EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2005)
Takeo Igarashi, Joaquim A. Jorge (Editors)

Object Snapping Method Using Multi-Resolution Fuzzy Grid Snapping Technique

Sumudu Dematapitiya, Masatoshi Kawazoe, Qamar Uddin Khand and Sato Saga

Department of Computer Science and Systems Engineering
Muroran Institute of Technology
27-1 Mizumoto-Cho, Muroran, 050-8585, Japan.
Phone:+81-143-46-5447 Facsimile:+81-143-46-5499 Email:sumudu@sagalab.csse.muroran-it.ac.jp

Abstract

We propose the automatic snapping of objects using the Multi-resolution Fuzzy Grid Snapping (MFGS) for a sketch based CAD system. The sketch based CAD system, which has already been realized, identifies sketch drawings as primitive geometric curve objects, based on the user's drawing manner. While drawing objects of varying fineness, misalignments arise due to unappropriate grid snapping resolution settings. Meanwhile, MFGS is an automatic grid snapping resolution switching mechanism, which snaps a cursor on a multiple grid system, based on the snapping intentions of the user. In this paper, we propose to solve the problem of misalignments of sketched objects of various fineness by applying MFGS to object snapping. To show the effectiveness of this method, we give experiments containing drawings of linear and circular objects of various sizes, drawn with different snapping intentions. By comparing these results with the results of the traditional snapping method, we can see that because of this newly proposed object snapping method, it has become possible to dispense with the manual switching of grid snapping resolutions. Therefore we can say that our snapping method enables the user to draw curves of different fineness, seamlessly and efficiently

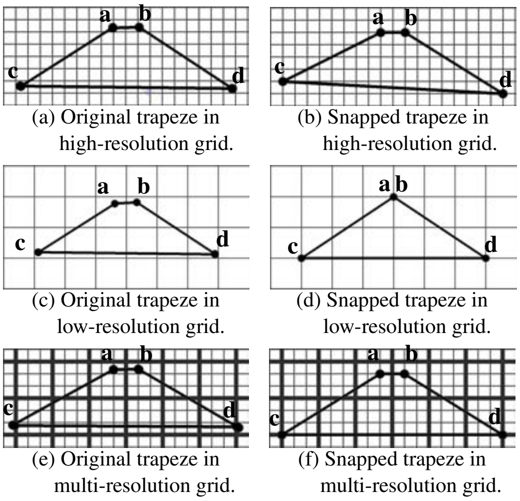
Categories and Subject Descriptors (according to ACM CCS): H.5.2 [User Interfaces]: Interaction styles

1. Introduction

We have already proposed a sketch based CAD system [Sag95, SYS03, KNS05] to help freehand drawing of objects composed of primitive geometrical curves. This system can identify freehand curves into 7 primitive geometrical curves, and align the objects by snapping them to a preset grid system.

This system works well when the grid snapping resolution is suitable to the fineness of the objects being intended. Here we use the term "fineness" to mean the degree of resolution in which the user intends to align the object. Nevertheless, problems in the setting of appropriate grid snapping resolution arise due to the fact that good resolution setting is sensitively dependent on the fineness of objects, and that the fineness of objects is different from drawing to drawing. For an example, Fig. 1(a) and Fig. 1(b) show sketch drawings on a high resolution grid. Figure 1(a) shows a sketched shape of a trapeze in the status before being snapped and Fig. 1(b) shows the status after being snapped. Coarse objects – objects intended with less fineness – drawn on a high resolution grid tend to be misaligned as seen on Fig. 1(b). On the other hand, Fig. 1(c) and Fig. 1(d) show sketch drawings on a low resolution grid. In this case, although the coarse objects are aligned correctly, the fine objects are aligned incorrectly. This implies that only if we switch the resolution when drawing, it becomes possible to draw curves of different fineness without misalignment. But practically, switching grid snapping resolution manually is very time consuming and as a result disturbs the seamless drawing process that comes naturally to us.

Now we propose the automatic switching of grid snapping resolution for each sketched object, so that we can realize the correct alignment of coarse and fine objects without manual procedures. This is illustrated in Fig. 1(e) and Fig. 1(f). Here, we are sketching on a grid system that consists of multiple grids, which have different resolutions. As seen in Fig. 1(f),



(a) Original trapeze in high-resolution grid.
(b) Snapped trapeze in high-resolution grid.
(c) Original trapeze in low-resolution grid.
(d) Snapped trapeze in low-resolution grid.
(e) Original trapeze in multi-resolution grid.
(f) Snapped trapeze in multi-resolution grid.

Figure 1: *Difference of snapping according to grid resolution.*

if the fine objects are snapped to the high resolution grid, and if the coarse objects are snapped to the low resolution grid, the sketched drawing will be aligned as we wish. The realization of this system will effectively solve our problem of the manual switching of grid snapping resolutions.

Meanwhile, we have already realized the automatic switching of a grid snapping resolution for a pointing cursor, where a careful pointing manner – the pointing manner when the user intends fine objects – is associated with high resolution cursor snapping, and a rough pointing manner – the pointing manner when the user intends coarse objects – is associated with low resolution cursor snapping. This snapping method based on fuzzy logic, is called MFGS (Multi-resolution Fuzzy Grid Snapping) [KSM04a, KSM04b]. In MFGS, the pointing manner is associated with the fuzziness of the cursor point, so that the fuzziness is small when the user's pointing manner is careful and large when the user's pointing manner is rough. Then MFGS snaps the fuzzy cursor point on a multiple grid system, which consists of grids of different resolutions. MFGS snaps the fuzzy cursor point to a higher resolution grid when the fuzziness is small and to a lower resolution grid when the fuzziness is large.

Now we go back to the snapping problem on the sketch based CAD system explained above. The sketch based CAD system is based on FSCI (Fuzzy Spline Curve Identifier) [SMS94a, SMS94b, SM93]. FSCI identifies a sketched object as one of 7 primitive curves. FSCI's identification process is based on the user's drawing manner. This is done by calculating a fuzziness which is associated with the drawing manner and then using fuzzy logic to identify the user's intentions. To achieve this, FSCI is designed so that the fuzziness generated is small when the user's drawing manner is careful and large when the user's drawing manner is rough.

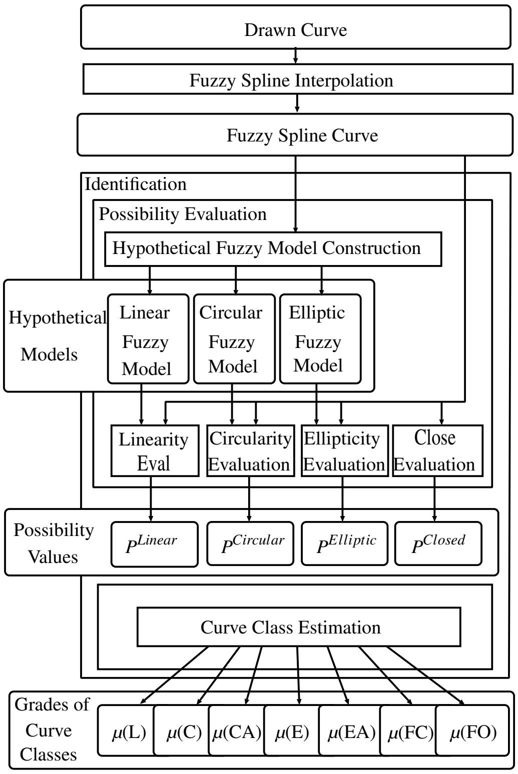


Figure 2: *Curve identification by FSCI.*

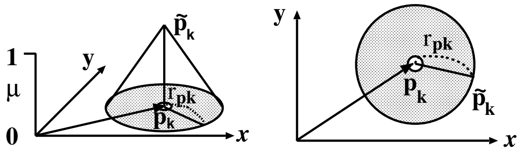
Therefore we can say that FSCI originally includes fuzziness in each point that reflects the drawing manner of the user. As a result, we propose to apply the essential idea of MFGS to objects identified by FSCI, to realize the automatic switching of grid snapping resolution.

2. FSCI

Before proposing MFGS of fuzzy objects, we need to understand the flow from hand-sketch to fuzzy primitive objects, and in what way fuzzy objects are presented in the system. Therefore, as a preparation, we introduce the outline of FSCI.

2.1. General flow of FSCI

The general flow of FSCI is shown in Fig. 2. The drawer uses a pen and a tablet to draw a curve, which is sampled and then given to the system as a sequence of points. Also the time stamps of the sampled points will be given to the system. According to the manner of drawing, this positional information will contain a vagueness. The vagueness of the positional information will increase as the roughness of the drawing manner increases. Therefore, according to the roughness of the drawing manner, the fuzziness is calculated



(a) Perspective view.

(b) Top view.

Figure 3: *Conical fuzzy point* $\tilde{\mathbf{p}}_k$.

by the fuzziness generator [SO98] so that the fuzziness will be small when the drawing manner is careful, and large when rough. Practically, the fuzziness $r_{\mathbf{p}_k}$ at a certain point $\mathbf{p}_k$ is calculated in the fuzziness generator by

$$r_{\mathbf{p}_k} = r\left(a_{\mathbf{p}_k}, v_{\mathbf{p}_k}\right), \tag{1}$$

where $a_{\mathbf{p}_k}$ and $v_{\mathbf{p}_k}$ is the acceleration and the velocity at $\mathbf{p}_k$, respectively. The concrete form of the function r is given in [SO98]. Now we will consider how a point containing fuzziness information is expressed.

To introduce vagueness into a point, it is proposed in FSCI, to express the point with a conical fuzzy point $\tilde{\mathbf{p}}_k =< \mathbf{p}_k, r_{\mathbf{p}_k} >$. Here, $\tilde{\mathbf{p}}_k$ is a fuzzy set which is characterized by the conical membership function

$$\mu_{\tilde{\mathbf{p}}_k}(\mathbf{v}) = \left(1 - \frac{\| \mathbf{v} - \mathbf{p}_k \|}{r_{\mathbf{p}_k}}\right) \vee 0, \tag{2}$$

where $\mathbf{p}_k$ is the current position of the point, $r_{\mathbf{p}_k}$ is the fuzziness that represents the vagueness of the point position and $\vee$ stands for a max operator. Figure 3 illustrates the conical fuzzy point.

After the sampled points' fuzziness is calculated, a fuzzy spline curve $\tilde{\mathbf{s}}^n(u)$, which interpolates the fuzzy points $\tilde{\mathbf{p}}_k$, is generated by the method proposed in [SM93, SMS94a]. $\tilde{\mathbf{s}}^n(u)$ is obtained as a barycentric combination of a set of conical fuzzy points $\tilde{\mathbf{p}}_k$,

$$\tilde{\mathbf{s}}^n(u) = \sum_{k=0}^{m} \tilde{\mathbf{p}}_k N_k^n(u). \tag{3}$$

This equation can also be expressed as,

$$\tilde{\mathbf{s}}^n(u) = \left\langle \sum_{k=0}^{m} \mathbf{p}_k N_k^n(u), \sum_{k=0}^{m} r_{\mathbf{p}_k} N_k^n(u) \right\rangle. \tag{4}$$

In the equations 3 and 4, $N_k^n(u)$ are normalized B-splines of degree n [Far88]. This fuzzy spline curve is an extension of an ordinary spline curve, and it can also be regarded as the locus of a fuzzy point that travels while changing its fuzziness as the parameter u changes.

FSCI will identify the fuzzy spline curve as one of seven primitive geometrical curves, and output a primitive curve. These curves are Line (L), Circle (C), Circular Arc (CA), Ellipse (E), Elliptic Arc (EA), Closed Free Curve (FC) and Open Free Curve (FO). The fuzziness, contained in the fuzzy spline curve, is used to infer the user's intentions and recognize which class of geometric curve was intended by the user. FSCI uses this fuzziness and fuzzy reasoning to produce a primitive curve. In order to identify and construct the primitive curve, FSCI calculates seven membership grades – $\mu(L)$, $\mu(C)$, $\mu(CA)$, $\mu(E)$, $\mu(EA)$, $\mu(FC)$ and $\mu(FO)$ – for each of the seven primitive curve classes and seven sets of curve shape parameters that are associated with them [SM93, SMS94b]. The identification process consists of two steps: the possibility evaluation process and the curve class estimation process.

In the possibility evaluation process, FSCI constructs three hypothetical models (see section 2.2) called linear fuzzy model, circular fuzzy model and elliptic fuzzy model. Then, each hypothetical fuzzy model is compared with the original fuzzy spline curve, and its validity is evaluated by a possibility value for each fuzzy model – P^{Linear}, $P^{Circular}$, $P^{Elliptic}$ – based on the possibility measure [L.A78]. These values show the degrees of linearity, circularity and ellipticity of the drawn curve. Apart from these three values, FSCI calculates "closedness", which expresses the degree to which the fuzzy spline curve is closed. The accordance between the fuzzy end points of the fuzzy spline curve is checked, and the closedness is evaluated by another possibility value P^{Closed}.

In the curve class estimation process, FSCI utilizes a set of fuzzy rules to calculate the seven membership grades – $\mu(L)$, $\mu(C)$, $\mu(CA)$, $\mu(E)$, $\mu(EA)$, $\mu(FC)$ and $\mu(FO)$ – from the four possibility values – P^{Linear}, $P^{Circular}$, $P^{Elliptic}$, P^{Closed}. The curve class which has the highest grade among the seven membership grades becomes the identified result. The fuzzy rules are designed to have a strategy that tries to find the simplest possible curve class. Lastly, a primitive curve model is constructed from the identified curve class and it's corresponding hypothetical model.

2.2. Fuzzy hypothetical model in FSCI

A hypothetical fuzzy model, used in FSCI is a fuzzy rational B$\acute{e}$zier curve with parameters adjusted to fit the given fuzzy spline curve as well as possible. A Fuzzy rational B$\acute{e}$zier curve is actually a special case of the fuzzy spline curve [SM93, SMS94a]. In other words, we can say that a hypothetical model is a barycentric combination of fuzzy points, which are actually the locus of a traveling fuzzy point, varying its vagueness while moving. Later on, we will be using the information derived from the hypothetical model, especially fuzziness, for snapping purposes. The general expression of the hypothetical model is

$$\tilde{\mathbf{b}}(t) = \sum_{i=0}^{n} Q_i(t)\tilde{\mathbf{q}}_i. \tag{5}$$

Here, $Q_i(t)$ is a function of time having barycentric property $\sum_{i=0}^{n} Q_i(t) = 1$, and $\tilde{\mathbf{q}}_i$ stands for a conical fuzzy point. The Hypothetical model has three variations which represent the

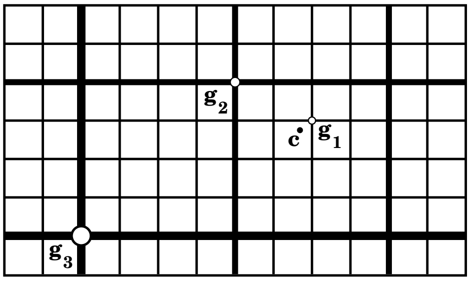


Figure 4: *Three snapping choices for point* **c**.

linear fuzzy model, the circular fuzzy model and the elliptic fuzzy model. The linear fuzzy model and the circular fuzzy model will be considered in detail in section 4.1.

3. MFGS of fuzzy points

Before considering the MFGS of objects, in this section we will consider the MFGS of fuzzy points.

In the single-resolution grid systems, a point is simply snapped with the nearest grid point. However, in the multi-resolution grid systems, there are multiple choices. Selection of the grid layer, in which the point should be snapped, depends upon the user's intention. For example, Fig. 4 shows a three layered multi-resolution grid system that includes a high-resolution grid system G_1, a middle-resolution grid system G_2 and a low-resolution grid system G_3. In this figure, $\mathbf{c}$ is the point to be snapped, while $\mathbf{g}_1$, $\mathbf{g}_2$ and $\mathbf{g}_3$ are the nearest grid points to $\mathbf{c}$ in G_1, G_2 and G_3, respectively. In this particular case, the user has three choices to snap the point, which are $\mathbf{g}_1$, $\mathbf{g}_2$ and $\mathbf{g}_3$. In addition, the user has another option, which is no snapping. In this case, if the user does not intend to snap, then the point will be snapped to itself. Thus the user has four choices that are $\mathbf{g}_1$, $\mathbf{g}_2$, $\mathbf{g}_3$ and $\mathbf{c}$. This leads to another problem of determining the intention of the user.

To overcome the problem, a snapping strategy in which the user's snapping intention is reflected on the fuzziness of a fuzzy point has been proposed [KSM04a,KSM04b]. In this strategy, low resolution snapping intentions are associated with larger fuzziness, and high resolution snapping intentions are associated with smaller fuzziness. In this way, it is possible to utilize the fuzziness to measure the intentions of the user.

MFGS first designs a fuzzy point model that represents not only the point position but also its vagueness. This is followed by the design of a multi-resolution grid system with fuzziness in its grid points. Finally, a fuzzy grid snapping technique named MFGS is realized, by embodying the above strategy.

3.1. Fuzzy point model

Here we will consider a fuzzy cursor [KSM04a, KSM04b] as an example of a fuzzy point. The fuzzy cursor can be expressed as $\tilde{\mathbf{c}} =< \mathbf{c}, r_{\mathbf{c}} >$ where $\mathbf{c}$ stands for the current position of the cursor and $r_{\mathbf{c}}$ stands for the fuzziness of $\mathbf{c}$. In the concept of the fuzzy cursor, the user's cursor pointing manner closely relates to the fuzziness of the cursor. When the user controls the cursor roughly, the generated fuzziness becomes large and when the cursor is controlled carefully, the generated fuzziness becomes small. Actually, the fuzziness of the fuzzy cursor is calculated in realtime using the speed and acceleration of the moving cursor. When the speed and acceleration of the cursor becomes large, the fuzziness generated grows large proportionally. And when the speed and the acceleration becomes small, the generated fuzziness becomes small. The reason for this association is that when the user points roughly, the vagueness of the cursor point is larger, and when the user points carefully the vagueness of the cursor point is small.

Moreover, when the user points roughly, the user is intending a coarse positioning of the cursor, implying that MFGS should use low resolution snapping for the cursor, if possible. As the generated fuzziness is large in this case, association of large fuzziness with low resolution snapping intentions in the MFGS strategy explained above can be accepted. And when the user points carefully, fine positioning of the cursor is intended, implying that MFGS should use a high resolution snapping for the cursor. Again, as the generated fuzziness is small in this case, association of small fuzziness with high resolution snapping intentions is also acceptable.

3.2. Multi-resolution grid system

In MFGS, an n-layered multi-resolution grid system is defined as a combination of single-resolution grid systems $G_i(i = 1,2,\ldots,n)$, each of which has two properties S_{G_i} and r_{G_i}. Here, S_{G_i} and r_{G_i} are the stride and the fuzziness of a grid G_i, respectively. In the grid system, it is assumed that $G_i(i = 1,2,\ldots,n)$ are in descending order of resolution. Therefore, MFGS simply gives the smallest value to S_{G_1} and the largest value to S_{G_n}. On the other hand, MFGS uses the fuzziness for each grid to represent the covering area of each grid point. The covered area is considered to be small for a high-resolution grid system, but large for a low-resolution grid system. Therefore, we assign the smallest amount of fuzziness to r_{G_1}, and the largest amount of fuzziness to r_{G_n}.

3.3. Fuzzy grid snapping

The snapping strategy for the multi-resolution grid system discussed above is realized through MFGS. For simplicity, we assume that the number of layers n is 3, without losing generality.

First, the system selects one grid point $\mathbf{g}_i$ that is nearest to

Table 1: *Rules of MFGS.*

Rule 1		
$\mu(\mathbf{g}_3)$	$=$	$N^{\mathbf{g}_3}$
Rule 2		
$\mu(\mathbf{g}_2)$	$=$	$(1-N^{\tilde{\mathbf{g}}_3}) \;\wedge\; N^{\mathbf{g}_2}$
Rule 3		
$\mu(\tilde{\mathbf{g}}_1)$	$=$	$(1-N^{\tilde{\mathbf{g}}_3}) \;\wedge\; (1-N^{\tilde{\mathbf{g}}_2}) \;\wedge\; N^{\tilde{\mathbf{g}}_1}$
Rule 4		
$\mu(\tilde{\mathbf{c}})$	$=$	$(1-N^{\tilde{\mathbf{g}}_3}) \;\wedge\; (1-N^{\tilde{\mathbf{g}}_2}) \;\wedge\; (1-N^{\tilde{\mathbf{g}}_1})$

the fuzzy point $\tilde{\mathbf{c}}$ from each grid system G_i, and uses it as a snapping candidate. Second, the system replaces each snapping candidate $\mathbf{g}_i$ with a conical fuzzy point $\tilde{\mathbf{g}}_i =< \mathbf{g}_i, r_{\mathbf{g}_i} >$, where $r_{\mathbf{g}_i}$ is the fuzziness that inherits the fuzziness of the grid G_i, which is r_{G_i}. Third, the system evaluates each snapping candidate with necessity $N^{\tilde{\mathbf{g}}_i}$. Here, $N^{\tilde{\mathbf{g}}_i}$ is the necessity of the fuzzy proposition "$\tilde{\mathbf{g}}_i$ is $\tilde{\mathbf{c}}$" [Zad78]. The necessity is defined as,

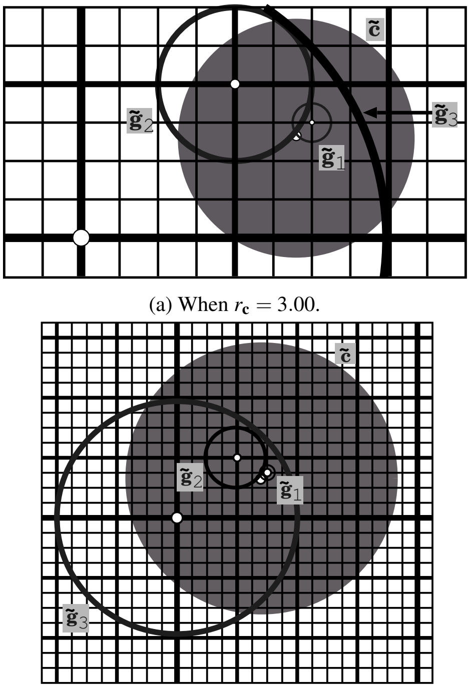
$$\begin{aligned} N^{\tilde{\mathbf{g}}_i} &= Nec_{\tilde{\mathbf{g}}_i}(\tilde{\mathbf{c}}) \\ &= \inf_{\mathbf{v}}((1-\mu_{\tilde{\mathbf{g}}_i}(\mathbf{v})) \vee \mu_{\tilde{\mathbf{c}}}(\mathbf{v})) \end{aligned} \tag{6}$$

according to [Zad78] and [KY95]. In this particular case where $\tilde{\mathbf{g}}_i$ and $\tilde{\mathbf{c}}$ have conical fuzzy membership functions, the system can easily calculate the necessity by

$$Nec_{\tilde{\mathbf{g}}_i}(\tilde{\mathbf{c}}) = \left(\frac{r_{\mathbf{c}} - \parallel \mathbf{g}_i - \mathbf{c} \parallel}{r_{\mathbf{c}} + r_{\mathbf{g}_i}} \right) \vee 0. \tag{7}$$

Fourth, the system performs fuzzy reasoning by applying the rules shown in Table 1, and then evaluates the snapping candidates with grades $\mu(\tilde{\mathbf{g}}_3)$, $\mu(\tilde{\mathbf{g}}_2)$, $\mu(\tilde{\mathbf{g}}_1)$ and $\mu(\tilde{\mathbf{c}})$. In this table, the $\wedge$ symbol stands for a min operator – that is logical *And* operator – and $(1-N^{\tilde{\mathbf{g}}_i})$ is the negation of $N^{\tilde{\mathbf{g}}_i}$. The rules shown in Table 1 can be translated as follows:

- Rule 1 recommends $\tilde{\mathbf{g}}_3$ as a snapping point $\mathbf{g}_s$ if $\tilde{\mathbf{g}}_3$ is necessarily in $\tilde{\mathbf{c}}$,
- Rule 2 recommends $\tilde{\mathbf{g}}_2$ as a snapping point $\mathbf{g}_s$ if $\tilde{\mathbf{g}}_3$ is not necessarily in $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{g}}_2$ is necessarily in $\tilde{\mathbf{c}}$,
- Rule 3 recommends $\tilde{\mathbf{g}}_1$ as a snapping point $\mathbf{g}_s$ if $\tilde{\mathbf{g}}_3$ is not necessarily in $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{g}}_2$ is not necessarily in $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{g}}_1$ is necessarily in $\tilde{\mathbf{c}}$,
- Rule 4 recommends $\tilde{\mathbf{c}}$ as a snapping point $\mathbf{g}_s$ if $\tilde{\mathbf{g}}_3$ is not necessarily in $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{g}}_2$ is not necessarily in $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{g}}_1$ is not necessarily in $\tilde{\mathbf{c}}$.



(a) When $r_{\mathbf{c}} = 3.00$.

(b) When $r_{\mathbf{c}} = 9.00$.

Figure 5: *Snapping candidates and fuzzy cursor in multi-resolution grid system.*

Table 2: *Evaluated grades of snapping candidates according to fuzziness* $r_{\mathbf{c}}$.

$r_{\mathbf{c}}$	$\mu(\tilde{\mathbf{g}}_3)$	$\mu(\tilde{\mathbf{g}}_2)$	$\mu(\tilde{\mathbf{g}}_1)$	$\mu(\tilde{\mathbf{c}})$
1.50	0.00	0.00	0.46	0.53
3.00	0.00	0.17	0.69	0.30
9.00	0.16	0.62	0.37	0.11
22.00	0.52	0.47	0.17	0.04

The rules imply that the system will try to snap the fuzzy cursor with the lowest resolution grid point as long as there is necessity. Finally, the system determines the grid candidate that has the highest grade and selects it as snapping point $\mathbf{g}_s$. At this point, if $\mu(\tilde{\mathbf{c}})$ has the highest grade, then the current point $\mathbf{c}$ is selected as snapping point $\mathbf{g}_s$. This is the case where the system inferred that the user doesn't want to snap the point with any grid point.

To demonstrate how MFGS works for the case shown in Fig. 4, let us set the strides as $S_{G_1} = 1.00$, $S_{G_2} = 4.00$, $S_{G_3} = 16.00$ and the fuzziness as $r_{G_1} = 0.50$, $r_{G_2} = 2.00$, $r_{G_3} = 8.00$. Then, let us set the fuzzy cursor's fuzziness $r_{\mathbf{c}}$ with four different values 1.50, 3.00, 9.00 and 22.00. Figures 5(a) and

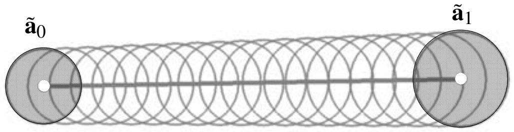


Figure 6: *Linear fuzzy model.*

5(b) illustrate the case of $r_{\mathbf{c}} = 3.00$ and the case of $r_{\mathbf{c}} = 9.00$, respectively.

Table 2 shows the evaluated grades of snapping candidates according to the fuzziness $r_{\mathbf{c}}$. On the basis of the table, we can say that the larger amount of fuzziness the cursor has, the lower resolution snapping the system selects. This fact confirms that MFGS coincides with the proposed snapping strategy.

4. MFGS of fuzzy objects

The Sketch based CAD System recognizes sketched strokes as objects using the FSCI mechanism as explained in section 2. In this paper, we propose to snap these objects using MFGS to achieve seamless sketching ability without the need of manual switching of grid snapping resolution. In this section, we will discuss how fuzzy points are extracted from the identified objects, and how the snapping of the objects is achieved.

4.1. Extraction of fuzzy points from objects

The actual snapping of the objects is done by snapping their control points to the present grid. These control points are points chosen so that they can express any unique instance of the particular object. If we consider these control points to be fuzzy points, we can apply MFGS to these control points and, as a result, use MFGS for object snapping.

First, we will focus on implementing MFGS to the simplest primitive curve, which is the line. To implement MFGS to a primitive line object, control points of the line have to contain fuzziness. The hypothetical model of a line is shown in Fig. 6 , and explained by

$$\tilde{\mathbf{b}}(t) = (1-t)\tilde{\mathbf{a}}_0 + t\tilde{\mathbf{a}}_1 \qquad t \in [0,1], \tag{8}$$

which is a special case of the general expression of the hypothetical model given in Eq. 5. Here, $\tilde{\mathbf{a}}_0$ and $\tilde{\mathbf{a}}_1$ are fuzzy end points on the linear fuzzy model. We will extract these two control points for the line and later use MFGS to decide which level of grid each control point should be snapped to.

Secondly, let us consider the snapping of a primitive object of a circle or a circular arc. It must be noted that a circle is a special case of a circular arc. Therefore, we will consider the circular arc mainly.

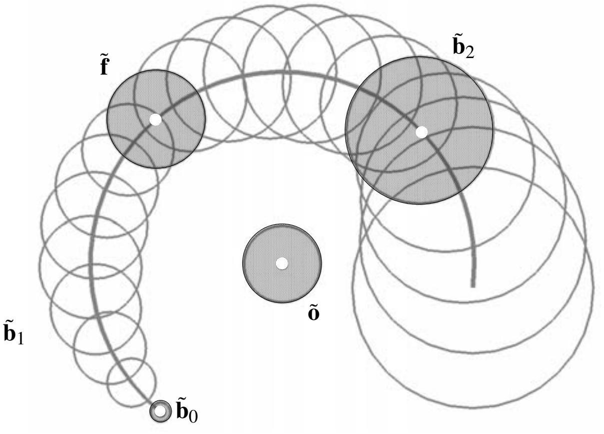


Figure 7: *Circular fuzzy model.*

In the case of the circular arc, the hypothetical model used is called the R-model, which was first introduced in [TYS02] based on [SM93]. In Fig. 7 we present the R-model of the circular fuzzy model

$$\tilde{\mathbf{b}}(t) = \frac{P_0(t)\tilde{\mathbf{b}}_0 + P_1(t)(1+w)\tilde{\mathbf{f}} + P_2(t)\tilde{\mathbf{b}}_2}{P_0(t) + P_1(t)(1+w) + P_2(t)} \qquad t \in [0,t_e], \tag{9}$$

which is also a special case of the general equation of the hypothetical model given in Eq. 5. In the above equation, $\tilde{\mathbf{b}}_0$, $\tilde{\mathbf{b}}_2$, and $\tilde{\mathbf{f}}$ are fuzzy points on the circular fuzzy model and w is a real value between 0 and 1. $P_0(t)$, $P_1(t)$ and $P_2(t)$ used in Eq. 9 are explained as,

$$P_0(t) = B_0^2(t) - \frac{1}{2}B_1^2(t) \tag{10}$$

$$P_1(t) = B_1^2(t) \tag{11}$$

$$P_2(t) = B_2^2(t) - \frac{1}{2}B_1^2(t). \tag{12}$$

Here, $B_i^2(t)$ are the Bernstein polynomials of the 2nd order. A detailed explanation of $\tilde{\mathbf{b}}_0$, $\tilde{\mathbf{b}}_2$, $\tilde{\mathbf{f}}$ and w is given in [TYS02].

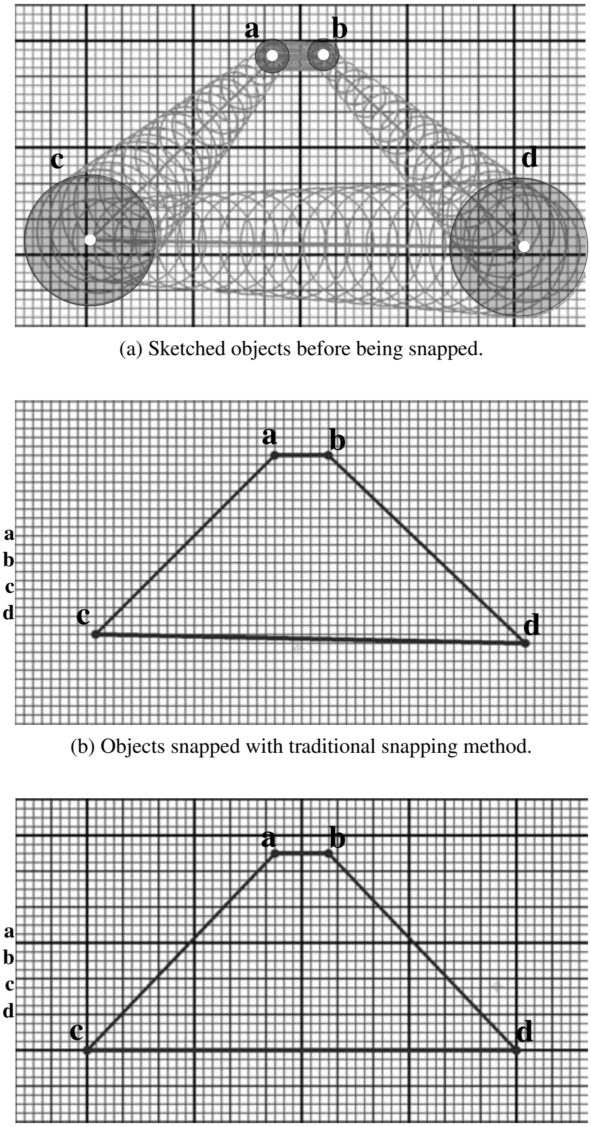
In Fig. 7, $\tilde{\mathbf{o}}$ stands for the center of the circle or the circular arc. The point $\tilde{\mathbf{o}}$ is explained as

$$\tilde{\mathbf{o}} = \frac{\tilde{\mathbf{b}}_0 + \tilde{\mathbf{b}}_2 - 2w\tilde{\mathbf{f}}}{2(1-w)}, \tag{13}$$

calculated based on the properties of the rational B*é*zier curves. From this circular fuzzy model, we extract the beginning point of the circular arc $\tilde{\mathbf{b}}_0$, and the center of the arc $\tilde{\mathbf{o}}$. These control points will be used when applying MFGS to circular objects.

4.2. Snapping of objects

As two control points have been extracted (see section 4.1) in both cases of linear and circular objects, we can use the same procedure, explained below, to snap them.



(a) Sketched objects before being snapped.

(b) Objects snapped with traditional snapping method.

(c) Objects snapped with MFGS.

Figure 8: *Snapping comparison of lines.*

By applying MFGS to each of the fuzzy control points extracted in section 4.1, we will calculate the most suitable grid level for each point and consequently, the most suitable grid point to be snapped to. Then, the corresponding object will be remodeled using the coordinate information of the original control points and the snapped points. The remodeling is based on a similar transformation matrix calculated using the control points' coordinate change. Then the result will be shown to the user.

5. Experimental results

In this section, we will consider the experimental results which show the effect of MFGS, and compare them with the traditional snapping method. We will explain the experimental results concerning the snapping of line objects and of circular objects in the next two subsections.

5.1. Experimental results of line objects

In Fig. 8 we compare the traditional snapping and MFGS of line objects.

As seen in Fig. 8(b), the traditional snapping system makes it difficult for the user to draw long line objects like line $[\mathbf{c},\mathbf{d}]$, drawn roughly on a high resolution grid, because it usually tends to result in misarrangements. If the grid snapping resolution is lowered to solve this problem, carefully drawn small lines cannot be drawn correctly.

As seen in Fig. 8(c), MFGS solves this problem by snapping each line to the appropriate grid level. We can see that the end points of line $[\mathbf{a},\mathbf{b}]$ are snapped to the highest resolution grid, and the end points of line $[\mathbf{c},\mathbf{d}]$ are snapped to the lowest resolution grid. In Fig. 8(a) we can see that the fuzziness of the end points of $[\mathbf{a},\mathbf{b}]$ are small due to the careful drawing manner, which in turn caused MFGS to snap to the high resolution grid. And we can see that the fuzziness of the end points of $[\mathbf{c},\mathbf{d}]$ has quite a large fuzziness due to the rough drawing manner, which in turn caused MFGS to snap to the low resolution grid.

If we closely look at Fig. 8(c), we can see that in the case of the lines $[\mathbf{a},\mathbf{c}]$ and $[\mathbf{b},\mathbf{d}]$, the end points are snapped to grids of different resolutions. If we look at Fig. 8(a) we can see the explanation for this phenomenon. In line $[\mathbf{a},\mathbf{c}]$ and $[\mathbf{b},\mathbf{d}]$, both careful and rough drawing manner appear in one line segement, causing the fuzziness to vary. As a result MFGS recognizes the appropriate grid snapping resolution for each end point and snaps them, resulting in a drawing that complies with the intentions of the user.

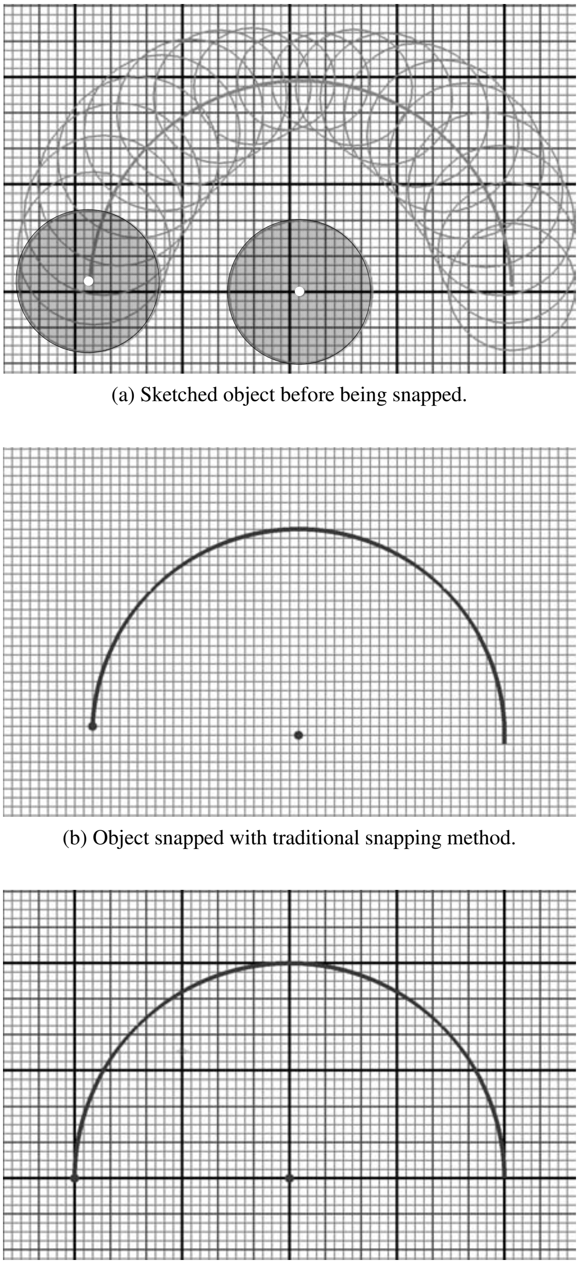
This clearly shows that while drawing line objects with different sizes, MFGS lets the user snap a specific line to a suitable grid level. The users intention is conveyed to the system purely by drawing manners which come naturally to the user.

5.2. Experimental results of circular objects

In Fig. 9 and Fig. 10, we compare the traditional snapping and MFGS of circular objects.

Figure 9 shows a drawing of a circular arc with a large radius. In the case of traditional snapping, we can see in Fig. 9(b) that it is difficult to snap this circular arc on a high resolution grid. The end points and the center point tend to drift from the point we want it to be snapped to, and result in misarrangement. In the case of MFGS, we can see in Fig.

9(c) that the circular arc has been correctly aligned. The user draws a large circular arc with a rough drawing manner making the fuzziness large. This fuzziness, in turn, makes MFGS snap the circular arc to a low-resolution grid.

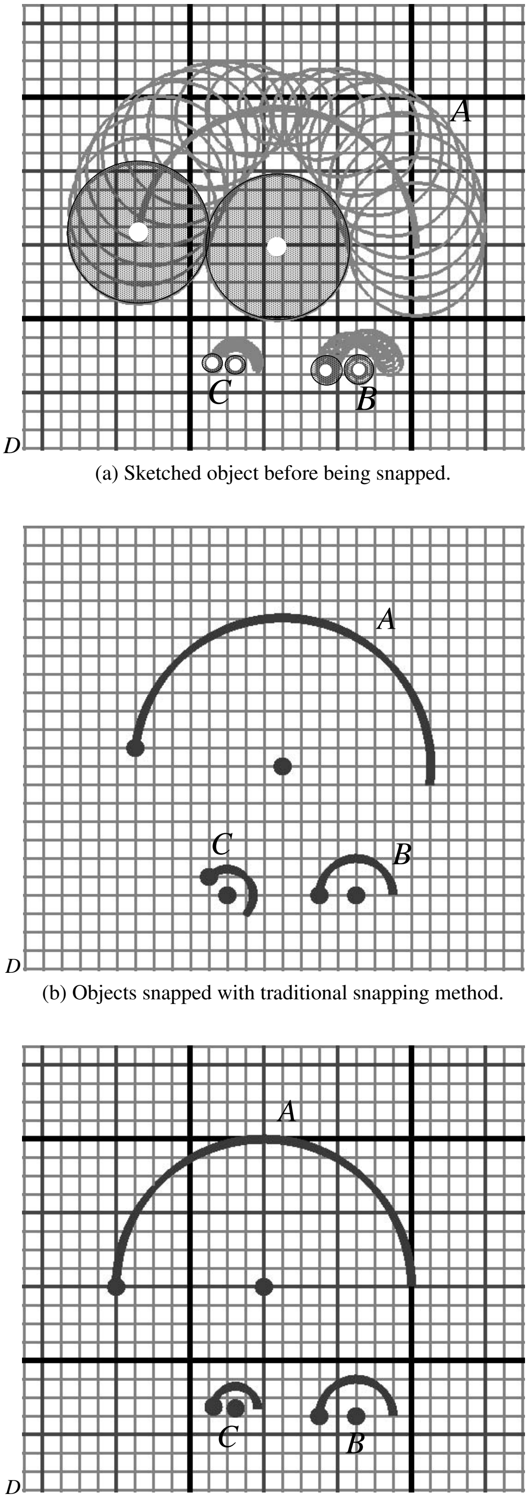

(a) Sketched object before being snapped.

(b) Object snapped with traditional snapping method.

(c) Object snapped with MFGS.

Figure 9: *Snapping comparison of circular arc of large size.*

Curve *A* in Fig. 10 is a circular arc with a medium radius. In the case of traditional snapping, we can see in Fig. 10(b) that it is difficult to snap this circular arc on a high resolution grid. The end points and the center point tend to



(a) Sketched object before being snapped.

(b) Objects snapped with traditional snapping method.

(c) Objects snapped with MFGS.

Figure 10: *Snapping comparison of circular arcs of different sizes.*

drift from the point we want it to be snapped to, and result in misarrangement. In the case of MFGS, we can see in Fig. 10(c) that the circular arc has been correctly aligned. The user draws a medium circular arc with less roughness than in the case of the large circular arc, making the fuzziness medium. This fuzziness, in turn, makes MFGS snap the circular arc to a medium-resolution grid.

Curve *B* in Fig. 10 is a circular arc with a small radius. In the case of traditional snapping, we can see in 10(b) that the circular arc has been snapped correctly. This is because by chance, the grid snapping resolution of the grid has been appropriate for the size of the circular arc. In the case of MFGS also, we can see in Fig. 10(c) that the circular arc has been correctly aligned. The user draws the small circular arc with a small roughness or, in other words, more carefully than the bigger curves, making the fuzziness small. This fuzziness, in turn, makes MFGS snap the circular arc to a high-resolution grid.

Curve *C* in Fig. 10 is a circular arc with a very small radius. In the case of traditional snapping, we can see in 10(b) that it is very difficult to snap this circular arc on a low-resolution grid. As the grid has a low resolution setting compared with the size of the circular arc, the snapping makes the circular arc completely misarranged. In the case of MFGS, we can see in Fig. 10(c) that the circular arc has been correctly aligned. The user draws a very small circular arc with no roughness or, in other words, much more carefully than the bigger curves, making the fuzziness very small. This fuzziness, in turn, makes MFGS choose not to snap the circular arc to any of the grids.

The above experimental results show that, while drawing circular objects with different sizes, MFGS snaps a specific circular object to a suitable grid with the appropriate level of grid snapping resolution, automatically. The user's intention of snapping has been conveyed to the system by means of a natural drawing manner.

6. Conclusions

In this paper, we have presented a new object snapping technique based on MFGS that realizes automatic object snapping for multi-resolution fuzzy grid systems. The presented object snapping technique, dynamically selects a grid snapping resolution appropriate for the sketched object, and snaps it according to the user's intentions, which is derived from the drawing manners. Experimental results clearly show that MFGS is an effective object snapping technique, which relieves the user from manual switching of grid snapping resolution, while drawing objects of different fineness. As a result, seamless sketching becomes possible without hampering the natural flow of drawing.

Although there are 7 classes of curves used in the sketch based CAD system, we have discussed only 3 of them in this paper – line, circle and circular arc. This is because these curves can be snapped using only two points as control points, while curves such as elliptical curves have more requirements such as the snapping of axes and so on. Therefore, more research has to be done in order to introduce Multi-resolution Fuzzy Grid Snapping to these objects. Furthermore, evaluation experiments have to be done, to quantitatively indicate how efficient this new technique of object snapping is, compared with the traditional snapping method.

References

[Far88] FARIN G.: *Curves and Surfaces for Computer Aided Geometric Design : A Practical Guide*. Academic Press, 1988. 3

[KNS05] KAWAI R., NISHIKAWA A., SAGA S.: A Freehand Sketch Input Front-End Processor: SKIT (in Japanese). *IEICE Trans. J88-D-II*, 5 (May 2005), 897–905. 1

[KSM04a] KHAND Q. U., SAGA S., MAEDA J.: Automatic Cursor Snapping into Muliti-Resolution Grid Systems Based on Fuzzy Model. *IPSJ 45*, 10 (Oct 2004), 2439–2442. 2, 4

[KSM04b] KHAND Q. U., SAGA S., MAEDA J.: Cursor Snapping into Multi-Resolution Grid Systems Based on Fuzzy Point Model. In *Proc. SCIS and ISIS 2004 (CD-ROM)* (2004). 4 pages (WE-1-3) , Yokohama, Japan. 2, 4

[KY95] KLIR G. J., YUAN B.: *Fuzzy Sets and Fuzzy Logic-Theory and Applications*. Prentice Hall, 1995. 5

[L.A78] L.A.ZADEH: Fuzzy Sets As a Basis for a Theory of Possibility. *Fuzzy Sets Systems 1*, 1 (1978), 3–28. 3

[Sag95] SAGA S.: A Freehand Interface for Computer Aided Drawing Systems Based on the Fuzzy Spline Curve Identifier. In *Proc. 1995 IEEE International Conference on Systems, Man and Cybernetics* (Oct. 1995), pp. 2754–2759. Vancouver, Canada. 1

[SM93] SAGA S., MAKINO H.: Fuzzy Spline Interpolation and Its Application to On-Line Freehand Curve Identification. In *Proc. 2nd IEEE International Conference On Fuzzy Systems* (1993), pp. 1183–1190. San Francisco, USA. 2, 3, 6

[SMS94a] SAGA S., MAKINO H., SASAKI J.: Method for Modeling Freehand Curves–The Fuzzy Spline Interpolation (in Japanese). *IEICE Trans. J77-D-II* (Aug 1994), 1610–1619. 2, 3

[SMS94b] SAGA S., MAKINO H., SASAKI J.: The Fuzzy Spline Curve Identifier (in Japanese). *IEICE Trans. J77-D-II* (Aug 1994), 1620–1629. 2, 3

[SO98] SAGA S., OHKAWA T.: Refinement of Fuzziness Generator for the Freehand Curve Identifier FSCI. In *Proc. 2nd IEEE International Conference of Intelligent Processing Systems, Gold Coast, Australia* (Aug 1998), pp. 323–329. 3

[SYS03] SATO Y., YASUFUKU N., SAGA S.: Sequential Fuzzy Spline Curve Generator for Drawing Interface by Sketch (in Japanese). *IEICE Trans. J86-D-II*, 2 (Feb 2003), 242–251. 1

[TYS02] TAKIKAWA H., YASUFUKU N., SAGA S.: Improvement of Identification Algorithms in the Freehand Curve Identifier FSCI (in Japanese). *IEICE Trans. J85-D-II*, 11 (Nov 2002), 1683–1691. 6

[Zad78] ZADEH L. A.: Fuzzy Sets as a Basis for a Theory of Possibility. *Fuzzy Sets and Systems 1*, 1 (1978), 3–28. 5