

# PressureFish: A Method to Improve Control of Discrete Pressure-based Input

**Kang Shi**

University of Manitoba  
Winnipeg, Canada  
kangshi@cs.umanitoba.ca

**Pourang Irani**

University of Manitoba  
Winnipeg, Canada  
irani@cs.umanitoba.ca

**Sean Gustafson**

University of Manitoba  
Winnipeg, Canada  
umgustal@cs.umanitoba.ca

**Sriram Subramanian**

University of Bristol,  
Bristol, U.K  
sriram@cs.bris.ac.uk

## ABSTRACT

Studies investigating user control of pressure input have reported time-accuracy trade-offs of, on average, over 30%, when interacting with a large number of pressure levels. To increase the level of control with pressure input, we designed and evaluated four different discretization functions: linear, fisheye, visual fisheye, and clustered. The fisheye discretization dynamically modifies the range of pressure values based on the position of the pressure cursor. Our results show that a fisheye function results in significantly lower error rates and a lower number of crossings than have been reported in the literature. Furthermore, the fisheye function improves control without compromising speed. We discuss the findings of our study and identify several design recommendations for integrating pressure control into common interface tasks.

## Author Keywords

Pressure discretization, fisheye-based pressure control.

## ACM Classification Keywords

H5.m. Information interfaces (e.g., HCI): Miscellaneous.

## INTRODUCTION

Recently, several studies have reported the benefits of pressure-based interaction as an alternative input channel [3,6–8]. Ramos et al. [7,8] explored the design space of pressure-based interactions with styli. Their results revealed that adequate control of pressure values is tightly coupled to a fixed number of discrete pressure levels (maximum of six levels). Mizobuchi et al. [5] investigated accurate control of pressure exerted on a pen-based device and showed that users can better control forces that are divided into discrete levels and smaller than 3N. Cechanowicz et al. [3] investigated augmenting a mouse with one or two pressure sensors and showed that with one sensor users can control between 8 and 10 discrete levels.

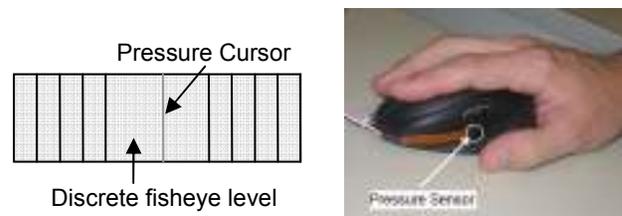
One common outcome reported by previous studies is the high number of errors that result from pressure-based input

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2008, April 5–10, 2008, Florence, Italy.

Copyright 2008 ACM 978-1-60558-011-1/08/04...\$5.00

[3,8]. As a result, pressure input may not be highly practical as a reliable input dimension. This limited ability to properly control pressure has made it difficult to introduce pressure input to facilitate tasks that require multiple levels of pressure control such as in menu navigation [3], scrolling, and high-precision zooming [7].



**Figure 1: (a) PressureFish: Fisheye pressure menu; (b) Mouse augmented with one pressure sensor.**

The discretization method that researchers employed for dividing the pressure range into discrete units or levels is an important aspect of the pressure-based systems in previous studies. However, since manufacturers of pressure-sensing devices apply different analog-to-digital (AtoD) converters there is no standard mechanism to discretize the number of pressure levels. As a result, there are many methods and mappings for discretizing the number of controllable levels using a pressure-based device [3,6,8]. Ramos et al. [7] and Mizobuchi et al. [6] used a linear discretization function, while Ramos et al. [8] used a parabolic-sigmoid discretization function that resulted in a slow response at low pressure levels, linear behaviour in the middle levels, and a slow response at the high levels of the pressure range. Cechanowicz et al. [3] used a quadratic discretization function that allocated larger pressure ranges at the lower levels and smaller pressure ranges at higher levels.

In this paper we present the design of *PressureFish*, a fish-eye discretization function (see Figure 1) and compare it to a variety of discretization methods proposed in the literature. We carry out our investigation on a pressure augmented mouse [3]. Our results show that the fisheye function increases accuracy without compromising speed. For example, at 10 pressure-levels with a fisheye function, users are significantly more accurate with 78% accuracy compared to 54% for linear and require significantly less target crossings (i.e., overshooting the target before acquiring it) with an average of 0.4 crossings per trial compared to 0.7 for linear. Overall, by using the fisheye discretization function users are able to exhibit better control of pressure input.

## DISCRETIZATION FUNCTIONS

The analog force exerted by the user on a pressure sensor is converted to a digital data stream through a manufacturer-specific AtoD converter. As a result, manufacturers provide 256, 512, or 1024 discrete integer pressure values. However, users cannot effectively control these large numbers of discrete values. Applications further discretize the raw integer values by grouping adjacent values into unique controllable pressure levels [3,8]. Here we describe the various discretization functions (henceforth referred to as ‘function’) we evaluated in this study.

In our descriptions we use the following variables:  $x$  – the raw pressure value from the sensor;  $l$  – the number of pressure levels the space is divided into;  $R$  – the total number of raw pressure values from the pressure sensor.

*Linear Discretization:* A linear function (L) partitions the entire pressure space into equal units. For instance, a pressure space of 600 units ( $R = 600$ ) divided into 10 levels ( $l = 10$ ) would produce levels consisting of 60 pressure units each. The linear function is given by

$$y = \text{floor}((x * l) / R)$$

Numerous studies have reported using a linear function to control pressure input [2,3,5].

*Clustered Discretization:* Some groups [3,8] have used functions that assign more pressure levels to the middle range of the pressure space by hand-picking various design parameters like the starting pressure unit for each level and the number of pressure units for each pressure level. Rather than hand-pick, we used a K-means clustering algorithm to discretize the space. Users were asked to select randomly highlighted pressure levels discretized using the linear function described above and a quadratic function described by Cechanowicz et al. [3]. We collected raw pressure values for 6 users (208 trials/user) and used the K-means-clustering algorithm to design an overlapping discretization for each pressure level. Following a pilot study that showed no significant difference between the quadratic and linear functions, we decided to proceed with the linear function only, to allow us to compare and contrast linear (L) with K-mean clustered linear (KC).

*PressureFish Discretization:* This fisheye function (FE) was inspired by the fisheye distortion functions introduced by Furnas [4] and applied to fisheye menus [1]. The idea of a fisheye function is to make the area near the point of interest highly visible. This results in a distortion with a variable amount of space reserved for the various elements in the pressure space. Items further away from the focal point occupy less space, while items closer to the focal point occupy more space, and the item of focus itself occupies the largest amount of space. While this distortion of the visual space offers enhanced visibility researchers have also reported targeting problems that arise from the constant change of control-to-display ratio [5].

However, the fisheye function could be particularly advantageous as a discretization function for three reasons. First,

when the pressure cursor is at the level of interest, the fish-eye function automatically increases the amount of pressure values assigned to that pressure level. As a result, when the user presses the pressure sensor and fixes it to a particular level, the selected pressure value remains selected despite small variations in the applied pressure value. Second, finger-tips have a tendency to exert inadvertent forces. Such forces directly affect the movement of the pressure cursor, thereby reducing the level of user control. Since the fisheye function reserves sufficient space for the active pressure item, minor forces from the tips of the finger do not significantly impact pressure control. Finally, since the control space involves depressing a sensor rather than moving a mouse, users are less likely to have targeting problems.

We use the following fisheye function ( $r$  = fisheye radius):

$$y = \begin{cases} \text{floor} \left( \frac{(x - r) * (l - 1)}{R - r} \right) + 1 & \forall x > r - \frac{R - r}{l - 1} \\ 0 & \forall x \leq r - \frac{R - r}{l - 1} \end{cases}$$

To effectively control the fisheye selection, several design choices are possible. Each of the design parameters consist of modifying the values for  $r$ ,  $R$ , and  $l$  given the equation above. In this study, we used values of  $R = 600$ ,  $l$  consisting of values 6, 8, 10, 12, and 16, and  $r$  was assigned a value of 120 pressure units. These values were selected based on a number of pilots we ran before starting the final study.

*Visual Fisheye Discretization:* Visual feedback is an essential element in pressure-based interaction [3,6–8]. While the fisheye function divides the entire pressure space into non-uniform units, the visual fisheye (VF) function uses an underlying linear function but presents the visualization as a fisheye menu. As a result, the users are controlling the pressure cursor using a linear function but are being led into believing that the pressure is being controlled using a fish-eye technique. The motivation behind the design of VF is that if such a technique were to be successful then developers could simply enhance the visual presentation of pressure input. We were interested in identifying whether the visual effects were sufficient to improve control without changing the underlying discretization function (i.e. identify the degree of importance of visual feedback on pressure input).

## COMPARISON OF DISCRETIZATION FUNCTIONS

Our experimental goal was to examine differences in accuracy, speed and number of crossings using different functions. The experimental design we used was adapted from two other studies [3,8].

The experimental software recorded trial completion time (MT), errors (E), and number of crossings (NC) as dependent variables. MT is the total time taken for the user to apply the appropriate amount of pressure and select the target. NC is the number of times the cursor leaves a target after first entry for a particular trial. E is the number of times the participant selects a location which is not a target. The trial ended only when the user selected the right target, so multi-

ple errors were possible for each trial. While MT gives us an overall success rate, E and NC provide information about the level of control achievable using each of the different pressure-control mechanisms.

We used an optical mouse with a pressure sensor mounted on the left side, where it is easy and comfortable to be accessed with the thumb (Figure 1). The sensor (model #IESF-R-5L from CUI Inc.) could measure a maximum pressure value of 1.5N and provided 1024 pressure levels. However in our experiment we only used the range from 0 to 600, as earlier studies suggest that user fatigue is common at higher pressure ranges [3]. The software was implemented in C# and the sensor was controlled using the Phidgets library. The experiment was conducted on a 1024×768 pixels screen with a P4 3.2 GHz, Windows XP.

#### Task and Stimuli

In the task, participants were asked to control a red cursor moving vertically in a gray rectangular menu. The cursor starts at the top of the gray menu when the pressure value is 0. The cursor moves down when participants press the pressure button and moves up when participants release the pressure button. The menu is divided into small units based on the selected function and the number of pressure levels. The system randomly highlights, in yellow, a menu item the user is required to select. In each trial, participants are required to move the red cursor into the yellow target area and select the target with a Dwell or Click selection mechanism, which have been shown to be the best selection mechanisms for a pressure mouse [3]. The trial ends when the participant selects the appropriate target. If the selected item is not the right target, then the item changes to a dark gray, and the trial continues until the participant selects the right target. To select using Dwell, users maintain the cursor within the target for 750 ms, whereas Click users click with the left mouse button.

The study used a 4×4×5×2 within-participants factorial design. The factors were:

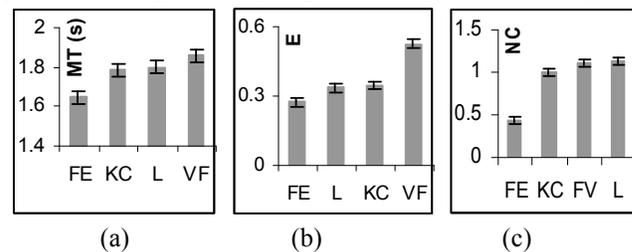
- Function: FE, VF, L, and KC.
- Relative Pressure Distance: 128, 256, 384, and 512.
- Pressure Level: 6, 8, 10, 12, and 16.
- Selection Mechanism: Dwell and Click.

The order of presentation was first controlled for function type, and then for pressure level. Levels of the other two factors were presented randomly. After explaining the selection mechanisms, participants were given ample time to practice the techniques. The experiment consisted of three blocks with two repetitions per block, per condition.

With 14 participants (9 male, 5 female; average age = 27 years), 4 functions, 4 distances, 5 pressure levels, 2 selection mechanisms, 3 blocks, and 2 trials, the system recorded a total of 13440 trials and the experiment took approximately 1 hour per user. None of the participants had any experience with pressure-based input.

#### Results

We used the univariate ANOVA and Tamhane post-hoc pair-wise comparisons (unequal variances) for our analyses.



**Figure 3: Average performance of the different functions (best on the left and worst on the right) with performance measures (a) Movement Time (MT); (b) Errors (E); (c) Number of Crossings (NC).**

**Movement Time:** We found no significant difference between functions ( $F_{(3,39)} = 1.898$ ,  $p=0.15$ ). FE was the fastest followed by KC, L and VF (see Figure 3a). As previously reported [2,7] we too found that Click was significantly faster than Dwell ( $F_{(1,13)} = 19.105$ ,  $p<0.001$ ).

**Errors:** Overall, we found a significant difference in E between the functions ( $F_{(3,39)} = 4.264$ ,  $p<0.01$ ) and selection techniques ( $F_{(1,13)} = 46.91$ ,  $p<0.001$ ). Post-hoc pair-wise comparison of the functions showed that FE had significantly fewest errors followed by L, KC, and VF (no significant difference between L and KC). Figure 3b shows the average E for each function. There was a significant interaction between selection technique and function ( $F_{(3,39)} = 7.654$ ,  $p<0.001$ ). In the case of Click, the ranking of the functions was similar to that reported above, while for Dwell the order was L, KC, FE, and VF. However we could not find any significant difference between the functions for Dwell selection technique.

**Number of Crossings:** We found a significant difference in NC between the functions ( $F_{(3,39)} = 19.606$ ,  $p<0.001$ ) and selection techniques ( $F_{(1,13)} = 7.77$ ,  $p<0.02$ ). Post-hoc comparison of the functions showed that FE had significantly fewer crossings than all other functions followed by KC, VF, and L (no significant difference between VF and L). Figure 3c shows the average NC for each function. We found no significant interaction between selection technique and function ( $F_{(3,39)} = 2.86$ ,  $p>0.05$ ).

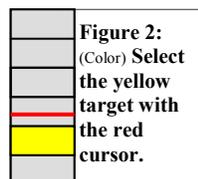
**Subjective Feedback:** FE was most preferred by nine users followed by L with three, KC with one and one with VF.

#### DISCUSSION

The results of our experiment reveal that a fisheye function allows users to select pressure levels with greater accuracy and with lower numbers of crossings without losing out on performance time.

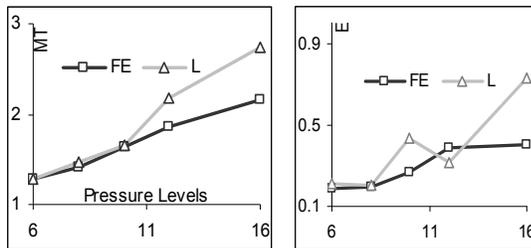
#### Fisheye Improves Pressure Control Across Levels

In line with our expectation, our results show that the method of discretizing the pressure space has a significant effect on the user's ability to control pressure. Additionally, users preferred this function over all the others. This effect is felt across all pressure levels (see Figure 4). We found that the PressureFish function is effective as it primarily



**Figure 2:**  
(Color) Select the yellow target with the red cursor.

reduces the amount of inadvertent crossings and allows the user to “lock” into a specific pressure level.



**Figure 4: Average performance of FE and L across various pressure levels for (a) Movement time and (b) Error rates.**

Results on speed, number of crossings, and accuracy, indicate that performance decreases gradually as the number of pressure levels increases beyond 6 (see Figure 4, KC and VF always perform no different from L and have been removed from the figure to avoid visual clutter). However, beyond 12 pressure levels, we observe a very sudden drop in performance with all functions except the fisheye. In the case of the fisheye function, users can control up to 16 pressure levels almost as comfortably as 12.

#### Effect of Fisheye Control on Selection Techniques

As reported in previous studies [3, 8] we too observed a larger number of errors with the Click selection technique in comparison to Dwell. One reason for this is that any force applied by one finger co-activates adjacent fingers simultaneously [9]. This effect is pronounced in the case of the Click selection technique as clicking the mouse button with the index finger activates muscles in the thumb, which in turn interferes with pressure control on the sensor. However, our results show that the fisheye function operates equally well with both selection techniques in terms of error rate, as well as the number of crossings. This suggests that fisheye functions can be universally applied across different selection mechanisms. Although untested, we believe this result is valid for a pressure sensitive stylus.

#### Effect of VF and KC on Pressure Control

Our results showed that users had difficulties controlling pressure in the VF condition. This result is consistent with other similar findings on desktop pointing and focus targeting with Fisheyes which suggest that distorting the control space results in better control [2] and distorting the visual space causes targeting problems if careful consideration is not given for the control space [5].

In all conditions, we found no significant difference in performance between KC and L. However, in most cases KC was marginally better than L. This can be attributed to the overlapping pressure units and the context-sensitive manner of deciding the pressure level. However, we believe that better segmentation of the pressure units could be achieved by careful analysis of the different types of errors (overshoot vs. undershoot) that users commit.

#### Recommendations to Designers

We provide the following recommendations:

- Designers should consider using a fisheye function to improve pressure control.
- A fisheye function will enable the use of a larger number of discrete pressure levels.
- Visual feedback is an essential but not sufficient factor for the enhancement of pressure control.

#### CONCLUSION

In this paper we report on the design and effectiveness of PressureFish, a fisheye discretization function that allows users to control pressure input with fewer errors than previously reported discretization techniques, without time penalties and with higher user preference. We believe our results will facilitate integrating pressure-based input with other input mechanisms. In the future, we will investigate the possibility of designing pressure menus that behave similarly and that share the common advantages of marking-menus. We will also investigate other fisheye functions to improve accuracy and facilitate the design of novel and improved navigation techniques such as pressure-scrolling, panning and zooming. Effective control of pressure input can also lead to designs that allow users to manipulate the control-to-display ratio in instances such as cursor control in multi-display or large display environments.

#### ACKNOWLEDGEMENTS

Thanks to Aoife Currid for editing and proof-reading this document and the reviewers for insightful comments. This work was partially funded by NSERC.

#### REFERENCES

1. Bederson, B. B., Fisheye mnus. *Proc. UIST 2000*, 217-226.
2. Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. Semantic pointing: Improving target acquisition with control-display ratio adaptation. *Proc CHI 2004*, 519-526.
3. Cechanowicz, J., Irani, P., and Subramanian, S., Augmenting the mouse with pressure sensitive input. *Proc. CHI 2007*, 1385-1394.
4. Furnas, G. W., Generalized fisheye views. *Proc. CHI 1986*, 16-23.
5. Gutwin, C. Improving focus targeting in interactive fisheye views. *Proc. CHI 2002*, 267-274.
6. Mizobuchi, S., Terasaki, S., Keski-Jaskari, T., Nousiainen, J., Ryyanen, M., and Silfverberg, M., Making an impression: Force-controlled pen input for handheld devices. *Proc. CHI 2005 Extended Abstracts*, 1661-1664.
7. Ramos, G. and Balakrishnan, R., Zliding: Fluid zooming and sliding for high precision parameter manipulation. *Proc. UIST 2005*, 143-152.
8. Ramos, G., Boulos, M., and Balakrishnan, R., Pressure widgets. *Proc. CHI 2004*, 487-494.
9. Schieber, M.H., and Santello, M., Hand function: Peripheral and central constraints on performance, *J. of App. Phys.*, 96 (2004) 2293-230.