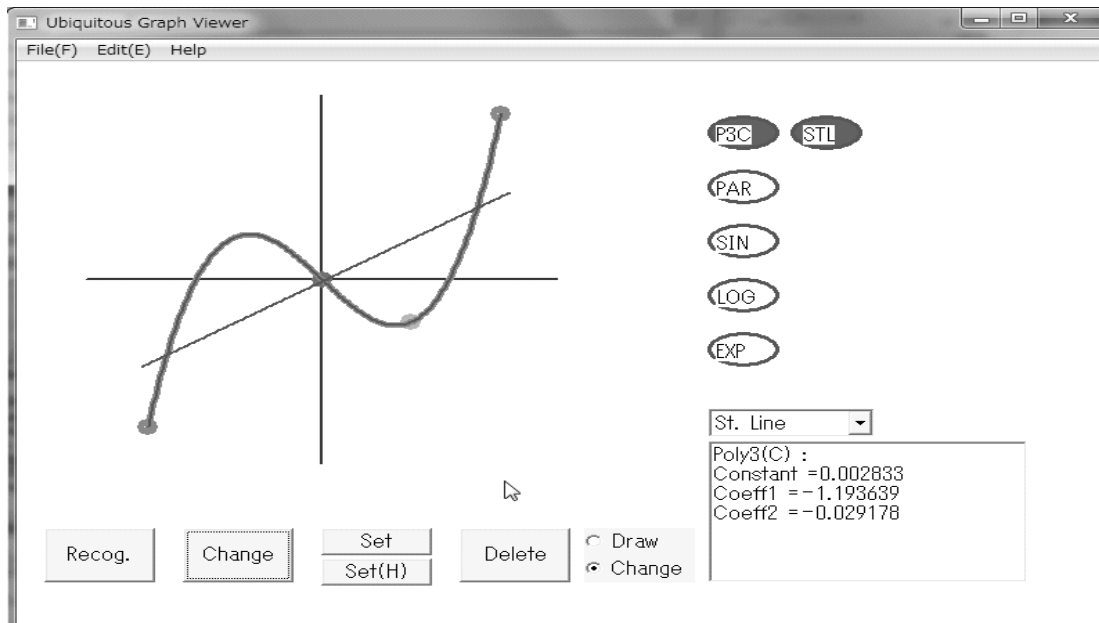


Figure 2.2 Screen shot of the system.

By left clicking an elliptical icon, we can switch between the modes “display” and “not display,” and the curves with the mode “display” are send to the tactile display.

3. Recognition Method

In our method for recognizing the curve types, qualitative measurements are quantified, and a Choquet integral with respect to a non-additive measure is a suitable tool for the analysis. In this section, we improve our recognition method for the systematic addition of recognition targets.

3.1 Recognition method using Choquet integral

For the recognition method in the previous version, we prepared nine feature values [5]:

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 points above the x -axis.
9. Number of points to the right of the y -axis.

Moreover, for each curve type, we defined a two-additive measure μ . Equivalently, we set real numbers $\{\mu_k, \nu_{j,k} : j \leq 9, j < k \leq 9\}$, and the corresponding Choquet integral is expressed by

$$\int f d\mu = \sum_{j=1}^9 f(j)\mu_j + \sum_{j < k} f(j) \otimes f(k)\nu_{j,k}, \quad (1)$$

where f is a $[0,1]$ -valued function defined on $\{1,2,\dots,9\}$. Let $\{y_k\}_{k=1}^9$ be a set of feature values, and set

$$\vec{y} = (y_k, y_j \otimes y_k)_{k \leq 9, j < k} \text{ and}$$

$$\vec{\mu} = (\mu_k, \nu_{j,k})_{k \leq 9, j < k}.$$

Then, the above integral can be expressed as

$$\int f d\mu = \vec{y} \cdot \vec{\mu}.$$

For a given curve, we obtain nine feature values. The curve type for which the corresponding Choquet integral takes the smallest value is the recognition result. Figure 2.1 shows a 1-dimensional analogy of the recognition method. There are three linear functions, and each area is expressed by an interval.

If we add a new curve type using this method, we have to control these linear functions to satisfy the condition that each feature value (vector) belongs to corresponding interval.

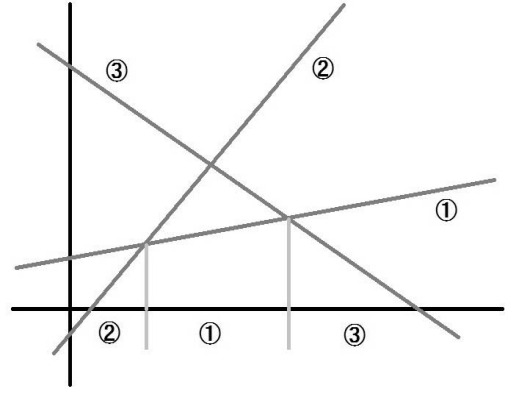


Figure 3.1 Example

3.2 Vector-type Choquet integral

Our targets are nine curve types, and we define 45 parameters $\{\mu_j, \nu_{j,k} : j \leq 9, k < j\}$ for each curve type. In other words, each μ_j or $\nu_{j,k}$ is a function defined on the set of all curve types, and the set of Choquet integrals expressed by (1) distinguish them. Functions are often treated as vectors and we use this generalization.

The generalization of our Choquet integral is given by replacing \wedge by a t-norm, and Choquet integral with respect to some simple set function expressed using \vee . Let A be a finite set, a and b be elements of A , and m be a set function defined by

$$m(B) = \begin{cases} \alpha & \text{if } (a \in B \text{ and } b \in B^c) \text{ or } (a \in B^c \text{ and } b \in B) \\ 0 & \text{otherwise} \end{cases}.$$

In general, the Choquet integral is defined for a set function whose value is zero for the empty set, and m clearly satisfies this property. Therefore, we can consider the Choquet integral of a function f defined on A , which is given by

$$\int_A f d m = \int_0^\infty m(\{x : f(x) > r\}) dr = (f(a) \vee f(b) - f(a) \wedge f(b)) \alpha. \quad (2)$$

Thus we consider the operator \vee and generalize this operator to a general t-conorm:

$$\int_A f d \vec{\mu} = \sum_{x \in A} f(x) \vec{\mu}_x + \sum_{\{x,y\} \subset A} f(x) \otimes f(y) \vec{\nu}_{x,y}^{(t)} + \sum_{\{x,y\} \subset A} f(x) \oplus f(y) \vec{\nu}_{x,y}^{(c)}, \quad (3)$$

where, \otimes is a t-norm and \oplus is a t-conorm. In this study, we use Dombi's t-norm, and the standard t-conorm defined by using Dombi's t-norm:

$$x \otimes y = \frac{1}{((1/x - 1)^\lambda + (1/y - 1)^\lambda)^{1/\lambda}} \quad (\text{in this case, } \lambda = 2.5; \text{ see [5] and [6]),}$$

$$x \oplus y = 1 - (1-x) \otimes (1-y).$$

We also use the basic unit vectors in 81-dimensional Euclidian space for the set of vectors $\{\vec{u}_j, \vec{v}_{j,k}^{(t)}, \vec{v}_{j,k}^{(c)} : j \leq 9, k < j\}$.

3.3 Adding new curve types

Using the vector-valued integral in (3) we change the recognition method as follows:

1. Obtain data for correct curve types.
2. Using feature values and the integral in (3), the curve data are transformed to groups of vectors.
3. For each group, we obtain the mean vector and the covariance matrix.
4. Define the Mahalanobis metrics for each group.
5. The nearest group is the recognition result for a given curve.

This is a standard recognition method using feature vectors, and adding new recognition targets is not difficult. We explain the method in detail.

The curve data are created using our system (previous version). We need sufficient data for each curve type (20 to 30 sets of data for each curve type). There must be several data for a standard curve shape, and we also need several data for mistakable shape comparing with other curve type. We obtain an integral vector defined by (3) for each curve. We use standard unit vectors for the definition in (3); however, these are not the optimal choice in general. We will need to obtain more adequate choices through more accurate analysis.

After we obtain a vector group for each curve type, we obtain the mean vector and the covariance matrix. The covariance matrix S is a non-negative definite symmetric matrix, and all eigenvalues are non-negative. In general, some feature values may take almost identical values in a certain group. Further, some eigenvalues take very small values and small errors will give large influences to the inverse matrix S^{-1} . To avoid this situation we consider $S + \epsilon I$ instead of S , where I is a unit matrix and $\epsilon > 0$ is a small number (we use $\{\text{largest eigenvalue}\} \times 10^{-8}$ in practice). $(S + \epsilon I)^{-1}$ is quite different from a pseudo-inverse. In this analysis, we may use a pseudo-inverse, ignoring eigenvectors corresponding to small eigenvalues. Then, the distance for the difference of these directions is zero with respect to the corresponding Mahalanobis metric. Originally, such differences in these directions are fatal errors when they belong the same group, that is, the distance must be very large.

3.4 Evaluation of recognition method

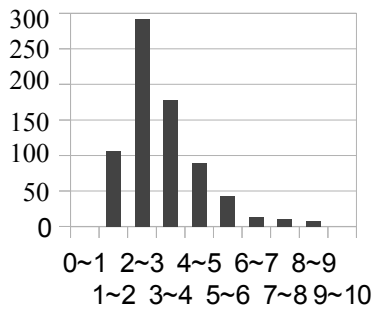


Figure 3.2 Correct Group Distances

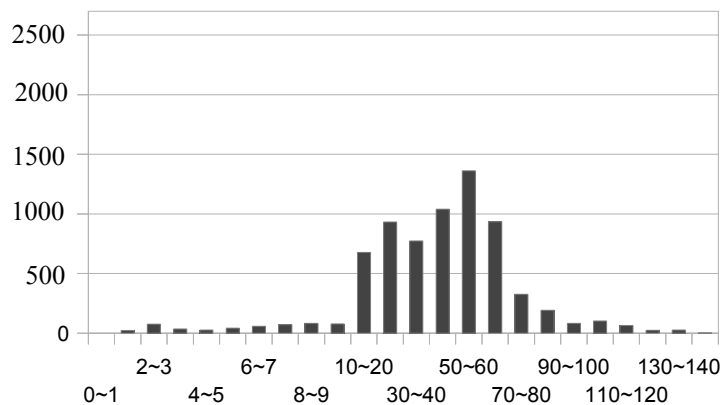


Figure 3.3 Wrong Group Distances

We create data for 780 curves. There are 10 groups, that is, 10 mean vectors and 10 Mahalanobis distances. Then we obtain 7800 distances, out of which 780 values are for correct groups and other values are for wrong groups. Figures 3.2 and 3.3 show histograms of the distances. For the incorrect group distances the corresponding set is nine times larger than that for the correct group.

4. Edit Functions

4.1 Edit functions

For the quick use of the system, we need adequate methods to change the position and shape of an input curve. Apart from the method for direct input of parameters, there are methods to change the input parameters by using a mouse.

To change all parameters, there are several reference points for each curve. The role of the mouse drag is determined according to the position of the mouse cursor. For a circle, straight line, parabola curve, and cubic curve, the first reference point is for a parallel shift in two-dimensional space. For a sine curve, an exponential and log function graph, and a Gauss function graph, the first reference point is for a one-dimensional shift. We consider only the case of the origin as an inflection point for a tangent curve, and the first reference point controls the frequency.

For sine curves, the second reference point is for changing frequency. For cubic functions the second reference point changes their local minimum (or maximum) points. In the case where there are no local minimum point or local maximum point, subtract a straight line (a linear function) crossing at inflection point, we obtain a similar reference point. For Gaussian function graphs, we change the variance parameters.

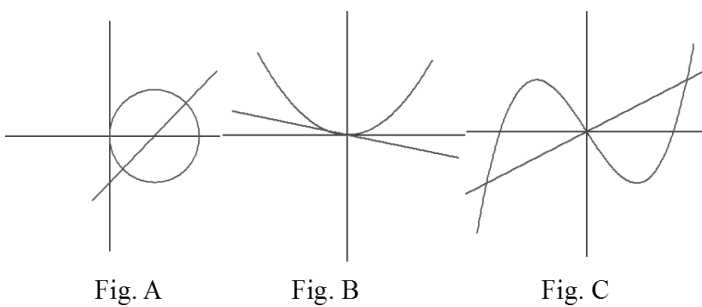


Figure 4.1 Move Test Targets

Table 4.1 Reference Points

Curve Type	1 st Ref. Pt	2 nd Ref. Pt	Edge Pt
Circle	Center	No Ref. Pt	No Edges
St. Line	Center		Left and Right Edges
Parabola Crv	Peak Pt		
Cubic Crv.	Inflection Pt	Local Min(Max) pt.	Left and Right Edges
Sine Crv.	X-Cross Pt	Local Min(Max) pt.	
Tangent Crv.	Quarter Cycle	No Ref. Pt	
Exp. Func.	Y-Cross Pt		
Log. Func.	X-Cross Pt		
Gauss Crv.	Peak Pt	Inflection Pt	

Table 4.2 Move Test Result

	Direct Input	Use Mouse
A-1	43 (sec.)	29
A-2	40	30
B-1	30	27
B-2	40	21
C-1	42	28
C-2	39	24
Average	38.2	26.5

When the mouse cursor is not on the reference points, other parameters are adjusted according to the mouse cursor position. Thus, we can change all parameters for all curve types by dragging the mouse.

4.2 Evaluation of edit function

For the evaluation of the move functions, we carried out a timekeeping test for creation of some

target figures. two persons created the three target figures. The target figures are presented in Figure 4.1, and Table 4.2 gives the results. In practice, the functions of drawing a curve, set to the system, and sending the curve to the tactile display are carried out using a mouse.

4.3 Evaluation of tactile output

The system is intended to communicate graphical information by presenting tactile information.

A concrete example is a rough shape figure. Usually, such a figure with some mathematical properties.

We examine communicative capability of tactile figures obtained with the system.

Figure 4.2 shows tactile figures obtained in the timekeeping test. Figure A includes two graphs of exponential function $y=e^{ax}$. One is for positive a and the other is for negative a . Figure B includes two log function graphs and Figure C includes a sin curve and cosine curve.

In a tactile display, one of them is output and a test subject judges it.

Figure E includes a parabola curve and a straight line. The objective of the figure is to find a graph point on the parabola curve where the gradient is same with the straight line.

Figures F to H show three positional relations between a circle and a straight line. Figures I and J show a circle and a straight line. The two straight lines passing through the origin are additional lines. These are not output in the display. Test subjects indicate the position of cross points or the touching point by the specifying area number 1 ~ 8.

The test subjects are 11 students with good eye sight, who were majoring in engineering. All of them understood the properties and shape of the graphs. After several explanations they wore an eye mask and participated in these tests.

There are a small number of mistakes (one cross point mistake, two touch point mistakes, and one same gradient point fatal mistake). Figure 4.3 shows a graph of average values of answer time for these tests.

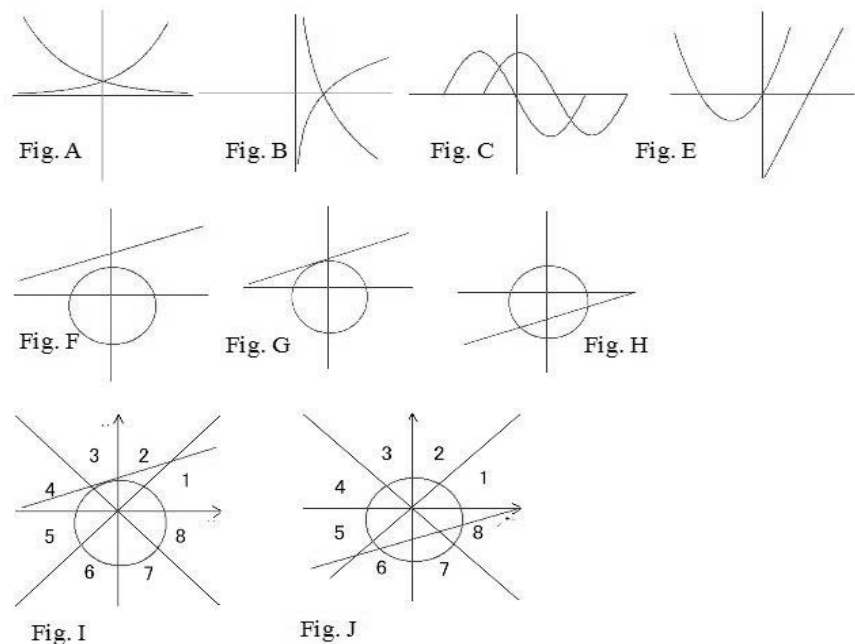


Figure 4.2 Move Test Targets

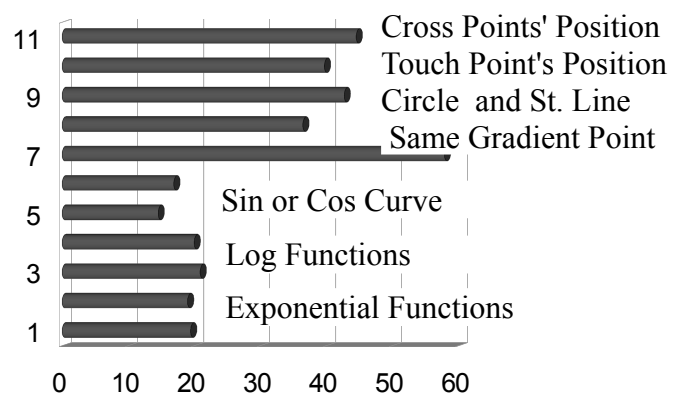


Figure 4.3 Answer Time (sec)

5. Conclusion

We improve our graphical input system by using a recognition method to increase the number of curve types by two. Some ad-hoc methods have been removed, and we can further increase the number of curve types by using this method. However, if we need to increase the number twofold or tenfold, the method will not work well. We need some general theories for the quantification of qualitative properties.

We created some edit functions for input curves, and these help us in easily providing tactile figures. We conducted a test for examining the communication capabilities of these tactile images. The test shows that they have the capability to convey the required properties. If adequate graphical content for visually impaired students is available, then this system works well in a science class. This may be a very difficult future problem.

References

- [1] Reichinger, A., Neumuller, M., Rest, F., Maierhofer, S., Pugathofer, W. Computer-Aided Design of Tactile Models Taxonomy and Case Studies, Lecture Notes in Computer Science 383 (ICCHP 2012) No.2, 2012, 497-504.
- [2] Yamazawa, K., Teshima, Y., Watanabe, Y., Ikegame, Y., Fujiyoshi, M., Oouchi, S., Kaneko, T. Three-Dimensional Model Fabricated by Layered Manufacturing for Visually Handicapped Persons to Trace Heart Shape, Lecture Notes in Computer Science 7383 (ICCHP 2012) No.2, 2012, 505-508.
- [3] Watanabe, T., Yamaguchi, T., Nakagawa, M. Development of Software for Automatic Creation of Embossed Graphs, Lecture Notes in Computer Science 7383 (ICCHP 2012) No.1, 2012,
- [4] Loitch, C., Weber, G. Viable Haptic UML for Blind People, Lecture Note in Computer Science 7383 (ICCHP 2012) No2, 2012, 509-516.
- [5] Fukuda, R. Miura, A. Ubiquitous Mathematical Graphic Viewer for Visually Impaired Students, Proceedings of the 16th Asian Technology Conference in Mathematics, 2011
- [6] Fukuda, R. Nii, K., Handwriting Tool For Note Takers in Mathematical Classes, Proceedings of the 14th Asian Technology Conference in Mathematics, 2009
- [7] 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, 1982, 149-163