

PAPER

Non-Linear Distance Filter for Modeling Effect of a Large Pointer Used in a Gesture-Based Pointing Interface

Kazuaki KONDO^{†a)}, *Member*, Takuto FUJIWARA^{†b)}, *Nonmember*, and Yuichi NAKAMURA^{†c)}, *Member*

SUMMARY When using a gesture-based interface for pointing to targets on a wide screen, displaying a large pointer instead of a typical spot pattern reduces disturbance caused by measurement errors of user's pointing posture. However, it remains unclear why a large pointer helps facilitate easy pointing. To examine this issue, in this study we propose a mathematical model that formulates human pointing motions affected by a large pointer. Our idea is to describe the effect of the large pointer as human visual perception, because the user will perceive the pointer–target distance as being shorter than it actually is. We embedded this scheme, referred to as non-linear distance filter (NDF), into a typical feedback loop model designed to formulate human pointing motions. We also proposed a method to estimate NDF mapping from pointing trajectories, and used it to investigate the applicability of the model under three typical disturbance patterns: small vibration, smooth shift, and step signal. Experimental results demonstrated that the proposed NDF-based model could accurately reproduced actual pointing trajectories, achieving high similarity values of 0.89, 0.97, and 0.91 for the three respective disturbance patterns. The results indicate the applicability of the proposed method. In addition, we confirmed that the obtained NDF mappings suggested rationales for why a large pointer helps facilitate easy pointing.

key words: *gesture-based pointing interface, human motion dynamics, pointer size*

1. Introduction

Large screens are prevalent in various scenarios, including discussions with slides and information announcements in public spaces. To facilitate open communication with large screens, we need a remote interface that enables us to indicate displayed contents easily and intuitively. A gesture-based pointing system is such an interface. It recognizes pointing postures via visual sensing to display pointers at the locations being pointed to. It can measure pointing postures non-invasively and does not require additional special devices like laser pointers. However, remote visual sensing is subject to considerable errors in the estimation of the pointing posture. Even small errors in the body coordinate are enlarged in the screen coordinate at a distance. Users perceive significant disturbances in the location of the displayed pointer, such as rapid vibration or moving by itself. Therefore, users must expend considerable effort to control pointer locations. To achieve smooth communications, it is

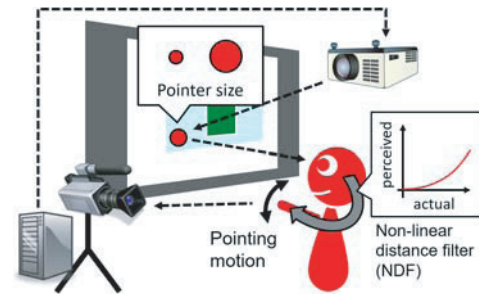


Fig. 1 An objective of this study: formulating pointing motions affected by a large pointer via embedding of NDF into a feedback loop, as indicated by the dashed lines.

necessary to reduce the influence of disturbance to alleviate the pointing effort.

Displaying a large pointer instead of a typical spot pattern can help to reduce pointing effort. This approach is evident from our daily experience. Consider indicating a target on a screen with a laser pointer. We often feel it is easier to indicate targets when projecting a circular pattern rather than a spot pattern. However, it remains unclear why a large pointer helps facilitate easy pointing. To develop a reliable pointing interface, the relationship between pointer size and interface usability must be scientifically analyzed, in conjunction with an end-to-end investigation that includes a subjective performance evaluation. To examine this issue, in this study we propose a mathematical model that formulates human pointing motions affected by a large pointer subjected to considerable disturbances. Our idea is to describe the effect of the large pointer as human visual perception. We consider that when displaying a large pointer, the user will perceive the pointer–target distance as being shorter than it actually is, or weakly activate body kinematics for pointing than the perceived distance. We implemented this scheme, referred to as non-linear distance filter (NDF), as a projection from the actual pointer–target distance on the screen to the subjective distance perceived by the user, and embedded it into a typical feedback loop model designed to formulate human pointing motions (Fig. 1).

We expect the proposed method to provide the following two approaches for analyzing interface usability.

(1) Perception-based approach

NDF quantitatively describes the effect of a large pointer as human visual perception. It works as an intermediate feature space mediating interface design

Manuscript received December 20, 2019.

Manuscript revised May 12, 2020.

Manuscript publicized August 3, 2020.

[†]The authors are with Academic Center for Computing and Media Studies, Kyoto University, Kyoto-shi, 606–8501 Japan.

a) E-mail: kondo@ccm.media.kyoto-u.ac.jp

b) E-mail: fujiwara@ccm.media.kyoto-u.ac.jp

c) E-mail: yuichi@ccm.media.kyoto-u.ac.jp

DOI: 10.1587/transinf.2019EDP7328

and usability. We can see how much and when the pointer-target distance is perceived differently from the actual distance, which enables us to scientifically analyze why a large pointer facilitates easy pointing. Aside from pointer size and disturbances, pointing motions are affected by other factors such as the pointing accuracy requirement and the amount of attention allocated to a pointing task. The proposed model has a possibility to describe the effect of those factors with NDF.

(2) Trajectory-based approach

The proposed model can simulate pointing trajectories as time-series values. Because the trajectory reflects interface usability and pointing performance, we can quantitatively evaluate them without actual experiments. Obviously, greater effort is required when performing a rapid pointing movement than when performing a smooth and slow pointing movement. Therefore, trajectory jerk [1] and trajectory torque [2] have been used as metrics of pointing effort and efficiency. Comparing the effort index values observed with and without disturbance demonstrates how much pointing effort is increased by the disturbance. Comparing spot pointer and large pointer conditions under the same disturbance indicates how much the effort is reduced by displaying the large pointer. This pointing effort analysis can be achieved by measuring actual pointing trajectories rather than by performing a simulation. However, the consistency in the results derived by these different methods is meaningful. When they are consistent, the proposed method provides rationales for the acquired results. When they are inconsistent, it suggests the existence of an unmodeled factor.

To show that the above approaches are viable, the applicability of the proposed model, i.e., how accurately the proposed model explains actual pointing motions, must be analyzed. In this study, we investigated applicability in terms of a pointer size and a disturbance pattern to focus on how the proposed method can describe large pointer's effectiveness against disturbance.

The remainder of this paper is structured as follows: In Sect. 2, the contribution of this study is clarified in comparison to previous works. In Sect. 3, we propose a mathematical model of pointing behaviors containing NDF and a way to analyze it. In Sect. 4, we discuss the input/output mapping of NDF estimated from the measured pointing trajectories, and in Sect. 5, its polyhedral approximation is proposed. Finally, in Sect. 6, the performance of the proposed model is evaluated through simulation of pointing trajectories.

2. Related Works

One of the representative works in earlier analysis of human pointing behavior is by Fitts et al. [3]. They investigated reaching motion in several experimental tasks, including pushing a target a distance by pen and moving rings or small pins. With D denoting the initial distance to the target

and W denoting the size of the target measured along the axis of motion, the movement time MT required to reach the target is given by $MT = a + b \log_2(D/W + 1)$, known as Fitts' law, where a and b are empirically or experimentally determined constants. The applicability of this fundamental formula has been investigated in various scenarios, such as pointing on a horizontal plane [4] and reaching the target with different aspect ratios [5]. These reports have contributed to the design of efficient user interfaces.

Only the target size W is controllable in the three variables focused on in Fitts' law. Based on the fact that a large target is easy to reach, various "making the target larger" methods have been proposed for electric display devices, such as changing the target size and shape according to the pointer-target distance [6] and zooming the neighborhood of the pointer [7]. Changing the pointer size has a similar effect to changing the target size. Ren et al. proposed a variable pointer size controlled by the pressure on a touch device [8]. A pointer called a "bubble cursor" [9] automatically deforms its shape based on the distribution of target candidates in the neighborhood. A deformable pointer proposed by Chapuis et al. [10] becomes large when it moves rapidly. It indirectly contributes to easy pointing by improving the visibility of pointers moving at high speed. A dynamic change of control-display gain is also used to improve pointing usability. It provides precise pointing with rough movement [11]. In terms of developing a pointing interface for a large screen, Nancel et al. investigated the performance of a two-step pointing strategy that employs rapid pointing with a large circular pointer and precise adjustments with a cross-shaped pointer [12]. As explained above, large pointers have been widely used to improve pointing usability. Rationales for these improvements were provided by Fitts' law.

In performance evaluations of those pointing interfaces, participants are usually requested to select small items displayed on a screen. A typical task is clicking an icon using a mouse to activate corresponding software. This experimental configuration assumes a pointing interface as input media for sending control commands. Evaluation criteria are how fast and accurately items can be selected. According to this assumption, conventional works mostly analyzed selection success rates, time required for selection, and pointer-item distances at selection time. For example, researchers have evaluated the performance of laser pointers in comparison with pointing gestures on a large display [13], analyzed the control of contents on a large screen using intuitive grasp/release gestures [14], and examined the use of wearable instruments such as wrist watches or cell phones as pointing devices [15], [16]. In contrast, we assume pointing behaviors used for smooth communication, not for content control. After a pointer reaches a target, the user must maintain the targeted location for a certain duration so that the audience can recognize the indicated target. Additionally, pointer users often draw lines to emphasize particular words or sentences, or draw circles to indicate targeted figures. In those cases, the pointing motions have significant meaning

rather than end of pointing. Therefore, conventional metrics based primarily on item selection are insufficient, and pointing motion analysis is necessary for evaluating the performance of pointing interfaces [17]. Unfortunately, Fitts' law does not meet this requirement because it only addresses static variables D , W , and MT . This is why we have proposed a new formulation focused on pointing motions. The proposed model is regarded as an expansion of Fitts' law into a temporally dynamic framework.

Researchers have developed approaches to formulate various human motions such as arm reaching [18], knob turning [19], and computer mouse control [20]. These works utilized a fundamental feedback loop model proposed by McRuer et al. [21], which contains the dynamics of the target apparatus and adopts human behavior as its controller. We have followed this approach and expanded the loop model to build a mathematical model of pointing motion [22]. In our previous study, the NDF describing effect of a large pointer was designed in a top-down manner [23]. In the current study we attempt to estimate the input/output mapping of the NDF from measured pointing trajectories in a bottom-up manner to analyze the NDF features in actual pointing. Furthermore, we address the influence of the pointing velocity on the NDF to build a more precise model.

3. Pointing Behavior Model with a Non-Linear Distance Filter

3.1 Model Structure

Researchers have formulated human behavior when controlling various equipment including arm reaching [18], turning knobs [19], and using a mouse [20]. These formulations were derived from a fundamental feedback loop model proposed by McRuer et al. [21]. It contains the dynamics of the target apparatus and human behavior as its controller. In the present study we follow this approach to build a mathematical model of pointing behavior.

A pointing system consists of a user and a pointing interface that estimates a pointing location via visual sensing and displays a pointer at that location (Fig. 1). When the user moves the pointer to the target, those components work in the following flow.

1. The user determines a target location p_t .
2. The user changes his or her pointing posture so that the pointer moves closer to the target.
3. The pointing interface displays the pointer at an estimated pointing location $p_e = p_g + p_d$, where p_g and p_d are the ground truth value of the pointing location and the estimation error, respectively.
4. The loop of steps 2-3 iterates until the user recognizes the pointer location p_e as being the same as the target location p_t .

This procedure is formulated as a negative feedback loop, shown in Fig. 2. H_a in the loop corresponds to motor dynamics around the shoulder or elbow joint. H_c works as

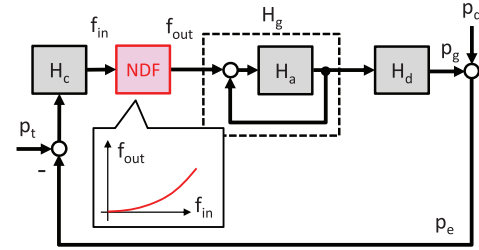


Fig. 2 Our proposed feedback loop model of pointing behavior. An estimated pointing location p_e consists of its ground truth p_g and estimation error p_d caused by visual sensing.

its controller. Conventionally, those human dynamics were integrated into one component. However, in our proposed model, the human dynamics is split into H_a and H_c because we want to analyze the NDF existing between them. The NDF controls the manipulation values f_{out} of the motion dynamics H_a according to the pointer–target distance f_{in} . It is expected to describe the change in pointing behavior when using a large pointer. Note that H_c and H_a describe the temporal dynamics, whereas the NDF relates its inputs and outputs in the spatial domain.

The output of H_a is an absolute pointing location, but the NDF gives H_a a relative value based on a pointer–target distance. Because of this inconsistency, a sequential chain of H_c , the NDF and H_a is not adequate to describe pointing behaviors. To solve this problem, the proposed model has a small positive feedback loop $H_g(s) = \frac{H_a(s)}{1-H_a(s)}$ that returns the current pointing location to the input value of H_a . In this way, H_a gets a sum of the manipulation values for “starting from the current location” and “moving closer to the target”. It results in the pointer reaching the target.

Considering human body movement activated by muscles, H_a is formulated as a second-order lag element:

$$H_a(s) = \frac{K_a}{T_a^2 s^2 + 2\zeta T_a s + 1}, \quad (1)$$

where K_a , T_a , and ζ correspond to the gain, the latency constant, and the damping parameter of the motor dynamics, respectively. Based on the human controller component proposed in [21], H_c is formulated as a first-order lag element:

$$H_c(s) = \frac{1}{T_c s + 1}, \quad (2)$$

where T_c determines the latency performance. Whereas [21] used a combination of a first-order lag and a lead element as a more general formulation, H_c includes only a lag feature. Why we configure it so is explained in Sect. 6. Since the estimation error is modeled as p_d , the pointing interface is assumed to be just a delay component. All the delays in the loop are integrated into $H_d(s) = e^{-s\tau}$, where τ is the total delay including the human motion dynamics, the controller, and the pointing interface.

In practical use of a pointing interface for communication, the user often performs a dual task in which they simultaneously indicate the targeted content on the screen

and verbally describe it to the audience. In this dual task, the attention of the user is divided into the two tasks, and at times insufficient attention is allocated to the pointing task; this results in ambiguities in visual perception and in the activation of pointing kinematics. Thus, to formulate pointing motions during a dual task, we must consider an additional factor: how much attention is being allocated to the pointing task. However, the amount of attention allocated to pointing is difficult to reliably control and measure. Deep consideration is necessary to describe the influence of the attention factor. Thus, in both the modeling and the experiments in this study, we assume that a pointer user is not disturbed by the verbal task and that the user's attention is sufficiently allocated to the pointing task.

3.2 Method to Analyze the NDF

How much the NDF converts a target–pointer distance to a manipulation value is described by its input/output mapping. It is estimated from actual pointing trajectories using the following method. We consider an accurate measurement of the pointing posture, because even invasive sensors are allowable in the analysis phase. It assumes the estimated pointing location to be equal to the ground truth p_g . Given p_g and artificial p_t, p_d , input values f_{in} of the NDF and their corresponding outputs f_{out} are derived through the pointing dynamics components H_g, H_c , and H_d . Under the assumption of those dynamics being always the same, we employed H_g, H_c , and H_d estimated beforehand using p_g, p_t , and p_d when a spot pointer is displayed. In the proposed model without the NDF, the input and output values of the chained dynamics $H_d H_g H_c$ are given as $p_t - (p_g + p_d)$ and p_g , respectively. The optimal parameters of H_g, H_c , and H_d are estimated by minimizing

$$\|p_g(t) - \mathcal{L}^{-1}\{H_d(s)H_g(s)H_c(s)(p_t(s) - p_g(s) - p_d(s))\}\|_2, \quad (3)$$

where the operator $\mathcal{L}$ denotes Laplace transform.

The input values f_{in} are easily derived by sequentially forward calculation: $f_{in}(t) = \mathcal{L}^{-1}\{H_c(s)(p_t(s) - p_g(s) - p_d(s))\}$. On the other hand, obtaining the output values f_{out} requires inverse processing corresponding to $(H_d H_g)^{-1}$. However, $(H_d H_g)^{-1}$ is not proper and f_{out} cannot be derived in a forward manner as for f_{in} . The proposed method estimates a sequence of f_{out} that minimizes the evaluation function over a total duration:

$$E_{dyn} = \sum_{t=0}^N \| \widehat{p_g}(t) - p_g(t) \|_2 + w_{sm} \sum_{t=1}^{N-1} \left\| \frac{d^2 f_{out}(t)}{dt^2} \right\|_2, \quad (4)$$

$$p_g(t) = \mathcal{L}^{-1}\{H_d(s)H_g(s)f_{out}(s)\}, \quad (5)$$

where N is the number of temporal samples. The first term of Eq. (4) is a residual between the actual pointing location and those generated by the model. The second term is a smoothness constraint that works as a regularizer with the weight w_{sm} .

A non-linear optimization is not adequate for estimating a large number of variables f_{out} because we must check various initial value combinations to acquire a global minimum. Therefore, Eq. (5) is transformed into a linear and discrete form in the following way, so that a linear solver is applicable. First, the delay τ is discretized into the sampling number N_{delay} . It replaces the delay component H_d in Eq. (5) by temporal shifting in the sampling index:

$$E_{dyn} = \sum_{t=0}^{N-N_{delay}} \|\widehat{p_g}(t + N_{delay}) - p_g(t)\|_2 + w_{sm} \sum_{t=1}^{N-N_{delay}-1} \left\| \frac{d^2 f_{out}(t)}{dt^2} \right\|_2, \quad (6)$$

$$p_g(t) = \mathcal{L}^{-1}\{H_g(s)f_{out}(s)\} \quad (7)$$

Padé approximation is also available, but it is not easy to determine the optimal order number when considering the trade-off between approximation accuracy and stability. Since human behavior is ambiguous and often fluctuating, configuring the accurate delay time in human dynamics is not meaningful. Approximating delay τ as the integral multiple of the sampling rate is sufficiently reasonable.

An additional approach for linearization uses a state space representation that converts a transfer function described in the frequency domain into a function in the real domain. In the state space representation, a complicated system is decomposed into elementary components using additional parameters called state variables. The input and output variables of the original system are related via a first-order differential equation $p_g(t) = H_g(f_{out}(t), \mathbf{x}(t), \dot{\mathbf{x}}(t))$ using the state variables $\mathbf{x}$. Discretizing the differential parts of the state space representation provides its linear form $p_g(t) = H_g(f_{out}(t), \mathbf{x}(t))$. Specifically, using the same sampling rate used in the delay approximation, $p_g(t) = H_g(f_{out}(t), \mathbf{x}(t))$ is represented as

$$\mathbf{x}(t+1) = \mathbf{A_g}\mathbf{x}(t) + \mathbf{B_g}f_{out}(t) \quad (8)$$

$$p_g(t) = \mathbf{C_g}\mathbf{x}(t) + \mathbf{D_g}f_{out}(t), \quad (9)$$

where $\mathbf{A_g}, \mathbf{B_g}, \mathbf{C_g}$ and $\mathbf{D_g}$ indicate a state transition matrix, an input matrix, an output matrix, and a direct feedthrough matrix, respectively. Given an initial state $\mathbf{x}(0)$, a general form of p_g is described as

$$p_g(t) = \mathbf{C_g}\{\mathbf{A_g}^t + \sum_{i=0}^{t-1} \mathbf{A_g}^i \mathbf{x}_0 \mathbf{B_g} f_{out}(t-1-i)\} + \mathbf{D_g}f_{out}(t). \quad (10)$$

It transcribes the residual part $p_g(t + N_{delay}) - \widehat{p_g}(t)$ in Eq. (4) in a matrix form. Finally, with the difference approximation of the quadratic differential part of Eq. (6), the evaluation function E_{dyn} becomes a sum of the square of the linear terms. Thus, the sequence of optimal f_{out} that minimizes E_{dyn} is easily acquired using a least-square method.

4. Analysis of the NDF when Displaying a Large Pointer

4.1 Experimental Environment

Measurement of actual pointing trajectories is necessary

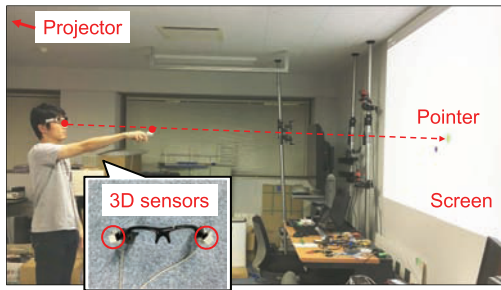


Fig. 3 Experimental environment for measuring pointing behavior. Two sensors are attached on the glasses to measure the positions of the participant's temples.

to analyze the NDF using the proposed method. The experimental environment for the measurement, as shown in Fig. 3, was configured as follows: A projector on the ceiling constructs a 1920×1080 pixels screen over an approximately 2.3 m width $\times$ 1.4 m height area on the white wall. The participants stood approximately 2.5 m away from the screen and were asked to straighten their arms while pointing. The screen served as an approximately 49° horizontal $\times$ 33° vertical degrees apparent field of view (FOV) for a participant. This is an intermediate size when considering the effective visual field is $30^\circ \times 20^\circ$ and the stable fixation field is $60^\circ\text{--}90^\circ \times 45^\circ\text{--}70^\circ$. In the former FOV, humans can detect, identify, and discriminate visual stimuli with natural eye movement, whereas the latter FOV requires additional head movement. For these viewpoints, our configuration assumed a visual-content-based communication with slight head movement.

Under the straight arm condition, the pointing location was estimated as being where the indicating vector, running from the virtual reference point inside the head to the fingertip, intersects with the screen [24], [25]. To accurately acquire the indicating vector, magnetic-field-based three-dimensional sensors were attached to the participants' temples and the tip of their right forefinger. The reference point is assumed to be located at the midpoint of the temples. Note that this sensing method is allowed when analyzing human pointing behavior, whereas remote sensing should be used in practical applications. The pointing target, a spot pointer, and a large pointer were displayed as filled circles with 2 cm, 3 cm, and 5 cm radius on the screen, respectively.

4.2 Measurement of Pointing Trajectories

The pointing task given to the participants was compensating for unexpected pointer movements to match the pointer location to the target one. It configures "keep pointing location for a certain duration" behaviors in a practical use of a pointing interface.

The target location $p_t = 0$ was fixed to the origin of the screen coordinate in front of the participants. Artificial disturbance signals assumed as estimation errors p_d were added to the horizontal coordinate of the estimated pointing location on the screen. This method enables us to clarify the

relation between the estimation errors and user responses against them. The following three patterns of artificial disturbance signals were used in the experiments.

a. Small vibration disturbance

This pattern of disturbance is caused by temporal perturbation of the image intensity. To generate the disturbances, a high frequency domain of random walk sequences was extracted and normalized so that their maximum magnitude became 4.0 cm. The maximum value corresponds to the pointing location error when practical 1 cm measurement errors appear on both the center of the head and the fingertip positions. The sampling rate of the random walk was 30 Hz, as generally used for image capture.

b. Smoothly shifting disturbance

The pointing location errors are accumulated when a tracking algorithm is used to estimate pointing postures. To generate this pattern of disturbance, we selected a similar method to that used for the small vibration disturbance. A low frequency domain of random walk sequences was extracted and normalized so that their maximum magnitude became 20.0 cm. This value corresponds to the maximum error when the estimated position of the center of the head lies inside of the head volume even with the error accumulation.

c. Step disturbance

When the interface fails to detect the particular feature point on the body, the estimated pointing location suddenly jumps. 10 cm step signals recurring with intervals of several seconds simulate such a case. The step magnitude corresponds to the pointing location error when only one eye is detected and reluctantly regarded as the center of the head instead of the midpoint of two eyes. Incidentally, this pattern of signal works as not only sudden disturbance also reference motions to indicate the target at 10 cm distance.

With different random seeds, the three sequential signals for each pattern of disturbance were generated. The duration of each signal was 10 s.

As mentioned in the introduction, this paper investigates the applicability of the proposed method in terms of a pointer size and a disturbance pattern. To achieve this, other pointing conditions were simplified and fixed in the experiment. Conditions to consider include the amount of attention allocated to a pointing task, and the pointing accuracy requirement. To meet the assumption that the attention of the user is sufficiently allocated to the pointing task, the participants silently performed the pointing experiments and no one spoke to them during the pointing.

The pointing accuracy requirement was configured in a similar manner. If assuming possible practical use cases, we need to configure various pointing accuracy requirements based on visual content displayed on the screen and the context in the communication. To simplify this pointing condition, we defined a general and comprehensive criterion of the pointing accuracy. It was described by the appearance of

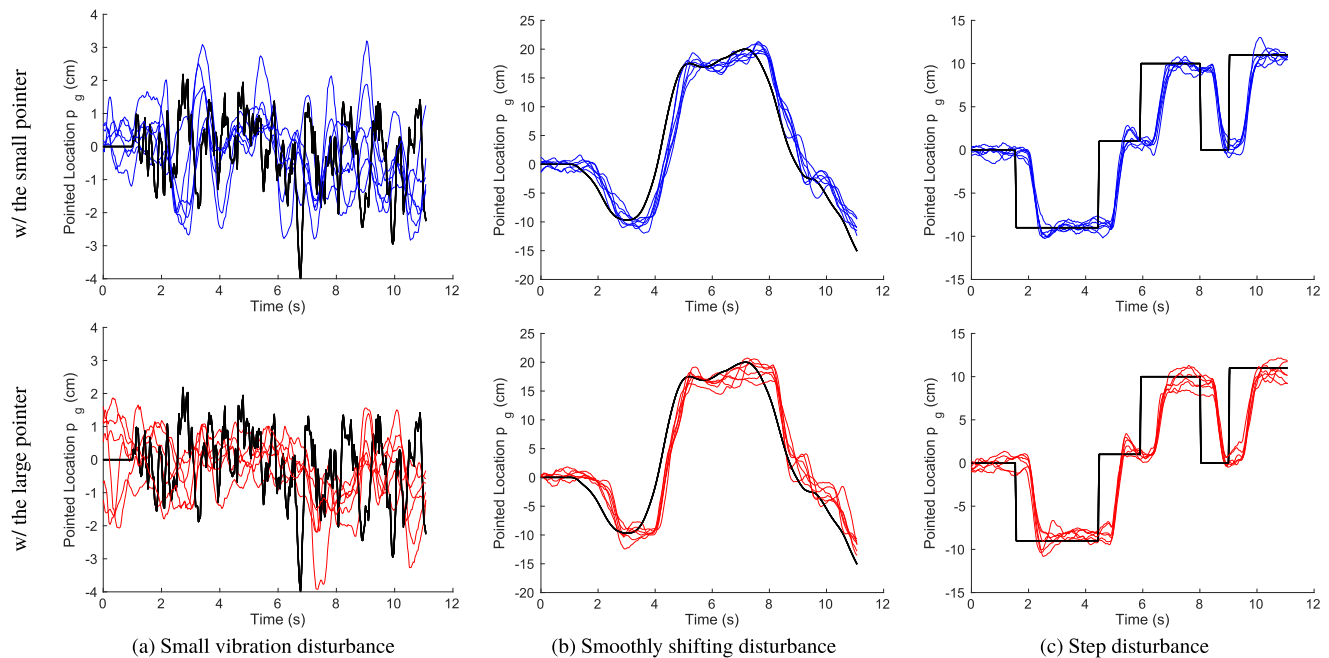


Fig. 4 Examples of measured pointing trajectories. The black trajectories are artificial disturbances given to the participants. The blue and red trajectories indicate their responses when displaying the spot pointer and the large pointer, respectively. The response trajectories are inverted in the figure to clarify the relation to the disturbances.

a target and pointer on the screen. Specifically, the requirement is met when the target is hidden by the pointer; the center locations of the target and pointer need not match precisely. This requirement was explained to the participants using the instructions written below. No requirement related to pointing motion velocity was assumed or explained to the participants. Note that these conditions were used to clarify pointing accuracy criterion in the participants' mind and did not necessarily approximate a particular pointing situation. The actual instructions given to the participants were:

- The pointing task starts when the participant puts the pointer on the target.
- The pointer moves by itself due to the disturbance. The participant changes his or her pointing posture to keep the pointer location at the target location.
- The pointing accuracy requirement is met when the target is hidden by the pointer. The participant does not need to match the center locations of the pointer and target, precisely.

We confirmed that the requirement given to the participants was not excessively difficult because they naturally performed the pointing tasks, and no participant reported psychological loads such as impatience or disconcertedness. Even if they felt minor amount of these psychological loads, it could be also in an actual pointing situation in which a user needs to precisely indicate a target. Thus, the pointing accuracy requirement configured in the experiment was reasonable.

The participants were 7 male and 1 female aged 22–40 years. The pointing trials were carried out 6 times under

each experimental condition defined by the combination of the disturbance signal and the pointer pattern (spot/large). Thus, each participant conducted 108 pointing trials in total. Each participant rehearsed for 5 minutes with disturbance signals generated with different random seeds from those used in the measurement trials. In the experiment, the signs of the disturbance signals were randomly inversed and the order of the experimental conditions was shuffled, so that participants could not predict what pattern of disturbance would appear in the next trial, as in a practical situation. Moreover, the three patterns of disturbance all have quite different signal forms, as shown in Fig. 4. Even the user who first sees those disturbances can instantly distinguish them. Therefore, the experimental condition in terms of how the participants can recognize the disturbance effectively approximates a pointing condition in a practical situation.

The measured pointing trajectories were differently influenced by the disturbance patterns and the pointer sizes. Several examples of the measured trajectories are shown in Fig. 4. Consider Fig. 4(a), corresponding to the small vibration disturbance. The pointing location strongly respond to the high frequency disturbances when displaying the spot pointer. It changes to the slow compensation motion when displaying the large pointer. Interestingly, these compensation motions did not disappear even when the participants used the large pointer. The magnitudes of given disturbance signals (drawn as black lines) were mostly within several centimeters. The participants did not need to move in most cases when using the 5 cm radius large pointer. However, the actual pointing motions (drawn as red lines) demonstrate

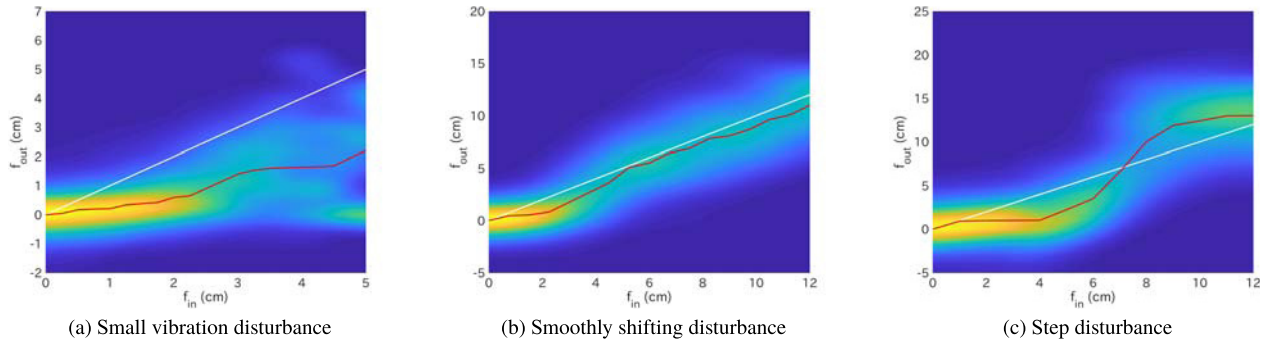


Fig. 5 Examples of estimated NDF mapping in density form. The red line in each figure indicates polygonal approximation of the mapping. The white straight line means the identity mapping $f_{out} = f_{in}$.

that they responded to the disturbance at almost every moment. It is difficult for a human to accurately predict and respond to high frequency disturbances. In such cases, making the center locations of the target and pointer closer is a reasonable strategy because it generates a margin to handle those disturbances. We believe this is why the participants responded to the disturbance even when they did not need to respond. In the case of smoothly shifting disturbance shown in Fig. 4 (b), we can see the spot pointer induced high tracking performance. Meanwhile, the large pointer provided insensitive responses for the small pointer–target distances. When the step disturbances were given, the large pointer reduced the overshoot values and made the pointer location stay at a little far from the target (Fig. 4 (c)).

4.3 Fundamental Analysis of the Input/output Mapping

To analyze the NDF input/output mapping, the method proposed in Sect. 3.2 was applied to the pointing trajectories measured in the experiments. The weight of the smoothness constraint in Eq. (6) was configured as $w_{sm} = 20$ based on preliminary investigations with the various w_{sm} .

The density forms of the estimated input/output mapping are shown in Fig. 5. In all patterns of disturbance, the NDF delivered smaller f_{out} than the actual pointer–target distance f_{in} when the f_{in} is small. As we expected, using the large pointer invokes insensitive responses to the disturbance. It induces smaller pointing motions and reduces the effort needed in pointing tasks compared with a typical spot pointer. NDF could be a useful feature space for analyzing the effects of pointer size when conducting a usability analysis.

The individual features caused by the different disturbance pattern are explained in the following analysis.

a. Small vibration disturbance

The mapping result had high density at almost zero or quite small f_{out} in the $f_{in} < 2\text{cm}$. Although it spreads over the wide range of f_{out} in larger f_{in} , $f_{out} < f_{in}$ is still maintained. The participants became insensitive to the disturbance when the target is almost covered by the large pointer.

b. Smoothly shifting disturbance

The NDF mapping density still had a single peak along f_{out} even when f_{in} was large. The participants responded steadily to the smoothly shifting disturbances. As f_{in} became larger than the pointer radius 5 cm, the NDF mapping got closer to $f_{out} = f_{in}$. It means the large pointer had almost no effect when the pointer–target distance was large. This result well reflects the pointing trajectories shown in the bottom of Fig. 4 (b): the pointing location stayed while the pointer–target is small, and then it moved rapidly.

c. Step disturbance

The pointing strategy against the suddenly jumping disturbances consists of two behaviors: (1) moving the pointer closer to the target as soon as possible, and (2) maintaining the pointer location after reaching the target. Those two behaviors emerged as $f_{out} > f_{in}$ in the $f_{in} > 8\text{cm}$ region and quite small f_{out} in the $f_{in} < 4\text{cm}$ region, respectively. In the intermediate range, the mapping described the transient movement until the pointer reaches the target. As the pointer became closer to the target, f_{out} got smaller rapidly and the pointing movement slowed down.

4.4 Consideration of Pointing Motion Velocity

Even when the particular f_{in} is given, the estimated input/output density map has a variance of several centimeters along f_{out} . Unfortunately, these magnitudes are significantly large compared with the absolute values of f_{out} . A scheme reducing this ambiguity is required to increase the modeling accuracy. We focused on pointing motion velocity because the conventional models contain the prediction term in their controller. Figure 6 shows the relationship between the pointing motion velocities and the modeling errors given by the polygonal approximation of the input/output density shown in Fig. 5. The pointing motion velocities were derived as differentiated outputs of H_g . Since the error distributions have strong positive correlations, the ambiguity of f_{out} will be reduced when the pointing motion velocity is used as an additional input variable to the NDF. In our coordinate definition, the direction from the target to the

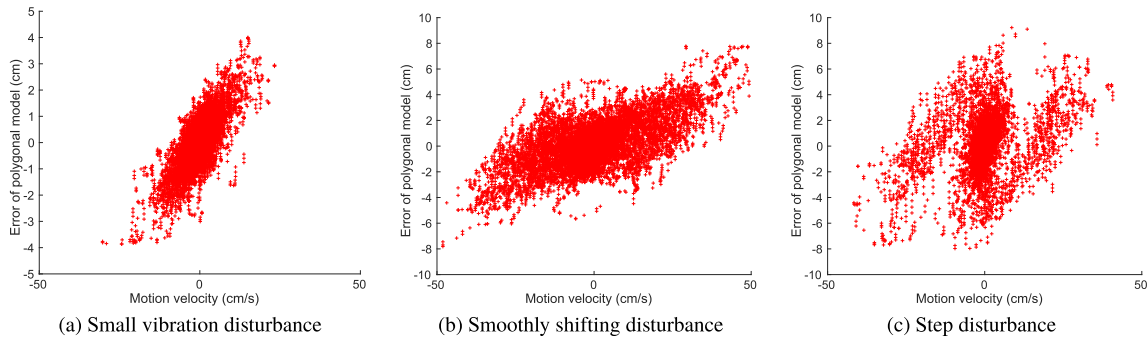


Fig. 6 Relationship between pointing motion velocity extracted from outputs of H_g and modeling errors caused by the polygonal approximations shown in Fig. 5.

pointer is positive. Thus, the distributions shown in Fig. 6 indicate that the manipulation values become larger when the pointer moves away from the target. This analysis is supported by the subjective pointing strategy, i.e. the participants predicted that the pointer would be moving away from the target and they attempted to move the pointer close to the target more strongly.

5. Polyhedron Model of the NDF

Based on the above analysis, we formulate the NDF as a non-linear mapping $f_{out} = f_{nd}(f_{in}, m_v)$, where m_v denotes the pointing motion velocity derived by differentiating the outputs of H_g . We assume a planar approximation in each local part of the NDF to formulate it with an assembly of multiple planes, i.e. a polyhedron as shown in Fig. 7. Our polyhedron model is defined by its vertices $P_v(i, j) = f_{out}(f_{in}^{step} i, m_v^{step} j)$, $i = 0, 1, \dots, N_{in} - 1$, $j = 0, 1, \dots, N_v - 1$ on the grid of the f_{in} - m_v plane. f_{in}^{step} , m_v^{step} and N_{in} , N_v denote the grid size and the grid numbers, respectively. It formulates f_{nd} as a bilinear interpolation

$$\begin{aligned} f_{nd}(f_{in}, m_v) &= (1-p)(1-q)P_v(i, j) + (1-p)qP_v(i, j+1) \\ &\quad + p(1-q)P_v(i+1, j) + pqP_v(i+1, j+1), \\ p &= \frac{f_{in} - f_{in}^{step} i}{f_{in}^{step}}, \quad q = \frac{m_v - m_v^{step} j}{m_v^{step}} \\ &\text{for } i, j \text{ s.t.} \\ &\quad f_{in}^{step} i < f_{in} \leq f_{in}^{step} (i+1), \\ &\quad m_v^{step} j < m_v \leq m_v^{step} (j+1). \end{aligned} \quad (11)$$

The vertices $P_v(i, j)$ are determined to minimize an expectation value of fitting errors

$$E_p = \sum_k \sum_l \sum_m NDF(k, l, m) (f_{out}(m) - f_{nd}(f_{in}(k), m_v(l))), \quad (12)$$

where $NDF(k, l, m)$ denotes the discretized NDF density map in the 3D space defined by f_{in} , m_v , and f_{out} .

Non-linear optimization is required to estimate the most likely value of $P_v(i, j)$ because $f_{nd}(f_{in}, m_v)$ contains

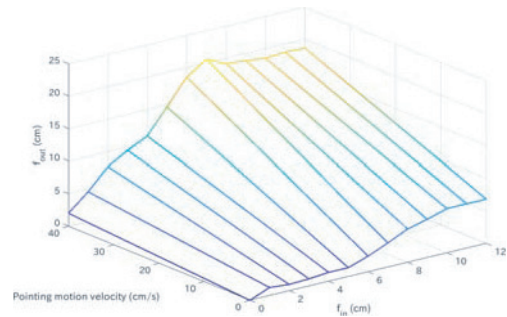


Fig. 7 Illustration of the proposed polyhedral model of the NDF mapping.

Table 1 Grid parameters used to estimate the polyhedron NDF.

Dist. pattern	f_{in}^{step} (cm)	m_v^{step} (cm/s)	N_{in}	N_v
Small vibration	0.25	15	20	2
Smooth sift	1.0	40	12	2
Step	0.75	40	16	2

thresholding to acquire the grid indexes i and j . In the current study, we configure the initial values of $P_v(i, j)$ that approximate the 2D density maps on the m_v - f_{out} plane at each $f_{in} = f_{in}^{step} i$ with a line. Those lines were defined by gravity centers of the density and their principal axes of inertia. The grid parameters used in the estimation are shown in Table 1. The examples of the estimated polyhedron NDFs for each disturbance pattern are shown in Fig. 8. When a particular value of m_v is given, ambiguity of f_{out} becomes smaller than that in the 2D NDF mapping shown in Fig. 5.

6. Simulative Evaluation of Model Performance

To validate the performance of the proposed model, we discuss the similarity between the simulated trajectories $\hat{p}_g$ and the actually measured trajectories p_g .

Pointing trajectories without the NDF were simulated as $\hat{p}_g = \mathcal{L}^{-1} H_d(s) H_g(s) H_c(s) (p_t(s) - p_e(s))$, whereas those with the estimated polyhedron NDF were derived according to the flow of $f_{in} = \mathcal{L}^{-1} \{H_d(s) (p_t(s) - p_e(s))\}$, $f_{out} = f_{nd}(f_{in}, m_v)$, and $\hat{p}_g = \mathcal{L}^{-1} \{H_d(s) H_g(s) f_{out}(s)\}$. The magnitude of the modeling errors depends on the starting time and

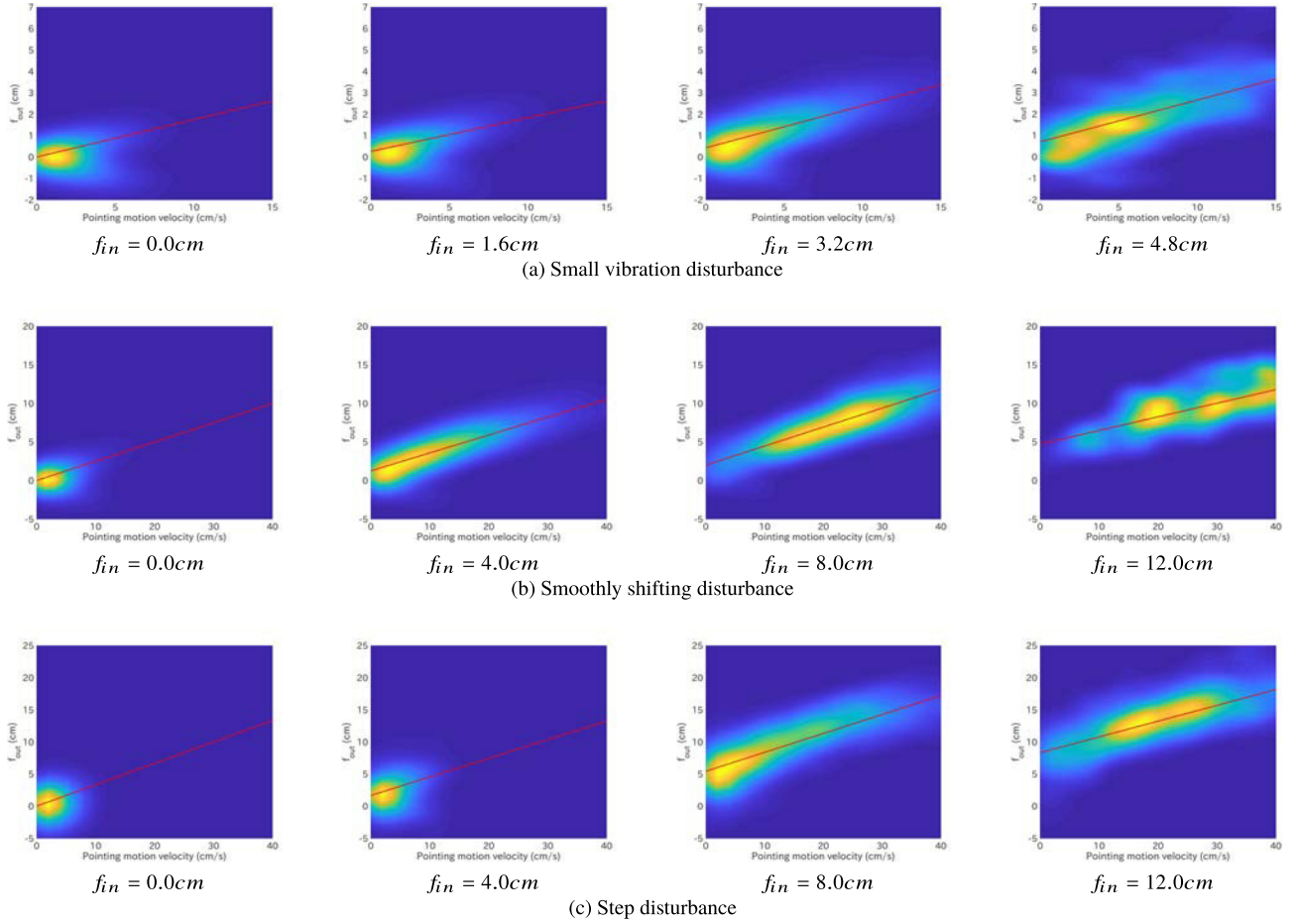


Fig. 8 Polyhedral approximation of the NDF mapping. The figures illustrate the NDF density maps and the sections of the estimated polyhedron NDF on the several m_v – f_{out} planes.

the duration of the simulation. In our evaluation, the trajectory simulations started at 1.0 s intervals and lasted for 3.0 s.

Given the examples of the trajectory comparison shown in Fig. 9, the proposed method containing NDF can produce closer trajectories than without NDF. The remaining modeling errors are explained below. In the responses to the small vibration disturbance, the magnitudes of the simulated trajectories are smaller than those of the actual trajectories, especially at the local peaks. Refer to the NDF density map of the $m_v > 7$ cm/s region shown in the top row of Fig. 8 with $f_{in} = 3.2, 4.8$ cm. The sections of the estimated polyhedron NDF illustrated by the red lines lie a little lower than the center lines of the density. It delivered smaller f_{out} than actual and resulted in the smaller magnitude in the simulated trajectories. The polyhedron NDF formulation with more degree of freedom, e.g. $N_v > 2$ will reduce those modeling errors. The NDF corresponding to the smoothly shifting disturbance lay at almost the center of its density. The simulated trajectories contained similar, smaller, and larger magnitudes in almost equal proportion. In the case of the step disturbance, the trajectory mismatch frequently appeared at the stationary parts, such as the 2.5–4.0 s and 7.0–8.5 s periods in Fig. 9(c). Those modeling errors were caused by forcedly applying the polyhedron NDF to the diffused den-

sity for small f_{in} . The NDF mapping densities shown in Fig. 8 with $f_{in} = 0.0, 4.0$ cm were difficult to approximate with the lines.

The signal coherence indicating similarity of the given two signals with 0–1.0 value was used to quantitatively evaluate the model performance. It enables us to deeply analyze how they are similar in the frequency domain, e.g. the coherence values at low frequency reflect similarity of global magnitude, whereas those at high frequency represent similarity of wave form in the short duration. The estimated NDF achieve significantly high coherence values as shown in Fig. 10. Their mean values in less than 4.0 Hz reach 0.89, 0.97, and 0.91 for the small vibration, smoothly shifting, and step disturbance, respectively. It certifies that, with human visual perception described by NDF, the proposed model well explains pointing motions when displaying the large pointer.

The drastic improvement seen in Fig. 10(a) reflects great influence of the large pointer to small vibration disturbance. The relatively low coherence values in the low frequency domain was probably due internal disturbances that are not supposed in the proposed model, such as body perturbations caused by rapid arm swings and following movements to correct them. The remaining two patterns of dis-

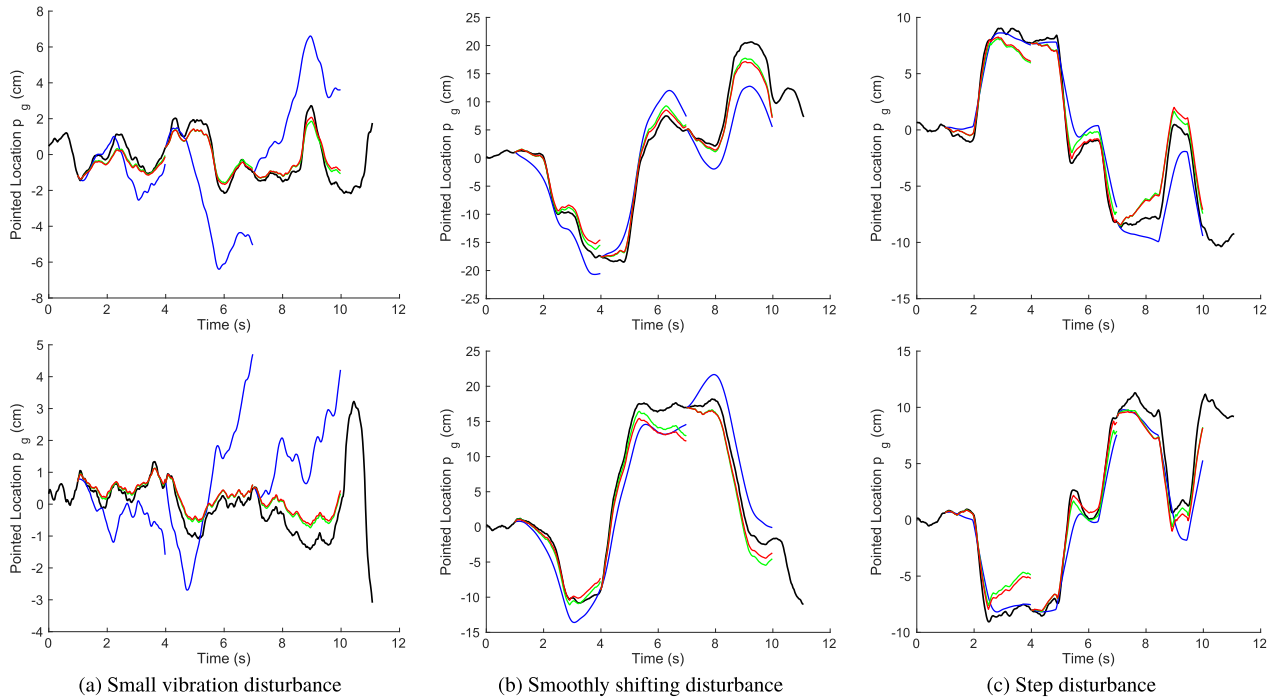


Fig. 9 Examples of simulated pointing trajectories starting from 1.0, 4.0, and 7.0 s. The blue, red, and green colored trajectories correspond to simulation results without NDF, with estimated polyhedron NDF, and with their mean NDF among the participants, respectively. The black ones are actual pointing trajectories as ground truth.

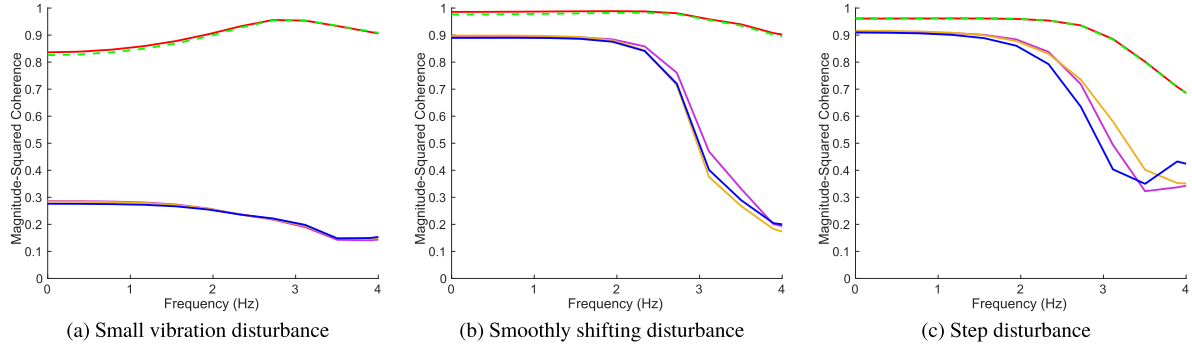


Fig. 10 Performance of the pointing model with the NDF evaluated by signal coherence between simulated and actual trajectories. The blue, red, and green lines have the same mapping as those used in Fig. 9. The purple and yellow lines indicate coherence when p_g and p_e are used to calculate pointing motion velocities, respectively. The frequency range is limited up to 4.0 Hz because human conscious behavior is included within it.

turbances did not provoke those unexpected disturbances. Thus, the high coherence values were maintained in the low frequency domain. In contrast, the coherence values gradually decreased as the frequency increased. It is in some ways unsurprising, because our proposed model considers only conscious body control; it does not assume quick responses derived by reactive motion. The relatively rapid decrease in the step disturbance case is explained by the insufficient polyhedron NDF approximation: applying the lines to the diffused densities discussed before. The smoothly shifting disturbance has such modeling ambiguity only in small f_{in} as shown in Fig. 8(b). Thus, the high coherence values achieved in the all-frequency domain.

As a deeper analysis, we investigated which position in the model should be used to derive the pointing motion velocity to acquire the best modeling performance. The pointing motion velocities derived from p_g and p_e instead of outputs of H_g resulted in little improvement or similar coherence values to those given by the model without NDF. We can say that the participants performed predictive controls based on the somesthetic sense of “how much the arm is going to move” rather than the observations of “how much the arm is moving” or “how much the pointer is moving”. Therefore, we did not include a prediction term in the controller H_c that receives $-p_e$. Most of the conventional works assumed control of vehicles or apparatus with complicated

dynamics. Predictive control should be based on the responses from the apparatus to avoid failures based on a false sense of confidence. On the other hand, in the pointing case, the users well recognize their own body dynamics and the pointing interface with only a certain delay. They can precisely predict pointing behavior before observing it. Furthermore, they are in no danger even if the pointing accuracy is not good, whereas driving a vehicle must maintain safety by accurate control. Therefore, the quick response based on the somesthetic sense is prior to the accurate but slow response via visual observation.

Additionally, we investigated the influence of personal traits on the change of pointing behavior. Contrary to our expectation, the simulated trajectories using the individually optimized polyhedron NDF and their mean were quite similar. Also, coherence values had almost no differences. At least in our experimental environment, instructions, and participants' variance, use of the large pointer similarly affects all participants' pointing behavior.

7. Conclusion

In this study, we proposed the feedback loop model of pointing behaviors when using a large pointer. The influence of the large pointer is described by the non-linear distance filter (NDF) that converts the actual pointer–target distance to subjective distance to use it as the manipulation values of human pointing dynamics. We also proposed a method to estimate input and output values of the NDF from pointing trajectories and to approximate obtained NDF density map using a polyhedron in the 3D space defined by the pointer–target distance, the pointing motion velocity, and the output manipulation value.

Under the three patterns of disturbance, the proposed method involving NDF accurately reproduced the actual pointing trajectories, which demonstrates its applicability for modeling pointing motions when displaying a large pointer. From the experimental results, we established the following observations regarding usability analysis. For all three patterns of disturbance, output values of the estimated NDFs tended to be smaller than the actual target–pointer distances. This indicates that a large pointer induces smaller pointing motions and reduces the effort needed in pointing tasks compared with a typical spot pointer. NDF appears to be a useful feature space for describing why a large pointer facilitates easy pointing. With the proposed pointing motion model, we can conduct a trajectory–based usability analysis, in which we will specifically develop a method to quantify pointing effort from pointing trajectories such as a jerk or torque, and then compare estimated effort values against a subjective effort index.

A remaining issue is validation under various conditions. Considerable conditions include other pointing accuracy required to users and the attention balance between pointing and explanation tasks. Other pointing tasks such as tracking of a moving target must also be examined. In addition, to generalize the proposed method, pointing conditions

such as pointer size should be defined as parameters in the model instead of discrete patterns.

References

- [1] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of Neuroscience*, vol.5, no.7, pp.1688–1703, 1985.
- [2] T. Uno, M. Kawato, and R. Suzuki, "Formation and control of optimal trajectory in human multijoint arm movement," *Biological Cybernetics*, vol.61, no.2, pp.89–101, 1989.
- [3] P.M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *Journal of Experimental Psychology*, vol.47, no.6, pp.381–391, 1954.
- [4] N. Smyrnis, I. Evdokimidis, T.S. Constantinidis, and G. Kastrinakis, "Speed-accuracy trade-off in the performance of pointing movements in different directions in two-dimensional space," *Experimental Brain Research*, vol.134, no.1, pp.21–31, 2000.
- [5] J. Accot and S. Zhai, "Refining fitts' law models for bivariate pointing," *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp.193–200, 2003.
- [6] M.J. McGuffin and R. Balakrishnan, "Fitts' law and expanding targets: Experimental studies and designs for user interfaces," *ACM Trans. Comput.-Hum. Interact.*, vol.12, no.4, pp.388–422, Dec. 2005.
- [7] L. Findlater, A. Jansen, K. Shinohara, M. Dixon, P. Kamb, J. Rakita, and J.O. Wobbrock, "Enhanced area cursors: Reducing fine pointing demands for people with motor impairments," *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, pp.153–162, 2010.
- [8] X. Ren, J. Yin, S. Zhao, and Y. Li, "The adaptive hybrid cursor: A pressure-based target selection technique for pen-based user interfaces," *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction*, vol.4662, pp.310–323, 2007.
- [9] T. Grossman and R. Balakrishnan, "The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area," *Proc. SIGCHI conference on Human factors in computing systems*, pp.281–290, 2005.
- [10] O. Chapuis, J.-B. Labrune, and E. Pietriga, "DynaSpot: Speed-Dependent Area Cursor," *CHI '09: SIGCHI conference on Human Factors in computing systems*, pp.1391–1400, 2009.
- [11] A. Worden, N. Walker, K. Bharat, and S. Hudson, "Making computers easier for older adults to use: area cursors and sticky icons," *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pp.266–271, 1997.
- [12] M. Nancel, E. Pietriga, O. Chapuis, and M. Beaudouin-Lafon, "Mid-air pointing on ultra-walls," *ACM Trans. Computer-Human Interaction*, vol.22, no.5, pp.1–62, 2015.
- [13] R. Jota, M.A. Nacenta, J.A. Jorge, S. Carpendale, and S. Greenberg, "A comparison of ray pointing techniques for very large displays," *Proc. Graphics Interface*, pp.269–276, 2010.
- [14] D. Vogel and R. Balakrishnan, "Distant freehand pointing and clicking on very large, high resolution displays," *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pp.33–42, 2005.
- [15] S. Koceski, N. Koceska, and I. Koccev, "Design and evaluation of cell phone pointing interface for robot control," *International Journal of Computer Applications*, vol.51, no.3, pp.27–32, 2012.
- [16] S. Siddhupuria, S. Malacria, M. Nancel, and E. Lank, "Pointing at a distance with everyday smart devices," *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp.173:1–173:11, 2018.
- [17] D.M. Wolpert, R.C. Miall, J.L. Winter, and J.F. Stein, "Evidence for an error deadzone in compensatory tracking," *Journal of Motor Behavior*, vol.22, no.4, pp.299–308, 1992.
- [18] B. Hoff, "A model of duration in normal and perturbed reach-

- ing movement,” *Biological Cybernetics*, vol.71, no.6, pp.481–488, 1994.
- [19] C.J. Hasser and M.R. Cutkosky, “System identification of the human hand grasping a haptic knob,” *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp.180–189, 2002.
- [20] S. Aranovskiy, R. Ushirobira, D.V. Efimov, and G. Casiez, “Modeling pointing tasks in mouse-based human-computer interactions,” *IEEE 55th Conference on Decision and Control*, pp.6595–6600, 2016.
- [21] D.T. McRuer and H.R. Jex, “A review of quasi-linear pilot models,” *IEEE Transactions on Human Factors in Electronics*, vol.8, no.3, pp.231–249, 1967.
- [22] K. Kondo, G. Mizuno, and Y. Nakamura, “Feedback control model of a gesture-based pointing interface for a large display,” *IEICE TRANSACTIONS on Information and Systems*, vol.E101-D, no.7, pp.1894–1905, 2018.
- [23] K. Kondo and Y. Nakamura, “Nonlinear visual perception for modeling a gesture-based pointing system using a large pointer,” *Proc. IEEE/SICE Int. Symposium on System Integration*, pp.638–644, 2019.
- [24] M. Fukumoto, Y. Suenaga, and K. Mase, “Finger-pointer: Pointing interface by image processing,” *Computers & Graphics*, vol.18, no.5, pp.633–642, Sept. 1994.
- [25] K. Kondo, G. Mizuno, and Y. Nakamura, “Analysis of human pointing behavior in vision-based pointing interface system: Difference of two typical pointing styles,” *Proc. 13th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, HMS 2016*, vol.49, pp.367–372, IFAC-PapersOnLine, 2016.



Yuichi Nakamura received B.E., M.E., and Ph.D. degrees in electrical engineering from Kyoto University, in 1985, 1987, and 1992, respectively. From 1990 to 1993, he worked as an instructor at the Department of Electrical Engineering of Kyoto University. From 1993 to 2004, he worked for Institute of Information Sciences and Electronics of University of Tsukuba, Institute of Engineering Mechanics and Systems of University of Tsukuba, as an assistant professor and an associate professor, respectively.

Since 2004, he has been a professor of Academic Center of Computing and Media Studies, Kyoto University. His research interests are on computer vision, multimedia, human-computer and human-human interaction including distance communication, and multimedia contents production.



Kazuaki Kondo received B.E. and M.E. degrees in Engineering Science, Ph.D. degree in Information Science and Technology, from Osaka university, in 2002, 2004, and 2007, respectively. He was a research associate in the Institute of Scientific and Industrial Research in Osaka University, from 2007 to 2009. He has been an senior lecturer at Kyoto University, Japan, from 2015. He has been working on the research of computer vision, human-computer interaction, and media computing. Current his

major work is to analyze human behaviors in various daily environments for designing human-friendly interfaces or supports, such as pointing, manipulating equipment's, object handover, and collaborative assembling.



Takuto Fujiwara received B.E. degree in electrical engineering from Kyoto University, in 2019. He is now a graduate student in master's course of Kyoto University. He has been working on analysis of pointing behavior.