HAL
open science

Designing Guiding Systems for Gesture-Based Interaction

William Delamare, Céline Coutrix, Laurence Nigay

To cite this version:

William Delamare, Céline Coutrix, Laurence Nigay. Designing Guiding Systems for Gesture-Based Interaction. EICS '15: Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, 2015, Duisburg, Germany, Germany. pp.44-53, 10.1145/2774225.2774847 . hal-01492570

HAL Id: hal-01492570

https://hal.science/hal-01492570

Submitted on 4 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Designing Guiding Systems for Gesture-Based Interaction

William Delamare, Céline Coutrix and Laurence Nigay
CNRS, Joseph Fourier University UJF-Grenoble 1
Grenoble Informatics Laboratory (LIG), UMR 5217, Grenoble, F-38041, France
{William.Delamare, Celine.Coutrix, Laurence.Nigay}@imag.fr

ABSTRACT
2D or 3D gesture commands are still not routinely adopted, despite the technological advances for tracking gestures. The fact that gesture commands are not self-revealing is a bottleneck for this adoption. Guiding novice users is therefore crucial in order to reveal what commands are available and how to trigger them. However guiding systems are mainly designed in an ad hoc manner. Even if isolated design characteristics exist, they concentrate on a limited number of guidance aspects. We hence present a design space that unifies and completes these studies by providing a coherent set of issues for designing the behavior of a guiding system. We distinguish Feedback and Feedforward and consider four questions: When, What, How and Where. In order to leverage efficient use of our design space, we provide an online tool and illustrate with scenarios how practitioners can use it.

Author Keywords
Gesture Interaction; Guiding System; Design Space; Feedforward; Feedback.

ACM Classification Keywords
D.2.2. Software Engineering: Design Tools and Techniques - User interfaces.

INTRODUCTION
Gesture commands have motivated numerous research studies regarding tracking mechanisms [7], recognition algorithms [15], gesture taxonomies [13], 2D [6] and 3D [14] applications to list a few. Moreover several commercial products exist for gesture interaction, such as the Microsoft Kinect, the Wii remote controller, and more recently the LeapMotion or the Atheer Air Platform presented during the 2015 Consumer Electronic Show (CES). However, despite all this technological progress, the adoption of gesture commands as an input modality is still moderate aside from entertainment applications. We argue that the fact that gestures are not self-revealing is a key bottleneck with respect to acceptance of gesture commands by a broader public [21] and for common everyday applications: users should namely know which commands are available and how to trigger them.

Two solutions to tackle the self-revealing obstacle exist. First, researchers have explored the gesture set design phase in order to study the association between gestures and commands. For instance, designers can propose expert-elicited [20] or user-elicited [16] gestures. However, these user studies highlight that there is no perfect consensus amongst users. Another solution is to propose interactive systems able to employ the users' own gestures [28]. Although conceptually appealing, this solution requires that all the interactive gesture-based applications are compatible. This is unlikely to occur soon. Second, users can benefit from a guiding system that shows available commands and their associated gestures [2, 3, 6, 9, 17]. Given the limitations of the first solution (i.e. design of gesture sets), guiding systems are essential for gesture commands. Yet, the design of guiding systems is still ad-hoc. We hence present a design space that specifies and organizes relevant design options for the design of guiding systems and therefore helps gesture-based interaction to be widely adopted.

Our design space builds on well-known, but scattered concepts in the Human-Computer Interaction domain, such as the notion of *mode* [11] or *spatial relationship* between interaction modalities [30]. We articulate and apply those concepts to the design of gesture guiding systems. Our design space defines a unifying framework for the design of gesture guiding systems by organizing a set of design issues along axes. These axes characterize the behavior of the guiding system. The scope of the design space is the design of the behavior of the guiding system, rather than the design of its concrete representation. Indeed having defined the behavior of the guiding system, the next step is the design of the concrete form of the guiding system (i.e., perceivable form of the provided guidance information). A salient feature of the design space is to highlight two fundamental aspects of guidance: the feedback and feedforward aspects (Figure 1). In order to efficiently use the design space for describing, comparing and designing new gesture guiding systems, an interactive tool allows practitioners and researchers to search and filter amongst the set of guiding systems that are described according to the design axes.

Our main contribution is the design space for characterizing guiding systems' behavior. Although our design space is

design-oriented rather than technology-oriented, we believe that the online tool is valuable for the engineering community as it promotes the use of our design space and gathers the knowledge about existing guiding systems.

DESIGN SPACE

Our design space is thus composed of two categories corresponding to two fundamental aspects for guiding systems: *feedback* and *feedforward*.

Feedback mechanisms provide information about the outcomes of the actions *already performed* by the user (i.e. in the past) [29]. For a gesture guiding system, the feedback mechanism should provide information about the performed gesture and how well this gesture is executed or recognized (i.e. intended/recognized gesture). In the simple 2D guide example of Figure 1, the feedback mechanism provides a dashed ink trail corresponding to the already performed portion of the gesture. Feedforward mechanisms provide information prior to any action (i.e. for future actions) [29]. For a gesture guiding system, the feedforward mechanism should show two types of information: (a) available commands to the user as well as (b), how to trigger the commands. In the example of Figure 1, the feedforward mechanism displays available gestures with their remaining paths following in order to trigger a command among commands '1', '2' and '3'. Although these two types of information are distinguished in [29], Kurtenbach et al. combine them as part of the Revelation phase [17]. In contrast to the feedback mechanism, we call the mechanism in charge of this Revelation phase the feedforward mechanism in the design space.

We describe both mechanisms with four groups of design axes related to four questions: *When* (temporal characteristics), *What* (content characteristics), *How* (medium characteristics) and *Where* (spatial characteristics). In particular, the *What* group of *feedforward* contains a subgroup highlighting the potential impact of the feedback mechanism onto the content of the feedforward mechanism. 35 axes compose our design space (Figure 2).

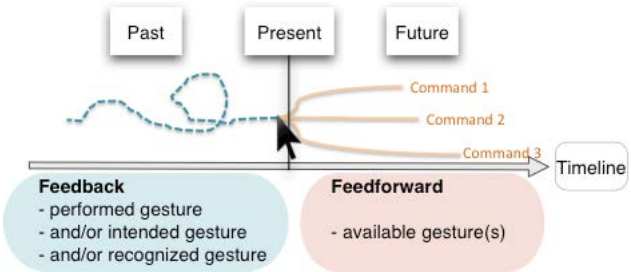


Figure 1: Gesture guiding systems: Feedback (i.e. performed/intended/recognized gestures) and Feedforward (i.e. available gestures) while the user is performing a gesture.

Group *When*

This group of axes characterizes the temporal features of the feedback and feedforward mechanisms. We consider the execution of one gesture as the referent unit of time. In this temporal reference frame, we distinguish the beginning (i.e. the *trigger*), the middle (i.e. the *execution*) and the end (i.e. the end of the guiding *mode*).

Trigger

We define three axes for describing the trigger event:

- *Initiative*: the mechanism can be triggered by the *user* or by the *system*. For instance, a *user-triggered* feedforward mechanism can display the gestures after touch and hold actions on a touch-sensitive surface [17]. A *system-triggered* mechanism can display more feedforward information if the user has difficulties executing the gestures [2].

A guide allows users to execute a gesture (e.g., 2D gesture or 3D mid-air gesture) using a given interaction modality that we name "primary" modality. We further refine the user-triggered option with two axes:

- *Modality*: the user can trigger the mechanism using the *same* modality as the primary one or using a *different* one. With a touch-based application for instance, touching the screen [3] uses the *same* modality as the primary one whereas a voice command in the case of a voice-activated gesture guide corresponds to a *different* modality than the primary one.

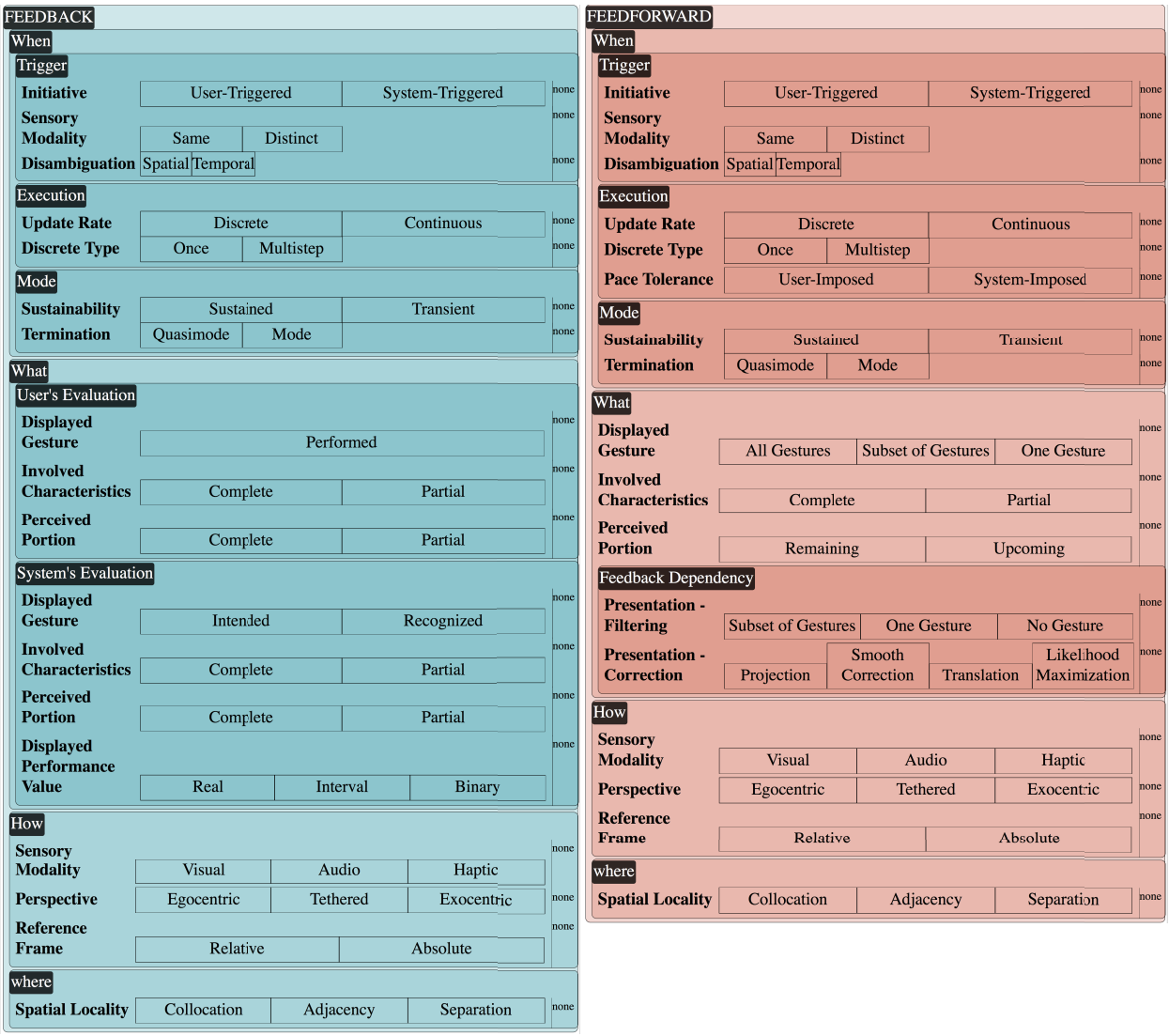
In case of a user-activated guide using the primary modality, a disambiguation mechanism is necessary to distinguish the triggering action from the gesture:

- *Disambiguation*: The system can detect the trigger using *temporal* criteria (e.g., holding a pose [3]) or *spatial* criteria (e.g., gesturing in a specific dedicated area [6] or touching a button [10]).

Execution

Once the user or the system has triggered the feedback or feedforward mechanism, we characterize the temporal features of the guiding system while executing a gesture:

- *Update Rate*: The information can be updated with a *continuous* or *discrete* mechanism. An example of a *continuous* update rate is a feedback mechanism drawing the 2D executed path in real time [1]. It allows the user to continuously judge her/his 2D gesture. Another example is a feedforward mechanism providing visual representations of the candidate gestures updated during the gesture execution [9]. A *discrete* update rate can be further described regarding its type:
- *Discrete Type*: The system can display the information only *once* or *many* times. A final feedback at the end of the gesture execution [9] or a display of the gestures only at the beginning (i.e., feedforward) as with a crib-sheet [3] illustrate the *once* option. In contrast a Hierarchical Marking Menu [18] updates the available commands



(a) (b)

Figure 2: Design space with 35 axes organized according two categories – (a) Feedback and (b), Feedforward. Each category is composed of four groups – *When* (temporal characteristics), *What* (content characteristics), *How* (medium characteristics) and *Where* (spatial characteristics). Each axis contains design options and the 'none' option.

each time the user crosses a level in the menu, and hence is a *multi-step* discrete type of feedforward mechanism.

Another axis describes the temporal constraints imposed on the execution of the gesture by the user. This axis is therefore specific to the feedforward mechanism:

- *Feedforward - Pace Tolerance*: the system can present the gestures to be performed with a *user-imposed* timing (e.g., the 3D self-guided arrow condition of LightGuide that allows the user to decide at which pace (s)he will execute the gestures [27]) or a *system-imposed* timing (e.g., a teacher avatar animation that the user has to imitate in real-time [12]). This dimension does not refer to the speed at which the gesture will be presented to the user (e.g., with a video [10]), but rather to the constraint imposed by the guide on the execution of the gestures by the user. The system-imposed option is mainly used when the speed of the gesture is a relevant concept of the application, such as when learning dance body movements [2].

Mode

We define how the feedback and/or feedforward mechanisms end after having been activated. Two axes characterize the end of the guiding mode:

- *Sustainability*: the mechanism can be *transient* or *sustained*. A transient feedback and/or feedforward can automatically disappear once the system recognizes a gesture [3]. Moreover if the system monitors the performance of the user, the system can judge that

feedback and/or feedforward is no longer needed [2]. For the case of a sustained mode, we can further refine the termination:

- *Termination*: The mode can be sustained as a *quasimode* (e.g., releasing a dedicated button to make the guide disappear [26]) or can be *persistent* (e.g., clicking on a 'close' button on screen [6]).

Group *What*

This group of axes characterizes the information conveyed by the feedback and feedforward mechanisms. While the feedforward mechanism provides information about the future actions only, the feedback mechanism involves two sources of information: the information about past actions performed by the user and the information about the processing of these past user actions by the system.

Group What: Gesture

A guiding system provides information concerning the past and future gestures. The following axes characterize such information about: the performed gestures (*feedback, 1*), the recognized or intended gestures (*feedback, 2*), and the gestures managed by the system (*feedforward, 3*).

1. *Gesture* axis of the *Feedback mechanism – User's Evaluation*: this axis includes only one option. In order to allow the user to evaluate her/his gesture, the system can display the *performed* gesture, such as an ink trail displaying the performed path [1].
2. *Gesture* axis of the *Feedback mechanism – System's Evaluation*: the system can represent the *intended* and/or the *recognized* gesture. The user can correct her/his gesture by comparing the executed and the intended gestures and by knowing what the system recognizes. A 3D teacher avatar showing the movement that the user should have performed [12] illustrates a display of the *intended* gesture. SimpleFlow [5], although not a gesture guiding system per se, displays the gesture that will be *recognized* if the user releases the mouse button.
3. *Gesture* axis of the *Feedforward mechanism*: the guiding system can reveal *all* available gestures, a *subset of gestures* or only *one* gesture. The first step of ShadowGuides [9] or the 3D Marking Menus [23] reveal *all* available gestures. The second step of ShadowGuides shows only a *subset of gestures* that are available from the registration pose executed during the first step [9]. Finally, GestureBar [6] proposes a toolbar presenting only the gesture associated with the desired command.

Having characterized the different types of gestures for feedback and/or feedforward mechanisms, we focus on the characteristics of the gesture that are made perceivable by the guiding system. When dealing with gestures, we often emphasize the shape characteristic (e.g., drawing a 'square' or a 'circle'). There is however a wide range of gesture characteristics. This is highly correlated with the recognition algorithm and the context in which the user is guided. For instance, in a context of gesture-based interaction, drawing a large vertical square in mid-air might be a different command than a small horizontal square. In the context of dance movement training, it might be relevant to process and evaluate the speed during the execution of a movement. In addition, some recognizers provide a way to ignore the scale and the orientation of the inputs [15]. But if these features are relevant for interaction, the guiding system can provide feedback and feedforward regarding the scale of the gesture, its orientation, but also its localization (e.g., performing a 'delete' gesture should be performed on a contextual object), the force (e.g., a 2D square can be drawn with thicker lines where the user has applied more pressure on the surface). There are also temporal characteristics that can be useful when guiding gestures such as dance movements, i.e. the whole duration, the local velocity or acceleration. We focus on a higher level of description and characterize the type of the guide. To do so, we define two axes: one regarding characteristics involved in the gestures' representation and another axis focusing on the quantity of information provided to the user.

- *Involved Characteristics*: the system can provide information regarding the *complete* set of characteristics (e.g. the shape, the force, the orientation, the name of the command, etc.) or a *partial* set. For instance, YouMove proposes distinct training stages that involve different characteristics, such as the stability while holding a pose or the smoothness while executing the gesture [2]. Note that a *complete* set of characteristics does not refer to an absolute exhaustive list of gesture characteristics. It refers only to the characteristics relevant for the application domain. Thus, if the guide aims to manage only the shape and the pressure characteristics for instance, its feedback mechanism will involve the complete set of characteristics if it provides information about the shape and the pressure.
- *Perceived Portion*: the feedback (resp. feedforward) mechanism can present the gestures so that the user perceives the complete available information, i.e. *complete* (resp. *remaining)* portion of the gesture or only a *partial* (resp. *upcoming)* portion (Figure 3). This axis describes how much the user will know about the gesture

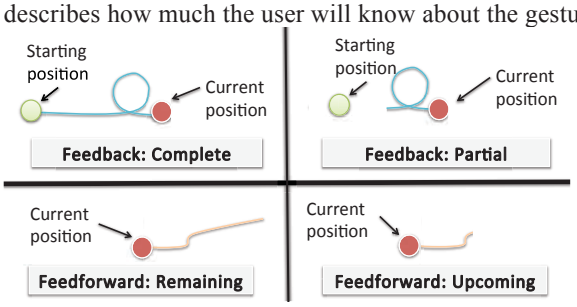


Figure 3: Illustration of the *Perceived Portion* axis. Top row: the feedback represents the complete performed gesture (left) or only a partial portion (right). Bottom row: the feedforward represents the remaining gesture to perform (left) or only an upcoming portion (right).

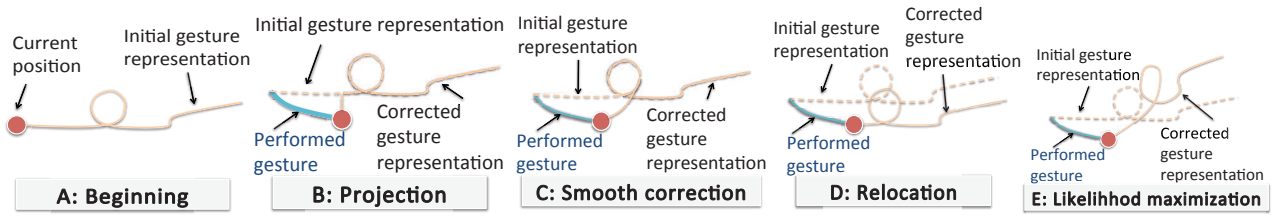


Figure 4: Illustration of the Presentation – Correction axis. A) Initial state at the beginning. B) Projection correction: the represented gesture includes an additional segment from the point of the initial path closest to the current position. C) Smooth correction: the represented gesture includes an additional segment that smoothly links the current position to the initial gesture. D) Relocation: the represented gesture is relocated according to the current position. E) Likelihood maximization: the combination of the represented gesture and the performed gesture maximizes the recognition score of the gesture.

portion. This design decision could be driven by minimizing the information overload on screen both for feedback and feedforward. For instance, users guided by a crib-sheet [3] perceive the *remaining* portions of the gestures as feedforward. In contrast, users guided by a Hierarchical Marking Menu perceive only the *upcoming* strokes [17], i.e. the direction towards the items of the current menu level. An ink trail provides a feedback by showing the *complete* path of the gesture [3]. Contrastingly, Gestu-Wan displays a partial portion by showing only the last 3D posture recognized during the recognition process [24].

Group What: Performance Value for Feedback

The guiding system might evaluate the execution of the gesture by the user. As part of the information provided by the feedback mechanism, we consider the *displayed performance value* resulting in a comparison between the performed portion of the gesture and the intended or recognized gesture:

- *Displayed Performance Value*: the system can provide a *binary* value, an *interval* or a *real* value to be interpreted by the user. A binary value can be the result of a recognizer such as 'recognized' or 'not' [18]. A real value can be the recognizer's score for a given gesture [12] or a custom metric such as the distance between the intended and current positions of the hands [22].

Group What: Impact of Feedback onto Feedforward

Two axes dedicated to Feedforward characterize how the feedforward informational content is impacted by previous actions, i.e. the feedback mechanism. Previous performed actions can have two impacts on the informational content provided by the feedforward mechanism:

- *Information – Filtering*: based on the already executed gesture, the guiding system can remove gestures from the initial set of gestures revealed at the beginning. During the execution of the gesture, the filtering process can be partial and defines a *subset of gestures* or only *one gesture*. The filtering can also be complete: *no gesture* is presented to the user. OctoPocus reveals at first all available gestures and makes the gestures with a recognition score under a threshold disappear, thus displaying only a *subset of gestures* [3]. We can also envision a version of OctoPocus that would display only *one gesture*, i.e. the gesture ranked best by the recognizer. The Adaptive Guide has a complete filtering process as it removes guidance during the execution of the gesture in order to enhance the learning aspect [1].
- *Presentation – Correction*: the guiding system can modify the content of the feedforward based on the already executed movement. If the trajectory of the gesture executed by the user deviates from the intended one, the guiding system can modify the content of the feedforward so that the current position is redirected to the closest point in space of the intended gesture, i.e. a *projection* (Figure 4, B). It can also *smoothly* redirect the current position onto the trajectory (Figure 4, C). The guide can also modify the position of the representation of the gesture by *relocating* it to the current position (Figure 4, D). Finally, the guide can modify the end of the gesture in order to *maximize* the likelihood of the gesture being recognized if the combination 'already performed deviated path' + 'modified remaining path' is executed (Figure 4, E). If the user deviates from the intended trajectory while being guided by the Follow Spot [27], the system will guide the hand according to the *projection* option. The 3D Arrow [27] or the motion retargeting mechanism for full body movements [25] allows a *smooth* correction of the hand towards the intended trajectory. OctoPocus [3] continuously relocates the representation of the gestures under the cursor.

Group How

This group of axes describes how the feedback or feedforward mechanism conveys guiding information: we focus on the type of output modalities and do not describe the final perceivable form, out of the scope of the design space. First we consider the human sensory modality involved for perceiving the guiding information.

- *Sensory Modality*: the guide can transmit information through a *visual, haptic* or *audio* sensory modality. For instance, the guide can visually display the shape of the gestures [6], use vibration to convey the scale of the deviation between intended and performed gestures [22] or use a sound to indicate the success or failure of a gesture [12].

We further describe visual options to present a gesture since the visual modality is commonly used in guiding systems. Rather than detailing the visual cues used for each characteristic of a gesture, we describe at a higher level of abstraction the design possibilities of the language of the visual modality, a modality being defined as the coupling of an interaction language with a physical device [30]. A second design step after defining the type of language is to define the concrete form for each characteristic of a gesture.

- *Perspective*: the display can be *egocentric* (i.e. from the user's point of view [27]) or *exocentric* (i.e. from a third person's point of view [22]). In between these options are *tethered* perspectives that are attached to the user, but still separated from her, such as an augmented mirror with a head-coupled mechanism [2].
- *Reference Frame*: the information can be displayed *relative* to the position of the user's limb to be guided or in an *absolute* reference frame. Several guiding systems display gestures centered on the cursor's position [3, 18]. Contrastingly, the first step of the interactive crib-sheet displays gestures in a predefined position regardless of the current cursor's position [17].

We emphasize that the *Perspective* and the *Reference Frame* are two independent axes. For instance LightGuide [27] displays the gesture to execute through visual cues as if on the user's hand (*egocentric*). However, this projection of the gesture does not follow the hand, but is anchored in a specific location in the world coordinates (*absolute*). While 3D Marking Menu [23] also proposes to display gestures on a distant screen from an *egocentric* point of view, gestures are centered on the cursor's position representing the user's hand (*relative*).

Group Where

This group contains only one axis and characterizes the spatial relationship between the location at which the user must focus to perceive guiding information and the location at which the user must focus to perform actions. With this definition, the spatial location of a guiding system for 2D gesture interaction using a mouse is defined with regards to the location of the cursor's representation. This axis will influence the choice of the physical device(s) for conveying feedback and feedforward information [30].

- *Spatial Locality*: the guide and the inputs can be *collocated*, *adjacent* or *separated.* LightGuide [27] projects collocated visual information on the user's hand to guide 3D gestures. ShadowGuides [9] displays the gestures in an area adjacent to the input area in order to prevent occlusion problems. For technical reasons, 3D gesture guiding systems often use a separate display in order to provide visual information without instrumenting the user [2, 12, 22].

We stress that the *Spatial Locality* and the *Reference Frame* are two independent axes. Indeed for instance ShadowGuides [9] creates a representation of the user's hands next to the physical hands (*adjacent*) and displays gestures centered on this representation (*relative*) while a static Crib Sheet [3] displays the gestures as static images next to the input area (*adjacent)* uncoupled from the user (*absolute*).

RELATED WORK

We build on previous work on design options for guiding systems as well as guiding systems.

Design Spaces

We focus only on studies that provide design options that explicitly address guiding systems and position them in our design space. We underline that we consider design options and not design recommendations, i.e. what the system can do rather than should do. For design recommendations, we refer the readers to Kurtenbach et al., who provide high-level design principles to support doing and learning gestures [17], Bragdon et al., who extend these principles by focusing on the approachability of the gesture-based interface [6], or Anderson et al., who provide design guidelines for body movement learning and training [2]. There are few design recommendations as few options have been identified. Indeed, design options have to be identified first before studying their impact on usability.

We describe the design options identified in the literature according to the two fundamental aspects of guidance that structure our design space: feedback and feedforward. While some studies consider both aspects [3], some focus only on one of them, either feedback [25] or feedforward [27].

Feedback

Bau and Mackay define four dimensions for characterizing the feedback mechanism for gesture-based interaction [3].

1. First, they define the *recognition value* presented to the user. This recognition value can be a binary value (e.g., yes/no), an interval or real value (e.g., the score from the recognizer). Our design space includes this axis.
2. Second, they define *filtering*, which describes how the feedback modifies the set of displayed gestures. We integrate this axis and extend it with the 'none' option illustrated by the Adaptive Guide [1] that can filter all gestures. Indeed, the Adaptive Guide removes the representations of the gestures earlier and earlier during the execution of the gesture to prevent users from relying too much on guidance and hence to enhance learning. Instead of including this axis in the 'feedback' group, we created a specific subgroup regarding the impact of feedback onto feedforward in the 'feedforward' group.
3. Third, they define the *update rate*. Indeed, the system can provide feedback continuously during the gesture execution, at multiple times or only once. In our design space this axis is split into two axes in order to distinguish continuous vs. discrete (multistep or once).
4. Finally, they define the *representation* axis, which describes how the feedback is presented to the user.

Since we focus on the behavior of the guiding system, we do not integrate this axis that defines the concrete representation of feedback based on visual features. This type of description is useful for the next step of the design process, regarding the concrete form of the guide, after the definition of its behavior.

Schönauer et al. perform an in-depth analysis of multimodal feedback for body movements [25]. They discuss nine aspects of guidance, of which six do not present design options of the behavior of a guiding system: For instance one aspect is the level of abstraction of a stimuli and its influence on the response time from a human information processing point of view. Three aspects are relevant for designing the behavior of a guiding system.

1. First, they consider the *modalities* (e.g., visual, (spatial-) audio, or haptic). The design space includes this axis applied to both feedback and feedforward.
2. Although they discuss the spatial locality of the feedback as it might cause a shift in attention or generate interference between stimuli (e.g., two haptic feedbacks too close from each other), they do not propose any design options. We proceed further by refining this *spatial locality* axis with three Allen's relations applied to spatial relations [30]: collocation, adjacency and separation. This allows us to distinguish between apparently similar guiding systems such as a static guide positioned under the cursor or at a corner of the screen [1].
3. Finally, Schönauer et al. characterize in *synchronicity and timeliness* a behavior for correcting the gesture of the user: the use of movement prediction to provide smooth corrections. Sodhi et al. [27] also propose this correction along with other options: the 'projection' and 'relocation' options. The design space includes the three options along an axis "Presentation Correction" (Figure 4). This axis contains an additional design option, namely. *Likelihood Maximization* option. This option defines a new correction in order to maximize the recognition score. As for the *filtering* axis, this "Presentation Correction" axis belongs to the subgroup that characterizes the impact of feedback onto feedforward.

Feedforward

Bau and Mackay define two dimensions for characterizing the feedforward mechanism [3].

1. First, they define the *level of details* provided by the feedforward mechanism. The guide can provide a minimal hint regarding the direction only, a portion of the gesture or the whole gesture. The design space includes this axis applied to both feedback and feedforward. To do so we use more generic terms for the options namely 'upcoming portion' and 'remaining portion' for the feedforward as well as 'complete portion' and 'partial portion' for the feedback. Our axis characterizes the level of knowledge about the past and future gestures from the user's point of view.
2. Second, as for the feedback mechanism, they define the *update rate* (once, multistep, continuous). As for feedback, in our design space this axis is split into two axes in order to distinguish continuous vs. discrete (multistep or once).

Sodhi et al. focus on design considerations for on-body guidance visual hints [27]. While they consider the feedback and feedforward separately, they mainly focus on the feedforward content and the gesture visualization. The authors also define two axes describing the guide's behavior: the timing and the perspective.

1. Timing refers to the pace at which the user will follow the content provided by the feedforward mechanism. It can be user-defined or system-defined. Our design space includes this axis.
2. Perspective refers to the viewpoint from which the user will see the gesture. Authors consider egocentric and exocentric viewpoints. Our design space includes this axis and extends it with the tethered viewpoint option [19].

The above studies define design axes and issues of our design space, but significant work was needed to gather, unify, organize and refine relevant design options in a comprehensible and usable design space. Thus, our design space (1) integrates these existing design axes/issues, (2) unifies them into a common and coherent structure and (3) extends them in order to provide a design space as thorough as possible for describing the behavior of the guide (When, What, How, Where). The distinction between the Feedback and Feedforward aspects emphasizes the fact that designers should consider design options for both aspects.

Guiding Systems

Due to space limitations, we do not provide a thorough description of all the 45 surveyed guiding systems, corresponding to 59 descriptions. However, the list of existing guiding systems and their descriptions along our design axes can be found online: http://iihm.imag.fr/delamare/guidance/. We focus on partial descriptions of systems that illustrate the new axes and options in our design space.

Existing guiding systems demonstrate different temporal designs. For instance, ShadowGuides, (Hierarchical) Marking Menu, OctoPocus and YouMove use a 'hold' action for the trigger of the guide [2, 3, 9, 17, 18]. The rationale behind this option is to make the trigger of the guide easy, in particular when the user hesitates during the gesture execution. In contrast, GestureBar [6] and the guides in [1] appear when the user clicks in a specific area (e.g., a button). We hence added the three axes – namely Initiative, Modality and Disambiguation – in order to characterize the *Trigger* in the *When* group. Similarly, we added the *Mode* subgroup in order to characterize how the guide disappears. Therefore, we can distinguish guiding

systems that automatically disappear (e.g., ShadowGuides [9]) from the systems that need an additional action from the user (e.g., GestureBar [6]).

Existing guiding systems are based on several spatial dimensions and options. For instance, we can identify two steps to describe ShadowGuides, namely the registration step and the execution step. The registration step provides information in the top-right corner of the screen about how to start the gestures (e.g., two hands on the surface or five fingers disposed in specific positions). The execution step displays command names and gestures, that are available according to the registration pose, in the bottom-left corner of the screen. In addition, while the execution step of ShadowGuides is inspired from OctoPocus [3], they have different priorities: OctoPocus aims at providing guidance in context (i.e. collocated with the on-screen cursor) while ShadowGuides aims to prevent occlusion on the touch-screen. These dimensions and options inspired our group *Where*.

Finally, existing guiding systems provide guidance information with several forms. Most 2D gesture guiding systems provide visual information by displaying the gestures [3, 6, 9, 17]. In addition to visual information, some 3D gesture guiding systems provide vibrotactile cues. Such vibrotactile stimulation during the execution of a gesture convey information about speed [25] or the distance between the executed and intended gesture [22]. Guiding systems using the visual modality can display the gestures in two different ways: while both YouMove [2] and the system in [12] demonstrate the gesture through a 3D digital skeleton, YouMove connects the skeleton to the user's posture (*relative*) whereas the skeleton in [12] is disconnected from the user (*absolute*). In order to distinguish these systems, we introduced the *Modality* and *Reference Frame* axes in the *How* group.

DESIGN SPACE VALIDATION & USAGE: SCENARIOS

A design space can be characterized along three dimensions [4]:

- Descriptive power: describe a significant range of systems.
- Evaluative power: help figuring out which design options to choose.
- Generative power: help designing new systems.

The completeness of a design space cannot be proven once and for all. However, its descriptive power ensures that all existing guiding systems are covered. We hence argue (1) that existing design issues or axes are included and/or extended in our design space that is based on a coherent structure along four questions: *When* (beginning, during, ending), *What*, *How* and *Where*, and (2), that a variety of existing systems are characterized in our design space highlighting their design specificities (http://iihm.imag.fr/delamare/guidance/).

The evaluative power of the design space is on-going work. In this section we further illustrate the descriptive power and focus on the generative power through two usage scenarios. These scenarios illustrate how our design space and its corresponding online tool facilitate the design phase of gesture guiding systems. The online tool offers four main services: (1) a description of existing systems one-by-one, (2) a high-level or detailed comparison of systems, (3) an interactive method to explore how often design options are used by existing systems and (4), a method to design a custom system and compare it to existing ones. In addition, the tool provides a repository of existing gesture guiding systems that can be kept updated. The following scenarios illustrate some of these services.

Scenario A: Finding a solution

John is an engineer developing an application for users to control Smart Environment appliances such as lighting sources with a centralized control (as in one of our collaborative project with industrial partners on a box for controlling light sources). John already has a solution for the selection phase that uses a distant pointing gesture [8] and is looking for a solution regarding the control phase (e.g., dimming or controlling the colors of selected light sources). Unfortunately, only a short time can be dedicated to interaction before the first prototype demonstration. John and his manager are interested in using gestural interaction, but cannot afford to spend too much time on the implementation of the gesture recognizer, the definition of the gesture set and the gesture guide. They are both clearly aware that they need a guiding system to let users know what commands are available and how to trigger them. For the design of the guiding system, our tool ('Interactive Design Space' tab) allows John to quickly find a guiding system that fulfills all his requirements without spending time to find the description of existing techniques: Since John has smartphone development skills, he wants to quickly implement a proof-of-concept of (i) a 3D gesture interaction system using sensors embedded in smartphones to capture gesture inputs. An early customers' survey showed that (ii) the visual modality would be preferably used. For this first iteration, he wants to focus on the feedforward behavior only, by showing (iii) all gestures (iv) with a relative reference frame. With only four click actions, John can see that a system complies with these criteria: 3D Marking Menu [23]. This guiding system represents 3D mid-air stroke gesture on a separated screen. He can now have a look at the corresponding system. Our tool allows John to quickly find – and benefit from – relevant previous works, i.e. reported implementation difficulties, detailed and low-level design options, results of experiments, etc. instead of developing yet another ad-hoc system from scratch. This ensures John will optimize his time on this part of the application. For this first iteration, John decides to implement a version of the 3D Marking Menu [23].

Summary: Our tool can help practitioners to design and develop a guiding system. The tool therefore also promotes the spreading of existing guiding systems in real applications.

Scenario B: Drawing inspiration and adapting solutions

After the first iteration, an extension of the external specifications prevents John from using an additional screen to present gestures as implemented in the original prototype of 3D Marking Menu [23]. Indeed, users should now be able to access the guide in every room of the smart environment but assuming a screen in every room is not a realistic solution. Since John has chosen to use a smartphone to capture the gesture inputs, he decides to display all gestures on the screen of the smartphone. However, he feels the need to integrate some vibrotactile feedback because of space limitation to present the gestures. The tool quickly reveals that only two systems use vibrotactile feedback [22, 25]. However, after clicking on the 'haptic feedback' option, John can see that these systems are designed for the feedforward of one gesture only. Thus, John should find a way to adapt the vibrotactile feedback for the case of several gestures. John removes the 'haptic feedback' filter by clicking on it and can see that the system SimpleFlow [5] allows a *Revelation Filtering* mechanism that leads to one gesture candidate only. John uses the prediction mechanism of SimpleFlow in order to determine which gesture is currently executed. Next, he adapts the vibrotactile pattern from [25] in order to make perceivable the distance between the current position of the smartphone and the closest boundary of the 3D zone of the elicited command. The closer the smartphone is to the boundaries, the stronger the vibration. John adds a hysteresis in order to prevent strong vibrations once the smartphone has crossed a boundary.

Summary: If no straightforward solution fulfills all requirements, our tool allow practitioners to quickly draw inspiration from previous work in order to adapt and/or to merge existing solutions.

CONCLUSION

This paper addresses the problem of the design of gesture guiding systems. Indeed, we argue that such gesture guiding systems can help to promote the use of gesture interaction. The main contribution of this work is to provide a design space that defines a unifying framework and organizes a set of design issues along axes. The design space enables the study of the design of gesture guiding systems in the light of interaction characteristics, in contrast to a technological-centered or representation-centered design approach. We illustrate how practitioners can efficiently use our design space through an interactive online tool.

We illustrate the descriptive and generative powers of the design space with scenarios. The evaluative power of the design space is one identified line of work. Assessing which options are better requires further systematic experiments,

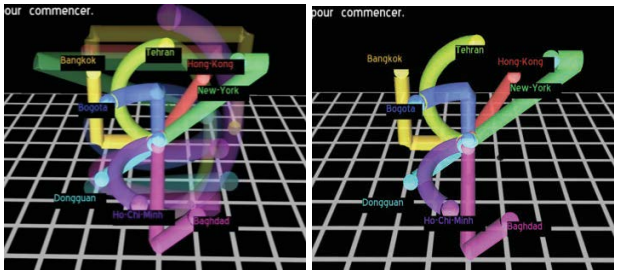


Figure 5: Illustration of two design options along the *Perceived Gesture* axis for guiding 3D gestures with 3D pipe visualization. Left: *Remaining* portion of the gesture. Right: *Upcoming* portion of the gesture.

Varying parameters along one axis only. The existing experiments described in other papers are difficult to use to bootstrap the evaluative power: for instance, the Crib Sheet Guide and the Static Guide evaluated in [1] vary along two axes, which prevents us from drawing evaluative conclusions on the corresponding options of our design space. The systematic experimental exploration of the design space is one future direction of work. As an ongoing work, we are currently developing a 3D gesture guiding system in order to explore design options along different design axes of our design space. Figure 5 shows an example of such design options by considering the *Perceived Gesture* axis. Exploring options of a given axis allows us to experimentally evaluate their effects and therefore to enrich the design space with concrete design recommendations.

ACKNOWLEDGMENTS

This work has been supported by the DELight project (French government's FUI -Single Inter-Ministry Fund-program). This work has also been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

REFERENCES

1. Anderson, F. and Bischof, W.F. Learning and performance with gesture guides. In *Proc. CHI '13*. (2013), 1109–1118.
2. Anderson, F., Grossman, T., Matejka, J. and George, F. YouMove: enhancing movement training with an augmented reality mirror. In *Proc. UIST '13*. (2013), 311–320.
3. Bau, O. and Mackay, W.E. OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proc. UIST '08*. (2008), 37–46.
4. Beaudouin-Lafon, M. Designing interaction, not interfaces. In *Proc. AVI '04*. (2004), 15–22.
5. Bennett, M., McCarthy, K., O'Modhrain, S. and Smyth, B. Simpleflow: enhancing gestural interaction with gesture prediction, abbreviation and autocompletion. In *Proc. INTERACT '11*. 6946, (2011), 591–608.

6. Bragdon, A., Zeleznik, R., Williamson, B., Miller, T. and LaViola, J.J. GestureBar: improving the approachability of gesture-based interfaces. In *Proc. CHI '09.* (2009), 2269–2278.

7. Colaço, A., Kirmani, A., Yang, H.S., Gong, N.-W., Schmandt, C. and Goyal, V.K. Mime: compact, low power 3D gesture sensing for interaction with head mounted displays. In *Proc. UIST '13.* (2013), 227–236.

8. Delamare, W., Coutrix, C. and Nigay, L. Mobile pointing task in the physical world: balancing focus and performance while disambiguating. In *Proc. MobileHCI '13.* (2013), 89–98.

9. Freeman, D., Benko, H., Morris, M.R. and Wigdor, D. ShadowGuides: visualizations for in-situ learning of multi-touch and whole-hand gestures. In *Proc. ITS '09.* (2009), 165–172.

10. Ghomi, E., Huot, S., Bau, O., Beaudouin-Lafon, M. and Mackay, W.E. 2013. Arpège: Learning Multitouch Chord Gestures Vocabularies. In *Proc. ITS '13* (2013), 209–218.

11. Hinckley, K., Guimbretiere, F., Baudisch, P., Sarin, R., Agrawala, M. and Cutrell, E. The springboard: multiple modes in one spring-loaded control. In *Proc. CHI '06.* (2006), 181–190.

12. Kamal, A., Li, Y. and Lank, E. Teaching motion gestures via recognizer feedback. In *Proc. IUI '14*. (2014), 73–82.

13. Karam, M. and Schraefel, m. c. A taxonomy of gestures in human computer interactions. (2005), 1–45.

14. Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L. and Marca, S. Di. Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing*. 10, 5 (2005), 285–299.

15. Kratz, S. and Rohs, M. A $3 gesture recognizer: simple gesture recognition for devices equipped with 3D acceleration sensors. In *Proc. IUI '10.* (2010), 341–344.

16. Kühnel, C., Westermann, T., Hemmert, F., Kratz, S., Müller, A. and Möller, S. I'm home: Defining and evaluating a gesture set for smart-home control. *International Journal of Human-Computer Studies*. 69, 11 (2011), 693–704.

17. Kurtenbach, G., Moran, T.P. and Buxton, W. Contextual Animation of Gestural Commands. *Computer Graphics Forum*. 13, 5 (1994), 305–314.

18. Kurtenbach, G., Sellen, A. and Buxton, W. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Human-Computer Interaction*. 8, 1 (1993), 1–23.

19. Milgram, P. and Colquhoun, H. A taxonomy of real and virtual world display integration. *Mixed reality: Merging real and virtual worlds*. (1999), 5–30.

20. Nacenta, M. a., Kamber, Y., Qiang, Y. and Kristensson, P.O. Memorability of pre-designed and user-defined gesture sets. In *Proc. CHI '13*. (2013), 1099–1108.

21. Norman, D.A. 2010. The way I see it: Natural user interfaces are not natural. *interactions*. 17, 3 (2010), 6–10.

22. Portillo-Rodriguez, O., Sandoval-Gonzalez, O.O., Ruffaldi, E., Leonardi, R., Avizzano, C.A. and Bergamasco, M. Real-time gesture recognition, evaluation and feed-forward correction of a multimodal tai-chi platform. *Haptic and Audio Interaction Design*. (2008), 30–39.

23. Ren, G. and O'Neill, E. 3D Marking menu selection with freehand gestures. In *Proc. 3DUI '12*. (2012), 61–68.

24. Rovelo, G., Degraen, D., Vanacken, D., Luyten, K. and Coninx, K. Gestu-Wan - An Intelligible Mid-Air Gesture Guidance System for Walk-up-and-Use Displays. In *Proc. INTERACT '15*, (2015), to appear.

25. Schönauer, C., Fukushi, K., Olwal, A., Kaufmann, H. and Raskar, R. Multimodal motion guidance: techniques for adaptive and dynamic feedback. In *Proc. ICMI '12* (2012), 133–140.

26. Schwaller, M., Kühni, J., Angelini, L. and Lalanne, D. 2014. Improving in-game gesture learning with visual feedback. *Human-Computer Interaction. Applications and Services*. 8512 LNCS, PART 3 (2014), 643–653.

27. Sodhi, R., Benko, H. and Wilson, A. Lightguide: projected visualizations for hand movement guidance. In *Proc. CHI '12.* (2012), 179–188.

28. Vatavu, R. Nomadic gestures: A technique for reusing gesture commands for frequent ambient interactions. *Journal of Ambient Intelligence and Smart Environments*. 4, (2012), 79–93.

29. Vermeulen, J., Luyten, K., van den Hoven, E. and Coninx, K. Crossing the bridge over Norman's Gulf of Execution: revealing feedforward's true identity. In *Proc. CHI '13.* (2013), 1931–1940.

30. Vernier, F. and Nigay, L. A framework for the combination and characterization of output modalities. *Interactive Systems Design, Specification, and Verification*. (2001), 35–50.