A Novel Taxonomy for Gestural Interaction Techniques Based on Accelerometers

Adriano Scoditti, Renaud Blanch and Joelle Coutaz
Laboratoire d'Informatique de Grenoble, Equipe IIHM, Université Grenoble I
385, rue de la Bibliotheque, BP 53, F-38041 Grenoble cedex 9, France
adriano.scoditti@imag.fr, renaud.blanch@imag.fr, joelle.coutaz@imag.fr

ABSTRACT

A large variety of gestural interaction techniques based on accelerometers is now available. In this article, we propose a new taxonomic space as a systematic structure for supporting the comparative analysis of these techniques as well as for designing new ones. An interaction technique is plotted as a point in a space where the vertical axis denotes the semantic coverage of the techniques, and the horizontal axis expresses the physical actions users are engaged in, i.e. the lexicon. In addition, syntactic modifiers are used to express the interpretation process of input tokens into semantics, as well as pragmatic modifiers to make explicit the level of indirection between users' actions and system responses. To demonstrate the coverage of the taxonomy, we have classified 25 interaction techniques based on accelerometers. The analysis of the design space *per se* reveals directions for future research.

Author Keywords

Handheld devices and mobile computing, Input and interaction technologies, Multi-modal interfaces, Recognition and interpretation of user input (face, body, speech etc.)

General Terms

Human Factors, Languages

ACM Classification Keywords

H.5.m Information Interfaces and Presentation (I.7): Miscellaneous

INTRODUCTION

Last generation mobile devices are enhanced with a diversity of sensors capable of probing real world physical properties in real time. The pioneering work on sensor-based interaction techniques [12, 18, 20, 26, 28] has paved the way for an active research area [1, 34, 35]. Although these results satisfy "the gold standard of science" [32], in practice, they are too "narrow truths" [6] to support designers decisions and researchers analysis. Designers and researchers need an overall systematic structure that helps them to reason, compare, elicit (and create!) the appropriate techniques for the problem at hand. Taxonomies, which provide such a structure, are good candidates for generalization in an emerging field. The challenge, however, is to provide a classification framework that is both complete and simple to use. Since completeness is illusory in a moving and prolific domain such as user interface design, we will not include it in our goals.

In this article, we propose a new taxonomy for gestural interaction techniques based on accelerometers. The motivation for limiting the coverage of the taxonomy to accelerometers-based interaction is that gestural interaction for mobile devices is a very vivid and unstructured area of research. Accelerometers are currently the most pervasive technology for sensing multiple dimensions of real world actions [18]. We will however see that our taxonomy is able to go beyond accelerometers-based techniques, covering a wide domain of issues related to interaction.

To develop our taxonomy, we have built a controlled vocabulary (i.e. primitives) obtained through an extensive analysis of the taxonomies that have laid the foundations for Human-Computer Interaction (HCI) more than twenty five years ago. For the most part, this early work in HCI has been ignored or forgotten by researchers driven by the trendy "technology push" approach.

Our taxonomy is based on the following principles:

(1) Interaction between a computer system and a human being is conveyed through input (output) expressions that are produced with input (output) devices, and that are compliant with an input (output) interaction language.

(2) As any language, an input (output) interaction language can be defined formally in terms of semantics, syntax, and lexical units.

(3) The generation of an input (output) expression involves using devices whose characteristics, from the human perspective, have a strong impact on the expressiveness and the effectiveness of the user interface [7].

Building on Foley's work [13] as well as on Buxton's pragmatics considerations of input structures [7], our taxonomy brings together the four aspects of interaction ranging from semantics to pragmatics with the appropriate human-motivated extensions for addressing the specificity of gestural interaction based on accelerometers. In contrast to Mackinlay *et al.*'s semantic analysis of the design space for input devices [21], we do not consider the transformation functions that characterize the system-oriented perspective of interaction techniques.

In this article we review the taxonomies that have served as sources of inspiration for our own work. This analysis offers the opportunity to clarify the terminology (after all, what is an interaction technique?) We then present our taxonomy illustrated with a survey of gestural interaction techniques. We conclude with future directions for research that our taxonomy has permitted to discover. Our expectation is to provide new insights and to start promising directions for the design of novel and powerful gestural interaction techniques.

REVIEW OF THE FOUNDATIONS

Classic HCI papers propose a wide spectrum of taxonomies falling into one of two main categories: **linguistics**-inspired taxonomies, and **morphological** taxonomies. Linguistics inspired taxonomies are driven by the lexical, syntactic, and semantic structures of languages. In the morphological approach, interaction techniques are points in a multi-dimensional space where each dimension represents a differentiating property. The next paragraph analyzes both approaches by referring to existing examples.

Linguistics-inspired taxonomies

Historically, the semantic-syntactic-lexical layers developed for artificial (formal) languages, have served as a useful tool for structuring the design process of user interfaces. In this approach, a user interface is assimilated to an artificial interaction language. This language is composed of an input interaction language that allows users to express their mental goals, and of an output interaction language that expresses the system state in terms that match users conceptual model. Foley, Wallace and Chan's taxonomy [13] and Moran's *et al.* Command Language Grammar (CLG) [22] are both examples of this approach.

The main contribution of Foley *et al.*'s taxonomy is twofold:

(1) **Six interaction tasks** that define the semantics of a canonical set of non-terminal symbols [words] for graphics (*select* "to make a selection from a set of alternatives", *position* to "indicate a position on an interactive surface", *orient* to "orient an entity in 2-D or 3-D space", *path* to "generate a path, which is a series of positions or orientations created over time", *quantify* to "specify a value, or quantify a measure", *text* to "enter a text string").

(2) **The cross product of these six interaction tasks with input devices** that shows the many ways each interaction task can be performed with existing devices.

To complement the six interaction tasks, which are tasks for specifying something, Foley introduces **four controlling tasks** such as *Stretch* and *Sketch* to express direct modifications of entities. Overall, Foley *et al.*'s taxonomy describes a large number of interaction techniques but does not define sharp boundaries between input devices, interaction tasks and controlling tasks resulting in an ambiguous definition of the notion of interaction technique *per se*.

For this reason, we define the terminology used in the linguistics approach to user interface design in the following way:

- a complete [input] sentence such as "<move> <entity> <position>" instructs the computer system to perform some functions (ideally, this function implements the semantics of the sentence).
- a sentence is composed of words (e.g., <move>, <entity>, <position>) whose assembly is compliant with a predefined syntax where each word is a symbol, that is, a primitive non-terminal that conveys a unit of semantics (or lexeme).

Using these definitions, "the entry of each symbol [word] by the user is an *interaction task* performed by means of an interaction technique" [13]. In other words, an *interaction technique* produces *non-terminal symbols [words]* by assembling terminals according to predefined lexical rules. These *terminals*, which belong to the digital world, result from the transformation of physical real world properties and actions sensed by physical input devices.

Our taxonomy re-uses Foley et al.*'s interaction tasks as a basis for non-terminal symbols: they are simple semantic units that have proven to be empirically valid.* In particular, Ballagas *et al.* use Foley *et al.*'s interaction tasks as a structuring framework to analyze smartphones viewed as input devices [1]. Although these semantic units are empirically sound, their lexical level "lumps together issues as diverse as: how tokens [words] are spelt, where devices are placed in the work station, the type of physical gesture used to articulate a token." [7].

Consequently, Buxton proposes to make a clear distinction between lexical issues as defined in artificial languages theory (e.g., spelling of words and choice of terminals) from "pragmatic issues of gesture, space and devices" which define "the primary level of contact of a user with a system" [7]. Drawing on the importance of pragmatics on users experience with systems, Buxton proposes a taxonomy of input devices that makes explicit pragmatic attributes including physical properties sensed by input devices (such as pressure, motion and position), the number of dimensions sensed (i.e. the number of degrees of freedom), as well as the sensing type (devices that work by touch vs. devices that require a mechanical intermediary). Buxton's taxonomy helps in finding equivalences between input devices, or in identifying places for the development of new devices. *Drawing on Buxton's analysis, our taxonomy, which is motivated by gestural interaction, incorporates some aspects of pragmatics.*

In the same vein as Buxton, Mackinlay *et al.* [21], then Card *et al.* [9], extend Buxton's work using a morphological approach to the analysis of input devices [7].

Morphological Taxonomies

Figure 1 shows Mackinlay *et al.*ś taxonomy considered by the scientific community as the archetypal morphological approach to device modeling. Mackinlay *et al.*ś model (improved later on by Card *et al.* [9]) uses a primitive movement vocabulary and a set of three composition operators (merge, layout, and connect) which, applied to the primitives produce a design space for reasoning about input devices. Formally, an input device is defined as a six-tuple composed by <the manipulation operator M, the input domain set In, the current state S, the resolution function R, the output domain set Out, behavior W>. Composition operators are used to combine inputs, outputs and devices, for example,*merge* such that the resulting input domain is the cross product of the input domain of the two devices, *connection* to combine two devices by cascading the output of one device to the input of the other, and *layout* to express spatial relationships between devices.

The resulting taxonomy (Figure 1) is a multidimensional parametric space where the y-axis denotes the physical property that can be manipulated by the device (In, M), while the x-axis corresponds to the dimensions of interest during the manipulation (Out, S). Interestingly, the taxonomy gives an idea of the continuity of the interaction technique supported by the device and the grain of the interaction itself by integrating information about the resolution function (R) that mapps the input domain set into the output domain set.

In [9], Card *et al.* present their exploitation of the taxonomy to reason about the effectiveness of input devices in terms of *Desk footprint*, *Pointing Speed*, *Pointing Precision*, *Errors*, *Time to Learn*, etc. These criteria form a sub-framework usable to compare apparently similar input devices. As important, Card *et al.* show how to reason about mappings between input devices and interaction tasks (e.g., pointing task, viewing task). For doing so, interaction tasks as well as input devices are plotted in the design space, while task-device mappings are represented as a connect operator from the device to the task. This representation completed with the parameters of the design space provides the designer with a sound and systematic apparatus for reasoning about the various design options. In other words, a morphological design space like this one can be used to integrate the results from several disciplines. Not only it supports reasoning on existing solutions, but also its structure *per se* is intended to foster the discovery of novel solutions.

More recently, Nancel *et al.* [23] proposed a morphological taxonomy for reasoning about menu-based techniques applicable to pen-driven interaction. This approach has been extended by considering additional input sensors. This work focuses on the property sensed by the considered devices and uses the vocabulary introduced by Mackinlay [21]. The originality of this taxonomy comes from the idea that by choosing the input device, the design space becomes a classification of input techniques. On the other hand, the proposed organization limits the discussion to the lexical aspect of interaction, leaving aside the syntactic and semantic dimensions. This limitation is too restrictive when considering gestural interaction techniques. In particular, context, which is key in mobile computing [10], as well as the distinction between foreground and background interaction [8] are ignored.

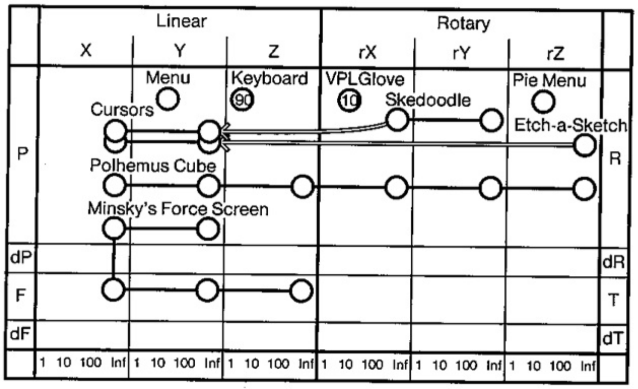

Figure 1. Physical, virtual and composite input devices can be classified using the taxonomy defined by Mackinlay *et al.*. A circle in a cell indicates that a device senses a physical property characterized by the coordinates of the grid. A black line represents a merge composition. An arrow represents a connect composition. A dashed line - no example shown here - a layout composition.

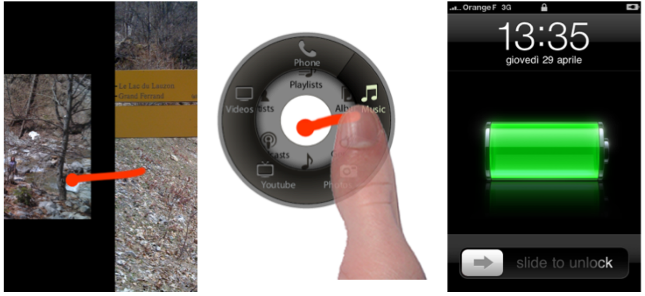

Figure 2. The "sliding" gesture is semantically multiplexed to achieve different meanings, depending on context.

A NEW TAXONOMY

As shown in Figure 2, the same gesture may convey very different meanings depending on the context in which it is produced: "go to previous photo" as for the Apple's photo album (or "go to next slide" as in Charade in [3]), "open a submenu" in Francone's Wavelet Menu [14], or "unlock" the iPhone screen. In addition, a gesture that makes sense for the system, may not be acceptable in a public social context [29] as it could be meaningful and interpreted by the public itself.

These observations lead us to define a new taxonomy according to the following principles: (1) Coverage of semantic, syntactic, lexical, and pragmatic issues of interaction where semantic granularity is that of Foley's et al. interaction tasks; (2) Adoption of a user centered perspective where physical human actions are premium, leaving aside the internal computational transformations; (3) Consideration for context; (4) Coverage of both foreground and background interaction (as defined by Buxton [8]). Figure 3 shows the elements of the framework that we describe in detail next. Figure 4 illustrates the use of the taxonomy for conventional WIMP techniques, whereas novel accelerometers hand-based interaction techniques are presented in Figure 5.

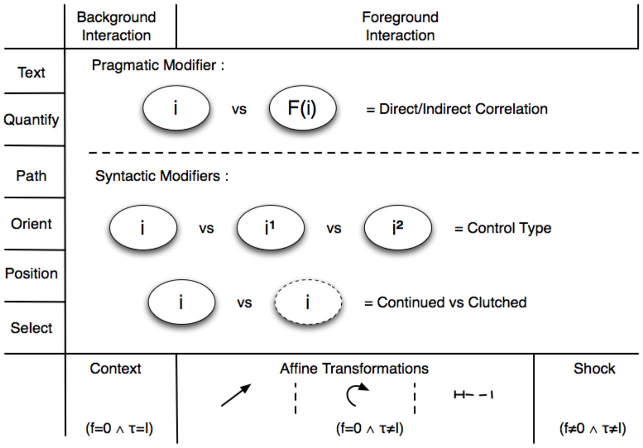

Figure 3. Our classification space for gestural interaction techniques based on accelerometers. The abscissa defines the lexicon in terms of the physical manipulations users perform with the device, with a clear separation between background and foreground interaction. The ordinate corresponds to Foley's interaction tasks. An interaction technique is uniquely identified by an integer *i* and plotted as a point in this space. Each point is decorated with the pragmatic and syntactic properties of the corresponding interaction technique.

Lexical Axis

Because of our focus on users' involvement in the interaction, the input lexicon corresponds to the physical actions users apply to devices. We divide human physical actions into two groups: (1) conscious actions that belong to the foreground interaction, and (2) unconscious actions that correspond to background interaction. The foreground interaction area contains the interaction techniques that require the user to consciously manipulate the device to reach some objective (as for the sliding gesture of Figure 2). The background interaction area corresponds to the interaction techniques where the system interprets user's unconscious actions together with contextual information to perform some system state change on behalf of the user. For example, during a phone call, the iPhone switches the screen backlight off to safe battery life as the user brings the device next to the ear.

Whether human actions are performed consciously to address the system or not, our classification space characterizes these actions with two additional variables: (τ) the geometrical transformation matrix that models user's movements in space, and (f) the frequency of these movements. The combinations of τ and f identify three sub-areas within the lexical axis: "Context", "Affine Transformations" and "Shock". The affine transformations group identifies the most common interaction techniques based on translations, rotations and/or scales (in this case, τ is different from the identity matrix I), and without any repetition (that is, f is equal to zero, meaning that the interaction is time driven). The sliding gesture of Figure 2 falls in this category. The shock category identifies those interaction techniques based on a combination of translations, rotations and/or scales (τ is different from the identity matrix) repeated over time (then, f is different from zero). The shake gesture exemplified by Shoogle [34] falls in this category. The context category corresponds to unconscious human manipulations that the system may interpret to feed into its own context model and, depending on this context, acts on behalf of the user. For this situation, we stipulate that τ is the Identity matrix and f is equal to zero.

As a simple example, consider the physical actions users need to perform in order to close a graphical window in a conventional WIMP environment. First, they have to move the mouse in the physical world, then to press the mouse button when the final position is reached in order to trigger the close window command. Both of these actions involve a translation: on the mouse plan for the first action and on the perpendicular axis when pressing the mouse button down. Yet both of them are time driven and do not involve frequency.

Syntactic Axis

Independently from the device used, we characterize the syntactic dimension of an interaction technique with the following two variables that we call *syntactic modifiers*: (1) the existence (or absence) of triggers to specify the begin/end of the interaction, and (2) the control type associated with the input token, which may be position-control, speed-control or acceleration-control. As a result, given that, in our taxonomy, an interaction technique is uniquely identified by an index *i*, the trigger syntactic modifier is represented as an oval that surrounds the interaction technique identifier using a dashed-line or a continuous line to respectively denote the presence (i.e. clutch) or absence (i.e unclutch) of a trigger. In addition, a derivative-like notation is used to convey the control type where *i* is decorated with an exponential number that expresses the derivative order with respect to time (i.e., no derivative for position, first order derivative for speed, and second order derivative for acceleration).

Analyzing the syntactic dimension of the close window example, mouse movements are translated into pointer movements. This pointer is position-controlled and the interaction is unclutched i.e. there is no explicit start/end action to bind the human physical movements with the software cursor movements. On the other hand, the selection of the close window button widget, which requires a shock action on the physical button of the mouse, is a clutched interaction technique.

Semantic Axis

As justified in our review about the foundational taxonomies developed in HCI, we re-use Foley's interaction tasks: Select, Position, Orient, Path, Quantify, and Text [13] (See the vertical axis of Figure 3).

Analyzing the semantic dimension of the close window example, the translation of the mouse corresponds to the user's goal to assign a new position to the mouse pointer motivated by the need to select the "close window" button widget. Finally, the click of the mouse physical button corresponds to the goal of confirming the selection of the widget soft button.

Pragmatic Axis

One of the originalities of our work is the attempt to classify gestural interaction techniques in close connection with their meaning in the user's real world. To do this, we introduce a *pragmatic modifier* that expresses the directness [25, 4] of the mapping between the user's expectation (i.e. goal) and the semantics of the interaction technique in the computer world. For indirect mapping, the identifier i of the interaction technique becomes the parameter of a function $F(i)$ to indicate the existence of one or several reinterpretation layers, whereas for direct mapping, i does not receive any additional decoration.

Analyzing the pragmatic dimension of the close window example, "positioning the pointer" has a direct pragmatic connection: every physical mouse translation is associated to a pointer translation. "Associating the mouse button to the software button widget" is also characterized by a direct pragmatic connection. The whole sequence of actions on the other hand is characterized by one degree of indirection since the user's objective is to close a graphical window, not to click a software button widget. Therefore, the button widget creates one level of indirection between human actions and the meaning of the sentence.

To illustrate the coverage of our taxonomy, we classify some mouse-driven interaction techniques. We will then move to gestural interaction techniques that involve accelerometers.

CLASSIFICATION OF WIMP TECHNIQUES

For conventional GUIs, cursor control is the paramount mouse-based interaction technique. As discussed above, when specifying a position with a mouse, physical translations are mapped directly into translations of the pointer: there is no pragmatic indirection function F. The interaction technique is position-based: there is no syntactic derivative modifier. In addition, it is continuously active. Fig. 4 shows its corresponding location as interaction technique (1).

Other familiar GUI interaction techniques include selecting an item in a linear menu, changing the orientation of a graphical object within a graphics editor, quantifying a dimension through a slider, or typing text with a virtual keyboard - respectively denoted in Fig. 4 as interaction techniques (2), (3), (5), and (6). All of them are characterized by some level of indirection due to the use of an intermediary graphical widget. All of them are position-based since they are built on the elementary positioning interaction technique. All of them are clutched since they need a trigger to specify the beginning and the end of the interaction (with a the mouse click button). As for Foley's path task *digitizing a sketch* (cf. (4) in Fig. 4), it consists of a temporal series of positions and orientations. It is thus position-controlled, clutched by the mouse button and indirect.

The classification of the common mouse-driven interaction techniques within our taxonomy calls for the following observations: (1) the mouse supports all of the interaction tasks identified by Foley, whether it be directly or indirectly. In other words, the mouse can be used to fully control a WIMP-based graphical system. (2) The taxonomy demonstrates the simplicity, the uniformity and the completeness of the interaction language supported by the mouse-driven WIMP interaction techniques since all of them are characterized by the same syntactic and pragmatic modifiers. (3) The taxonomy highlights the limited initiative the system has in such interactions as they are always explicitly performed by users (Foreground Interaction). (4) From the user's perspective, the interaction techniques that are characterized by some degree of indirection are more complex, while those that are pragmatically direct are simpler. This last observation brings forward a fundamental property of our taxonomy: *the less modifiers an interaction technique is characterized by, the simpler it is from the user's perspective*;

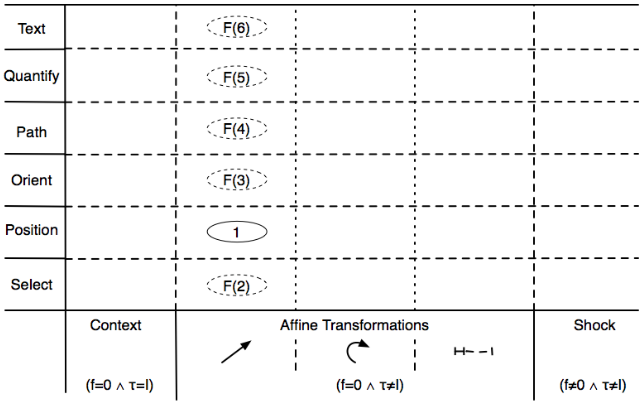

Figure 4. Classical mouse-driven interaction techniques within our taxonomy: (1) Positioning the cursor; (2) Menu item selection; (3) Defining orientation in a graphics editor; (4) Sketching using a drawing tool; (5) Defining a quantity through a slider; (6) Typing text with a virtual keyboard.

The classical interaction techniques for desktop devices have demonstrated the capacity of our taxonomy to provide designers with a synthetic view of the GUI paradigm. We now switch to mobile systems and post-WIMP interaction techniques that use accelerometers as input devices, and see what lessons can be drawn about them from the analysis of our taxonomy.

CLASSIFICATION OF ACCELEROMETERS-DRIVEN TECHNIQUES

The accelerometers-based input interaction techniques considered in this section are presented in chronological order and plotted in Figure 5. For the sake of completeness, all of the variations of an interaction technique are discussed. For example, an interaction technique that exists as continuous (i.e. un-clutched) and clutched appears twice in the taxonomic space, each one denoted with the appropriate syntactic modifiers.

Chameleon

Chameleon is one of the pioneering examples of immersive interaction techniques based on accelerometers *et al.* [12]. It is a palmtop solution where gestural interaction techniques are aware of the spatial position and orientation of the device. Three basic interaction techniques are proposed (denoted respectively as (7), (8), (9) in Fig. 5): (7) is based on the translation of the device whereas (8) and (9) involve tilting. All of them are available either as continuous or clutched where the begin/end if the interaction is specified through the press/release of a phisycal button. Consequently, they are represented twice in Fig. 5 respectively with the continuous line oval syntactic modifier as well as with the dashed-line oval. In the context of a spread-sheet application, selecting a cell with interaction technique (7) is performed by translating the device in the (x,y) plane. A series of translations along the z axis permits to select successive undo and redo commands. Tilting the device in a direction (say, left) permits to preview the (left-)adjacent cell (see (8) in Fig. 5). Selection and Preview do not imply any level of indirection as the control is directly connected to the item of interest. Interaction technique (9) supports the manipulation of a circular contextual menu to control a text browser or a movie player, thus introducing one level of indirection.

Rekimoto's experiments

Rekimoto analyzes clutched tilting interaction to control linear and circular menus (denoted as (10) in Fig. 5) as well as a map application according to interaction technique (11) [28]. Interaction technique (10) associates a tilt angle to each menu item to select a command through the use of the menu widget leading to an indirection. (11) directly associates a physical position to specify a position in a map. There is no indirection. Both techniques are position-controlled and clutched using a press/release of a physical button to mark the begin/end of the interaction.

Harrison's scenarios

Harrison *et al.* address the problem of navigation tasks within calendars and text-based applications. The originality of the solution (denoted as (12) in Fig. 5) relies on the use of a speed control type that changes the syntax of the technique: the larger the tilt angle is, the faster pages are scrolled. It is a clutched interaction technique since the begin/end of the interaction is marked either by repositioning the device to the initial position or by squeezing it. The squeeze solution, denoted as (13), is particularly original: it is a meta-interaction technique intended to stop the page-selection interaction technique by shocking the device.

Sketching with accelerometers

In his experiments reported in [28], Rekimoto motivates the use of tilt by the simplicity of sensing motion as variations in angle rather than by changes of position. Levin *et al.* propose an original method to sense positions through accelerometers by using acceleration first- and second-order derivatives (called respectively, *jerk* and *jounce* gestures) [20]. Their paper describes a fine tuned interaction (14) where physical translations are mapped into system translations as a series of positions. The algorithm is applied to a sketching tool to show the sharpness of the approach. This example illustrates the expressive power of our notation where jerk and jounce are represented with ^{3} and ^{4} exponents syntactic modifiers. It is a continuous interaction technique (absence of clutch to start and stop the interaction). On the other hand, the interaction technique introduces one level of indirection since the gesture acts on a virtual pen (which in turn draws on the canvas).

Rock'n'Scroll

The interaction techniques we have analyzed so far are concerned with foreground interaction only. At the opposite, in [2], Bartlett focuses primarily on background interaction: the system tries to understand user gestures in order to adapt dynamically to context changes such as screen orientation (15) [2]. For foreground interaction, Bartlett mimics familiar gestures to control the orientation of pictures by tilting the device vertically (16), while a horizontal tilt is used to select the next/previous picture by the way of a menu (17). Thus, (17) introduces a level of indirection. The tilt gesture (18) is also proposed for a 3D game to control the movements (position) of an avatar which in turn controls the game, thus introducing a level of indirection. While (15), (16) and (17) are position-controlled, (18) uses a speed control type.

TiltType

TiltType (19) supports text input by combining tilt angles to select characters organized in five position-controlled circular menus multiplexed through the use of five buttons [26]. The buttons also serve to trigger the interaction which therefore denotes a clutched interaction technique. In addition, selecting characters from circular menus introduces an indirection with regard to the text entry task.

Hinckley's state of the art

Hinckley *et al.* have developed an early classification of interaction techniques for mobile devices [18]. From this taxonomy, they propose a couple of interesting applications that are consistent with our own work by enhancing the background interaction already introduced in Barlett [2]. First, the paper models the screen orientation control by defining the bezels each zone should have in order to prevent from unstable situations; second it applies the same concept to the power management of the device by defining situation where the screen should switch off, letting the user to implicitly select power management options (20).

XWand

XWand allows users to select a device within a multimedia environment by pointing at the device of interest (21) and then to control it with gestures such as tilting the wand to control the volume level (22) *et al.* [35]. The paper proposes a selection interaction defined by complex physical gestures that combine translation and rotation. This is why (22) lies at the frontier between translation and rotation. The same role (and indirection level) widgets propose in desktop or mobile devices metaphors, is reproduced by XWand with real world objects thus introducing a level of indirection.

Peephole displays

A peephole is a spatially aware handheld display used as a window on a large (virtual) workspace [36]. Yee introduces two interaction techniques for this purpose based on accelerometers: (23) uses physical translations to position the peephole over the workspace; (24) enables the selection

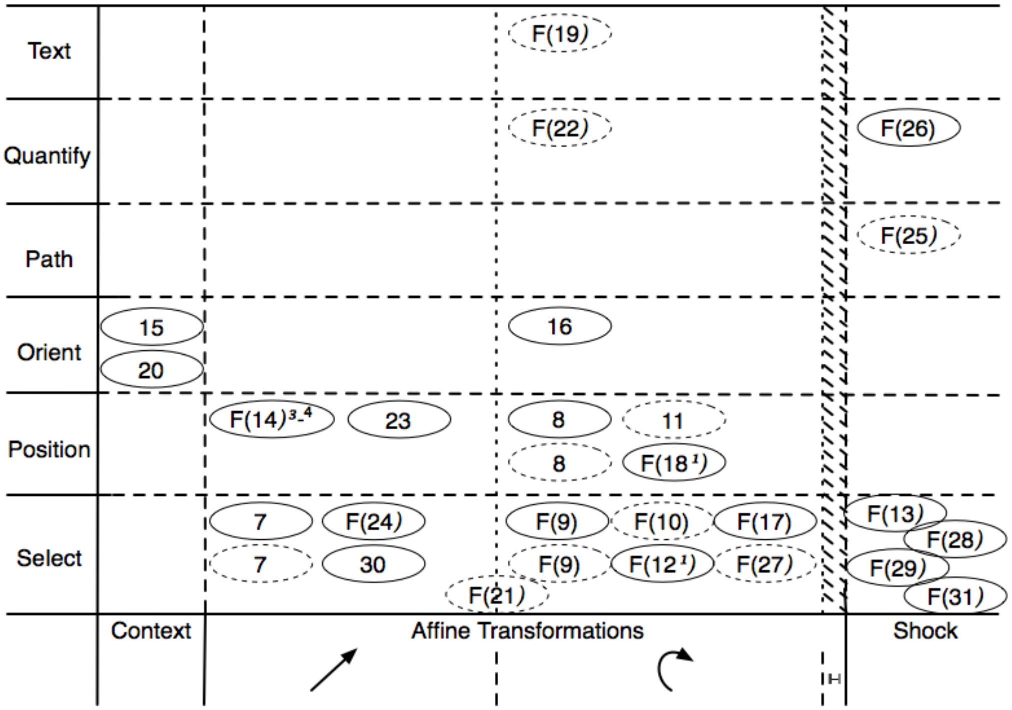

Figure 5. A state of the art of accelerometers-based interaction techniques. An interaction technique is identified by an integer i: (7) successive undo/redo as well as active cell selection through translations; (8) tilt to preview adjacent cells; (9) tilt to select a command in a pie menu; (10) tilt to select commands in linear and pie menus; (11) tilt to control position on a map; (12) tilt to browse a calendar; (13) squeeze to stop an interaction; (14) drawing through physical translations; (15) passive screen orientation adaptation; (16) active screen orientation control; (17) tilt to select pictures; (18) tilt to control first person shooter game; (19) tilt to enter text; (20) passive control of screen orientation and power energy saving; (21) tilt and translation to select physical world object; (22) control volume through tilt; (23) translation of virtual workspace through physical translation; (24) selection of the level of user interface details through translation; (25) gestural authentication with shock durations over time; (26) shake to quantify device status; (27) tilt to select graphical views; (28) shock to trigger an interaction; (29) shock to select the previously active application; (30) gesture recognition; (31) shake to select the next song.

among different views of the same content through physical translations. Both interactions are position-based and continuous. Whereas (23) directly maps physical translations into the desired position, (24) interprets user gestures to control view changes, thus introducing a level of indirection.

Gestural Authentication

Path is one of the least explored interaction tasks. Even though this task is defined as a composition of orientations and positions, it is unique because it also considers time as a component. The only work classified under this category comes from the studies conducted by Patel *et al.* [27]. The paper describes a public authentication method driven by a series of well-defined shock gestures applied to the device (25). The series of gestures constitutes the authentication path for a particular user. It is a clutched interaction technique since it is activated by the user on the authentication public terminal. It introduces a level of indirection since the user has to follow instructions proposed by the terminal screen.

Shoogle

Williamson *et al.* develop the idea to use shock gestures to sense a quantity [34]. By shaking their mobile phone, users can quantify the number of unread messages, of lost calls, or evaluate battery life (26). Each application of the Shoogle interaction technique is based on a metaphor which then introduces some indirection. Messages are assimilated to balls and the battery charge to a liquid quantity while the container is the device itself. Shoogle uses audio feedbacks correlated to the quantity of balls, or to the amount of liquid contained in the device.

TimeTilt and TapTap

Inspired from real world objects, Roudaut *et al.* propose two TimeTilt interaction techniques based on accelerometers [30]. (27) is a position-controlled interaction technique that enables choosing among different views by rotating the device. The interaction is triggered by tapping the back of the device (28). (29) supports switching through applications by shocking the device. It does not need to be triggered (29). (27) introduces a level of indirection since it depends on the graphical widgets that renders the different views.

Miscellaneous

Kratz *et al.* focused on feedback implications for gestural interaction techniques [19]. They propose a position-controlled gestural menu to select an option through physical gestures (30). We decided to classify this interaction technique as pragmatically direct as the paper does not associate any system command to the proposed gestures. The last interaction technique we analyze is the Apple's shake gesture available on the iPod to skip to the next song (31).

DISCUSSION AND RESEARCH DIRECTIONS

Figure 5 provides an overall picture of about 30 representative accelerometer-based input interaction techniques. The visual structure of this representation reveals three interesting facts: the absence of scale-based interaction techniques, *the dominance of the Select and Position interaction tasks*, with a *majority for pragmatic indirection*.

The absence of scale-based gestures is likely due to current technological limitations: typically, mobile devices are currently made of rigid material that limits the development of deformation-based interaction techniques. Nevertheless, the scale affine transformation opens the way for future research: in the near future, users will be able to shape their own devices as demonstrated in the early prototypes developed by [31, 17]. In addition, it is reasonable to envision twist- and scale-based interactions using accelerometers as input devices to propose a candidate language to perform, among others, the Path interaction task in a simple manner.

As made obvious by Figure 5, the centre of gravity is located in the lower part of the taxonomy. Clearly, *most interaction techniques based on accelerometers are concerned by the Select and Position interaction tasks*. Surprisingly, the use of accelerometers for specifying Orientation has not been explored extensively. Therefore, our classification suggests to concentrate research efforts on the development of interaction techniques to support Orientation, as well as Path, Quantity and Text input interaction tasks.

Most proposed accelerometer-based interaction techniques are characterized by *indirect I/O pragmatic connections*. Interestingly, Selection and Position are rarely implemented through pragmatically direct techniques. This is in contrast with mouse-based Foley's Select and Position atomic tasks. It results form such observation that accelerometers-based Interaction Techniques are not necessarily well suited to conventional WIMP interactors. The taxonomy *brings forward the difference between mouse-based interaction techniques and accelerometer-driven ones*. By construction, accelerometers sense acceleration, i.e., the direction of gravity essentially. These observations suggests to consider Orientation as an atomic tasks for accelerometers-based gestural interactions.

In addition to the three main observations revealed by Figure 5 (i.e. absence of scale, dominance of Select and Position, as well as primacy of indirection), the fine-grained structure of the taxonomy provides researchers and designers with the appropriate apparatus for sound reasoning. An indication of this is that our taxonomic space has allowed us to understand intrinsic and implicit differences even among apparently similar interaction techniques such as for example between (14) and (23) which, at first, appear to be very similar.

From the researcher's point of view, the classification shows a transparent state of the art where each interaction technique is classified without ambiguity. Typically, reference taxonomies such as [13] or [7] do not consider the role of time (cf. frequency and duration), nor do they cover unconscious interaction (cf. background interaction) and unstructured interaction such as device shaking. In addition, they do not explicitly consider whether an interaction technique is clutched or unclutched introducing ambiguities and mixing up different aspects of human interaction behavior.

From the designer's point of view, the dimensions of our taxonomy can be used as a framework for decision making. For example, an unclutched interaction technique may be considered for default tasks, while different clutched interaction techniques can be multiplexed through the use of standard or ad-hoc widgets. By proposing at least an interaction technique for each of the proposed task while designing an application, designers will be able to offer a complete and uniform user experience similar to the WIMP one. Furthermore, designers can predict the difficulties that final users will encounter by analyzing the pragmatic and syntactic modifiers that characterize the interaction techniques they envision. Thus, they will be able to choose interaction techniques that best suit the targeted representative users (novice, intermediate, expert).

CONCLUSIONS

We propose a novel taxonomy for accelerometer-based interaction techniques that uses the lexical, syntactic, semantic and pragmatic dimensions of languages to characterize the physical actions users need to perform to enter a command. We have not addressed any system-oriented issues, as we did not want, at this point, to differentiate interaction techniques by their implementation characteristics. Granularity, resolution function as well as state machines have already been taken into account by others [9, 21]. Our goal is to complement these taxonomies rather than acting as a substitute.

As demonstrated in this article, our taxonomy is radically centered on human physical actions. Our research hypothesis is that the physical action is the appropriate atomic level from which novel interaction techniques can be designed to provide system-wide consistent languages with specific attention for gestures involving scale as well as for gestures to specify ”orphans” commands such as Path, Quantity, and Text. In addition, the syntactic and pragmatic modifiers of our classification space provide a sound predictive measure for the learning curve users have to go over when approaching a new interaction technique. We hope that our taxonomic space, as a design framework, will encourage the development of novel interaction techniques that will support users through a more agreeable, *natural* [7] and *intuitive* system.

REFERENCES

1. R. Ballagas, J. Borchers, M. Rohs, and J. G. Sheridan. The smart phone: A ubiquitous input device. *IEEE Pervasive Computing*, 5(1):70, 2006.

2. J. F. Bartlett. Rock 'n' scroll is here to stay. *IEEE*

Comput. Graph. Appl., 20(3):40–45, 2000.

3. T. Baudel and M. Beaudouin-Lafon. Charade: remote control of objects using free-hand gestures. *Commun. ACM*, 36(7):28–35, 1993.

4. M. Beaudouin-Lafon. Instrumental interaction: an interaction model for designing post-wimp user interfaces. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 446–453, New York, NY, USA, 2000. ACM.

5. R. A. Bolt. "put-that-there": Voice and gesture at the graphics interface. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270, New York, NY, USA, 1980. ACM.

6. F. P. Brooks. Grasping reality through illusion—interactive graphics serving science. In *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1–11, New York, NY, USA, 1988. ACM.

7. W. Buxton. Lexical and pragmatic considerations of input structures. *SIGGRAPH Comput. Graph.*, 17(1):31–37, 1983.

8. W. Buxton. Integrating the periphery and context: A new model of telematic. *Proceedings of Graphics Interface*, pages 239–246, 1995.

9. S. K. Card, J. D. Mackinlay, and G. G. Robertson. A morphological analysis of the design space of input devices. *ACM Trans. Inf. Syst.*, 9(2):99–122, 1991.

10. J. Coutaz, J. L. Crowley, S. Dobson, and D. Garlan. Context is key. *Commun. ACM*, 48:49–53, March 2005.

11. J. Coutaz, L. Nigay, D. Salber, A. Blandford, J. May, and R. Young. Four easy pieces for assessing the usability of multimodal interaction: The CARE properties. In *Proceedings of the INTERACT'95 conference*, pages 115–120, 1995. S. A. Arnesen & D. Gilmore Eds., Chapman & Hall Publ., Lillehammer, Norway.

12. G. W. Fitzmaurice, S. Zhai, and M. H. Chignell. Virtual reality for palmtop computers. *ACM Trans. Inf. Syst.*, 11(3):197–218, 1993.

13. J. D. Foley, V. L. Wallace, and P. Chan. The human factors of computer graphics interaction techniques. *IEEE Comput. Graph. Appl.*, 4(11):13–48, 1984.

14. J. Francone, G. Bailly, L. Nigay, and E. Lecolinet. Wavelet menu: une adaptation des marking menus pour les dispositifs mobiles. In *IHM '09: Proceedings of the 21st International Conference on Association Francophone d'Interaction Homme-Machine*, pages 367–370, New York, NY, USA, 2009. ACM.

15. D. Goldberg and C. Richardson. Touch-typing with a stylus (abstract). In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, page 520, New York, NY, USA, 1993. ACM.

16. B. L. Harrison, K. P. Fishkin, A. Gujar, C. Mochon, and R. Want. Squeeze me, hold me, tilt me! an exploration of manipulative user interfaces. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 17–24, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

17. F. Hemmert, G. Joost, A. Knörig, and R. Wettach. Dynamic knobs: shape change as a means of interaction on a mobile phone. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2309–2314, New York, NY, USA, 2008. ACM.

18. K. Hinckley, J. Pierce, M. Sinclair, and E. Horvitz. Sensing techniques for mobile interaction. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 91–100, New York, NY, USA, 2000. ACM.

19. S. Kratz and R. Ballagas. Unravelling seams: improvoing mobile gesture recognition with visual feedback techniques. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 937–940, New York, NY, USA, 2009. ACM.

20. G. Levin and P. Yarin. Bringing sketching tools to keychain computers with an acceleration-based interface. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, pages 268–269, New York, NY, USA, 1999. ACM.

21. J. Mackinlay, S. K. Card, and G. G. Robertson. A semantic analysis of the design space of input devices. *Hum.-Comput. Interact.*, 5(2):145–190, 1990.

22. T. P. Moran. The command language grammar: a representation for the user interface of interactive computer systems. *International Journal of Man-Machine Studies*, 15(1):3 – 50, 1981.

23. M. Nancel, S. Huot, and M. Beaudouin-Lafon. Un espace de conception fondé sur une analyse morphologique des techniques de menus. In *IHM '09: Proceedings of the 21st International Conference on Association Francophone d'Interaction Homme-Machine*, pages 13–22, New York, NY, USA, 2009. ACM.

24. L. Nigay. *Modalit d'Interaction et Multimodalit*. PhD thesis, 2001. Thse habilitation diriger des recherches. prpare au Laboratoire de Communication Langagire et Interaction Personne-Systme (IMAG), Universit Joseph Fourier 74 pages.

25. D. Norman. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., 1986.

26. K. Partridge, S. Chatterjee, V. Sazawal, G. Borriello, and R. Want. Tilttype: accelerometer-supported text entry for very small devices. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 201–204, New York, NY, USA, 2002. ACM.

27. S. N. Patel, J. S. Pierce, and G. D. Abowd. A gesture-based authentication scheme for untrusted public terminals. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 157–160, New York, NY, USA, 2004. ACM.

28. J. Rekimoto. Tilting operations for small screen interfaces. In *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 167–168, New York, NY, USA, 1996. ACM.

29. J. Rico and S. Brewster. Usable gestures for mobile interfaces: evaluating social acceptability. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 887–896, New York, NY, USA, 2010. ACM.

30. A. Roudaut, M. Baglioni, and E. Lecolinet. Timetilt: Using sensor-based gestures to travel through multiple applications on a mobile device. In *INTERACT '09: Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction*, pages 830–834, Berlin, Heidelberg, 2009. Springer-Verlag.

31. C. Schwesig, I. Poupyrev, and E. Mori. Gummi: a bendable computer. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 263–270, New York, NY, USA, 2004. ACM.

32. M. Shaw. What makes good research in software engineering? *International Journal of Software Tools for Technology*, 4(1):1–7, 2002.

33. J. Wang, S. Zhai, and J. Canny. Camera phone based motion sensing: interaction techniques, applications and performance study. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 101–110, New York, NY, USA, 2006. ACM.

34. J. Williamson, R. Murray-Smith, and S. Hughes. Shoogle: excitatory multimodal interaction on mobile devices. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 121–124, New York, NY, USA, 2007. ACM.

35. A. Wilson and S. Shafer. Xwand: Ui for intelligent spaces. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 545–552, New York, NY, USA, 2003. ACM.

36. K.-P. Yee. Peephole displays: pen interaction on spatially aware handheld computers. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1–8, New York, NY, USA, 2003. ACM.

37. S. Zhai, P. Milgram, and D. Drascic. An evaluation of four 6 degree-of-freedom input techniques. In *CHI '93: INTERACT '93 and CHI '93 conference companion on Human factors in computing systems*, pages 123–125, New York, NY, USA, 1993. ACM.