Cursor Snapping into Multi-Resolution Grid Systems Based on Fuzzy Point Model

Qamar Uddin Khand, Sato Saga and Junji Maeda
Department of Computer Science and Systems Engineering
Muroran Institute of Technology
27-1 Mizumoto Cho, Muroran, 050-8585
Japan
email: qamar@sagalab.csse.muroran-it.ac.jp, {saga, junji}@csse.muroran-it.ac.jp

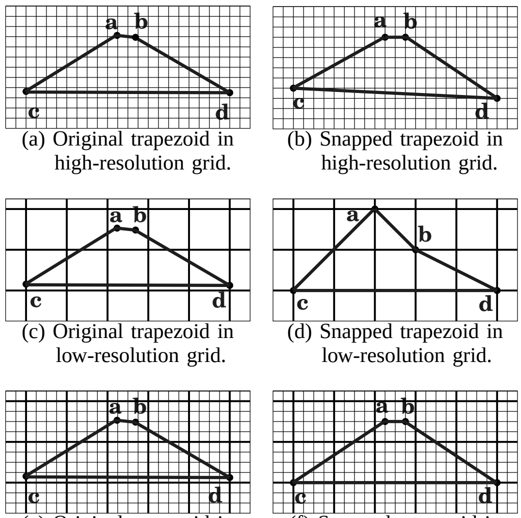
Abstract— **This paper presents a new grid snapping technique named Multi-resolution Fuzzy Grid Snapping (MFGS) that realizes automatic mouse cursor snapping for multi-resolution grid systems. In order to make drawings that include both fine and rough structures, quick switching between high and low resolution grid snapping is essential. MFGS dynamically selects a snapping resolution level from the multi-resolution grid system according to user's pointing manners and dispenses with manual switching of the snapping resolution. Our experimental results demonstrate that MFGS is an effective grid snapping technique, which speeds up the low-resolution grid snapping while keeping ability for the high-resolution grid snapping.**

I. Introduction

Grid snapping is one of the most commonly used techniques for object alignment in CAD applications. In presently available CAD applications, appropriate grid resolution depends upon the structural fineness of objects that a user intends to draw, and the user has to switch the resolution frequently. Since such switching is done manually, automatic switching of snapping resolution has been expected. Therefore, in this paper, we propose a new grid snapping technique, named Multi-resolution Fuzzy Grid Snapping (MFGS), that realizes snapping for multi-resolution grid systems, which automatically switches the snapping resolution.

Automatic control of grid size for object snapping has been given in HyperSnapping [1], where a user can control the snapping resolution level only by dragging objects. In this technique, the snapping resolution depends upon structures of already drawn objects, and user's intention is only known to the system if a user can set anchor and sub-anchor points properly. In contrast, MFGS is a simple cursor snapping technique that infers user's intention about snapping resolution directly from his/her pointing manners.

To illustrate why the switching is required, let us consider the example of forming a trapezoid that is snapped as shown in Fig. 1. In the case where a user intends to snap the line **ab** as a fine structure shown in Fig. 1(f), a high-resolution grid system shown in Fig. 1(a) is required. However, user's slight miss-arrangement for the line **cd** results in miss-alignment as shown in Fig. 1(b). On the other hand, in the case where the user intends to snap the line **cd** as a rough structure shown in Fig. 1(f), a low-resolution grid system shown in Fig. 1(c) is preferred for easy pointing operations. Nevertheless, user's arrangement for the line **ab** results in incorrect alignment as shown in Fig. 1(d). The observations show that high-resolution snapping with precise pointing is required when fine structures are intended and low-resolution snapping with easy pointing is good when rough structures are intended.



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

Fig. 1. Difference of snapping according to grid resolution.

Since usual drawing includes both fine and rough structures, quick switching between high-resolution snapping and low-resolution snapping is essential for making drawings efficiently. MFGS provides automatic resolution switching in the multi-resolution grid system, as shown in Fig. 1(e), that is a combination of several grid systems each of which has different resolution. For the automatic resolution switching, MFGS associates rough pointing manners with low-resolution snapping and careful pointing manners with high-resolution snapping. Using MFGS, a user can get correct cursor snapping, as shown in Fig. 1(f), by expressing his/her intention about snapping resolution through only varying pointing manners. Since MFGS provides the automatic switching of snapping resolution, the manual switching becomes unnecessary.

II. Multi-Resolution Fuzzy Grid Snapping

In the single-resolution grid systems, a cursor will just be 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 cursor should be snapped depends upon the user's intention. For example, Fig. 2 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 current cursor point, 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 cursor, which are $\mathbf{g}_1$, $\mathbf{g}_2$ and $\mathbf{g}_3$. This leads us to another problem of getting the intention of the user.

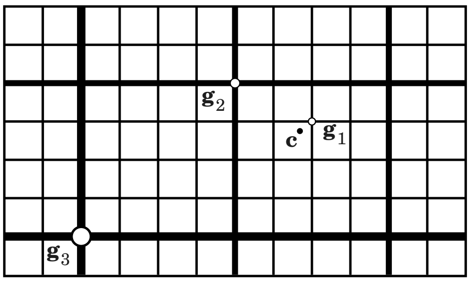


Fig. 2. Three choices for snapping cursor.

To overcome the problem, we propose a snapping strategy to utilize user's pointing manners. In this strategy, we associate rough pointing manners with low-resolution snapping and careful pointing manners with high-resolution snapping. The reason for this association is that the position a cursor represents is considered to be vague in the case of rough pointing manners while it is considered to be precise in the case of careful pointing manners.

We first design a fuzzy cursor model that represents not only the cursor position but also its vagueness. Then, we design a multi-resolution grid system that has fuzziness in its grid points. Finally, we propose a fuzzy grid snapping technique named MFGS, which embodies the above strategy.

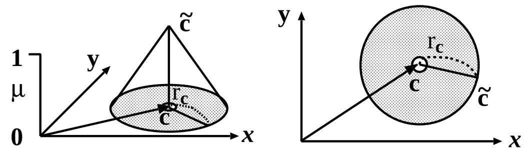
A. Fuzzy Cursor Model

To introduce vagueness into a cursor, we propose to express the cursor with a conical fuzzy cursor $\tilde{\mathbf{c}} =< \mathbf{c}, r_{\mathbf{c}} >$. Here, $\tilde{\mathbf{c}}$ is a fuzzy set which is characterized by the conical membership function

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

where $\mathbf{c}$ is the current cursor position, $r_{\mathbf{c}}$ is fuzziness that represents vagueness in the cursor position and $\vee$ stands for a $\max$ operator. Figure 3 illustrates the conical fuzzy cursor. We further establish the following method for calculating the amount of fuzziness $r_{\mathbf{c}}$ according to roughness of pointing manners.

The first step is to provide the system all the recent cursor positions and their time stamps for a certain period of time



(a) Perspective view. (b) Top view.

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

T as a sequence $(\mathbf{c}_i, t_i) \sim (\mathbf{c}_{i-m+1}, t_{i-m+1})$, where $(\mathbf{c}_i, t_i)$ corresponds to the current cursor. Secondly, the system applies spline interpolation [2] to the sequence, and then it checks acceleration a_i and velocity v_i at each cursor position $\mathbf{c}_i$. Then the system assigns an appropriate amount of fuzziness $r^*_{\mathbf{c}_i}$ to each cursor position $\mathbf{c}_i$ by using the fuzziness generator

$$r^*_{\mathbf{c}_i} = C_a a_i + C_v v_i, \tag{2}$$

that is proposed in [3]. Here, C_a and C_v are positive constant values. Thirdly, the system calculates recent average fuzziness as

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

Finally, the system calculates the fuzziness $r_{\mathbf{c}}$ by

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

where α is a constant value between 0 and 1. The final step is to adjust response speed of fuzziness variation in the fuzzy cursor.

B. Multi-Resolution Grid System

We define an n-layered multi-resolution grid system as a combination of single-resolution grid systems $G_i (i = 1, 2, \ldots, n)$, 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, we assume $G_i (i = 1, 2, \ldots, n)$ 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 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}.

C. Fuzzy Grid Snapping

The snapping strategy for the multi-resolution grid system discussed above is realized through the following method. We name this method Multi-resolution Fuzzy Grid Snapping (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 the fuzzy cursor $\tilde{\mathbf{c}}$ 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 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 necessity of a fuzzy proposition "$\tilde{\mathbf{g}}_i$ is $\tilde{\mathbf{c}}$" [4]. The necessity is defined as

$$\begin{aligned} N^{\tilde{\mathbf{g}}_i} &= Nec_{\tilde{\mathbf{g}}_i}(\tilde{\mathbf{c}}) \\ &= \inf_{\mathbf{v}}((1-\mu_{\tilde{\mathbf{g}}_i}(\mathbf{v})) \vee \mu_{\tilde{\mathbf{c}}}(\mathbf{v})) \end{aligned} \quad (5)$$

according to [4] and [5]. 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

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

Fourth, the system performs fuzzy reasoning by applying the rules shown in Table I, 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 a $\min$ operator that implies a logical operator AND, and $(1 - N^{\tilde{\mathbf{g}}_i})$ is negation of $N^{\tilde{\mathbf{g}}_i}$. 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 cursor $\mathbf{c}$ is selected as snapping point $\mathbf{g}_s$. This is the case where the system inferred that a user doesn't want to snap the cursor with any grid point.

TABLE I

RULES OF MFGS.

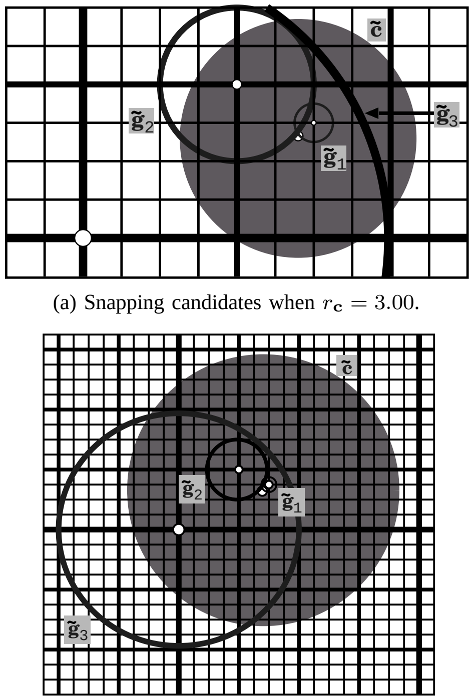
$\mu(\tilde{\mathbf{g}}_3)$	=	$N^{\tilde{\mathbf{g}}_3}$						
$\mu(\tilde{\mathbf{g}}_2)$	=	$(1 - N^{\tilde{\mathbf{g}}_3})$	$\wedge$	$N^{\tilde{\mathbf{g}}_2}$				
$\mu(\tilde{\mathbf{g}}_1)$	=	$(1 - N^{\tilde{\mathbf{g}}_3})$	$\wedge$	$(1 - N^{\tilde{\mathbf{g}}_2})$	$\wedge$	$N^{\tilde{\mathbf{g}}_1}$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

To demonstrate how MFGS works for the case shown in Fig. 2, 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 fuzzy cursor's fuzziness $r_{\mathbf{c}}$ with four different values 1.50, 3.00, 9.00 and 22.00. Figure 4(a) and 4(b) illustrates the case of $r_{\mathbf{c}} = 3.00$ and the case of $r_{\mathbf{c}} = 9.00$, respectively.

Table II shows the evaluated grades of snapping candidates according to the fuzziness $r_{\mathbf{c}}$. On the basis of the table, 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.

III. EXPERIMENTAL RESULTS

We performed some experiments to demonstrate effectiveness of MFGS and need of the fuzziness in MFGS.



(a) Snapping candidates when $r_{\mathbf{c}} = 3.00$.

(b) Snapping candidates when $r_{\mathbf{c}} = 9.00$.

Fig. 4. Snapping candidates and fuzzy cursor in multi-resolution grid system.

TABLE II

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

A. Effectiveness of MFGS

To evaluate the performance of MFGS, we made a target picking experiment with a three-layered multi-resolution grid system that includes G_1, G_2 and G_3. We let five users perform tasks to keep on picking consecutively appeared target grid points on a computer display[1] for five minutes and calculated average picking time. The task was repeated three times, where all the settings for MFGS were unchanged but only target point generation was changed. Each time, one grid layer was specified and the target grid points were randomly selected from grid points of the specified layer. The picking was regarded to be done when the mouse button was pressed while $\mathbf{g}_s$ was snapped with the target grid point.

For MFGS, we set the strides as $S_{G_1} = 5[pixels]$, $S_{G_2} = 20[pixels]$, $S_{G_3} = 80[pixels]$ and the fuzziness as $r_{G_1} = 2.5[pixels]$, $r_{G_2} = 10[pixels]$, $r_{G_3} = 40[pixels]$, respectively.

[1]Resolution of the display was $3.79[pixels/mm]$.

Then, we set the properties for the fuzzy cursor $\tilde{\mathbf{c}}$ as $T = 0.5[sec]$, $C_a = 0.036[sec^2]$, $C_v = 0.014[sec]$ and $\alpha = 0.5$.

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

Figure 5 shows that, although the picking time for targets from high-resolution layer G_1 using MFGS is slightly longer than the picking time using SGS, the picking time for targets from low-resolution layer G_3 using MFGS is considerably shorter than the picking time using SGS. In the experiment for MFGS, we observed that users achieved easy and quick picking for targets from G_3 by expressing their intention to the system for low-resolution snapping through rough pointing manners. The results clearly shows that MFGS is an effective snapping technique which speeds up the low-resolution grid snapping while keeping ability for high-resolution grid snapping.

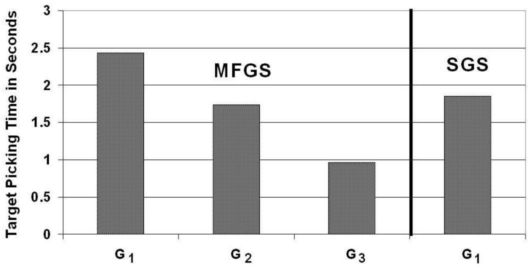


Fig. 5. Comparison of target picking time between MFGS and SGS.

B. Need of Fuzziness in MFGS

In order to demonstrate the need of fuzziness in the multi-resolution grid system, we established another three-layered multi-resolution grid snapping method named MGS-1, which has no fuzziness in grids, and performed a similar target picking experiment. In MGS-1, all the settings were same as MFGS except for $r_{\mathbf{g}_i}$'s, which were set to be zero. As shown in Fig. 6, the picking time for targets from low-resolution layer G_3 using MGS-1 is just about the same as that of MFGS, but it is noticeably longer for the high-resolution layer G_1. This indicates that the elimination of fuzziness from the grids of MFGS results in worse snapping for a high-resolution grid system.

To show the need of fuzziness in the cursor, we designed yet another three-layered multi-resolution grid snapping method named MGS-2, which has no fuzziness in the cursor, and performed a similar target picking experiment. In MGS-2, all the settings were same as MGS-1 except for the cursor settings. Here, instead of fuzzy cursor $\tilde{\mathbf{c}}$, we introduced a crisp (or non-fuzzy) cursor $\hat{\mathbf{c}}$, which is expressed as an ordinary circular set, whose center is $\mathbf{c}$ and radius is $r_{\mathbf{c}}$. In the case of the non-fuzzy cursor, necessity is simply calculated by

$$Nec_{\tilde{\mathbf{g}}_i}(\hat{\mathbf{c}}) = \begin{cases} 1 & if \;\; \mathbf{g}_i \subseteq \hat{\mathbf{c}} \\ 0 & otherwise. \end{cases} \tag{7}$$

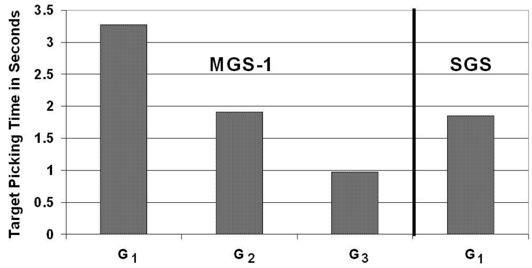


Fig. 6. Comparison of target picking time between MGS-1 and SGS.

Results in Fig. 7 illustrate that the picking time for targets from low-resolution layer G_3 using MGS-2 is almost the same as that of MFGS, however it is considerably longer for the high-resolution layer G_1. This clearly indicates that the elimination of fuzziness from the cursor of MFGS results in worse snapping for a high-resolution grid system.

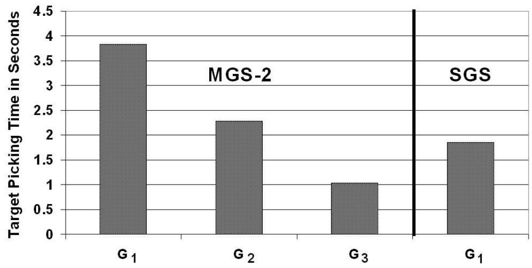


Fig. 7. Comparison of target picking time between MGS-2 and SGS.

The results in Fig. 6 and Fig. 7 clearly show the need of fuzziness in both the cursor and the grid systems.

IV. Conclusions

In this paper, we have presented a new grid snapping technique named MFGS that realizes automatic cursor snapping for multi-resolution grid systems. MFGS dynamically selects a resolution level of the grid and snaps a cursor according to user's pointing manners. Experimental results demonstrated that MFGS is an effective grid snapping technique which speeds up low-resolution grid snapping while keeping ability for high-resolution grid snapping.

References

[1] T. Masui, "HyperSnapping," *Proc. of IEEE 2001 Symposia on Human Centric Computing Languages and Environments*, Stresa, pp. 188-194, 2001.

[2] S. Saga, and H. Makino, "Fuzzy Spline Interpolation and Its Application to On-Line Freehand Curve Identification," *Proc. of 2nd IEEE Int. Conf. On Fuzzy Systems*, San Francisco, pp. 1183-1190, 1993.

[3] S. Saga, and T. Ohkawa, "Refinement of Fuzziness Generator for the Freehand Curve Identifier FSCI," *Proc. of 2nd IEEE Int. Conf. On Intelligent Processing Systems*, Gold Coast, pp. 323-329, 1998.

[4] L. A. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility," *Fuzzy Sets and Systems*, Vol. 1, No. 1, pp. 3-28, 1978.

[5] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic-Theory and Applications*, Prentice Hall, 1995.