

## A Multi-Resolution Grid Snapping Technique Based on Fuzzy Theory

QAMAR UDDIN KHAND,<sup>†</sup> SUMUDU DEMATAPITIYA,<sup>†</sup> SATO SAGA<sup>†</sup>  
and JUNJI MAEDA<sup>†</sup>

This paper presents a grid snapping technique, called multi-resolution fuzzy grid snapping (MFGS), that enables automatic mouse cursor snapping for a multi-resolution grid system. Quick and frequent switching between high- and low-resolution grid snapping is essential to make geometrical drawings that include both fine and coarse structures when using CAD systems with ordinary single-resolution grid systems. MFGS is intended to relieve users of this tedious manual switching. MFGS dynamically selects an appropriate snapping resolution level from a multi-resolution grid system according to the pointing behavior of the user. MFGS even supports an extremely fine grid resolution level, which is referred to as the no-snapping level. We show experimental results which demonstrate that MFGS is an effective grid snapping technique that speeds up low-resolution grid snapping while retaining the ability to snap to high-resolution grids. Furthermore, we examine the role of fuzziness in MFGS and its effect on snapping performance.

### 1. Introduction

We have developed a cursor snapping technique — multi-resolution fuzzy grid snapping (MFGS) — which snaps a cursor into a multi-resolution grid system by automatically selecting an appropriate grid resolution level. This selection takes into account the pointing behavior of the user as well as the cursor position.

As the alignment of objects is essential for geometrical drawings, most currently available drawing systems are equipped with object alignment functions based on various constraints. To enhance the alignment functions, many studies have examined constraint-based drawing systems, such as Sketchpad<sup>1)</sup>, Briar<sup>2)</sup> and Snap-Dragging<sup>3)</sup>, that increase the variety of constraints. However, these studies have not focused on automatic resolution setting for the constraints. When alternately drawing fine structures and coarse structures with current drawing systems, the user must frequently switch the grid resolution to achieve optimum snapping, which makes drawing a time-consuming task. Automatic grid resolution selection promises to become a key technology that will improve alignment functions.

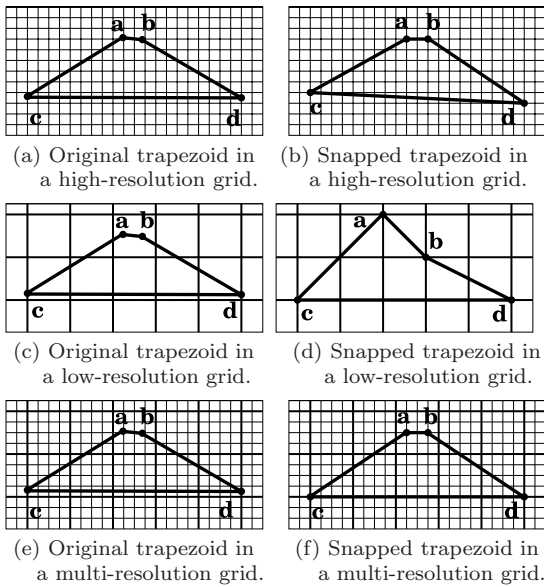
Automatic control of grid resolution for object snapping is provided by HyperSnapping<sup>4)</sup>, which allows a user to control the snapping resolution level by simply dragging objects. How-

ever, because this technique switches the snapping resolution depending on the travel distance of dragged objects from their previous position, the user occasionally must repeat the dragging operation several times to obtain the intended resolution. In contrast, MFGS is a cursor snapping technique that infers the intention of the user with regard to snapping resolution directly from the pointing behavior.

When the fineness of object alignment varies from coarse to fine, the natural pointing behavior of the user varies from rough movements to careful movements. Therefore, MFGS is designed to select low-resolution grids when the user points with rough movements and high-resolution grids when the user points with careful movements. As a result, the user is able to control the snapping resolution through natural pointing behavior.

Here, to illustrate why the selection of grid resolution is required, let us consider the example of forming a trapezoid, as shown in **Fig. 1**. For the user to align the fine parts of the structure, **a** and **b**, the grid must be set to a high resolution. However, while aligning the coarse parts of the structure, **c** and **d**, such a high-resolution grid setting hinders easy pointing operations. In other words, the grid resolution is too high for the purpose of aligning **c** and **d**. As a result, slight inaccuracies in the pointing of the user will cause points **c** and **d** to be misaligned, as shown in Fig.1 (b). On the other hand, if the grid is set with a low reso-

<sup>†</sup> Muroran Institute of Technology

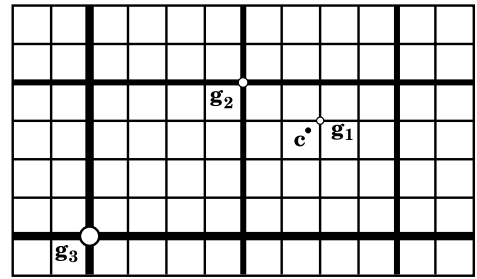


**Fig. 1** Differences in snapping according to grid resolution.

lution, as shown in Fig. 1 (c), so that the user can easily align **c** and **d**, it becomes impossible to correctly align **a** and **b** as intended, as shown in Fig. 1 (d). Therefore, to achieve easy and correct alignment, the user needs to switch the grid resolution according to the fineness or coarseness of the structure.

In MFGS, we consider drawing on a multi-resolution grid system that is a combination of several grids of different resolution, as shown in Fig. 1 (e) and (f). MFGS will automatically select the appropriate grid resolutions for the points **a**, **b**, **c** and **d**, according to the pointing behavior of the user. Here, careful pointing operations by the user for **a** and **b** cause MFGS to select the high-resolution grid for these points. On the other hand, rough pointing operations for **c** and **d** cause MFGS to select the low-resolution grid for these points. As a result, the user will be able to align the points as intended, as shown in Fig. 1 (f).

The core concept of MFGS was introduced in Ref. 5). The present paper reorganizes MFGS and extends it to include both no-snapping and multi-resolution snapping within the same framework. This paper also discusses the characteristics of MFGS with regard to simulations to show the MFGS mechanism. Furthermore, we have done experiments to test the efficiency of MFGS compared to single-resolution grid snapping (SGS) and to determine whether fuzziness should be incorporated in the multi-



**Fig. 2** Choices for cursor snapping in a multi-resolution grid.

resolution grid system.

## 2. Multi-Resolution Fuzzy Grid Snapping

In single-resolution grid systems, the cursor will simply be snapped to the nearest grid point. In multi-resolution grid systems, though, there are multiple choices for grid points. Selection of the grid layer in which the cursor should be snapped depends on the user's intention. For example, **Fig. 2** shows a three-layer 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, **c** is the current cursor point, while  $g_1$ ,  $g_2$ , and  $g_3$  are the nearest grid points to **c** in  $G_1$ ,  $G_2$ , and  $G_3$ , respectively. In this particular case, the user has three choices for snapping the cursor:  $g_1$ ,  $g_2$ , and  $g_3$ . In addition, the user has another option which we call no-snapping. No-snapping can be regarded as snapping of the cursor **c** to itself. Thus, the user now has four choices:  $g_1$ ,  $g_2$ ,  $g_3$ , and **c**. This leads to the problem of determining which choice the user intended.

To overcome this problem, we propose a snapping strategy that uses the pointing behavior of the user. In this strategy, we associate rough pointing behavior with low-resolution snapping, and careful pointing behavior with high-resolution snapping. The reason for this association is that the position the cursor represents is considered vague in the case of rough pointing, but precise in the case of careful pointing.

As the first step toward realizing MFGS, we designed a fuzzy cursor model and a multi-resolution grid system. We then developed a fuzzy grid snapping technique, MFGS, which embodies the above strategy.

### 2.1 Fuzzy Cursor Model

To introduce vagueness into a cursor's posi-

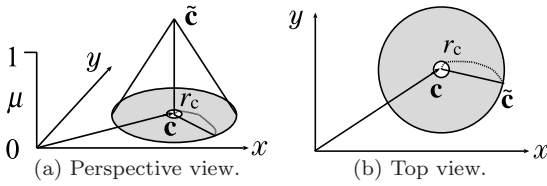


Fig. 3 Conical fuzzy cursor  $\tilde{\mathbf{c}}$ .

tion, we use a conical fuzzy cursor, which is expressed by a conical fuzzy point  $\tilde{\mathbf{c}} = \langle \mathbf{c}, r_{\mathbf{c}} \rangle$ . Here,  $\tilde{\mathbf{c}}$  is a fuzzy set characterized by a conical membership function

$$\mu_{\tilde{\mathbf{c}}}(\mathbf{v}) = \left(1 - \frac{\|\mathbf{v} - \mathbf{c}\|}{r_{\mathbf{c}}}\right) \vee 0, \quad (1)$$

where  $\mathbf{v}$  is a variable position vector in point space  $E^2$ ,  $\mathbf{c}$  is the current cursor position,  $r_{\mathbf{c}}$  is the fuzziness that represents the vagueness of the cursor position, and  $\vee$  represents for the max operator. **Figure 3** is a diagram of the conical fuzzy cursor. We then set the amount of fuzziness  $r_{\mathbf{c}}$  according to the roughness of the pointing behavior. Although there is no well established formula that directly associates fuzziness with roughness, we propose the following method based on the fuzziness generator empirically obtained in Refs. 6) and 7).

The first step is to provide the system with all of the recent cursor positions and the corresponding time stamps for a certain period of time  $T$  as a sequence  $(\mathbf{c}_i, t_i) \sim (\mathbf{c}_{i-m+1}, t_{i-m+1})$ , where  $(\mathbf{c}_i, t_i)$  represents the current cursor position. Second, the system applies spline interpolation<sup>8),9)</sup> to the sequence and then checks the acceleration  $a_i$  and the velocity  $v_i$  at each cursor position  $\mathbf{c}_i$ . The system next assigns an appropriate degree of fuzziness  $r_{\mathbf{c}_i}^*$  to each cursor position  $\mathbf{c}_i$  by using the fuzziness generator<sup>6),7)</sup>

$$r_{\mathbf{c}_i}^* = C_a a_i + C_v v_i, \quad (2)$$

where  $C_a$  and  $C_v$  are positive constant values that are determined empirically. Third, the system calculates the recent average fuzziness as

$$\bar{r}_{\mathbf{c}_i}^* = \frac{1}{m} \sum_{j=0}^{m-1} r_{\mathbf{c}_{i-j}}^*. \quad (3)$$

This step is to smooth the fluctuation of the fuzziness. Finally, the system calculates the fuzziness  $r_{\mathbf{c}}$  of the current cursor — that is,  $r_{\mathbf{c}_i}$  — as

$$r_{\mathbf{c}_i} = \alpha \bar{r}_{\mathbf{c}_i}^* + (1 - \alpha) r_{\mathbf{c}_{i-1}}, \quad (4)$$

where  $\alpha$  is a constant value between 0 and 1.

The final step is first-order lag filtering to adjust the response speed of the fuzziness variation of the fuzzy cursor.

## 2.2 Multi-Resolution Grid System

We define an  $n$ -layered multi-resolution grid system as a combination of single-resolution grid systems  $G_i (i = 1, 2, \dots, n)$ , each of which has two properties,  $S_{G_i}$  and  $r_{G_i}$ . Here,  $S_{G_i}$  and  $r_{G_i}$  are, respectively, the stride and the fuzziness of a grid  $G_i$ . In the grid system, we assume that  $G_i (i = 1, 2, \dots, n)$  are in descending order of resolution. Therefore, we simply give the smallest value to  $S_{G_1}$  and the largest value to  $S_{G_n}$ . On the other hand, we let the fuzziness represent the area covered by each grid point. The area covered 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  $r_{G_n}$ .

## 2.3 Fuzzy Grid Snapping

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

First, the system selects the grid point  $\mathbf{g}_i$  that is nearest to the fuzzy cursor  $\tilde{\mathbf{c}}$  from each grid system  $G_i$ , and uses it as a snapping candidate. Next, the system replaces each snapping candidate  $\mathbf{g}_i$  with a conical fuzzy point  $\tilde{\mathbf{g}}_i = \langle \mathbf{g}_i, r_{\mathbf{g}_i} \rangle$ , where  $r_{\mathbf{g}_i}$  is set to  $r_{G_i}$ , inheriting the fuzziness from the grid  $G_i$ . Then, for each snapping candidate, the system evaluates the necessity of the fuzzy proposition “ $\tilde{\mathbf{g}}_i$  is in  $\tilde{\mathbf{c}}$ ” by

$$N^{\tilde{\mathbf{g}}_i} = \inf_{\mathbf{v} \in E^2} ((1 - \mu_{\tilde{\mathbf{g}}_i}(\mathbf{v})) \vee \mu_{\tilde{\mathbf{c}}}(\mathbf{v})), \quad (5)$$

according to the definition of the necessity measure in fuzzy set theory<sup>10),11)</sup>. Note that in this particular case, where  $\tilde{\mathbf{g}}_i$  and  $\tilde{\mathbf{c}}$  have conical fuzzy membership functions, the necessity defined in Equation (5) can easily be calculated as

$$N^{\tilde{\mathbf{g}}_i} = \left( \frac{r_{\mathbf{c}} - \|\mathbf{g}_i - \mathbf{c}\|}{r_{\mathbf{c}} + r_{\mathbf{g}_i}} \right) \vee 0. \quad (6)$$

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 symbol  $\wedge$  stands for the min operator and is translated as a logical *and*, while  $(1 - N^{\tilde{\mathbf{g}}_i})$  is translated as the negation of  $N^{\tilde{\mathbf{g}}_i}$ . Therefore, the rules shown in Table 1 can be translated as

**Table 1** Fuzzy rules of MFGS.

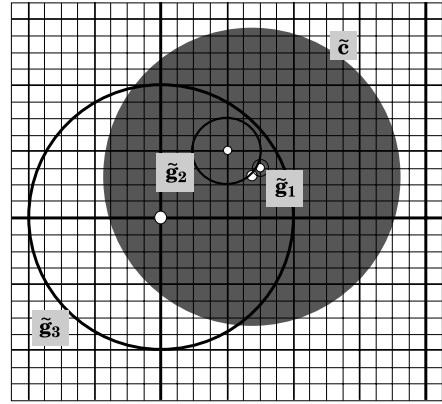
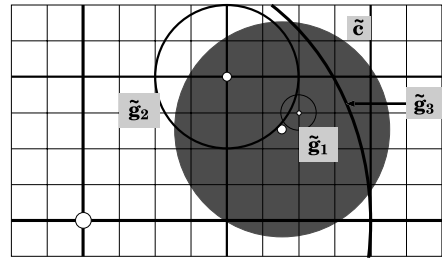
Rule	Fuzzy logical operations
1	$\mu(\tilde{\mathbf{g}}_3) = N\tilde{\mathbf{g}}_3$
2	$\mu(\tilde{\mathbf{g}}_2) = (1 - N\tilde{\mathbf{g}}_3) \wedge N\tilde{\mathbf{g}}_2$
3	$\mu(\tilde{\mathbf{g}}_1) = (1 - N\tilde{\mathbf{g}}_3) \wedge (1 - N\tilde{\mathbf{g}}_2) \wedge N\tilde{\mathbf{g}}_1$
4	$\mu(\tilde{\mathbf{c}})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

follows:

- Rule 1 recommends  $\tilde{\mathbf{g}}_3$  as the snapping point  $\mathbf{g}_s$  if (the necessity of “ $\tilde{\mathbf{g}}_3$  is in  $\tilde{\mathbf{c}}$ ” is high),
- Rule 2 recommends  $\tilde{\mathbf{g}}_2$  as the snapping point  $\mathbf{g}_s$  if (the necessity of “ $\tilde{\mathbf{g}}_3$  is in  $\tilde{\mathbf{c}}$ ” is *not* high) *and* (the necessity of “ $\tilde{\mathbf{g}}_2$  is in  $\tilde{\mathbf{c}}$ ” is high),
- Rule 3 recommends  $\tilde{\mathbf{g}}_1$  as the snapping point  $\mathbf{g}_s$  if (the necessity of “ $\tilde{\mathbf{g}}_3$  is in  $\tilde{\mathbf{c}}$ ” is *not* high) *and* (the necessity of “ $\tilde{\mathbf{g}}_2$  is in  $\tilde{\mathbf{c}}$ ” is *not* high) *and* (the necessity of “ $\tilde{\mathbf{g}}_1$  is in  $\tilde{\mathbf{c}}$ ” is high),
- Rule 4 recommends  $\tilde{\mathbf{c}}$  as the snapping point  $\mathbf{g}_s$  if (the necessity of “ $\tilde{\mathbf{g}}_3$  is in  $\tilde{\mathbf{c}}$ ” is *not* high) *and* (the necessity of “ $\tilde{\mathbf{g}}_2$  is in  $\tilde{\mathbf{c}}$ ” is *not* high) *and* (the necessity of “ $\tilde{\mathbf{g}}_1$  is in  $\tilde{\mathbf{c}}$ ” is *not* high).

The combined application of the four rules as a whole is to try to snap the fuzzy cursor to the candidate from the lowest resolution as long as the necessity is high. Finally, the system determines which snapping candidate has the highest grade and selects it as the snapping point  $\mathbf{g}_s$ . If, in an extreme case,  $\mu(\tilde{\mathbf{c}})$  has the highest grade, then the current cursor  $\mathbf{c}$  is selected as the snapping point  $\mathbf{g}_s$ . In this case, the system infers that the user wants no-snapping.

To demonstrate how MFGS works, let us again consider the case shown in Fig. 2. Now, let us set the strides as  $S_{G_1} = 1.00$ ,  $S_{G_2} = 4.00$ , and  $S_{G_3} = 16.00$  and the fuzziness as  $r_{G_1} = 0.50$ ,  $r_{G_2} = 2.00$ , and  $r_{G_3} = 8.00$ , and let us assume the exact positions as  $\mathbf{c} = (5.6, 2.6)$ ,  $\mathbf{g}_1 = (6.0, 3.0)$ ,  $\mathbf{g}_2 = (4.0, 4.0)$ , and  $\mathbf{g}_3 = (0.0, 0.0)$ . Then, let us set the fuzziness  $r_c$  of the fuzzy cursor with four different values: 1.50, 3.00, 9.00, and 22.00. **Figure 4(a)** and **Fig. 4(b)** illustrate the case of  $r_c = 9.00$  and the case of  $r_c = 3.00$ , respectively. The grades evaluated by MFGS with the settings are summarized in **Table 2**. As indicated in the table, as the fuzziness of the cursor increases, the resolution of the snapping selected by the system decreases. This confirms that MFGS follows the proposed snapping strategy.

(a)  $r_c = 9.00$ .(b)  $r_c = 3.00$  (enlarged portion of (a)).**Fig. 4** Relationships between snapping candidates and the fuzzy cursor in a multi-resolution grid system.**Table 2** Evaluated grades of snapping candidates according to fuzziness  $r_c$ .

$r_c$	$\mu(\tilde{\mathbf{g}}_3)$	$\mu(\tilde{\mathbf{g}}_2)$	$\mu(\tilde{\mathbf{g}}_1)$	$\mu(\tilde{\mathbf{c}})$
1.50	0.00	0.00	0.47	<u>0.53</u>
3.00	0.00	0.17	<u>0.70</u>	0.30
9.00	0.17	<u>0.62</u>	0.38	0.11
22.00	<u>0.53</u>	0.47	0.17	0.05

To demonstrate the characteristics of MFGS and show how MFGS evaluates the snapping grades at each cursor position, let us consider a fuzzy cursor  $\tilde{\mathbf{c}}$

$$\tilde{\mathbf{c}} = \langle (c_x, 0), r_c \rangle, \quad (7)$$

which is shown in **Fig. 5**. Figure 5 illustrates a fuzzy cursor  $\tilde{\mathbf{c}}$  that travels along the x-axis of a three-layer multi-resolution grid system while changing its fuzziness  $r_c$ . The properties of the multi-resolution grid system are set to be identical to those in the above demonstration.

**Figure 6** shows graphs of the grades that are evaluated by MFGS for the fuzzy cursor  $\tilde{\mathbf{c}}$ . In **Fig. 6(a)**, the variance of the grades  $\mu(\tilde{\mathbf{g}}_3)$ ,  $\mu(\tilde{\mathbf{g}}_2)$ ,  $\mu(\tilde{\mathbf{g}}_1)$  and  $\mu(\tilde{\mathbf{c}})$  are plotted three-dimensionally with respect to the change of  $c_x (\in [0.0, 16.0])$  and  $r_c (\in [0.0, 16.0])$ . Note that four surfaces exist, all of which intersect with

each other. When a certain surface is topmost, its corresponding snapping candidate is selected as the snapping point. Thus, let us refer to the area where the surface for a certain snapping candidate  $\tilde{\mathbf{g}}_i$  is topmost as the dominant range of  $\tilde{\mathbf{g}}_i$ . The change of the dominant range is more clearly shown in Fig. 6 (b), which is a top view of Fig. 6 (a).

In Fig. 6 (b), we can see that the number of dominant ranges that appear at every position of  $c_x$  is always four, which is the number of snapping candidates. This means that the user always has a chance to choose any snapping candidate, no matter what the cursor position, by controlling  $r_c$  through his or her pointing behavior.

Next, considering the width and position of each dominant range, we can see that they vary according to the shift of the position  $c_x$  and form a pattern that resembles saw blades. This saw-blade pattern implies the gravitational characteristics of MFGS. For example, if we consider the dominant range of  $\tilde{\mathbf{g}}_2$  in the graph, we see that it is considerably wider when  $\mathbf{c}$  is near the grid points of  $G_2$  (where  $c_x$  is near 0, 4, 8, 12 or 16) than when  $\mathbf{c}$  is far from these grid points. This means that the probability of  $\mathbf{c}$  being snapped to  $\tilde{\mathbf{g}}_2$  becomes greater as  $\mathbf{c}$  approaches a grid point of  $G_2$ . These characteristics can be thought of as the gravitation characteristics of  $G_2$ . Regarding the dominant ranges of the other snapping candidates, we can see similar saw-blade patterns. Therefore, the user will feel the gravitation characteristics of all the grid system  $G_i$  while moving the cursor.

From the above observations, we conclude that MFGS provides the user with a full choice of snapping resolution, allowing the user to feel the gravitation characteristics from the multi-resolution grid system.

### 3. Experimental Results

In this section, we present the results of experiments done to evaluate the performance of the MFGS technique. The first experiment was to test the effectiveness of MFGS, and the second was to determine the usefulness of fuzziness in MFGS.

#### 3.1 Effectiveness of MFGS

To evaluate the performance of MFGS, we did a target selection experiment with a three-layer multi-resolution grid system consisting of  $G_1$ ,  $G_2$ , and  $G_3$ . We had five users perform tasks that required them to select consecutively

**Table 3** Target selection time (s).

User	MFGS			SGS	MGS		
	G1	G2	G3		G1	G2	G3
A	2.34	1.70	0.95	2.13	3.09	2.06	0.91
B	2.33	1.56	0.95	1.61	2.55	1.69	0.98
C	2.50	2.14	1.15	1.81	2.78	2.18	1.19
D	2.57	1.77	0.91	1.91	2.94	1.81	0.93
E	2.41	1.52	0.88	1.82	2.98	1.80	0.87
Avg.	2.43	1.74	0.97	1.86	2.87	1.91	0.98

appearing target grid points on a computer display, as shown in Fig. 7, for five minutes. We then calculated the average selection time. The task was repeated three times, changing the target points but keeping all of the MFGS settings unchanged. Each time, when the user was required to select a target point, one grid layer was specified and the target grid point was selected randomly from the grid points of the specified layer. The selection was considered completed when the mouse button was pressed while the snapping point was over the target grid point.

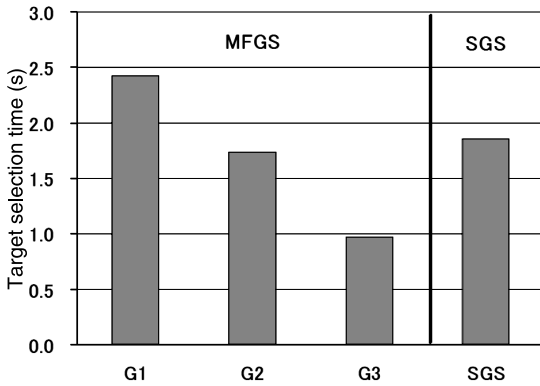
For MFGS, we set the strides as  $S_{G_1} = 5[\text{pixels}]$ ,  $S_{G_2} = 20[\text{pixels}]$ , and  $S_{G_3} = 80[\text{pixels}]$  and the fuzziness as  $r_{G_1} = 2.5[\text{pixels}]$ ,  $r_{G_2} = 10[\text{pixels}]$ , and  $r_{G_3} = 40[\text{pixels}]$ , respectively. We set the properties for the fuzzy cursor  $\tilde{\mathbf{c}}$  as  $T = 0.5[\text{s}]$ ,  $C_a = 0.036[\text{s}^2]$ ,  $C_v = 0.014[\text{s}]$  and  $\alpha = 0.5$ .

For comparison, we also performed a similar experiment to obtain the target selection time using single-resolution grid snapping (SGS), which is commonly used in current CAD applications. For SGS, we set the resolution of the grid system to the same values as for the high-resolution grid  $G_1$  of MFGS.

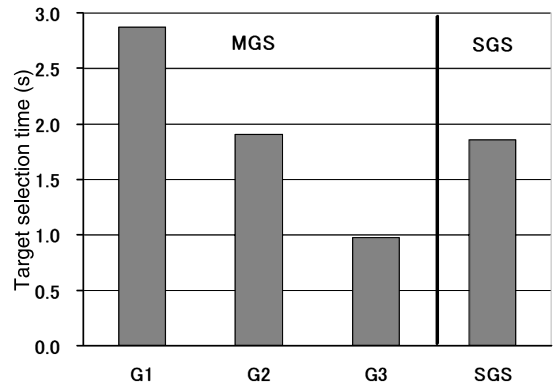
The results are summarized in Table 3 and graphically illustrated in Fig. 8. Figure 8 shows that the selection time for targets from low-resolution layer  $G_3$  with MFGS was shorter (0.89 s) than that with SGS, while the selection time for targets from high-resolution layer  $G_1$  with MFGS was slightly longer (0.57 s). In the MFGS experiment, we observed that the users achieved quick and easy target selection from  $G_3$  by expressing their intention for low-resolution snapping through rough pointing behavior. On the other hand, for targets from  $G_1$ , the users were able to achieve high-resolution snapping, but needed a slight pause

The resolution of the display was 3.79[pixels/mm].





**Fig. 8** Comparison of average target selection time between MFSGS and SGS.



**Fig. 9** Comparison of average target selection time between MGS and SGS.

to control and reduce the fuzziness of the cursor through careful pointing behavior. These results show that MFSGS is an effective snapping technique that speeds up low-resolution grid snapping while retaining the ability to snap to high-resolution grids.

### 3.2 Need for Fuzziness in MFSGS

To test the need for fuzziness in a multi-resolution grid system, we set up a non-fuzzy version of the multi-resolution grid snapping method, called MGS, and performed a similar target selection experiment using MGS. In MGS, all of the settings were the same as those in MFSGS, except that the values of  $r_{g_i}$  were set to zero. As shown in **Fig. 9**, with MGS the selection time for targets from low-resolution layer  $G_3$  was approximately the same as that of MFSGS shown in **Fig. 8**, but noticeably longer for higher-resolution layers  $G_1$  and  $G_2$ . A paired  $t$ -test shows that the differences for  $G_1$  and  $G_2$  are significant at significance levels of 2% and 6%, respectively. This indicates that eliminating the fuzziness from the grids of MFSGS reduced snapping performance for higher-resolution grid systems. Thus, fuzziness in the grid systems is important for achieving better snapping performance with MFSGS.

To analyze how the elimination of fuzziness from MFSGS grids affected snapping performance, we plotted the grades evaluated by MGS for the case of fuzzy cursor  $\tilde{c}$  shown in **Fig. 5**. The result is shown in **Fig. 10**. The graph patterns of **Fig. 10** and **Fig. 6** show that the dominant range of each snapping candidate varied more sensitively according to the shift of the position  $c_x$  in MGS, as compared to the MFSGS case. In addition, the number of dominant ranges varied from one to four, accord-

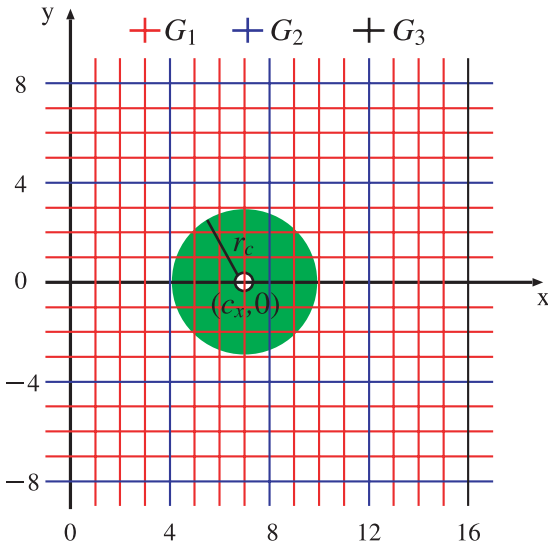
ing to the position  $c_x$  in MGS, while it was constant (four) in MFSGS, as shown in **Fig. 6**. The characteristics of MGS are likely to get users into tricky situations, where the snapping point moves around quickly as the user moves the fuzzy cursor. Such tricky situations reduce the snapping performance of MGS compared to that of MFSGS.

## 4. Conclusion

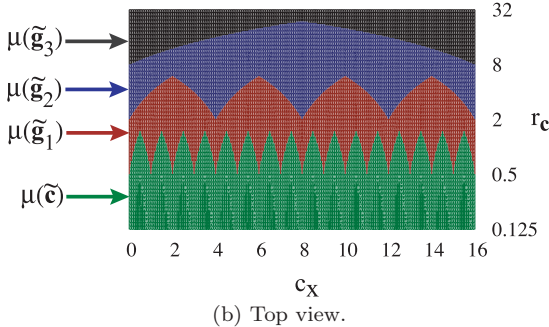
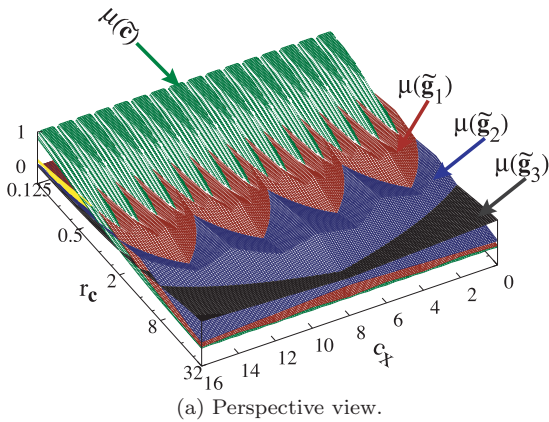
We have described MFSGS, a grid snapping technique that implements automatic cursor snapping for multi-resolution grid systems. The MFSGS technique dynamically selects a grid snapping resolution level and snaps the cursor to the grid layer with that resolution level, which depends on to the roughness or carefulness of the user's pointing behavior. This includes no-snapping, in which the cursor is not snapped to any layer of the multi-resolution grid system.

Our experimental results show that MFSGS is an effective grid snapping technique, which speeds up low-resolution grid snapping while retaining the ability to snap to high-resolution grids. Furthermore, experiments in which we eliminated fuzziness showed that using fuzziness contributes to better snapping performance with MFSGS.

As an advanced application of MFSGS, we plan to apply MFSGS to object snapping in the sketch-based CAD system proposed in Refs. 12) and 13). Fundamentally, the fuzzy cursor discussed here is just one instance of the conical fuzzy point. In fact, MFSGS can generally be applied to any conical fuzzy point. In the sketch-based CAD system, drawn objects have fuzzy feature points, the fuzziness of which is associ-

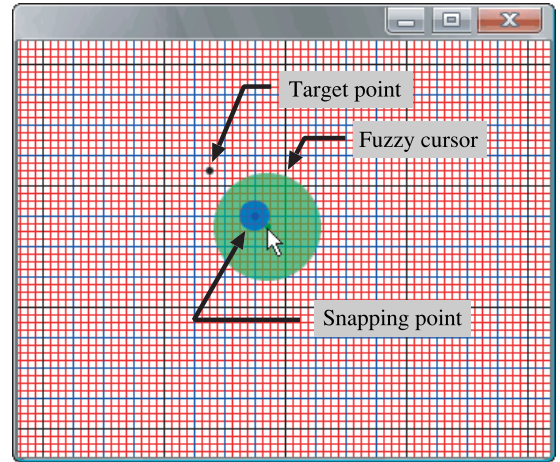


**Fig. 5** Fuzzy cursor  $\tilde{c}$  traveling along the x-axis while changing the fuzziness in a multi-resolution grid system.

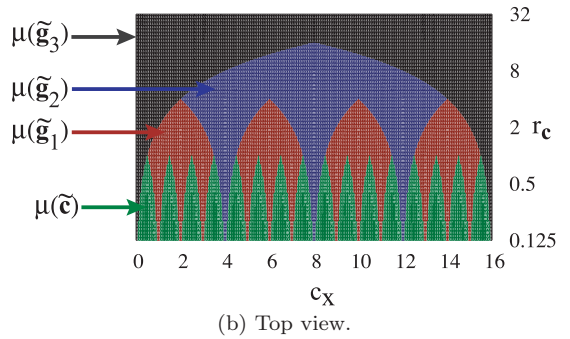
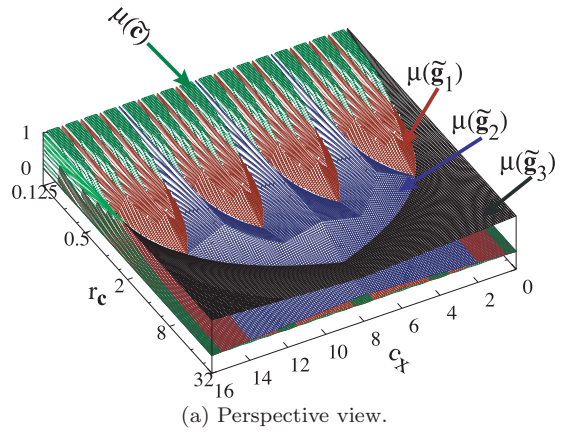


**Fig. 6** Grades evaluated by MFGS for fuzzy cursor  $\tilde{c}$  shown in Fig. 5.

ated with the roughness of the drawing behavior of the user, and they are also instances of the conical fuzzy point. Therefore, we believe MFGS can be extended for application to au-



**Fig. 7** Computer display for target selection experiment.



**Fig. 10** Grades evaluated by MFGS for fuzzy cursor  $\tilde{c}$  shown in Fig. 5.

tomatic snapping of the feature points of CAD objects to a multi-resolution grid system.

## References

- 1) Edward, S.I.: *Sketchpad: A man-machine graphical communication system*, PhD Thesis, Massachusetts Institute of Technology (1963).

- 2) Gleicher, M. and Witkin, A.: Drawing with constraints, *The visual Computer*, Vol.11, No.1, pp.39–51 (1994).
- 3) Bier, E.A.: Snap-dragging, *Computer Graphics*, Vol.20, No.4, pp.233–240 (1986).
- 4) Masui, T.: HyperSnapping, *Proc. IEEE 2001 Symposia on Human Centric Computing Languages and Environments, Stresa*, pp.188–194 (2001).
- 5) Khand, Q.U., Saga, S. and Maeda, J.: Automatic Cursor Snapping into Multi-Resolution Grid Systems Based on Fuzzy Model, *IPSJ*, Vol.45, No.10, pp.2439–2442 (2004).
- 6) Saga, S. and Ohkawa, T.: Refinement of Fuzziness Generator for the Freehand Curve Identifier FSCI, *Proc. 2nd IEEE International Conference On Intelligent Processing Systems, Gold Coast*, pp.323–329 (1998).
- 7) Saga, S. and Ohkawa, T.: Refinement of Fuzziness Generator in the Freehand Curve Identifier FSCI (in Japanese), *IEICE Trans.*, Vol.J82-D-I, No.5, pp.634–643 (1999).
- 8) Saga, S. and Makino, H.: Fuzzy Spline Interpolation and Its Application to On-Line Freehand Curve Identification, *Proc. 2nd IEEE International Conference On Fuzzy Systems*, San Francisco, USA, pp.1183–1190 (1993).
- 9) Saga, S., Makino, H. and Sasaki, J.: Method for Modeling Freehand Curves—The Fuzzy Spline Interpolation (in Japanese), *IEICE Trans.*, Vol.J77-D-II, pp.1610–1619 (1994).
- 10) Zadeh, L.A.: Fuzzy Sets as a Basis for a Theory of Possibility, *Fuzzy Sets and Systems*, Vol.1, No.1, pp.3–28 (1978).
- 11) Klir, G.J. and Yuan, B.: *Fuzzy Sets and Fuzzy Logic-Theory and Applications*, Prentice Hall (1995).
- 12) Saga, S.: A Freehand Interface for Computer Aided Drawing Systems Based on the Fuzzy Spline Curve Identifier, *Proc. 1995 IEEE International Conference on Systems, Man and Cybernetics*, Vancouver, Canada, pp.2754–2759 (1995).
- 13) Sato, Y., Yasufuku, N. and Saga, S.: Sequential Fuzzy Spline Curve Generator for Drawing Interface by Sketch (in Japanese), *IEICE Trans.*, Vol.J86-D-II, No.2, pp.242–251 (2003).

(Received February 13, 2006)

(Accepted January 9, 2007)

(Online version of this article can be found in the IPSJ Digital Courier, Vol.3, pp.198–206.)



**Qamar Uddin Khand** is an Associate Professor in the Information Technology department at Sukkur Institute of Business Administration, Sukkur, Pakistan. He received B.Sc. (Hons.) and M.Sc. (Hons.) degrees in Computer Science from the University of Sind, Jamshoro, Pakistan in 1996 and 1997, respectively. He was a research student at Muroran Institute of Information Technology (Muroran IT), Japan from October 1998 to March 2000. He received M.E. and Ph.D. degrees from Muroran IT, Japan in 2002 and 2005, respectively. His research interests includes mainly sketch-based interfaces and modeling, human interface based on soft computing, and CAD-based freehand drawing systems. He is an ACM and IEEE member.



**Sumudu Dematapitiya** received B.E. and M.E. degrees in computer science and systems engineering from Muroran Institute of Technology, Hokkaido, Japan in 2004 and 2006, respectively. He now works in NEC Corporation as a software developer. His research interests include human interfaces and artificial intelligence.



**Sato Saga** received M.E. and Ph.D. degrees from Hokkaido University, Hokkaido, Japan. From 1987 to 1989, he taught computer science in the Philippines as a member of the Japan Overseas Cooperation of Volunteers. From 1989 to 1994, he was a researcher at Tecnova Corporation. In 1994, he joined Muroran Institute of Technology, where he is currently an Associate Professor in the Department of Computer Science and Systems Engineering. His research interests include human interfaces based on soft computing.





**Junji Maeda** received a B.S. degree in physics and his M.E. degree and Ph.D. in applied physics from Hokkaido University, Japan, in 1972, 1974, and 1985, respectively. He was an Assistant Professor in the Department of Applied Physics, Hokkaido University, and in 1988 joined Muroran Institute of Technology, Japan, where he has been a Professor in the Department of Computer Science and Systems Engineering since 1994. From 1990 to 1991, he was with Linköping University, Sweden, as a Visiting Researcher. He was the General Co-chair of the IEEE International Workshop on Soft Computing in Industry 1999. His research interests include image processing, computer vision, and pattern recognition.

---